PEDOMAN TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB

BADAN SIBER DAN SANDI NEGARA

VERSI 1 - 2019

PEDOMAN TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB

BADAN SIBER DAN SANDI NEGARA

2019

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

TIM PENYUSUN

PEDOMAN TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB VERSI 1

- Pengarah : Akhmad Toha, S.AP
- Ketua Tim Penyusun : Rizkal, Sos., M.M.
- Anggota Tim Penyusun :
    1. Taufik Nurhidayat
    2. Supapri Situmorang
    3. Luhut Parulian
    4. Aris Munandar
    5. Yan Hadynoer
    6. Reikal Taupaani
    7. Sandy Rachman

- Tim Konsultan :
    1. Wahyu Winarno, ST. MT, ITIL
    2. Resdy Benyamin, MM, CGEIT, CISSP, CRISC, ITIL
    3. Surya Lesmana, ST, MT, CISA
    4. K. Satria Yudistira, ST, IRCA ISO27001
    5. Kumoro Wisnu Wibowo, M.Sc, CEH, CISSP
    6. Nuki Agustino, S.Kom, CISSP

DEPUTI BIDANG PROTEKSI

Sebagai Instansi Pemerintah, upaya yang dilakukan BSSN untuk mewujudkan keamanan siber nasional adalah dengan memastikan terjaminnya keamanan siber di berbagai sektor seperti Sektor Pemerintah, sektor Infrastruktur Informasi Kritikal Nasional (IIKN), dan sektor Ekonomi Digital yang salah satunya melalui pelaksanaan kegiatan proteksi keamanan siber. Dalam mendukung penyelenggaraan keamanan SPBE khususnya pada sektor Pemerintah, diperlukan berbagai jenis kebijakan turunan, standar, dan prosedur salah satunya terkait dengan keamanan aplikasi. Tidak sedikit aplikasi umum dan aplikasi khusus SPBE yang dibangun dan dikembangkan di lingkungan pemerintahan adalah aplikasi yang berbasis web.

Domain keamanan merupakan aspek penting pada aplikasi SPBE, khususnya aplikasi berbasis web yang dapat berjalan di Internet dan diakses oleh banyak orang. Oleh karena itu, BSSN dalam hal ini Deputi Bidang Proteksi guna memberikan informasi dan panduan kepada publik mengenai keamanan aplikasi berbasis web menyusun Pedoman Tata Kelola Keamanan Aplikasi Berbasis Web.

Secara umum ada 5 aspek keamanan dasar yang termuat dalam pedoman ini yaitu: *Authentication, Authorization, Confidentiality, Data Integrity* dan *Non-Repudiation*. Pedoman Tata Kelola Keamanan Aplikasi Berbasis Web memberikan konteks pengamanan secara end-to-end, dimana setiap proses harus dapat dijamin keamanannya mulai dari asal transaksi sampai dengan penyelesaian akhir transaksi sehingga dapat mempertahankan keamanan yang konsisten di semua tahapan pengolahan setiap prosesnya.

Apresiasi dan ucapan terima kasih yang sebesar-besarnya kepada berbagai pihak yang telah turut berpartisipasi dalam penyusunan dan penyempurnaan Pedoman Tata Kelola Keamanan Aplikasi Berbasis Web Tahun 2019 ini. Semoga Pedoman ini dapat memberikan manfaat. Terima kasih.

AKHMAD TOHA, S.AP.

PEDOMAN TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB

PENDAHULUAN

KEBIJAKAN TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB

STANDAR TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB

PROSEDUR TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB

CHECKLIST PENANGANAN KERENTANAN & WEB SERVER

PEDOMAN TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB

PENDAHULUAN

- Istilah/definisi
- Latar Belakang
- Peruntukan
- Lingkup Umum
- Tujuan
- Manfaat

KEBIJAKAN TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB

- Kategorisasi & Manajemen Risiko Pengamanan Aplikasi Berbasis Web
- Pengendalian Preventif Keamanan Aplikasi Berbasis Web
- Pengelolaan Akses, Otorisasi & Otentikasi
- Pengembangan & Pengelolaan Aplikasi Berbasis Web Oleh Pihak Ketiga
- Pengendalian Insiden Keamanan Aplikasi Berbasis Web
- Peran & Tanggungjawab Pengelolaan

STANDAR TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB

- Standar Arsitektur Keamanan Aplikasi Berbasis Web
- Standar Pengendalian Keamanan Web Server
- Standar Penempatan Aplikasi Berbasis Web & Infrastruktur Pendukung
- Standar Pengujian Aplikasi Berbasis Web
- Standar Pengendalian Insiden Keamanan Aplikasi Berbasis Web
- Standar Secure-Software Developement Life Cycle
- Standar Tools Pengamanan Aplikasi Berbasis Web

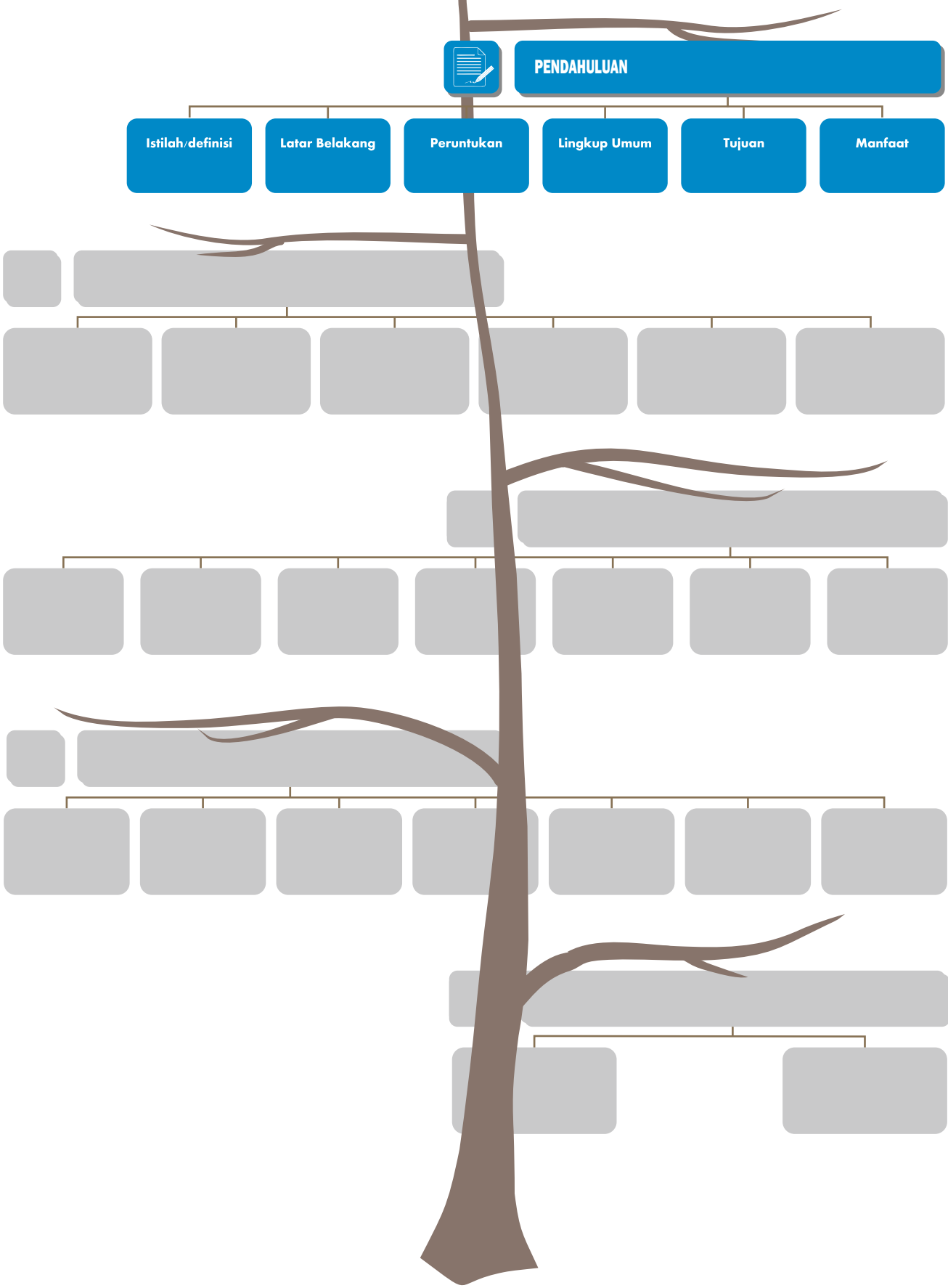
PROSEDUR TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB

- Prosedur Pengendalian Hak Akses Pengguna
- Prosedur Pengendalian Insiden Keamanan Aplikasi Berbasis Web
- Prosedur Pengujian Keamanan Aplikasi Berbasis Web
- Prosedur Monitoring Keamana Aplikasi Berbasis Web
- Prosedur Manajemen Kerentanan Aplikasi Berbasis Web
- Prosedur Manajemen Pihak Ketiga
- Prosedur Penguatan Keamanan Web Server (Hardening)

CHECKLIST PENANGANAN KERENTANAN & WEB SERVER

- Checklist Penanganan Kerentanan Berbasis Owasp
- Checklist Pengelolaan Keamanan Web Server

PEDOMAN TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB



I. PENDAHULUAN

I.1 ISTILAH/DEFINISI

a. Instansi adalah Kementerian/Lembaga, instansi pemerintah pusat dan daerah.

b. Aplikasi berbasis web atau web adalah sebuah aplikasi yang dapat diakses menggunakan web browser atau penjelajah web melalui jaringan internet atau intranet.

c. Kontrol akses adalah praktik untuk menentukan seluruh siklus otorisasi individu terkait transaksi, fungsi, dan aktivitasnya yang sah berkenaan dengan sumber daya informasi Instansi.

d. Otentikasi adalah proses dimana pengguna, proses, atau layanan memberikan bukti identitas mereka.

e. Ancaman (*Threat*) adalah kemungkinan bahaya yang dapat mengeksploitasi kerentanan suatu sistem untuk melanggar keamanan dan karenanya menyebabkan kemungkinan bahaya. Ancaman dapat berupa "disengaja" (misal: seorang cracker perorangan atau organisasi kriminal) atau "tidak disengaja" (misal: kemungkinan kerusakan komputer, bencana alam seperti gempa bumi, kebakaran, atau banjir).

f. Kerentanan (*Vulnerability*) adalah kelemahan yang dapat dimanfaatkan oleh aktor ancaman, untuk melakukan tindakan tidak sah dalam sistem komputer. Untuk mengeksploitasi kerentanan, penyerang harus memiliki setidaknya alat atau teknik yang dapat diterapkan yang terhubung ke kelemahan sistem.

g. Gangguan/Insiden adalah hal atau usaha yang berasal dari luar yang bersifat atau bertujuan melemahkan atau menghalangi secara tidak konsepsional (tidak terarah).

h. *Cookie* adalah sepotong kecil data yang dibuat dan disimpan di browser pengguna yang akan melacak informasi penting mengenai informasi sesi untuk situs tertentu.

i. *Content Management System* (CMS) adalah adalah perangkat lunak yang digunakan untuk menambahkan/ memanipulasi (mengubah) isi dari suatu situs web. Umumnya, sebuah CMS (*Content Management System*) terdiri dari dua elemen:

   - Aplikasi manajemen isi (Content Management Application, CMA)
   - Aplikasi pengiriman isi (Content Delivery Application, CDA)

j. *Intrusion Detection and Prevention* (IDPS), adalah suatu bentuk sistem keamanan jaringan/aplikasi yang berfungsi untuk mendeteksi dan mencegah ancaman yang teridentifikasi dengan karakteristik:

   - Memantau jaringan secara terus-menerus, mencari kemungkinan insiden berbahaya dan menangkap informasinya.
   - Melaporkan kejadian ancaman ini ke administrator sistem dan mengambil tindakan pencegahan, seperti menutup titik akses dan mengonfigurasi firewall untuk mencegah serangan di masa depan.

- Mengidentifikasi masalah dengan kebijakan keamanan, menghalangi karyawan dan tamu jaringan melanggar aturan yang terkandung dalam kebijakan ini.

k. *Web Application Firewall* (WAF), adalah sebuah sistem firewall yang dikhususkan untuk menyaring, memantau, dan menghalangi trafik HTTP/S dari dan oleh aplikasi web.

l. *Security Information and Event Management* (SIEM), adalah produk dan layanan perangkat lunak yang menggabungkan security information management (SIM) dan security event management (SEM). Tools ini menyediakan analisis real-time peringatan keamanan yang dihasilkan oleh aplikasi dan perangkat keras jaringan.

m. *Network Behavioral Analysis* (NBA) adalah cara untuk meningkatkan keamanan jaringan dengan memantau trafik dan mencatat tindakan atau penyimpangan yang tidak biasa dari operasi normal dengan karakteristik sebagai berikut:

- Solusi NBA memantau apa yang terjadi di dalam jaringan, mengumpulkan data dari banyak titik untuk mendukung analisis luring (offline).
- Setelah menetapkan tolok ukur untuk trafik normal, program NBA secara pasif memantau aktivitas jaringan dan menKitai pola yang tidak diketahui, baru atau tidak biasa yang mungkin mengindikasikan adanya ancaman.
- NBA juga dapat memantau dan mencatat tren dalam penggunaan lebar-pita (bandwidth) dan protokol.
- NBA sangat baik untuk menemukan malware baru dan eksploitasi 0-hari.

n. *IP Reputation Engine* (RE) - fraud/phishing control IP RE adalah sebuah sistem yang berbasis pada kepercayaan melalui reputasi sebagai sarana mengidentifikasi kemungkinan ancaman keamanan siber dengan karakteristik:

- IP RE dapat dibentuk oleh para pengguna dengan saling menilai dalam komunitas daring (online) untuk membangun kepercayaan melalui reputasi.
- Atau IP RE dapat dilakukan oleh sebuah institusi yang kemudian akan membagikannya kepada pengguna melalui perangkat.

o. *Anti Distributed Denial of Services* (DDoS), adalah tools yang digunakan untuk memitigasi serangan DDoS.

p. *Aplikasi Berbasis WebDefacement Monitoring*, adalah tool berupa sotware/appliance untuk memantau usaha peretas mengubah konten, tampilan visual seluruh situs, berKita, atau halaman web tertentu, sekaligus mengembalikan tampilan web ke kondisi semula sebelum diubah.

q. *Unified Threat Management*, adalah suatu pendekatan pengamanan informasi dimana suatu perangkat keras/perangkat lunak tunggal menyediakan berbagai fungsi keamanan. Ini kontras dengan metoda tradisional yang memiliki solusi titik untuk setiap fungsi keamanan. UTM menyederhanakan manajemen keamanan informasi dengan menyediakan satu manajemen dan titik pelaporan untuk administrator keamanan melaui pengelolaan beberapa produk dari beberapa vendor yang berbeda.

r. *Hidden Field Manipulation* (Manipulasi field-tersembunyi) merupakan salah satu peretasan dengan cara mengubah input pada field-tersembunyi. Serangan ini terutama berfokus pada situs web e-niaga.

s. *Cookie Poisoning*, merupakan tindakan memanipulasi atau memalsukan cookie untuk tujuan melewati langkah-langkah pengamanan atau mengirim informasi palsu ke server.

t. *Parameter Tampering* (Pengubahan Parameter), serupa dengan HFM, merupakan salah satu peretasan dengan cara mengubah input pada parameter-parameter web dalam URL, untuk mendapatkan akses otorisasi tanpa seijin pengguna.

u. *Buffer Overflow Attacks* (Serangan Buffer Overflow), merupakan peretasan dengan menggunakan input-cacat untuk menyebabkan kondisi dimana ektra informasi tidak dapat ditampung pada buffer dan meng-overwrite ruang memori bersebelahan yang dapat mengakibatkan sistem-terhenti, bocornya informasi, teraksesnya kode tertentu, atau memasukkan kode pada aplikasi.

v. *Cross Site Scripting* (XSS)/serangan XSS adalah peretasan dengan injeksi, di mana skrip diinjeksikan ke situs web yang tepercaya. Serangan XSS terjadi ketika penyerang menggunakan aplikasi web untuk mengirim kode berbahaya, umumnya dalam bentuk skrip sisi peramban, ke pengguna-akhir yang berbeda.

w. *Backdoor or Debug options* (Opsi debug), merupakan peretasan dengan memanfaatkan backdoor atau opsi debug tersedia yang dibuat secara sengaja oleh developer/pemrogram untuk memeriksa aplikasi, sehingga peretas dapat mengakses informasi sensitif secara mudah.

x. *Stealth Commanding* (Perintah Senyap), merupakan peretasan dengan memanfaatkan kelemahan inheren pada eksekusi perintah yang terdapat pada web server.

y. *Forced Browsing* (Jelajah Paksa) adalah serangan untuk mengakses sumber daya yang tidak dirujuk oleh aplikasi, tetapi masih dapat diakses. Peretasan manual ini biasanya menggunakan prediksi bila indeks direktori aplikasi berbasis pada generator angka atau nilai yang terprediksi.

z. *Third Party Misconfigurations* (Miskonfigurasi Pihak Ketiga), merupakan kondisi yang sangat umum, dimana kontrol sekuriti untuk aplikasi web dan/atau server gagal diimplementasikan dengan baik atau salah diimplementasikan.

aa. *Known Vulnerabilities* (Kerentanan yang diketahui), merupakan kondisi dimana pembuatan dan/atau implementasi sebuah aplikasi web menggunakan kode yang secara umum diketahui memiliki celah keamanan, menggunakan kode yang diambil dari orang lain tanpa verifikasi terlebih dahulu, atau menggunakan kode yang diambil dari sumber yang tingkat keamanannya rendah.

bb. *Web Deface* adalah adalah teknik mengganti atau menyisipkan file pada server, teknik ini dapat dilakukan karena terdapat lubang pada sistem security (vulnerability) yang ada di dalam sebuah situs Aplikasi Berbasis Web. Hal ini bertujuan untuk melakukan perubahan tampilan pada Aplikasi Berbasis Webkorban dengan tampilan yang dimiliki oleh defacer

cc. *Distributed Denial of Service* (DDoS) adalah salah satu jenis serangan Denial of Service yang menggunakan banyak host penyerang (baik itu menggunakan komputer yang didedikasikan untuk melakukan penyerangan atau komputer yang "dipaksa" menjadi zombie) untuk menyerang satu buah host target dalam sebuah jaringan.

dd. *Social Engineering* adalah manipulasi psikologis dari seseorang dalam melakukan aksi atau menguak suatu informasi rahasia. Rekayasa sosial umumnya dilakukan melalui telepon atau Internet. Rekayasa sosial merupakan salah satu metoda yang digunakan oleh peretas untuk memperoleh informasi tentang targetnya, dengan cara meminta informasi itu langsung kepada korban atau pihak lain yang mempunyai informasi itu. Rekayasa sosial mengkonsentrasikan diri pada rantai terlemah sistem jaringan komputer, yaitu manusia. Tidak ada sistem komputer yang tidak melibatkan interaksi manusia. Dan parahnya lagi, celah keamanan ini bersifat universal, tidak tergantung platform, sistem operasi, protokol, perangkat lunak, ataupun perangkat keras. Artinya, setiap sistem mempunyai kelemahan yang sama pada faktor manusia. Setiap orang yang mempunyai akses kedalam sistem secara fisik adalah ancaman, bahkan jika orang tersebut tidak termasuk dalam kebijakan kemanan yang telah disusun.

ee. *Pharming* adalah serangan dunia maya yang ditujukan untuk mengarahkan lalu lintas situs web ke situs palsu lainnya. Pharming dapat dilakukan dengan mengubah file host pada komputer korban atau dengan mengeksploitasi kerentanan dalam perangkat lunak server DNS. Server DNS adalah komputer yang bertanggungjawab untuk menyelesaikan nama Internet menjadi alamat IP asli mereka. Server DNS yang dikompromikan kadang-kadang disebut sebagai "diracuni". Pharming memerlukan akses tanpa perlindungan ke komputer target.

ff. *Phising* adalah usaha seorang attacker untuk mendapatkan informasi dari korban seperti username dan password dengan cara memancing korban untuk login melalui halaman palsu yang telah disiapkannya. Halaman palsu tersebut di buat dengan tampilan yang mirip dengan halaman aslinya, tujuannya agar korban tidak curiga. Setelah korban login, maka informasi yang di butuhkan oleh seorang attacker tercatat dalam sebuah log. Biasanya korban akan sadar bahwa dia telah ditipu setelah mengisikan username dan password kemudian login namun tidak dapat masuk ke akun yang dituju, justru malah terdirect ke halaman lain.

gg. *Ransomware* adalah jenis perangkat perusak yang dirancang untuk menghalangi akses kepada sistem komputer atau data hingga tebusan dibayar. Jenis yang sederhana bekerja dengan mengunci sistem dengan cara yang tidak sulit untuk ditangani oleh orang yang ahli, sedangkan jenis yang lebih canggih akan mengenkripsi berkas sehingga tidak dapat diakses. Serangan perangkat pemeras umumnya dilakukan melalui Trojan yang disamarkan sebagai berkas yang sahih.

hh. Pembangkit Konten (*Content Generator*) adalah suatu program pada Web server yang secara dinamis menghasilkan halaman-halaman HTML untuk para pengguna; halaman-halaman tersebut mungkin dihasilkan dengan menggunakan informasi yang diambil dari suatu server back end. Teknologi pembangkit konten di sisi server antara lain CGI, ASP .NET, Java EE, dan interface-interface server lainnya. Penggunaan umum eksekusi sisi server termasuk adalah akses database, aplikasi e-commerce/e-government, chat room dan lain sebagainya.

ii. Web Statis adalah Aplikasi Berbasis Webdimana informasi yang terkandung di dalamnya tidak bisa diupdate melalui aplikasi Aplikasi Berbasis Webtersebut melainkan harus merubah script yang ada di dalamnya.

jj. Web Dinamis adalah Aplikasi Berbasis Webdimana informasi yang terkandung di dalamnya dapat diupdate melalui aplikasi

I.2 LATAR BELAKANG

Ada 3 (tiga) hal yang melatarbelakangi penyusunan pedoman ini yakni:

1. Adanya kelemahan/kerentanan penanganan keamanan web di Instansi yang secara umum dapat dilihat dari tiga aspek yaitu:
   a. Aspek Manusia (*Peoples*) berupa:
      - Tak ada rasa memiliki dan *sense of crisis* dari pemilik aplikasi/web
      - Minimnya kompetensi dan pelatihan petugas keamanan informasi
   b. Aspek Proses (*Procesess*) berupa:
      - Integrasi keamanan yang buruk ke dalam siklus hidup pengembangan sistem
      - Kurang pemeliharan aplikasi (tak ada *update/patching*)
      - Pemrograman yang tidak aman (*insecure programming*)
      - Konfigurasi buruk (*default/no fine tune/hardening*)
      - Minim monitoring/pengujian rutin (*penetration/performance testing*)
      - Kurang manajemen dengan tidak ada supervisi atau pengawasan/kebijakan/SOP (*careless management*)
      - Respons penanganan insiden yang tidak memadai
   c. Aspek Teknologi (*Technology*) berupa:
      - Desain arsitektur keamanan yang lemah (tak ada lapisan/proteksi)
      - Minim penggunaan teknologi untuk proteksi minimum
2. Ancaman (*threat*) terhadap keamanan yang semakin variatif, kompleks dan maju (*sophisticated*) yang dapat berupa:
   a. Hidden Field Manipulation (Manipulasi field-tersembunyi)
   b. Cookie Poisoning
   c. Parameter Tampering
   d. Buffer Overflow
   e. Cross Site Scripting (XSS)
   f. Backdoor atau Debug Options (Opsi Debug)
   g. Stealth Commanding (Perintah Senyap)
   h. Forced Browsing (Jelajah Paksa)
   i. Third Party Misconfigurations (Miskonfigurasi Pihak Ketiga)

j. Web Defacement

k. Distributed Denial of Service (DDoS)

l. Social Engineering

m. Pharming

n. Phising

o. Dan lain-lain

3. Akibat dari 2 (dua) aspek diatas menimbulkan dampak terhadap:
   a. Terganggunya operasional internal dan pelayanan kepada masyarakat
   b. Pencurian data rahasia/sensitif
   c. Rusak/hilangnya reputasi instansi

I.3 PERUNTUKAN

Pedoman Tata Kelola Keamanan Aplikasi Berbasis Web ini diperuntukkan bagi seluruh instansi di semua level yaitu :

1. Kementerian dan Lembaga
2. Propinsi
3. Kabupaten/Kota

I.4 LINGKUP UMUM

Pedoman Tata Kelola Keamanan Aplikasi Berbasis Web ini berlaku untuk semua jenis web diantaranya:

1. Web Statis
2. Web Dinamis
3. Web Dinamis dengan Aplikasi Transaksional

I.5 TUJUAN

Pedoman Tata Kelola Keamanan Aplikasi Berbasis Web ini digunakan sebagai:

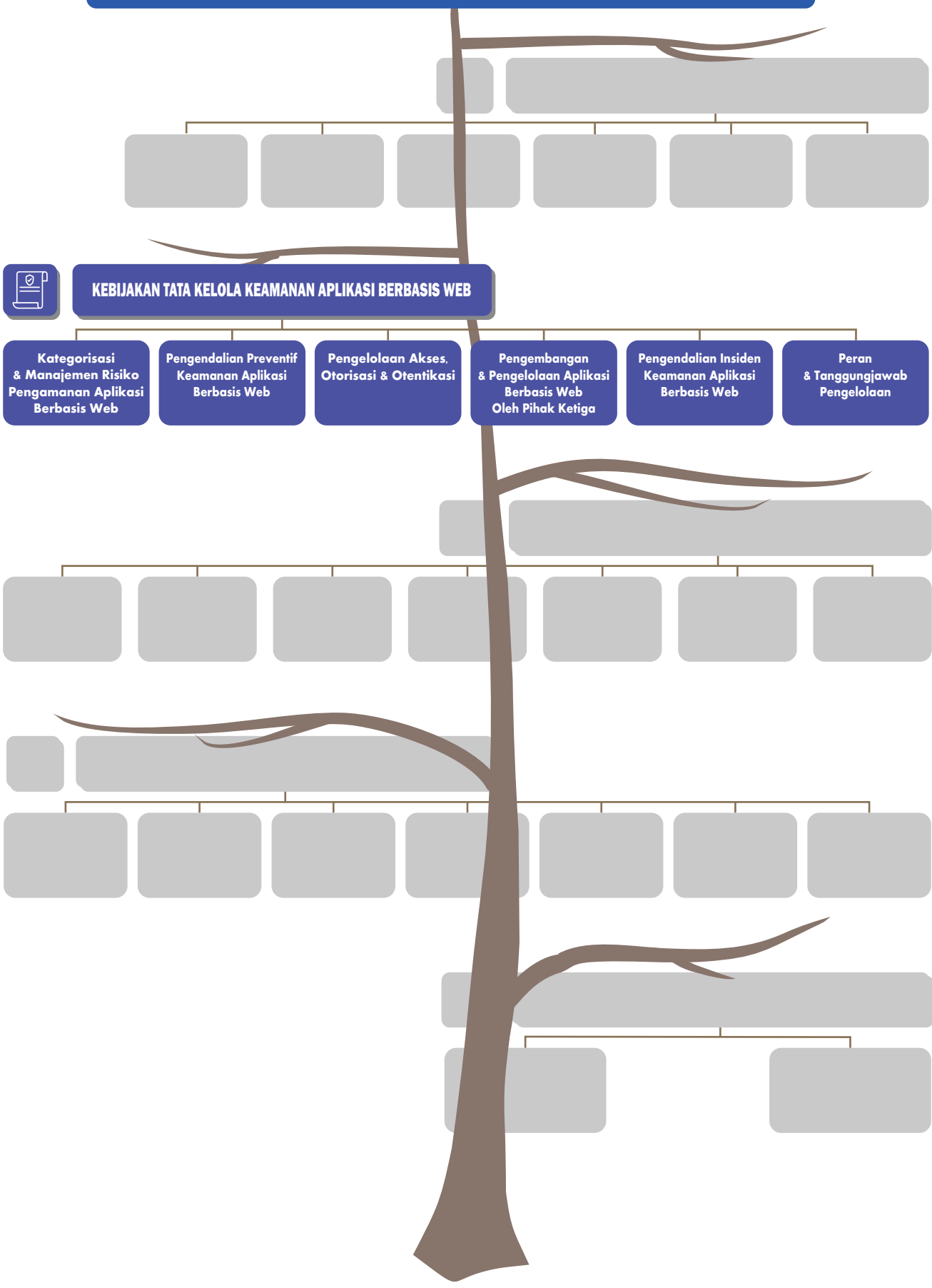
1. Prinsip dan panduan bagi setiap instansi dalam penggunaan sumber daya web di institusi masing-masing, sehingga memenuhi asas keamanan informasi yaitu: *Confidentiality, Availibility, & Integrity*
2. Batasan bagi instansi dan entitas pengambil keputusan di dalamnya dalam pengelolaan keamanan Aplikasi Berbasis Web.

I.6 MANFAAT

Manfaat penerapan Tata Kelola Keamanan Aplikasi Berbasis Web di instansi dapat dilihat dalam 3 perspektif yaitu:

1. Nasional. Untuk level nasional, berikut ini adalah manfaat yang akan dapat dirasakan:
    - Mendapatkan standar rujukan kualitas pengelolaan keamanan Aplikasi Berbasis Web di seluruh instansi.
    - Memudahkan monitoring dan evaluasi pengelolaan keamanan Aplikasi Berbasis Web di seluruh instansi.
2. Institusional. Bagi setiap instansi akan memberikan manfaat:
    - Membangun kesadaran terhadap risiko keamanan di masing-masing lingkungan institusi
    - Mendapatkan batasan dan panduan sesuai best practice dalam pelaksanaan tata kelola keamanan Aplikasi Berbasis Web di lingkungan masing-masing instansi.
    - Meningkatkan kinerja penyelenggaraan TIK di lingkungan instansi masing-masing baik terkait dengan manajemen internal maupun pelayanan publik
    - Menjaga keamanan Aplikasi Berbasis Web di setiap lingkungan instansi agar tetap berfungsi untuk:
        - Menampilkan informasi kelembagaan sesuai tugas pokok dan fungsi
        - Menyediakan informasi kepada publik (keterbukaan publik)
        - Memberikan layanan publik secara daring: pengaduan
        - Membangun keterhubungan (engagement), kerjasama dengan mitra kerja serta pemangku kepentingan yang lebih luas
        - Memberikan sarana diseminasi, sosialisasi, kampanye dan interaksi
3. Publik. Masyarakat diharapkan mendapat manfaat:
    - Kualitas pelayanan publik yang lebih baik.
    - Transparansi/keterbukaan dan tingkat ketersediaan informasi yang disediakan didalam Aplikasi Berbasis Web instansi, sehingga masyarakat dapat melakukan fungsi *social control.*

PEDOMAN TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB



II. KEBIJAKAN TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB

II.1 KATEGORISASI & MANAJEMEN RISIKO PENGAMANAN APLIKASI BERBASIS WEB

II.1.1 KATEGORISASI APLIKASI BERBASIS WEB

Pada umumnya instansi memiliki Aplikasi Berbasis Web dengan kategori sebagai berikut:

1. Web Statis memiliki ciri umum sebagai berikut:
    - Lebih cenderung bersifat informatif dimana pengguna/pengunjung hanya dapat melihat – lihat informasi di Web tersebut, tidak dapat mengisi data.
    - Interaksi yang terjadi antara pengguna dan server hanyalah seputar pemrosesan link saja.
    - Halaman-halaman Web tidak memliki database, data dan informasi tidak berubah-ubah kecuali diubah sintaksnya. Dokumen Web yang dikirim kepada client akan sama isinya dengan yang ada di web server.
    - Untuk menambah halaman harus menambah file baru.
2. Web Dinamis memiliki ciri umum sebagai berikut:
    - Pengguna dapat mengupdate informasi langsung dari Aplikasi Berbasis Webnya.
    - Mengubah tampilan Aplikasi Berbasis Web melalui *Content Management System* (CMS)
    - Menggunakan database yang digunakan untuk menampung banyaknya data, sehingga Aplikasi Berbasis Web tinggal mengambil data dari database.
3. Web Dinamis dengan Aplikasi Transaksional, memiliki ciri umum hampir sama dengan web dinamis tetapi terdapat fasilitas lain sebagai berikut:
    - Akses ke suatu aplikasi yang bersifat publik atau internal instansi (yang dapat diakses melalui internet atau intranet atau menggunakan VPN/MPLS/Leased Line).
    - Umumnya terkait dengan data yang bersifat konfidensial dan sensitif.

II.1.2 MANAJEMEN RISIKO PENGAMANAN APLIKASI BERBASIS WEB

1. Instansi harus menetapkan kajian risiko penyelenggaraan layanan aplikasi Aplikasi Berbasis Web, yang dapat merujuk pada manajemen risiko keamanan informasi yang telah disusun.
2. Unit Pengelola TI harus melakukan identifikasi, analisis, pengukuran, pengendalian dan penetapan rencana penanggulangan risiko.

3. Pengkajian risiko harus mencakup identifikasi dampak karena kelemahan aspek kerahasiaan (*confidentiality*), keutuhan (*integrity*), dan ketersediaan (*availability*) dari aplikasi Aplikasi Berbasis Webserta estimasi penilaian risiko setelah penerapan kontrol. Hasil pengkajian risiko dituangkan dalam dokumen daftar risiko TI (*IT risk register*).
4. Instansi menetapkan bahwa risiko yang dapat diterima adalah risiko yang bernilai "Rendah". Nilai risiko lainnya ("Sedang" dan "Tinggi") harus ditanggulangi dengan perbaikan atau penerapan kontrol baru sehingga nilai risiko setelah penerapan kontrol menjadi "Rendah".
5. Risiko yang tidak diterima akan ditindaklanjuti dengan menyusun rencana penanggulangan risiko (*risk treatment plan*) berupa:
    - Diterima: menerima risiko tanpa perlu menerapkan kontrol tambahan dari yang sudah ditetapkan.
    - Dikurangi: menerapkan kontrol untuk mengurangi risiko
    - Ditransfer: mengalihkan risiko dan pengendaliannya kepada pihak ketiga
    - Dihindari: menghindari penggunaan sumber daya TI yang mendatangkan risiko
6. Dokumen Daftar Risiko TI (*IT Risk Register*) dan rencana penanggulangan risikonya *(Risk Treatment Plan*) harus ditetapkan oleh Pimpinan Instansi sebagai pemilik risiko (*Risk Owner*).
7. Kriteria penilaian risiko dan hasil pengkajian risiko harus ditinjau ulang minimum 1 (satu) tahun sekali atau jika terjadi perubahan signifikan terhadap penyediaan layanan TI, aset TI yang digunakan, atau adanya perubahan peraturan atau kebijakan yang menimbulkan risiko baru.
8. Setiap intansi dapat menentukan pendekatan dalam penilaian risiko yaitu berdasarkan pendekatan yang umum dilakukan (*likelihood dan impact*) atau melalui pendekatan yang rutin dikeluarkan oleh OWASP melalui *OWASP Top 10 - The Ten Most Critical Web Application Security Risks*.

II.1.2.1 JENIS RISIKO KEAMANAN APLIKASI BERBASIS WEB

1. Risiko Reputasi

Opini publik yang negatif dapat timbul karena tampilan Aplikasi Berbasis Web yang sudah diubah oleh pihak yang tidak berwenang (*web deface*) dan kegagalan sistem untuk dapat diakses. Sehingga muncul opini negatif akan ketidakmampuan instansi memberikan dukungan layanan publik pada saat terjadi kegagalan sistem (*downtime*). Opini negatif ini juga dapat menurunkan tingkat kepercayaan publik/masyarakat terhadap instansi tersebut.

2. Risiko Operasional

Risiko operasional melekat di setiap layanan yang disediakan Instansi. Penggunaan aplikasi Aplikasi Berbasis Web dapat menimbulkan terjadinya risiko operasional yang disebabkan oleh ketidakcukupan/ketidaksesuaian

desain, implementasi, pemeliharaan sistem Aplikasi Berbasis Web dan infrastruktur pendukung Aplikasi Berbasis Web, metoda pengamanan, testing dan standar audit eksternal serta penggunaan jasa pihak lain dalam pengembangan dan operasional pengelolaan Aplikasi Berbasis Web.

3. **Risiko Hukum**

   Instansi menghadapi risiko hukum yang disebabkan adanya tuntutan hukum, ketiadaan peraturan perundangan yang mendukung atau kelemahan perikatan seperti tidak dipenuhinya syarat sah suatu kontrak.

4. **Risiko Finansial**

   Instansi menghadapi risiko finansial yang disebabkan hilangnya potensi finansial yang harusnya dapat dihimpun sebagai pendapatan negara atau daerah. Dan juga hilang/rusaknya aset negara yang berdampak pada finansial.

II.1.2.2 KEMUNGKINAN ANCAMAN KEAMANAN APLIKASI BERBASIS WEB

Kemungkinan (*likelihood*) terjadinya ancaman terhadap jenis Aplikasi Berbasis Web di atas dapat dikategorikan sebagai berikut:

Tabel 2.1 Kemungkinan Ancaman Keamanan Aplikasi Berbasis Web

No.	Kemungkinan	Uraian
1.	Hampir pasti	>90% akan terjadi dalam periode waktu satu tahun
2.	Sering (*Likely)*	Antara 50-90% akan terjadi dalam periode waktu satu (1) tahun
3.	Mungkin (*Possible)*	Antara 10-50% akan terjadi dalam periode waktu satu (1) tahun
4.	Jarang (*Rare)*	<10% akan terjadi dalam periode waktu satu (1) tahun

II.1.2.3 TINGKAT ANCAMAN KEAMANAN APLIKASI BERBASIS WEB

Pengendalian (*controlling*) terhadap dampak ancaman dapat dikategorikan menjadi ; tinggi, sedang, rendah yang didefinisikan sebagai berikut:

Tabel 2.2 Tingkat Risiko & Pengendaliannya

Tingkat Risiko	Pengendalian Risiko
Tinggi	Risiko tidak diterima dan perlu ditanggulangi segera (jangka pendek) untuk mengurangi dampak risiko yang ditumbulkannya.
Sedang	Risiko tidak diterima tetapi tidak harus ditanggulangi segera, karena dampaknya dalam jangka pendek belum kritikal atau masih dapat ditoleransi
Rendah	Risiko dapat diterima tetapi harus tetap memelihara efektivitas penerapan kontrol yang ada.

Dari dampak dan kemungkinan terjadinya ancaman keamanan Aplikasi Berbasis Web dapat dipetakan tabel kategori tingkat dampak sebagai berikut:

Tabel 2.3 Dampak Ancaman Keamanan Aplikasi Berbasis Web

KEMUNGKINAN	DAMPAK			
	Sangat Kecil	Ringan	Menengah	Besar
Hampir Pasti	Sedang	Sedang	Tinggi	Tinggi
Sering	Rendah	Sedang	Tinggi	Tinggi
Mungkin	Rendah	Sedang	Sedang	Tinggi
Jarang	Rendah	Rendah	Sedang	Tinggi

Berikut contoh penentuan tingkat risiko berdasarkan kategori web :

Tabel 2.4 Dampak Gangguan/Ancaman Keamanan Aplikasi Berbasis Web

Kategori	Sangat Kecil	Ringan	Menengah	Berat
1. Gangguan/Ancaman terhadap web statis				
• Gangguan terhadap Web statis	Web tidak dapat diakses <1 jam di luar jam kerja	Web tidak berfungsi > 1 – 4 jam selama jam kerja	Web tidak berfungsi > 4 – 24 jam selama jam kerja	Tampilan Web berubah signifikan terhadap tampilan sebelumnya (web deface)
2. Gangguan/Ancaman terhadap web dinamis				
• Gangguan/Ancaman terhadap web dinamis	Web tidak dapat diakses <1 jam di luar jam kerja	Web tidak berfungsi > 1 – 4 jam selama jam kerja	Web tidak berfungsi > 4 – 24 jam selama jam kerja	• Tampilan Web berubah signifikan terhadap tampilan sebelumnya (web deface). • Web tidak berfungsi lebih dari 24 jam selama jam kerja.
• Keluhan user	Keluhan kecil dan tidak signifikan	Keluhan dialami dan disampaikan oleh sejumlah pengguna	Keluhan dimuat di media lokal	Keluhan dimuat di media lokal dan nasional

3. Gangguan/Ancaman terhadap aplikasi web untuk transaksional				
• Gangguan/ancaman terhadap aplikasi-aplikasi web pelayanan publik	Aplikasi tidak dapat diakses <1 jam di luar jam kerja	Aplikasi tidak berfungsi > 1 – 4 jam selama jam kerja	Aplikasi tidak berfungsi > 4 – 24 jam selama jam kerja	• Tampilan Aplikasi Berbasis Webberubah signifikan terhadap tampilan sebelumnya (web deface). • Aplikasi tidak berfungsi lebih dari 24 jam selama jam kerja.
• Gangguan/ancaman terhadap aplikasi-aplikasi web internal	Sistem tidak dapat diakses <1 jam di luar jam kerja	Sistem tidak berfungsi > 1 – 4 jam selama jam kerja	Sistem tidak berfungsi > 4 – 24 jam selama jam kerja	Sistem tidak berfungsi lebih dari 24 jam selama jam kerja.
• Keluhan user	Keluhan kecil dan tidak signifikan	Keluhan dialami dan disampaikan oleh sejumlah pengguna	Keluhan dimuat di media lokal	Keluhan dimuat di media lokal dan nasional.
• Kebocoran/Kehilangan/ kerusakan data	Tidak ada kehilangan data	Ada kebocoran /kehilangan/kerusakan data tetapi bukan data sensitive/rahasia	Ada kebocoran /kehilangan/kerusakan data sensitif/rahasia dan berdampak signifikan secara internal	Ada kebocoran /kehilangan/kerusakan data sensitif berdampak signifikan secara nasional.

II.1.2.4 RISIKO KERENTANAN KEAMANAN APLIKASI BERBASIS WEB BERDASARKAN OWASP

OWASP (*Open Web Application Security Project*) merupakan organisasi nirlaba internasional yang mempunyai visi untuk menjaga keamanan cyber termasuk Aplikasi Berbasis Web. OWASP memastikan bahwa semua informasi dan materi pembelajarannya dapat diakses dengan mudah dan gratis sehingga semua pihak dapat meningkatkan keamanan Aplikasi Berbasis Web mereka. OWASP menentukan tingkat risiko atas setiap jenis kerentanan keamanan Aplikasi Berbasis Web sesuai dengan tabel berikut ini.

Tabel 2.5 Risiko Kerentanan Keamanan Aplikasi Berbasis Web Berdasarkan OWASP

Dapat Dieksploitasi	Prevalensi Kelemahan	Kelemahan Deteksi	Dampak Teknis
Mudah: 3	Tersebar Luas: 3	Mudah: 3	Parah: 3
Rata - rata: 2	Umum: 2	Rata - rata: 2	Sedang: 2
Sulit: 1	Tidak Umum: 1	Sulit: 1	Kecil: 1

Tabel 2.6 Identifikasi Tingkat Risiko atas Kerentanan Keamanan Aplikasi Berbasis Web

Tingkat Risiko	Ancaman/ Kerentanan	Penjelasan	Kendali Preventif
Dapat Diekploitasi: 3 Prevalensi Kelemahan: 2 Kelemahan Deteksi: 3	Injeksi	Kelemahan injeksi, seperti injeksi SQL, OS, dan LDAP, terjadi ketika data yang tidak dapat dipercaya dikirim ke suatu *interpreter* sebagai bagian dari suatu perintah atau *query*. Data berbahaya dari penyerang tersebut dapat mengelabui *interpreter* untuk mengeksekusi perintah yang tidak direncanakan, atau untuk mengakses data yang tidak terotorisasi.	• Memverifikasi bahwa semua penggunaan *interpreter* secara tegas harus memisahkan data yang tidak dapat dipercaya dari perintah atau *query*. Untuk *SQL calls*, menggunakan *bind variables* dalam semua *prepared statements* dan *stored procedures*, serta menghindari *dynamic queries*. • Memeriksa kode adalah cara cepat dan akurat untuk melihat apakah suatu aplikasi menggunakan *interpreter* yang aman. Perangkat untuk analisis kode dapat membantu analis keamanan mencari penggunaan *interpreter* dan melacak aliran data yang melalui aplikasi. Penguji penetrasi dapat memvalidasi isu-isu ini dengan membuat eksploitasi yang mengkonfirmasi kerentanan ini. • Pemindaian dinamis - otomatis yang menguji aplikasi dapat memberikan gambaran mengenai

Dampak Teknis: 3			keberadaan cacat injeksi yang dapat dieksploitasi. Pemindai tidak selalu dapat mencapai *interpreter,* dan memiliki kesulitan mendeteksi suatu serangan. *Error handling* yang buruk membuat cacat injeksi semakin mudah ditemukan. • Menggunakan API yang aman dengan menghindari penggunaan interpreter secara keseluruhan atau menyediakan interface yang berparameter. Selau waspada terhadap API, seperti stored procedures, yang meskipun berparameter, namun masih tetap dapat menimbulkan injeksi. • Mewaspadai dalam meloloskan karakter-karakter khusus yang menggunakan *escape syntax (routines*) khusus untuk interpreter. • Validasi input positif atau "daftar putih" ("*white list*") dengan kanonikalisasi yang tepat, tetapi bukan merupakan pertahanan yang lengkap karena banyak aplikasi membutuhkan karakter-karakter khusus dalam inputnya
Dapat Diekploitasi: 3 Prevalensi Kelemahan: 2	*Broken Authentication*	Adanya kelemahan otentikasi karena aplikasi: • Mengizinkan serangan otomatis seperti isian kredensial, dimana penyerang memiliki daftar nama pengguna dan kata sandi yang valid. • Mengizinkan kata sandi yang masih standar, lemah, atau terkenal, seperti "Kata Sandi1" atau "admin/admin".	• Menerapkan otentikasi multi-faktor untuk mencegah isian otomatis, kredensial, brute force, dan serangan karena penggunaan ulang kredensial yang dicuri. • Tidak mengirim atau menggunakan kredensial default apapun, terutama untuk pengguna admin. • Pengecekan kata sandi yang lemah, seperti menguji kata sandi baru atau yang diubah terhadap daftar 10.000 kata sandi terburuk.

Kelemahan Deteksi: 2 Dampak Teknis: 3		• Menggunakan pemulihan kredensial yang masih lemah atau tidak efektif serta proses ketika lupa kata sandi, seperti "jawaban berbasis pengetahuan", yang tidak dapat dibuat aman. • Menggunakan teks biasa, terenkripsi, atau kata sandi dengan hash yang lemah • Mengekspos ID Sesi di URL (mis., Penulisan ulang URL). • Tidak merotasi ID Sesi setelah berhasil masuk. • Tidak benar membatalkan ID Sesi. • Sesi pengguna atau token autentikasi (terutama SSO) tidak divalidasi dengan benar selama logout atau periode tidak aktif.	• Mengikuti standar panjang kata sandi, kompleksitas, dan kebijakan rotasi • Membatasi upaya login yang gagal. mencatat semua kegagalan dan memberitahukan administrator ketika isian kredensial, brute force, atau serangan lainnya terdeteksi. • ID sesi tidak boleh ada di URL, tetapi disimpan dengan aman, dan tidak valid setelah logout, idle, dan timeout absolut.
Dapat Diekploitasi: 3 Prevalensi Kelemahan: 2	*Sensitive Data Exposure*	Adanya kelemahan karena: • Data yang dikirim masih dalam teks yang jelas (clear text), yang umumnya menggunakan protokol seperti HTTP, SMTP, dan FTP • Data sensitif disimpan tanpa enkripsi termasuk data backup Masih adanya penggunaan	• Klasifikasi data yang diproses, disimpan, atau dikirim oleh suatu aplikasi. Identifikasi data mana yang sensitif menurut undang-undang privasi, persyaratan peraturan, atau kebutuhan instansi • Menerapkan kontrol sesuai klasifikasi. Tidak menyimpan data sensitif jika betul - betul tidak diperlukan. Membuang sesegera mungkin atau

Kelemahan Deteksi: 2 Dampak Teknis: 3		algoritma kriptografi lama yang lemah yang digunakan secara default • Apakah enkripsi tidak ditegakkan, mis. apakah ada arahan atau header keamanan agen pengguna (browser) yang hilang? Apakah agen pengguna (mis. Aplikasi, klien email) tidak memverifikasi apakah sertifikat server yang diterima valid?	menggunakan tokenisasi sesuai dengan standar PCI DSS • Mengenkripsi semua data sensitif saat idle. • Memastikan algoritma, protokol, dan kunci standar terkini dan kuat • Menggunakan manajemen kunci yang tepat. • Enkripsi semua data pengiriman dengan protokol yang aman seperti TLS dengan sandi kerahasiaan, prioritas sandi oleh server, dan parameter aman. • Menerapkan enkripsi menggunakan direktif seperti HTTP Strict Transport Security (HSTS). • Menonaktifkan caching untuk respons yang berisi data sensitif. • Menyimpan kata sandi menggunakan fungsi hashing adaptif dan kuat dengan faktor kerja (faktor penundaan), seperti Argon2, scrypt, bcrypt, atau PBKDF2. • Memverifikasi secara independen efektivitas konfigurasi dan pengaturan.
Dapat Diekploitasi: 3 Prevalensi Kelemahan: 3 Kelemahan Deteksi: 3 Dampak Teknis: 2	*Cross-Site Scripting* (XSS)	Kelemahan XSS terjadi ketika aplikasi mengambil data yang tidak dapat dipercaya dan mengirimnya ke suatu *web browser* tanpa validasi yang memadai. XSS memungkinkan penyerang mengeksekusi *script-script* di dalam *browser* korban, yang dapat membajak sesi pengguna, mengubah tampilan *Aplikasi Berbasis Web*, atau mengarahkan	• Menyaring semua data yang tidak dapat dipercaya dengan tepat berdasarkan konteks HTML (body, atribut, JavaScript, CSS, atau URL) tempat diletakkannya data. Para pengembang perlu menyertakan penyaringan ini dalam aplikasi mereka • Validasi input positif (*whitelist*) dengan kanonikalisasi dan decoding yang tepat sehingga dapat membantu melindungi dari XSS; tetapi itu bukan pertahanan yang menyeluruh sehingga perlu

		pengguna ke situs-situs jahat.	memvalidasi sebanyak mungkin untuk aplikasi yang membutuhkan karakter khusus dalam input mereka. mendekodekan setiap encoded-input, lalu memvalidasi panjang, karakter, format, dan setiap aturan bisnis pada data sebelum menerima input tersebut.
Dapat Diekploitasi: 3 Prevalensi Kelemahan: 2 Kelemahan Deteksi: 3 Dampak Teknis: 3	*XML External Entities (XXE)*	Adanya kelemahan karena: ▪ Aplikasi menerima XML secara langsung atau unggahan XML, terutama dari sumber yang tidak terpercaya, atau menyisipkan data yang tidak tepercaya ke dalam dokumen XML, yang kemudian diurai oleh prosesor XML. ▪ Setiap prosesor XML dalam aplikasi atau layanan Aplikasi Berbasis Webberbasis SOAP memiliki definisi tipe dokumen (DTD) yang diaktifkan. Karena mekanisme yang tepat untuk menonaktifkan pemrosesan DTD bervariasi menurut prosesor. ▪ Aplikasi menggunakan SAML untuk pemrosesan identitas dalam Single Sign On (SSO). Mungkin akan rentan jika SAML menggunakan XML untuk pernyataan identitas. ▪ Aplikasi menggunakan SOAP sebelum versi 1.2, kemungkinan rentan terhadap serangan XXE jika entitas XML	▪ Menggunakan format data yang kurang kompleks seperti JSON, dan hindari serialisasi data sensitif. ▪ Menambal atau memutakhirkan semua prosesor dan perpustakaan XML yang digunakan oleh aplikasi atau pada sistem operasi yang mendasarinya. Gunakan checker dependensi. ▪ Memperbaharui versi SOAP ke versi 1.2 atau lebih tinggi. ▪ Menonaktifkan entitas eksternal XML dan pemrosesan DTD di semua parser XML dalam aplikasi.. ▪ Memvalidasi input sisi-server, penyaringan, atau sanitasi untuk mencegah data yang bermusuhan dalam dokumen XML, header, atau node. ▪ Memverifikasi bahwa fungsionalitas unggah file XML atau XSL memvalidasi XML yang masuk menggunakan validasi XSD atau serupa. Alat SAST dapat membantu mendeteksi XXE dari kode sumber, meskipun tinjauan kode manual adalah alternatif terbaik dalam aplikasi yang besar dan kompleks dengan banyak integrasi. ▪ Jika kontrol - kontrol diatas tidak memungkinkan, perlu dipertimbangkan untuk menggunakan tambalan virtual,

		diteruskan ke kerangka kerja SOAP.	gateway keamanan API, atau malah Web Application Firewalls (WAFs) untuk mendeteksi, memantau, dan memblokir serangan XXE.
Dapat Diekploitasi: 2 Prevalensi Kelemahan: 3 Kelemahan Deteksi: 3 Dampak Teknis: 2	Cross-Site Request Forgery (CSRF)	Suatu serangan CSRF memaksa *browser* korban yang sudah log-on untuk mengirim *HTTP request* yang dipalsukan, termasuk di dalamnya *session cookie* korban dan informasi otentikasi lain yang otomatis disertakan, ke suatu aplikasi Aplikasi Berbasis Web yang rentan. Hal ini memungkinkan penyerang untuk memaksa *browser* korban menghasilkan *request* yang dianggap sah oleh aplikasi rentan tadi.	• Membutuhkan penyertaan unpredictable token dalam body atau URL setiap permintaan HTTP. Token tersebut harus unik untuk setiap sesi pengguna, atau juga untuk setiap permintaan. • Menyertakan token unik dalam field tersembunyi. Hal ini membuat nilainya dikirim dalam tubuh permintaan HTTP, sehingga tidak ada di dalam URL, yang rentan terekspos. • Token unik dapat juga disertakan dalam URL, atau parameter URL. Namun, penempatan tersebut berisiko karena URL akan terekspos ke penyerang, karenanya mengungkap token rahasia.
Dapat Diekploitasi: 3 Prevalensi Kelemahan: 3 Kelemahan Deteksi: 3 Dampak Teknis: 2	Kesalahan Konfigurasi Keamanan	Keamanan yang baik mensyaratkan dimilikinya suatu konfigurasi keamanan yang terdefinisi dan diterapkan untuk aplikasi, *framework*, server aplikasi, *web server*, *server* database, dan platform. Semua pengaturan ini harus didefinisikan, diimplementasikan,dan dipelihara, karena terdapat banyak aplikasi yang dirilis tanpa konfigurasi *default* yang aman. Hal ini juga mencakup menjaga semua software *up-to-date,*	• Lingkungan pengembangan, Quality Assurance/QA, dan produksi harus dikonfigurasi secara identik. Proses ini harus dilakukan untuk mensetup lingkungan baru yang aman. • Proses untuk memudahkan update dan men-deploy seluruh software update dan patch secara cepat ke lingkungan operasi. Hal ini mencakup juga seluruh pustaka kode, yang seringkali diabaikan. • Arsitektur aplikasi yang kuat yang menyediakan pemisahan dan

		termasuk semua pustaka kode yang digunakan aplikasi tersebut.	keamanan yang tegas antar komponen. • Menjalankan scanning dan melakukan audit secara periodik untuk membantu mendeteksi kesalahan konfigurasi atau patch yang hilang di masa mendatang.
Dapat Diekploitasi: 3 Prevalensi Kelemahan: 2 Kelemahan Deteksi: 2 Dampak Teknis: 3	*Insecure Deserialization*	• Aplikasi dan API akan rentan jika mereka membatalkan deserialisasi objek yang dirusak atau dipasok oleh penyerang. • Serangan terhadap Obyek dan struktur data tdi mana penyerang memodifikasi logika aplikasi atau eksekusi kode jarak jauh yang tidak diinginkan. • Serangan pengrusakan data yang umum, seperti serangan kontrol akses, di mana struktur data yang ada digunakan tetapi kontennya diubah. • Serialisasi dapat digunakan dalam aplikasi untuk: Komunikasi jarak jauh dan antar proses (RPC / IPC) Protokol kawat, layanan Aplikasi Berbasis Web, broker pesan Caching / Kegigihan Database, server cache, sistem file Cookie HTTP, parameter bentuk HTML, token otentikasi API	• Melakukan pemeriksaan integritas seperti tanda tangan digital pada objek berseri untuk mencegah pembuatan objek yang berbeda atau gangguan data. • Menjaga batasan tipe objek secara ketat selama deserialisasi sebelum pembuatan objek karena kode biasanya mengharapkan sekumpulan kelas yang dapat didefinisikan. • Mengisolasi dan menjalankan kode yang deserialize di lingkungan privilege rendah. Pengecualian deserialisasi dan kegagalan dimana jika terjadi deserialisasi dapat dilempar ke pengecualian. • Membatasi atau memantau konektivitas jaringan keluar masuk dari server yang deserialisasi. • Pemantauan deserialisasi secara terus menerus.

Dapat Diekploitasi: 2 Prevalensi Kelemahan: 2 Kelemahan Deteksi: 2 Dampak Teknis: 3	Kegagalan Membatasi Akses	Banyak aplikasi Aplikasi Berbasis Webmemeriksa hak akses URL sebelum memberikan *link* dan tombol-tombol yang diproteksi. Bagaimanapun juga, aplikasi perlu melakukan pemeriksaan kendali akses yang serupa setiap kali halaman-halaman ini diakses, atau penyerang akan dapat memalsukan URL untuk mengakses halaman-halaman yang tersembunyi	• Kebijakan otentikasi dan otorisasi dibuat berbasis-peran, untuk meminimalisasi upaya yang dibutuhkan untuk memelihara kebijakan tersebut. • Kebijakan tersebut harus dapat dikonfigurasi, dalam rangka meminimalisasi berbagai aspek hard code kebijakan itu. • Mekanisme penegakan kebijakan harus secara baku menolak semua akses, mensyaratkan dikabulkannya secara eksplisit pemberian akses pada pengguna dan peran tertentu ke setiap halaman. • Jika halaman tersebut sedang terlibat dalam suatu alur kerja, harus memastikan bahwa kondisi-kondisinya ada dalam keadaan yang tepat untuk memperkenankan akses.
Dapat Diekploitasi: 2 Prevalensi Kelemahan: 3	*Using Components with Known Vulnerabilities*	Adanya kelemahan karena: • Tidak mengetahui versi semua komponen yang digunakan (sisi klien dan sisi server). • Perangkat lunak yang sudah expired termasuk OS, server web / aplikasi, sistem manajemen basis data (DBMS), aplikasi, API dan semua komponen, lingkungan runtime, dan library	• Menghapus dependensi yang tidak digunakan, fitur yang tidak perlu, komponen, file, dan dokumentasinya. • Menginventarisir versi komponen baik disisi klien maupun server secara berkelanjutan (mis. Kerangka kerja, pustaka) dan ketergantungannya menggunakan alat seperti Dependency Check, retire.js, dll. • Menggunakan alat analisis komposisi perangkat lunak untuk mengotomatiskan proses. Berlangganan peringatan email

Kelemahan Deteksi: 2 Dampak Teknis: 2		• Tidak memindai kerentanan secara teratur dan berlangganan buletin keamanan terkait dengan komponen yang digunakan. • Tidak secara teratur memperbaiki atau meningkatkan platform, kerangka kerja, dan dependensi secara tepat waktu berbasis risiko. Ini biasanya terjadi di lingkungan ketika penambalan dilakukan bulanan atau triwulanan yang membuat kondisi software terbuka untuk berhari-hari atau berbulan-bulan. • Pengembang perangkat lunak tidak menguji dan memperbaharui, kompatibilitas library.	untuk kerentanan keamanan terkait dengan komponen yang digunakan. • Mendapatkan komponen dari sumber yang resmi melalui tautan aman. Direkomendasikan mendapatkan paket yang ditandatangani untuk mengurangi kemungkinan masuknya komponen jahat yang dimodifikasi. • Memonitor library dan komponen yang tidak dirawat atau tidak membuat tambalan keamanan dengan versi yang lebih lama. • Jika penambalan tidak memungkinkan, perlu dipertimbangkan untuk menggunakan tambalan virtual untuk memantau, mendeteksi, atau melindungi terhadap masalah yang ditemukan. • Instansi harus memastikan bahwa ada rencana yang berkelanjutan untuk memantau, menentukan prioritas, dan menerapkan pembaruan atau perubahan konfigurasi selama masa aplikasi/software.
Dapat Diekploitasi: 2 Prevalensi Kelemahan: 3	*Insufficient Logging & Monitoring*	Adanya kelemahan karena: ▪ Peristiwa yang dapat diaudit, seperti login, gagal login, dan transaksi bernilai tinggi tetapi tidak tercatat. ▪ Peringatan dan kesalahan menghasilkan pesan log yang tidak memadai, atau tidak jelas.	• Memastikan semua login, kegagalan kontrol akses, dan kegagalan validasi input sisi server dicatat dimana dari sisi pengguna cukup mengidentifikasi akun mencurigakan atau berbahaya, dan ditahan untuk waktu yang cukup untuk memungkinkan analisis forensik. • Memastikan bahwa log dihasilkan dalam format yang mudah dikonsumsi oleh manajemen log terpusat.

Kelemahan Deteksi: 2 Dampak Teknis: 3		▪ Log aplikasi dan API tidak dipantau untuk aktivitas yang mencurigakan. ▪ Log hanya disimpan secara lokal. ▪ Ambang waspada yang tepat dan proses eskalasi respons tidak ada atau tidak efektif. ▪ Pengujian penetrasi dan pemindaian oleh alat DAST (seperti OWASP ZAP) tidak memicu peringatan. ▪ Aplikasi tidak dapat mendeteksi, meningkatkan, atau memperingatkan serangan aktif secara real time atau mendekati waktu nyata.	• Memastikan transaksi dengan volume dan frekuensi tinggi memiliki jejak audit dengan kontrol integritas untuk mencegah gangguan atau penghapusan. • Melakukan pemantauan dan peringatan yang efektif sehingga kegiatan yang mencurigakan terdeteksi dan ditanggapi secara tepat waktu. • Menetapkan atau mengadopsi respons insiden dan rencana pemulihan. • Ada kerangka kerja perlindungan aplikasi komersial dan open source seperti OWASP AppSensor, WAF seperti ModSecurity dengan OWASP ModSecurity Core Rule Set atau SIEM

II.2 PENGENDALIAN PREVENTIF KEAMANAN APLIKASI BERBASIS WEB

Pengendalian keamanan yang bersifat preventif sangat penting untuk meminimalisasi kerentanan (*vulnerability*) dan meningkatkan level ketahanan dari aplikasi Aplikasi Berbasis Web yang dimiliki oleh instansi terhadap ancaman serangan. Setiap lubang keamanan (*security holes*) pada aplikasi Aplikasi Berbasis Web beserta infrastruktur pendukungnya sangat berpotensi untuk masuknya ancaman yang akan menyerang aplikasi Aplikasi Berbasis Web. Ancaman serangan akan selalu mencari titik kerentanan yang paling lemah, oleh karena itu harus ada persyaratan minimum dalam pengendalian preventif pengamanan Aplikasi Berbasis Web.

II.2.1 UPAYA PREVENTIF TERHADAP KEAMANAN KONTEN APLIKASI BERBASIS WEB

II.2.1.1 INFORMASI/DATA YANG TIDAK DAPAT DIPUBLIKASI

Setiap instansi harus memiliki daftar informasi/data yang tidak dapat dipublikasikan melalui Aplikasi Berbasis Web diantaranya:

1. Informasi atau rekaman (*record*) yang sensitif dan berklasifikasi.
2. Aturan dan prosedur personil internal.

3. Informasi pribadi tentang para personil dan klien Instansi.
4. Nomor telepon, alamat e-mail, atau daftar umum dari staf kecuali jika diperlukan untuk memenuhi persyaratan Instansi. Ketika suatu alamat e-mail harus dipublikasikan pada suatu situs Aplikasi Berbasis Web, harus mempertimbangkan penggunaan alamat e-mail umum atau alias (misalnya webmaster@namadomain.go.id sebagai pengganti badu@namadomain.go.id).
5. Jadwal para pemimpin instansi atau lokasi aktivitasnya.
6. Informasi tentang komposisi atau persiapan materi yang sensitif.
7. Informasi sensitif yang berkaitan dengan keamanan nasional.
8. Catatan investigasi.
9. Catatan-catatan keuangan, diluar yang sudah tersedia untuk publik.
10. Catatan-catatan medis.
11. Prosedur keamanan fisik dan informasi dari instansi.
12. Informasi tentang jaringan dan infrastruktur sistem informasi dari instansi.
13. Informasi yang berimplikasipada kerentanan keamanan.
14. Rencana-rencana, peta-peta, diagram-diagram, foto udara, dan rencana arsitektural pembangunan, properti, atau instalasi Instansiyang bersifat sensitif dan strategis.
15. Informasi tentang rencana pemulihan bencana, atau rencana kelanjutan operasi kecuali yang mutlak diperlukan.
16. Rincian prosedur tanggap darurat, rute evakuasi, atau personil Penanggungjawab Instansi untuk persoalan-persoalan tersebut.
17. Materi hak cipta tanpa ijin tertulis dari pemilik.

II.2.1.2 PERSYARATAN PUBLIKASI INFORMASI/DATA

Untuk menjaga kerahasian data/informasi instansi maka harus:

1. Mengidentifikasi informasi yang semestinya dipublikasikan pada Aplikasi Berbasis Web.
2. Tidak menempatkan informasi yang *proprietary*, terklasifikasi, dan informasi sensitif pada suatu Web server yang dapat diakses, kecuali jika dilakukan perlindungan informasi dengan otentikasi pengguna dan enkripsi.
3. Mengidentifikasi sasaran audiens.
4. Mengidentifikasi dampak negatif yang memungkinkan dari publikasi informasi.

5. Mengidentifikasi siapa yang harus bertanggungjawab untuk pembuatan, publikasi, dan pemeliharaan informasi sensitif.
6. Membuat atau menyusun informasi untuk publikasi Aplikasi Berbasis Web.
7. Meninjau kembali informasi dalam hal sensitifitas dan kontrol distribusi/pengeluaran (termasuk sensifitas informasi secara keseluruhan).
8. Menentukan akses yang tepat dan kontrol keamanan.
9. Memverifikasi informasi yang dipublikasikan.
10. Secara berkala meninjau kembali informasi yang dipublikasikan untuk mengkonfirmasi kepatuhan berkelanjutan terhadap ketentuan yang berlaku.

II.2.1.3 MENGURANGI SERANGAN TIDAK LANGSUNG TERHADAP KONTEN

1. Serangan Phising. Untuk meminimalisasi dampak dari serangan phising, maka setiap instansi harus:
   a. Memastikan kewaspadaan pengguna akan bahaya serangan *phishing* dan menghindarinya dengan cara:
      - Tidak membalas pesan email atau iklan *popup* yang meminta informasi pribadi atau keuangan.
      - Tidak mempercayai nomor telepon dalam email atau iklan *popup*
      - Menggunakan antivirus, anti-spyware, dan perangkat lunak firewall. Perangkat lunak tersebut dapat mendeteksi malware dalam suatu mesin pengguna yang terindikasi serangan phishing.
      - Tidak mengirimkan informasi pribadi atau keuangan lewat email.
      - Meninjau kembali pernyataan kartu kredit dan nomer rekening bank secara teratur.
      - Berhati-hati dalam mengakses situs Aplikasi Berbasis Web yang tidak dipercaya karena beberapa kerentanan Web browser dapat dieksploitasi hanya dengan mengunjungi situs semacam itu.
      - Berhati-hati dalam membuka suatu attachment atau men-download file apapun dari email atau situs web yang tidak dipercaya.
   b. Memvalidasi komunikasi resmi dengan mempersonalkan email (membuat email menurut selera) dan menyediakan informasi identifikasi yang unik yang seharusnya hanya diketahui oleh instansi dan pengguna.

c. Menggunakan signature pada e-mail sekalipun signature tidak dapat divalidasi secara otomatis oleh aplikasi e-mail pengguna.

d. Menjalankan validasi konten dalam lingkup aplikasi Aplikasi Berbasis Web. Kerentanan dalam aplikasi Aplikasi Berbasis Web mungkin digunakan dalam suatu serangan phishing.

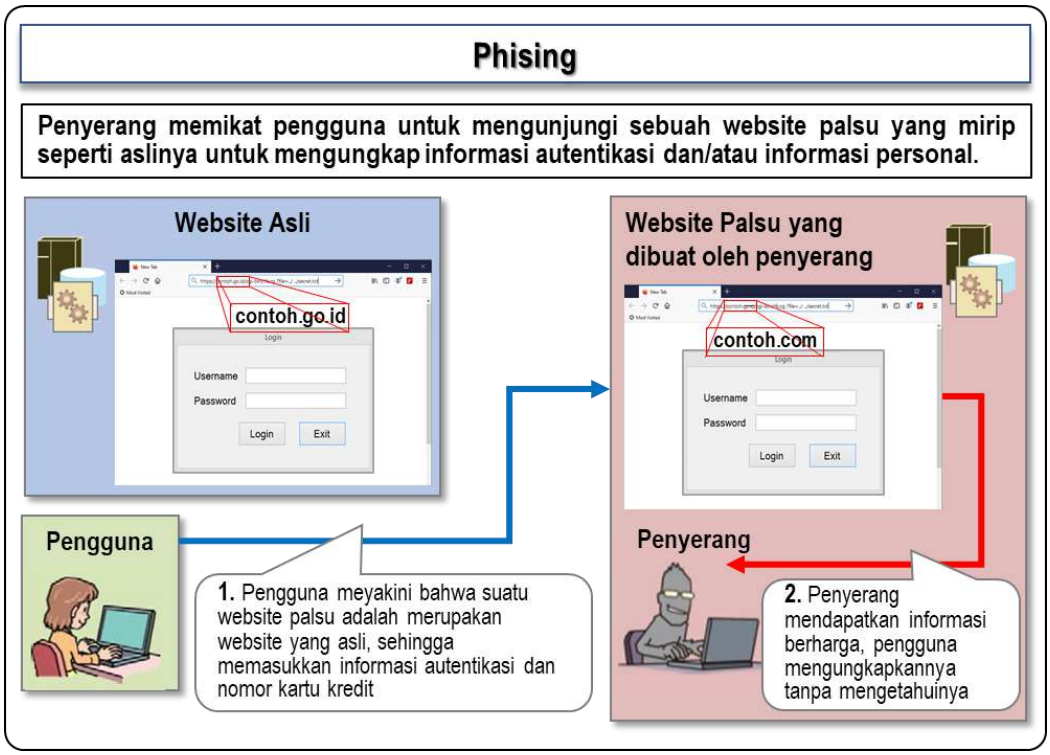
e. Membuat konten Aplikasi Berbasis Web tambahan sebagai identifikasi atas situs web yang sah.

f. Menggunakan otentikasi berbasis token atau otentikasi dua arah (mutual authentication) pada situs web untuk mencegah para pelaku phising dari penggunaan kembali informasi.

g. Situs Web yang menyimpan *Personally Identifiable Information*/PII harus mempertimbangkan dengan cermat untuk menerapkan anti-phishing yang lebih kuat.

h. Situs Web dengan informasi publik yang tidak sensitif tidak perlu mengimplementasikan anti-phishing canggih yang menghabiskan banyak biaya.

Gambar 2.1 Visualisasi Mekanisme Phising



2. Serangan Pharming. Untuk meminimalisasi dampak dari serangan pharming, maka setiap instansi harus:

a. Menggunakan versi terkini perangkat lunak DNS yang mengaplikasikan *patches* keamanan terkini—Suatu server DNS yang bobol akan mengijinkan penyerang untuk mengarahkan pengguna kepada suatu server yang berniat jahat sambil tetap mempertahankan suatu nama DNS yang resmi.

b. Mengimplementasikan Proteksi Server-Side DNS Terhadap Pharming—Ada beberapa tool yang tersedia untuk mengurangi ancaman terhadap perangkat lunak DNS, seperti DNS Security Extensions

c. Mengawasi domain instansi dan registrasi domain-domain yang serupa—Serangan-serangan pharming mungkin mengambil keuntungan dari para pengguna yang salah mengeja nama domain Instansi saat mengakses situs.

d. Menyederhanakan struktur dan jumlah nama domain instansi—Struktur penamaan domain bagi pemerintah mengikuti ketentuan yang berlaku berbasis domain go.id. Namun banyak domain pemerintah yang telah menyederhanakan struktur domainnya seperti misalnya http://www.pajak.go.id

e. Menggunakan koneksi yang aman (yaitu, HTTPS) untuk login, yang dapat memverifikasi keabsahan sertifikat server dan keterkaitannya dengan suatu situs web yang sah—browser yang modern akan memberitahukan seorang pengguna jika nama DNS tidak sesuai dengan yang diberikan oleh sertifikat, namun beberapa situs pharming dapat memiliki suatu sertifikat yang sah.

f. Menggunakan Pre-shared Secret— Pre-shared secret dapat digunakan untuk melawan serangan pharming. Suatu implementasi yang umum dari pre-shared secret adalah agar para pengguna yang sah membuat pertanyaan tertentu dan menjawab dengan apa yang seharusnya mereka tahu.

II.2.1.4 PROTEKSI KONTEN AKTIF (*ACTIVE CONTENT*)

Ketika memeriksa atau menulis suatu konten aktif yang dapat dieksekusi atau berupa script, harus mempertimbangkan hal-hal berikut:

1. Kode yang dapat dieksekusi sebaiknya sesederhana mungkin. Semakin panjang atau kompleks suatu kode sangat dimungkinkan memiliki masalah.
2. Kemampuan kode yang dapat dieksekusi untuk membaca dan menulis program sebaiknya dibatasi. Kode yang membaca file memungkinkan melanggar pembatasan akses atau meneruskan informasi sistem yang sensitif. Kode yang menulis file dapat memodifikasi atau merusak dokumen atau memunculkan *trojan horse*.
3. Interaksi kode dengan program atau aplikasi lain sebaiknya dianalisis untuk mengidentifkasi kerentanan keamanan. Sebagai contoh, banyak script CGI mengirim email sebagai tanggapan atas input formulir dengan membuka suatu koneksi dengan program sendmail. Memastikan interaksi ini dijalankan dengan cara yang aman.
4. Pada host Linux/Unix, kode sebaiknya tidak berjalan dengan suid (set-user-id).
5. Kode harus menggunakan nama path eksplisit ketika meminta program eksternal. Tidak direkomendasikan penggunaan variabel lingkungan path untuk menyelesaikan nama-nama path parsial.

6. Kode pembangkit konten Aplikasi Berbasis Web sebaiknya discan dan/atau diaudit (tergantung pada sensitifitas dari Web server dan kontennya). Tool yang ada secara komersil dapat men-scan .NET atau kode Java. Sejumlah entitas komersil menawarkan layanan peninjauan kembali kode.
7. Kode pembangkit konten Aplikasi Berbasis Web sebaiknya dikembangkan mengikuti keamanan terkini.
8. Untuk formulir data entry, tentukan suatu daftar karakter yang diharapkan dan memfilter karakter yang tidak diharapkan dari data input yang dimasukkan oleh seorang pengguna sebelum memproses suatu formulir. Sebagai contoh, pada sebagian formulir, data yang diharapkan digolongkan dalam kategori: huruf a-z, A-Z, dan 0-9. Kehati-hatian harus dilakukan ketika menerima karakter-karakter khusus semacam &,',",@,dan !. Simbol-simbol ini mungkin memiliki arti khusus dalam bahasa pembangkitan konten atau komponen lain dari aplikasi Aplikasi Berbasis Web.
9. Memastikan bahwa halaman yang dibangkitkan secara dinamis tidak berisi meta karakter berbahaya. Dimungkinkan bagi seorang pengguna yang berniat jahat untuk menempatkan tag-tag ini dalam database atau suatu file.Ketika suatu halaman dinamis dihasilkan menggunakan data yang diubah, kode yang berniat jahat yang ditambahkan dalam tag mungkin diteruskan ke klien browser. Selanjutnya browser pengguna dapat diperdaya untuk menjalankan suatu program yang dipilih penyerang. Program ini akan mengeksekusi dalam konteks keamanan browser untuk berkomunikasi dengan Web server yang sah, bukan konteks keamanan browser untuk berkomunikasi dengan penyerang. Jadi, program tersebut akan mengeksekusi dalam suatu konteks keamanan yang tidak tepat dengan hak istimewa akses yang tidak tepat.
10. Perangkat karakter pengkodean harus diatur secara eksplisit dalam setiap halaman, kemudian data pengguna harus di scan rangkaian byte yang merepresentasikan karakter khusus untuk skema pengkodean yang diketahui.
11. Setiap karakter dalam perangkat karakter tertentu dapat dikodekan menggunakan nilai numeriknya. Pengkodean output dapat digunakan sebagai suatu pengganti bagi pem-filteran data. Pengkodean menjadi penting secara khusus ketika karakter khusus semacam simbol copyright dapat menjadi bagian dari data dinamis. Meskipun demikian, pengkodean data dapat diberdayakan intensif, dan suatu keseimbangan harus ditemukan antara pengkodean dan metoda lain untuk mem-filter data.
12. Cookie harus selalu diperiksa untuk karakter khusus apapun dan karakter tersebut harus difilter.
13. Mekanisme enkripsi sebaiknya digunakan untuk mengenkripsi password yang masuk berbentuk script
14. Untuk aplikasi Aplikasi Berbasis Web yang dibatasi oleh username dan password, tak satupun dari website dalam aplikasi tersebut dapat diakses tanpa mengeksekusi proses login yang tepat.

II.2.2 UPAYA PREVENTIF TERHADAP KEAMANAN WEB SERVER

II.2.2.1 KEAMANAN SISTEM OPERASI (*OPERATING SYSTEM*)

Setidaknya ada beberapa kegiatan yang perlu dilakukan dalam memelihara keamanan Sistem Operasi di antaranya:

1. Merencanakan instalasi dan penyebaran Sistem Operasi dan komponen lain untuk Web Server.
2. Melakukan *patch* dan *update* Sistem Operasi secara regular.
3. Mengkonfigurasikan Sistem Operasi dengan konfigurasi yang paling aman (*Operating System Hardening*).
4. Menginstal dan mengkonfigurasi tambahan kontrol-kontrol keamanan, jika dibutuhkan.
5. Menguji Sistem Operasi untuk memastikan bahwa keempat langkah sebelumnya cukup mengatasi seluruh pokok persoalan keamanan.

II.2.2.2 KEAMANAN APLIKASI APLIKASI BERBASIS WEB

a. Manajemen keamanan sistem informasi harus terintegrasi penuh dengan proses Software Development Life Cycle (SDLC). Proses pengamanan sistem menjadi satu proses utuh mulai dari identifikasi kebutuhan keamanan (*security requirement*), perancangan keamanan, pembangunan fitur-fitur keamanan sistem, dan pengujian keamanan.

b. Pada tahap pengembangan sistem pada tahap identifikasi kebutuhan sistem, setiap instansi tidak hanya memperhatikan kebutuhan fungsionalitas dan non fungsionalitas, tetapi juga harus memasukkan kebutuhan keamanan.

c. Identifikasi kebutuhan keamanan dapat dikategorikan menjadi *functional security requirement, non-functional security requirement, derived security requirement, user stories, dan abuse case*

d. Perancangan keamanan sistem perlu dimulai dengan kajian risiko keamanan sistem.

e. Menentukan dan mendokumentasikan peran dan tanggungjawab keamanan informasi dalam SDLC.

f. Menentukan individu-individu yang memiliki peran dan tanggungjawab dalam keamanan informasi dan mengintegrasikan proses manajemen risiko Instansi ke dalam aktivitas SDLC.

g. Lingkungan Pengembangan (Development Environment) harus menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak dengan fasilitas sebagai berikut:

   - Editor, yaitu fasilitas untuk menuliskan kode sumber dari perangkat lunak.
   - Compiler, yaitu fasilitas untuk mengecek sintaks dari kode sumber kemudian mengubah dalam bentuk binari yang sesuai dengan bahasa mesin.

- Linker, yaitu fasilitas untuk menyatukan data binari yang beberapa kode sumber yang dihasilkan compiler sehingga data-data binari tersebut menjadi satu kesatuan dan menjadi suatu program komputer yang siap dieksekusi.
- Debuger, yaitu fasilitas untuk mengetes jalannya program, untuk mencari *bug*/kesalahan yang terdapat dalam program.
- Sampai tahap tertentu dapat membantu memberikan saran yang mempercepat penulisan. Pada saat penulisan kode, IDE juga dapat menunjukan bagian-bagian yang jelas mengandung kesalahan atau keraguan.

h. Lingkungan Produksi: mencakup semua aplikasi, sistem, dan infrastruktur sistem pendukung yang dapat diakses dan secara operasional digunakan untuk menyelesaikan proses dan transaksi oleh pengguna akhir. Akses ke lingkungan ini harus dibatasi hanya untuk pengguna yang berwenang

i. Lingkungan Pra-Produksi/QA/Staging Environment memiliki gambaran sebagai berikut:
   - Konfigurasi menuju lingkungan produksi
   - Verifikasi akhir dalam rilis aplikasi dan layanan
   - Diperuntukan untuk User Acceptance Testing & Operational Acceptance Testing
   - Data yang digunakan hampir menyerupai data produksi

j. Lingkungan Pengujian:
   - Lingkungan untuk uji integrasi (integration testing) dan jaminan kualitas (quality assurance).
   - Mengelola rilis versi aplikasi untuk menjaga stabilitas

k. Standarisasi penggunaan tools pemrograman yang dapat mengendalikan ancaman injeksi SQL baik bagi vendor ataupun internal berupa penggunaan API / Interpreter yang aman.

l. Memasukan kegiatan verifikasi keamanan aplikasi untuk ancaman injeksi SQL dalam setiap kegiatan ujicoba aplikasi yang sedang dikembangkan dan akan dioperasionalkan

m. Skenario ujicoba yang mengevaluasi hasil konstruksi aplikasi berdasarkan pedoman keamanan aplikasi harus dibuat dan dilaksanakan oleh fungsi audit.

n. Aplikasi harus melakukan otentikasi dan otorisasi hak akses user terhadap halaman web yang mengakses referensi obyek (file atau gambar). Untuk mengakses semua halaman web dan sumber daya, dibutuhkan otentikasi, kecuali memang ditujukan kepada publik

o. Penggunaan referensi obyek tidak langsung untuk menghindari perubahan parameter URL secara manual oleh pengguna. Menggunakan referensi obyek tidak langsung per pengguna atau sesi. Hal ini mencegah penyerang langsung mengarah ke sumber daya tidak terotorisasi

p. Penggunaan token unik terenkripsi untuk setiap URL permintaan persetujuan yang dikirimkan melalui suatu email yang kode parameternya tidak mudah dipahami user.

q. Adanya konfirmasi kembali dari sistem atas data yang berhasil / gagal dicatat oleh sistem atas tindakan yang dilakukan oleh pengguna. Memastikan pencatatan kontrol keamanan menyediakan kemampuan untuk mencatat peristiwa keberhasilan dan kegagalan yang diidentifikasi sebagai catatan yang terkait keamanan.

r. Standar konfigurasi server web aplikasi yang telah memperhatikan faktor keamanan dari ancaman serangan.

s. Arsitektur aplikasi yang kuat yang menyediakan pemisahan dan keamanan yang tegas antar komponen.

t. Menjalankan scan dan melakukan audit secara periodik untuk membantu mendeteksi kesalahan konfigurasi atau patch yang hilang di masa mendatang.

u. Setiap halaman web harus diotentikasi dan diotorisasi berdasarkan peran (Role) yang diperiksa berdasarkan sesi login, bukan berdasarkan halaman URL. Untuk mengakses semua halaman web dan sumber daya, dibutuhkan otentikasi, kecuali memang ditujukan kepada publik.

v. Instansi mengembangkan, menyebarluaskan, dan melakukan tinjauan dan pembaruan terhadap:

   - Kebijakan pelatihan dan sosialisasi keamanan secara formal dan terdokumentasi yang mencatat tujuan, lingkup, peran, tanggungjawab, komitmen manajemen, koordinasi antar entitas instansi.
   - Prosedur formal dan terdokumentasi yang memfasilitasi implementasi dari kebijakan pelatihan dan sosialisasi keamanan serta kendali yang berkaitan dengan pelatihan dan sosialisasi keamanan tersebut.

w. Instansi menyediakan sosialisasi dasar mengenai keamanan informasi kepada pengguna (termasuk pimpinan dan rekanan pengembang):

   - Sebagai bagian dari pelatihan pertama pengguna baru.
   - Ketika dibutuhkan karena perubahan sistem.
   - Pengulangan berkala.

x. Instansi menyediakan pelatihan keamanan informasi berbasis peran kepada pengguna sistem informasi:

   - Sebelum diberikan akses ke dalam sistem informasi dalam tugas kerjanya.
   - Ketika dibutuhkan saat ada perubahan sistem.

y. Patching Perangkat Lunak / Aplikasi: aplikasi keamanan yang disediakan vendor harus secara konsisten menyediakan patching untuk melindungi sistem, dan data dari kerusakan atau kehilangan karena ancaman worm, virus, kehilangan data, atau jenis serangan eksternal atau internal lainnya.

z. Instansi mengembangkan, melaksanakan, melakukan peninjauan dan pembaruan terhadap:

- Kebijakan formal dan terdokumentasi untuk proses akuisisi sistem dan layanan yang elah mempertimbangkan keamanan informasi yang mencakup tujuan, lingkup, peran, tanggungjawab, komitmen pimpinan, kordinasi kerja antar unit dalam Instansi.
- Prosedur formal dan terdokumentasi untuk memfasilitasi implementasi kebijakan di atas.

II.2.3 UPAYA PENGUJIAN KEAMANAN APLIKASI BERBASIS WEB

Kegiatan pengujian keamanan Aplikasi Berbasis Web adalah untuk melakukan evaluasi tingkat kerentanan Aplikasi Berbasis Web yang telah dibangun, untuk selanjutnya dilakukan perbaikan atas kelemahan yang ditemukan. Prinsip dan metoda dalam pengujian Aplikasi Berbasis Web adalah:

1. Instansi secara regular harus melakukan pengujian terhadap aplikasi Aplikasi Berbasis Web yang dimilikinya untuk melihat tingkat ancaman (*threat*) dan kerentanan (*vulnerability*).
2. Metoda yang dapat dilakukan untuk pengujian setidaknya meliputi: pemindaian (*scanning*) kerentanan dan penetration testing.
3. Pemindaian (*scanning*) kerentanan memerlukan suatu software pemindai kerentanan yang otomatis melakukan scan terhadap suatu host atau grup host pada suatu jaringan untuk menemukan kerentanan aplikasi, jaringan, dan OS.
4. Pemindaian (*scanning*) kerentanan harus dilakukan secara berkala setidaknya 1 (satu) bulan sekali.
5. Sebaiknya menggunakan lebih dari satu jenis vulnerability scanner. Karena tidak ada vulnerability scanner yang dapat mendeteksi seluruh kerentanan yang ada.
6. Penetration Testing didesain dengan melakukan penetrasi pada suatu jaringan/sistem target, menggunakan tool dan metoda serangan.
7. Penetration Testing harus dilakukan secara berkala minimal 1 (tahun) sekali.
8. Pengujian sebaiknya tidak dijalankan pada *appplication/web server* produksi tetapi dijalankan pada suatu sistem yang terpisah atau *application/web server* simulasi.
9. Hasil pemindaian dan penetration testing harus didokumentasikan dan kekurangan yang ditemukan harus diperbaiki.

II.2.4 TOOLS PENGAMANAN APLIKASI BERBASIS WEB

Setiap Instansi dalam melakukan pengamanan pada Aplikasi Berbasis Webnya dapat menggunakan tools sebagai berikut:

1. Intrusion Detection and Prevention (IDPS), sebagai rules based protection.
2. Web Application Firewall (WAF), sebagai safeguarding access.
3. Security Information and Event Management (SIEM), sebagai event management.

4. Network Behavioral Analysis, sebagai malicious transaction.
5. IP Reputation Engine (RE), sebagai fraud/phishing control.
6. Anti Distributed Denial of Services (DDOS), sebagai flood mitigation.
7. Aplikasi Berbasis Web Defacement Monitoring/Fie Integrity Monitoring , sebagai deface recovery.
8. Unified Threat Management (UTM), sebagai pendekatan keamanan informasi di mana satu perangkat keras atau instalasi perangkat lunak menyediakan beberapa fungsi keamanan
9. Time Synchronization, sebagai master waktu setiap event keamanan
10. Dan lain - lain

II.3 PENGELOLAAN AKSES, OTORISASI, & OTENTIKASI

Pengelolaan akses, otorisasi, dan otentikasi atas aplikasi Aplikasi Berbasis Web dalam suatu instansi perlu mengikuti prinsip-prinsip sebagai berikut:

1. Aplikasi Berbasis Web Instansi dapat memiliki konten publik dan konten terbatas.
2. Aplikasi Berbasis Web Instansi yang memiliki pembatasan akses konten harus lengkapi dengan hsak akses
3. Aplikasi Berbasis Web Instansi yang memiliki konten terbatas hanya boleh digunakan melalui mekanisme otentikasi dan otorisasi terenkripsi.
4. Instansi harus mengatur kendali keamanan kata sandi (password) berupa:
    - Kejadian Kompromi kata sandi: Password harus segera diubah jika kata sandi telah diungkapkan dengan tidak benar, diakses, atau digunakan oleh seorang yang tidak berwenang,
    - Masa Kadaluwarsa kata sandi: Instansi harus merekomendasikan kepada pengguna untuk mengubah kata sandi setidaknya 6 bulan sekali.
    - Penguncian kata sandi: Penguncian dilakukan sejumlah upaya login yang tidak sah dilakukan.
    - Penggunaan kembali kata sandi: Instansi merekomendasikan kepada pengguna untuk tidak menggunakan kembali kata sandi saat memperbarui atau mengubah kata sandi setidaknya empat perubahan kata sandi.
5. Instansi harus mengatur kendali akses ketika/berupa:
    - Sebelum suatu sistem baru dikembangkan atau dibeli, tim internal pemilik dan pengguna sistem harus secara spesifik menetapkan persyaratan keamanan sistem informasi yang diperlukan.
    - Pegawai yang menjalankan operasional sistem tidak boleh diberi akses melebihi apa yang mereka butuhkan untuk melakukan pekerjaan mereka.
    - Menonaktifkan software tambahan yang berpotensi membahayakan keamanan, dan tidak ada hubungannya dengan penggunaan dalam lingkungan operasional.

6. Instansi harus mengatur kendali akses terhadap Operating System berupa:

   - Pembatasan akses koneksi di lokasi tertentu, maka akses fisik ke terminal kabel data harus dibatasi hanya untuk karyawan sesuai kebutuhan.
   - Jika terjadi kegagalan memasukkan password setelah tiga kali maka user ID harus dikunci sementara agar tidak bisa digunakan sampai dilakukan reset oleh administrator sistem.
   - Penggunaan semua perangkat termasuk komputer atau perangkat pribadi, laptop dan perangkat portabel lain yang digunakan untuk mengakses jaringan dan aplikasi Aplikasi Berbasis Web internal di Instansi harus mendapat persetujuan atasan.
   - Jika tidak ada aktivitas di perangkat komputer atau server setelah jangka waktu tertentu, sistem harus secara otomatis mengunci dan mengosongkan tampilan layar, memutuskan session dan memerlukan password untuk mengembalikan session.

7. Instansi harus mengatur kendali akses terhadap aplikasi & database berupa:

   - Penggunaan dan/atau pendistribusian software, alat atau aplikasi pendeteksi kerentanan yang mungkin berpotensi mengganggu keamanan sistem informasi harus mendapat persetujuan dari atasan. Jika software, alat atau aplikasi khusus tersebut tidak diperlukan lagi secara aktif, maka harus dihapus dari sistem.
   - Membatasi penggunaan software dapat berpotensi menyebabkan kerusakan sistem.
   - Akses ke hardware dan software untuk keperluan diagnosa harus diawasi ketat oleh Petugas Keamanan Informasi dan hanya digunakan oleh petugas yang berwenang melakukan pengujian, pemecahan masalah, dan tujuan pengembangan.
   - Akses ke software atau alat khusus harus dibatasi hanya untuk pengguna terpercaya yang disetujui, dan setiap kali alat dijalankan, harus dimonitor aktivitas yang dihasilkan.
   - Unit Pengelola TI harus membangun dan membatasi penggunaan fasilitas khusus untuk digunakan dalam keadaan luar biasa di mana kontrol dapat dikompromikan demi menjaga kelangsungan operasional instansi.
   - Administrator hanya boleh memberikan hak akses ke aplikasi atau database setelah mendapat persetujuan tertulis dari Petugas Keamanan Informasi.
   - Akses pengguna harus dibatasi hanya sesuai keperluannya.
   - Perubahan terhadap database di lingkungan operasional harus dilakukan melalui jalur yang ditetapkan oleh Petugas Keamanan Informasi.
   - Systems log atau detail aktifitas aplikasi untuk keperluan audit (audit trail) tidak boleh diungkapkan kepada orang luar, kecuali yang bertugas terkait hal tersebut atau yang menyelidiki insiden keamanan sistem informasi dan telah menKitatangani perjanjian kerahasiaan.

- Akses terhadap aset informasi rahasia hanya diberikan untuk individu-individu tertentu, dan bukan kepada kelompok.
- Database atau aplikasi di Instansi atau Instansi dapat ditetapkan sebagai aplikasi terbatas, jika pemberian akses yang tidak tepat berpotensi timbulnya pelanggaran hukum, pengungkapan informasi rahasia dan timbulnya resiko kerusakan data bisnis yang kritikal atau akses tidak sah terhadap data atau informasi pribadi karyawan.

8. Instansi harus mengatur kendali akses pengguna terhadap Aplikasi Aplikasi Berbasis Web berupa:
   - Prosedur registrasi dan penghapusan dari daftar pengguna formal dilakukan untuk memberikan dan mencabut akses ke sistem dan layanan informasi.
   - ID pengguna unik digunakan untuk setiap pengguna yang selanjutnya bertanggungjawab atas tindakan yang dilakukan dalam penggunaan ID tersebut.
   - Pengecekan dengan pemilik sistem akan dilakukan untuk memastikan bahwa pengguna memiliki kebutuhan bisnis yang valid sebelum diberikan akses.
   - Metoda untuk memastikan bahwa pengguna mengetahui dan mengakui tanggungjawab dan ketentuan akses mereka.
   - Catatan atas seluruh hak akses yang diberikan kepada individu dibuatkan dan dirawat.
   - Pemeriksaan berkala dilakukan untuk mengunci atau menghapus akun pengguna yang berlebihan dan tinjauan kebutuhan bisnis yang berkelanjutan diselesaikan secara teratur.
   - Hak akses sekurang-kurangnya terdiri username dan password dikelola oleh pemilik Aplikasi Berbasis Web dan dijaga kerahasiaaanya oleh pengelola Aplikasi Berbasis Web.
   - Username yang digunakan harus bersifat unik.
   - Password yang digunakan harus menggunakan strong password.
   - Pengguna dilarang memberikan hak aksesnya kepada pihak lain.
   - Aplikasi Berbasis Web harus dilengkapi dengan modul pengelolaan pengguna.
   - Aplikasi Berbasis Web harus dilengkapi dengan kemampuan pendaftaran user baru dan reset password.
   - Penyampaiaan password baru dan reset passwor wajib dilakukan dengan cara aman.
   - Aplikasi Berbasis Web harus dilengkapi dengan kemampuan mencatat akses dan aktivitas tiap pengguna, minimal mencatat akses masuk, akses keluar sistem, kesalahan pemasukan password.
   - Aplikasi Berbasis Web harus memiliki kemampuan mengingat 10 password terdahulu dari pengguna dan dapat mengantisipasi penggunaan password dimaksud.
   - Basis data pengguna wajib dilindungi dengan enkripsi.

- Lingkungan komputasi berupa server, storage, dan jaringan dalam pengembangan Aplikasi Berbasis Web harus dipisahkan.
- Pemisahan lingkungan dimaksud dapat berbentuk pemisahan logik dan atau fisik didasarkan pada pertimbangan risiko.
- Akses ke lingkungan dan data pengembangan harus dibatasi hanya bagi tim pengembang dan pihak-pihak yang berwenang.
- Data yang digunakan dalam lingkungan pengembangan harus menggunakan data dummy dan mencerminkan bentuk data operasi.
- Data dummy harus memperhatikan aspek kerahasian dan privacy data.
- Lingkungan pengembangan Aplikasi Berbasis Web wajib berlokasi di wilayah negara indonesia.
- Aplikasi Berbasis Web wajib dilengkapi dengan sertifikat yang valid.
- Aplikasi Berbasis Web instansi yang mengandung konten rahasia dan atau sensitif wajib dilengkapi dengan enkripsi dan sertifikat yang dikeluarkan oleh BSSN.
- Pemilik Aplikasi Berbasis Web wajib memasitikan keamanan sertifikat dan memastikan validitasnya.
- Setiap personel dalam proses pengembangan Aplikasi Berbasis Web harus menKitatangani perjanjian kerahasiaan.
- Setiap kegiatan pengembangan Aplikasi Berbasis Web wajib mencantumkan tahapan pengujian keamanan dalam lingkup pekerjaan.
- Pengujian keamanan dilakukan berdasarkan identifikasi potensi ancaman keamanan.
- Koneksi basis data wajib dilakukan dengan mekasnisme koneksi terpisah.
- Kode sumber Aplikasi Berbasis Web tidak boleh mengandung akun koneksi basis data.
- Setiap koneksi ke basis data wajib menggunakan akun khusus koneksi basis data dengan pembatasan akses sesuai dengan fungsinya.

9. Penghapusan atau penyesuaian hak akses berupa:
   - Semua sistem yang menangani aset informasi sensitif harus dapat mencatat log yang berisi setiap tambahan, modifikasi dan penghapusan informasi. Log yang berisi aktifitas akses dan penggunaan sistem, harus disimpan setidaknya selama tiga bulan.
   - Log terkait upaya memasuki sistem dan jaringan internal, harus dipertahankan untuk periode 2 minggu terakhir.
   - Password yang tidak dienkripsi tidak boleh dicatat dalam log sistem.

- Mekanisme pendeteksi dan pencatat aktifitas terkait keamanan sistem informasi harus tahan terhadap upaya menonaktifkan, memodifikasi, atau menghapus log secara tidak sah.
- Semua komputer yang terhubung ke jaringan internal harus memiliki jam akurat yang disinkronkan dengan jam server.
- Alat untuk memantau atau mengamati aktivitas pengguna komputer tidak boleh digunakan, kecuali digunakan untuk penyelidikan dugaan aktivitas kriminal.
- Pemantauan aktifitas komputer atau ID pengguna untuk tujuan investigasi atau tujuan penegakan disiplin, harus diinformasikan kepada atasan langsung pengguna bersangkutan dan semua aktifitas pemantauan harus dicatat dan ditinjau.

II.4 PENGEMBANGAN & PENGELOLAAN APLIKASI BERBASIS WEB OLEH PIHAK KETIGA

1. Instansi yang pengelolaan aplikasi Aplikasi Berbasis Webnya dilakukan oleh pihak ketiga harus memperhatikan prinsip – prinsip keamanan informasi, yaitu: Confidentiality, Integrity, dan Availibility.
2. Setiap paket software pihak ketiga yang digunakan Instansi untuk keperluan operasional Instansi atau Instansi harus dipastikan bebas dari resiko penonaktifan tanpa diketahui oleh Instansi atau Instansi, yang mungkin bisa dilakukan oleh vendor penyedia.
3. Perancang dan pengembang sistem di Instansi tidak boleh menggunakan software, alat dan bahasa pemrograman yang keamanannya belum terbukti secara luas.
4. Perjanjian kerjasama antara pemberi kerja dan pengembang Aplikasi Berbasis Webwajib mengandung ketentuan yang mengatur kerahasiaan informasi pemberi kerja.
5. Pengembang Aplikasi Berbasis Web dan personel pengembang Aplikasi Berbasis Web dilarang menggunakan kode sumber yang mengandung kerentanan.
6. Instansi mensyaratkan agar pengembang sistem informasi:
    - Membuat dan melaksanakan uji keamanan dan rencana evaluasi yang mengakomodasi pengujian atau evaluasi, berbasis kedalaman dan ketelitian
    - Membuat bukti pelaksanaan rencana pengujian atau evaluasi dan hasil dari pengujian dan evaluasi tersebut.
7. Pemberi kerja wajib melakukan analisis dan menguji keamanan kode sumber.
8. Pengembang Aplikasi Berbasis Webwajib menggunakan platform Aplikasi Berbasis Web yang aman dan menyampaikan keterbatasan platform yang berpotensi mengandung kerentanan sistem kepada pemberi kerja.

9. Pemberi kerja wajib melakukan Analisis keamanan platform Aplikasi Berbasis Web yang akan dikembangkan.
10. Sistem operasi pada lingkungan pengembangan dan lingkungan produksi adalah versi sistem operasi yang paling sedikit mengandung kerentanan sistem.
11. Pemberi kerja wajib memastikan penggunaan sistem operasi yang paling sedikit mengandung kerentanan sistem dan melakukan update atau pathes secara berkala.
12. Memverifikasi resolusi host pihak ketiga—Sejumlah vendor menyediakan plug-in web browser yang mendukung pencocokan alamat Internet Protocol (IP) dari suatu situs Web, sehingga dapat memberikan suatu peringatan kepada para pengguna jika situs Web mencurigakan.
13. Melakukan peninjauan terhadap proses pengembangan, standar, perangkat dan pilihan-pilihan serta konfigurasi dari perangkat untuk memastikan bahwa proses, standar, perangkat dan pilihan-pilihan konfigurasi yang dipilih akan memenuhi kebutuhan keamanan Instansi.
14. Instansi mensyaratkan pengembang sistem informasi untuk mematuhi proses pengembangan yang terdokumentasi, yaitu:
    - Secara jelas menuliskan kebutuhan keamanan
    - Mengidentifikasikan standar dan perangkat lunak pengembangan yang digunakan dalam proses pengembangan dan mendokumentasikan secara spesifik pilihan – pilihan dan konfigurasi perangkat lunak yang digunakan dalam pengembangan sistem informasi.
15. Instansi sebaiknya memasukan kebutuhan-kebutuhan, keterangan, dan kriteria baik secara terang atau melalui suatu referensi di dalam kontrak pengadaan sistem informasi sesuai dengan hukum perundangan, peraturan pemerintah, kebijakan, peraturan, standar, pedoman dan misi instansi sebagai berikut:
    - Kebutuhan fungsional keamanan
    - Kebutuhan jaminan keamanan
    - Kebutuhan dokumentasi terkait keamanan
    - Deskripsi lingkungan pengembangan sistem informasi dan lingkungan saat sistem dioperasionalkan.
    - Syarat uji terima

II.5 PENGENDALIAN INSIDEN KEAMANAN APLIKASI BERBASIS WEB

II.5.1 JENIS INSIDEN DAN PENGENDALIANNYA

Beberapa survei menunjukkan pengendalian insiden keamanan Aplikasi Berbasis Web dapat menjadi tugas yang sulit sekalipun bagi organisasi yang telah memiliki kapabilitas TI yang baik. Oleh karena itu diperlukan pengembangan kemampuan respon yang tepat dengan pendekatan yang sistematis dan terstruktur terhadap insiden keamanan Aplikasi Berbasis Web ini. Untuk membangun kemampuan dalam merespon insiden keamanan Aplikasi Berbasis Web yang efektif maka instansi perlu mempersiapkan dan melakukan identifikasi kebutuhan dalam proses persiapan, pengendalian, dan pemeliharaan setelah terjadinya serangan keamanan Aplikasi Berbasis Web. Berikut ini identifikasi atas ancaman dan respon yang harus dilakukan jika terjadi insiden.

Tabel 2.7 Identifikasi Ancaman/Kerentanan & Respon yang Diperlukan

Tingkat Risiko	Jenis Insiden	Respon Insiden	Tindak Lanjut/Solusi
Tinggi	- DDOS, DOS - Intrusions: a. Web Deface b. Account Compromise - Cyber Harashment: a. Cyber Bully b. Cyber Stalking - Fraud: a. Phishing/Pharming b. Online Scam	- Membuat laporan insiden oleh tim Security Incident Response kepada tim teknis - Melakukan pengecekan terhadap service pada sistem oleh tim teknis - Melakukan Backup Log anomaly dari monitoring - Melakukan Analisis dari serangan tersebut - Melakukan koordinasi dengan stakeholder dan pihak terkait - Membuat laporan insiden oleh tim Security Incident Response kepada tim teknis - Melakukan pendokumentasian terhadap barang bukti digital	- Melakukan manajemen patching - Melakukan Vulnerability Assesment secara berkala pada sistem - Melakukan filtering terhadap traffik yang masuk - Melakukan pendokumentasian dari bukti – bukti insiden - Mengidentifikasi dan mengurangi dari kerentanan - Melakukan pengecekan kembali apakah sistem sudah berjalan dengan normal

		- Melakukan koordinasi dengan stakeholder dan low enforcement serta pihak terkait lainnya. - Backup log header email - Lakukan analisis terhadap bukti digital yang telah didapatkan	- Meningkatkan kesadaran terkait keamanan internet: - Tidak menampilkan informasi yang sensitive terkait organisasi, lembaga atau pribadi di internet - Melakukan validasi terhadap informasi yang diterima di internet - Menggunakan layanan DNS filtering - Melakukan pengecekan kembali terhadap sistem yang terinfeksi dan memastikan sistem sudah kembali normal. - Membuat advisory terkait insiden yang terjadi - Membuat dokumentasi untuk pembelajaran sebagai referensi untuk insiden dimasa yang akan datang - Membuat advisory terkait insiden yang terjadi - Membuat dokumentasi untuk pembelajaran sebagai referensi untuk insiden dimasa yang akan datang

Sedang	- Intrusions attempt: a. Port Scanning b. Brute Force - Malicous code: a. Botnet CNC b. Malware c. Trojan	- Membuat laporan insiden oleh tim tim Security Incident Response kepada tim teknis - Melakukan estimasi terhadap pengaruh insiden yang ada secara teknis serta potensi dari akibat insiden yang terjadi. - Melakukan identifikasi terhadap sistem yang terinfeksi atau yang menjadi target serangan. - Melakukan isolasi terhadap sistem yang terinfeksi malicious code. - Melakukan analisis terhadap log server, network dan malware. - Melakukan koordinasi dengan stakeholder dan pihak terkait	- Melakukan pemasangan dan update antivirus yang ada - Disable port dan layanan (service) yang tidak di butuhkan - Melakukan pemasangan firewall dan filtering packet - Disinfeksi, mengkarantina, menghapus dan melakukan recovery terhadap file yang terinfeksi - Mengurangi kerentanan dieksploitasi untuk host lain di dalam organisasi - Apabila diperlukan, menjalankan monitoring tambahan untuk melihat kemungkinan terjadinya aktititas yang sama yang akan terjadi dimasa depan. - Membuat advisory terkait insiden yang terjadi - Membuat dokumentasi untuk pembelajaran sebagai referensi untuk insiden dimasa yang akan datang

Rendah	- Spam: a. Junk mail b. Spam Relay c. Fake email	- Membuat laporan insiden oleh tim tim Security Incident Response kepada tim teknis - Backup dan Identifikasi log header email - Analisis header email - Melakukan pengecheckan legitimasi domain dan ip yang didapat dari header email - Melakukan koordinasi dengan stakeholder dan pihak terkait.	- Menggunakan antispam pada mail server - Menggunakan layanan DNS server yang aman - Membuat advisory terkait insiden yang terjadi - Membuat dokumentasi pembelajaran sebagai referensi untuk insiden dimasa yang akan datang

II.5.2 LANGKAH-LANGKAH PENGENDALIAN UMUM INSIDEN KEAMANAN APLIKASI BERBASIS WEB

Pengendalian insiden keamanan Aplikasi Berbasis Web dilakukan melalui langkah-langkah umum sebagai berikut:

1. Pihak/orang yang menemukan insiden harus menghubungi Pusat Pengendalian Insiden. Pihak/orang tersebut harus dilengkapi dengan prosedur kontak dan daftar kontak, diantaranya:
    a. Helpdesk
    b. Personel pemantauan deteksi intrusi
    c. Administrator sistem
    d. Administrator firewall
    e. Mitra/Pihak ketiga
    f. Manajer
    g. Petugas keamanan
    h. Sumber lainnya

    Mereka yang di Unit Pengelola TI mungkin memiliki prosedur kontak yang berbeda dari yang di luar unit pengelola TI.
2. Jika orang yang menemukan insiden tersebut adalah anggota Unit Pengelola TI atau departemen yang terdampak-insiden, mereka akan melanjutkan ke langkah 5.
3. Jika orang yang menemukan insiden itu bukan anggota departemen TI atau departemen yang terdampak-insiden, mereka harus menghubungi Pusat Penanganan Insiden di Nomor Telepon Penanganan Insiden atau orang yang berwenang untuk 24x7.

4. Pusat Pengendalian Insiden akan merujuk ke daftar kontak darurat TI atau daftar kontak departemen yang terkena dampak-insiden dan memanggil nomor yang ditunjuk dalam urutan pada daftar. Pusat Penanganan Insiden akan mencatat:
    a. Nama pelapor
    b. Waktu panggilan
    c. Informasi kontak tentang pelapor
    d. Jenis kejadian
    e. Peralatan atau orang yang terlibat
    f. Lokasi peralatan atau orang yang terlibat
    g. Bagaimana insiden itu terdeteksi
    h. Ketika peristiwa pertama kali diperhatikan, gagasan apakah yang mendukung bahwa ada terjadi sebuah insiden
5. Anggota staf TI atau anggota staf departemen yang terkena dampak insiden yang menerima panggilan (atau mengetahui insiden itu) akan merujuk ke daftar kontak mereka untuk menghubungi personel manajemen yang berwenang dan anggota respons insiden. Anggota staf juga harus memastikan cadangan personel dan manajer lainnya yang ditunjuk untuk dapat dihubungi. Anggota staf akan mencatat informasi yang diterima dalam format yang sama dengan Pusat Penanganan Insiden pada langkah sebelumnya. Anggota staf mungkin dapat menambahkan yang berikut:
    a. Apakah peralatan mempengaruhi bisnis yang kritis?
    b. Seberapa parah dampak potensial?
    c. Nama sistem yang terdampak, sistem operasi, alamat IP, dan lokasi
    d. Alamat IP dan informasi apa pun tentang asal serangan
6. Anggota tim respons yang dihubungi akan meninjau atau membahas situasi melalui telepon dan menentukan strategi respons.
    a. Apakah insiden itu nyata atau diduga?
    b. Apakah insiden itu masih berlangsung?
    c. Data atau apa saja yang terancam dan seberapa kritisnya?
    d. Apa dampaknya terhadap bisnis jika serangan berhasil? Minimal, serius, atau kritis?
    e. Sistem atau bagian apa yang terdampak, di mana mereka ditempatkan secara fisik dan di struktur jaringan?
    f. Apakah insiden di dalam jaringan-terpercaya (trusted network)?
    g. Apakah responsnya mendesak?
    h. Bisakah insiden itu dengan cepat terkendali?
    i. Apakah respons akan memperingatkan penyerang dan apakah kita peduli bila penyerang tahu?
    j. Jenis insiden apa ini? Contoh: virus, worm, intrusi, penyalahgunaan, kerusakan

7. Tiket insiden dikategorikan ke dalam level tertinggi yang berlaku dari salah satu kategori berikut:
    a. Kategori tinggi - Ancaman terhadap keselamatan publik, reputasi instansi & data sensitif
    b. Kategori medium - Ancaman terhadap sistem
    c. Kategori rendah - Gangguan layanan
8. Anggota tim akan menetapkan dan mengikuti salah satu dari prosedur berikut berdasarkan respons mereka pada penilaian insiden, misalnya:
    a. Prosedur respons worm
    b. Prosedur respons virus
    c. Prosedur kegagalan sistem
    d. Prosedur respons intrusi aktif
    e. Prosedur respons intrusi tidak aktif
    f. Prosedur penyalahgunaan sistem
    g. Prosedur respons pencurian properti
    h. Prosedur penolakan situs web terhadap layanan
    i. Prosedur penolakan layanan database atau file
    j. Prosedur respons spyware
    k. Prosedur respons lainnya

    Tim juga dapat membuat prosedur tambahan jika tidak ada prosedur yang berlaku. Tim harus mendokumentasikan apa yang telah dilakukan dan kemudian menetapkan prosedur untuk insiden tersebut.
9. Anggota tim akan menggunakan teknik forensik, yaitu:
    a. Mengidentifikasi
    b. mempreservasi
    c. Mengkoleksi
    d. Mengeksaminasi
    e. Menganalisis
    f. Mempresentasi
    g. Memutuskan

    Hanya personel yang berwenang yang boleh melakukan wawancara atau memeriksa bukti, dan personel yang berwenang dapat berbeda-beda tergantung situasi dan organisasi.
10. Anggota tim akan merekomendasikan perubahan untuk mencegah terjadinya infeksi pada sistem lainnya.
11. Setelah persetujuan manajemen, perubahan akan diterapkan.
12. Anggota tim akan mengembalikan sistem yang terdampak ke kondisi kerja-normal. Mereka dapat melakukan salah satu atau lebih hal berikut ini:
    a. Menginstal ulang sistem yang terkena dampak dari awal dan memulihkan data dari backup/cadangan jika perlu. Simpan bukti sebelum melakukan ini.

b. Membuat pengguna mengubah kata sandi jika kata sandi mungkin telah diretas.

c. Memastikan sistem telah diperkeras (harden) dengan mematikan atau menghapus instalasi layanan yang tidak digunakan.

d. Memastikan sistem telah ditampal (patch) sepenuhnya.

e. Memastikan perlindungan virus dan deteksi intrusi waktu-nyata berjalan.

f. Memastikan sistem mencatat peristiwa yang benar dan ke tingkat yang tepat.

13. Dokumentasi, berikut ini harus didokumentasikan:

    a. Bagaimana insiden ditemukan

    b. Kategori insiden

    c. Bagaimana insiden itu terjadi, baik melalui email, firewall, dll

    d. Dari mana insiden itu berasal, seperti alamat IP dan informasi terkait lainnya tentang insiden

    e. Bagaimana rencana responsnya

    f. Tindakan apa yang dilakukan sebagai respons?

    g. Efektif atau tidaknya dari respons

14. Preservasi/Perlindungan Bukti, tim akan membuat:

    a. Bagaimana insiden ditemukan

    b. Membuat salinan log, email, dan komunikasi lainnya

    c. Menyimpan daftar saksi

    d. Menyimpan bukti selama diperlukan

15. Memberi tahu pihak eksternal yang tepat bila diperlukan, misalnya:

    a. Memberi tahu polisi

    b. Memberi tahu agensi lain yang sesuai jika penuntutan terhadap penyusup dimungkinkan

16. Membuat penilaian kerusakan dan biaya, yaitu:

    a. Menilai kerusakan pada organisasi

    b. Memperkirakan biaya kerusakan

    c. Menghitung biaya upaya pencegahan

17. Meninjau respons dan memperbarui kebijakan untuk merencanakan dan mengambil langkah pencegahan agar gangguan tidak terjadi lagi, yaitu antara lain:

    a. Mempertimbangkan apakah kebijakan tambahan dapat mencegah intrusi

    b. Mempertimbangkan apakah suatu prosedur atau kebijakan tidak diikuti yang memperbolehkan intrusi, dan kemudian mempertimbangkan apa yang dapat diubah untuk memastikan bahwa prosedur atau kebijakan tersebut diikuti di masa depan

    c. Apakah respons insiden sesuai? Bagaimana itu bisa diperbaiki?

    d. Apakah setiap pihak yang tepat mendapat informasi tepat waktu?

e. Apakah prosedur penanganan insiden terperinci dan apakah mereka mencakup seluruh situasi? Bagaimana mereka dapat ditingkatkan?

f. Sudahkah perubahan dibuat untuk mencegah infeksi ulang? Apakah semua sistem telah ditampal, sistem dikunci, kata sandi diubah, pembaruan anti-virus, kebijakan email ditetapkan, dll?

g. Apakah telah dilakukan perubahan untuk mencegah infeksi baru dan serupa?

h. Haruskah ada kebijakan keamanan diperbarui?

i. Pelajaran apa yang bisa dipetik dari pengalaman ini?

II.6 PERAN & TANGGUNG JAWAB PENGELOLAAN

Dalam pengelolaan keamanan Aplikasi Berbasis Web perlu didefinisikan peran dan tanggungjawab di lingkungan instansi. Peran tanggungjawab ini tidak terlepas pada rujukan tentang Peta Okupasi Nasional Keamanan Siber, yang didalamnya memuat 30 Okupasi pada level 5-9. Berdasarkan jenjang kualifikasi KKNI, ragam okupasi Keamanan Siber didefinisikan sebagai berikut:

- Kualifikasi 5

  Pada kualifikasi 5 diidentifikasi 4 okupasi pada fase before dan during, yakni Teknisi Perangkat Keras Kriptografi/ Cryptographic Technician, Cryptographic Administrator dan Junior Cyber Security dan Cyber Security Operator.

- Kualifikasi 6

  Pada kualifikasi 6 diidentifikasi 10 okupasi ada fase before, during, dan after attack, yang meliputi ICT Security Product Evaluator, Cryptographic Analyst, Cryptographic Module Analyst, Vulnerability Assessment Analyst, Network Security Administrator, Cyber Security Administrator, Cyber Security Awareness Officer, Cyber Security Analyst/Cyber Security Incident Analyst, dan Digital Evidence First Responder.

- Kualifikasi 7

  Pada kualifikasi 7 diidentifikasi 12 okupasi pada seluruh fase meliputi Auditor Keamanan Informasi/Siber, Cybersecurity Governance Ofificer, Threat Hunter, Penetration Tester, Cyber Incident Manager, Cyber Security Awareness Lead Officer, Senior Cyber Security, ICT Security Product Lead Evaluator, Cryptographic Lead Engineer, Manajer Keamanan Jaringan, dan Digital Forensic Analyst.

- Kualifikasi 8

  Pada kualifikasi 8, diidentifikasi 5 okupasi pada fase before attack dan after attack meliputi Cyber Risk Specialist, Security Architect, Cryptographic Specialist, Cyber Incident Investigation Manager, dan Cyber Forensic Specialist.

- Kualifikasi 9

Pada kualifikasi 9, diidentifikasi 1 okupasi pada seluruh fase Chief of Information Security Officer.

Dari 30 okupasi diatas yang ada keterkaitan dalam pengelolaan keamanan Aplikasi Berbasis Webadalah sebagai berikut:

II.6.1 CHIEF INFORMATION OF SECURITY OFFICER

II.6.1.1 DEFINISI

Merupakan seorang eksekutif senior yang memiliki kompetensi dan keahlian manajemen dan teknis yang berwenang dan bertanggungjawab pada keamanan informasi organisasi dan operasional seluruh sistem elektronik dengan mempertimbangkan risiko operasional (termasuk misi, fungsi, citra atau reputasi), aset organisasi, individu, dan bagian organisasi lainnya.

II.6.1.2 LINGKUP BIDANG PEKERJAAN

1. Memimpin dan bertanggungjawab terhadap keamanan informasi organisasi termasuk di dalamnya manajemen risiko, arsitektur enterprise, manajemen aset, manajemen perubahan, manajemen keberlangsungan bisnis, SOC, kepatuhan akan tata kelola keamanan informasi dan perlindungan data pribadi
2. Pembentukan dan pelaksanaan tata kelola keamanan informasi organisasi
3. Mengelola tim, menunjuk personel yang bertanggungjawab
4. Evaluasi strategi dan tata kelola keamanan informasi organisasi
5. Memimpin dan mengarahkan di bidang portofolio, program, dan proyek keamanan informasi
6. Memimpin dan mengarahkan di bidang peningkatan kompetensi dan peningkatan kesadaran keamanan informasi
7. Melakukan perencanaan strategis organisasi
8. Melakukan perencanaan dan aktifitas keuangan termasuk penganggaran dan investasi yang berhubungan dengan keamanan informasi
9. Melakukan manajemen vendor dan rantai pasok terkait dengan strategi keamanan informasi

II.6.1.3 PROFIL

1. Berintegritas
2. Analitis
3. Mengatasi masalah (problem-solving)
4. Merencanakan dan mengorganisasi pekerjaan

5. Memimpin tim
6. Bertanggungjawab
7. Mampu mengarahkan dan mempunyai visi
8. Memberi motivasi
9. Memimpin perubahan

II.6.1.4 TANGGUNGJAWAB

1. Mengelola sumber daya yang diperlukan termasuk dukungan kepemimpinan, sumber daya keuangan dan personel keamanan utama untuk mendukung tujuan dan sasaran keamanan teknologi informasi dan mengurangi risiko organisasi secara keseluruhan.
2. Mengelola kebutuhan kebijakan dan berkolaborasi dengan pemangku kepentingan untuk mengembangkan kebijakan untuk mengatur kegiatan siber
3. Menetapkan dan / atau menerapkan kebijakan dan prosedur untuk memastikan perlindungan infrastruktur penting yang sesuai.
4. Merancang / mengintegrasikan strategi siber yang menguraikan visi, misi, dan tujuan yang selaras dengan rencana strategis organisasi.
5. Mengembangkan dan memelihara rencana strategis.
6. Memastikan ada rencana aksi dan tonggak atau rencana remediasi untuk kerentanan yang diidentifikasi selama penilaian risiko, audit, inspeksi, dll.
7. Melakukan upaya perencanaan strategis jangka panjang dengan mitra internal dan eksternal dalam kegiatan siber
8. Menunjuk dan memandu tim pakar keamanan TI
9. Mengelola penilaian risiko keamanan informasi.
10. Berkoordinasi dengan pemangku kepentingan sumber daya organisasi untuk memastikan alokasi dan distribusi aset modal manusia yang tepat.
11. Berkolaborasi dengan semua unsur organisasi yang terkait dalam pembuatan kebijakan dan prosedur privasi dan keamanan informasi.
12. Berkolaborasi dengan personel keamanan siber dalam proses penilaian risiko keamanan untuk menangani kepatuhan privasi dan mitigasi risiko.
13. Berkolaborasi dengan pemangku kepentingan utama untuk membangun program manajemen risiko keamanan siber.

14. Mengawasi pelatihan dan program peningkatan kesadaran keamanan informasi.
15. Mengawasi atau mengelola tindakan protektif atau korektif ketika insiden atau kerentanan keamanan informasi ditemukan.
16. Advokasi posisi resmi organisasi dalam proses hukum dan legislative.

II.6.1.5 WEWENANG

1. Membuat aturan, memberikan penghargaan, menjatuhkan sanksi berdasar aturan.
2. Mengangkat dan memberhentikan pegawai.
3. Melakukan perjanjian kerja sama dengan pihak lain.
4. Menentukan dan mengelola sumber daya yang dibutuhkan di bidang keamanan informasi.
5. Mewakili organisasi dalam hal keamanan informasi.

II.6.1.6 PERSYARATAN

1. Memiliki KKNI Level 8.
2. Pengalaman minimal 5 tahun pada 3 dari 5 domain yang terdiri atas:
    - ketatakelolaan di bidang keamanan informasi
    - manajemen risiko dan audit keamanan informasi
    - manajemen dan operasional keamanan informasi
    - aktivitas dalam keahlian keamanan informasi
    - perencanaan strategis, keuangan, dan manajemen vendor
3. Memiliki sertifikasi okupasi, salah satunya sebagai Security Architect, atau Cyber Risk Analyst

II.6.2 SECURITY ARCHITECT

II.6.2.1 DEFINISI

Merupakan seseorang yang memiliki kompetensi dan keahlian untuk memastikan security requirement dari pemangku kepentingan yang diperlukan untuk melindungi misi dan proses bisnis organisasi ditangani secara memadai dalam seluruh aspek dari enterprise architecture. Aspek-aspek tersebut meliputi *reference model, solution architecture* dan sistem yang dihasilkannya.

II.6.2.2 LINGKUP BIDANG PEKERJAAN

1. Menyusun arsitektur keamanan siber berdasarkan visi dan misi organisasi
2. Menyusun arsitektur keamanan siber berdasarkan kebijakan keamanan informasi, klasifikasi informasi serta berbagai peraturan dan standard yang sesuai dengan kebutuhan organisasi
3. Menyusun arsitektur keamanan siber sesuai kebutuhan fungsional organisasi
4. Melakukan analisis kesenjangan dan peta jalan implementasi arsitektur keamanan siber
5. Memastikan implementasi arsitektur keamanan siber sesuai dengan visi dan misi, kebutuhan fungsional dan kepatuhan pada peraturan dan standard.
6. Melakukan evaluasi antara arsitektur keamanan siber dengan kebutuhan dan persyaratan organisasi

II.6.2.3 PROFIL

1. Berintegritas
2. Analitis
3. Berfikir Strategis
4. Sintesis
5. Memimpin tim
6. Bertanggungjawab
7. Merencanakan dan mengorganisasi pekerjaan

II.6.2.4 TANGGUNGJAWAB

1. Memahami kebutuhan dan persyaratan keamanan siber sesuai visi, misi dan konteks organisasi
2. Menyusun arsitektur keamanan siber berdasarkan klasifikasi informasi, kebutuhan fungsionalitas bisnis dan kepatuhan pada peraturan dan standard yang berlaku.
3. Menyusun peta jalan implementasi arsitektur keamanan siber
4. Memastikan sistem yang dibeli dan/atau dikembangkan sesuai dengan arsitektur keamanan siber
5. Melakukan evaluasi atas arsitektur keamanan siber

II.6.2.5 WEWENANG

1. Menentukan persyaratan keamanan siber
2. Mengevaluasi penerapan persyaratan keamanan siber

II.6.2.6 PERSYARATAN

1. Memiliki KKNI Level 7
2. Memiliki sertifikasi okupasi *Cryptographic Specialist*
3. Memiliki pengalaman minimal 5 tahun, diantaranya dalam hal:
    - implementasi kendali keamanan dalam organisasi baik dalam aspek data, perangkat lunak dan infrasruktur
    - evaluasi efektifitas kendali keamanan melalui uji penetrasi
    - pengembangan perangkat lunak yang aman (Secure SDLC)
    - project manager

II.6.3 INCIDENT RESPONSE TEAM MANAGER

II.6.3.1 DEFINISI

Merupakan seseorang yang memiliki kemampuan teknis dan keahlian untuk mengelola dan memantau penanganan insiden serta ancaman keamanan siber dalam suatu organisasi, serta menyediakan koordinasi, umpan balik, dan komunikasi yang dibutuhkan.

II.6.3.2 LINGKUP BIDANG PEKERJAAN

1. Perencanaan dan pengelolaan kapabilitas tim penanganan insiden
2. Perencanaan program penanganan insiden dan pendelegasian wewenang
3. Investigasi insiden keamanan
4. Koordinasi penanganan insiden dan manajemen krisis
5. Evaluasi proses penanganan insiden
6. Penyediaan laporan penanganan insiden
7. Koordinasi penegakan hukum insiden keamanan

II.6.3.3 PROFIL

1. Berintegritas
2. Merencanakan dan mengorganisasikan pekerjaan
3. Analitis
4. Berorientasi pada detail
5. Bekerja dalam tim

II.6.3.4 TANGGUNGJAWAB

1. Pengawasan penanganan insiden
2. Berkoordinasi dengan semua pihak yang terkait dalam penanganan insiden, baik internal maupun eksternal (aparat penegak hukum dan/atau media)
3. Merumuskan prosedur keamanan
4. Menetapkan protokol komunikasi
5. Merumuskan rencana pengembangan program

II.6.3.5 WEWENANG

Mengambil tindakan yang diperlukan yang dianggap sesuai dalam merespon insiden

II.6.3.6 PERSYARATAN

1. KKNI Level 6
2. Lulusan S1
3. Memiliki pengalaman sekurang-kurangnya 4 tahun sebagai tim CSIRT
4. Memiliki Sertifikasi okupasi *Cybersecurity Analyst/Cybersecurity Incident Analyst, Threat Hunter, atau Digital Evidence First Responder*

II.6.4 CYBERSECURITY OPERATOR

II.6.4.1 DEFINISI

Merupakan seseorang yang memiliki kemampuan dan keterampilan untuk mengkategorikan dan mengenali tingkat kerentanan suatu insiden keamanan siber, bertugas untuk melaksanakan prosedur-prosedur dan perintah dari pejabat diatasnya pada pusat operasi keamanan /*Security Operation Center.*

II.6.4.2 LINGKUP BIDANG PEKERJAAN

1. Kategorisasi Insiden Keamanan Siber
2. Peninjauan notifikasi
3. Pembuatan tiket berdasarkan tingkat kerentanan
4. Pelaporan notifikasi

II.6.4.3 PROFIL

1. Berintegritas
2. Mematuhi prosedur
3. Berorientasi pada detail
4. Komunikatif
5. Mampu bekerja dalam tim

II.6.4.4 TANGGUNGJAWAB

Menjalankan prosedur dan perintah

II.6.4.5 WEWENANG

Mengambil tindakan yang diperlukan yang dianggap sesuai dalam merespon insiden

II.6.4.6 PERSYARATAN

KKNI Level 4

II.6.4.7 TUGAS UTAMA

1. Mendeteksi Kerentanan
2. Mengumpulan data yang diperlukan untuk memenuhi persyaratan pelaporan insiden keamanan siber
3. Mematuhi prosedur terminasi sistem dan tata cara pelaporan insiden

II.6.5 CYBERSECURITY ADMINISTRATOR

II.6.5.1 DEFINISI

Merupakan seseorang yang memiliki kemampuan dan keterampilan untuk melaksanakan implementasi dan membuat laporan pelaksanaan program keamanan siber sesuai rencana implementasi manajemen risiko yang sudah ditetapkan.

II.6.5.2 LINGKUP BIDANG PEKERJAAN

1. Mengimplementasikan program keamanan siber
2. Mendokumentasikan pelaksanaan program keamanan siber
3. Menyusun laporan pelaksanaan program keamanan siber

II.6.5.3 PROFIL

1. Berintegritas
2. Mematuhi prosedur
3. Berorientasi pada detail
4. Komunikatif
5. Mampu bekerja dalam tim

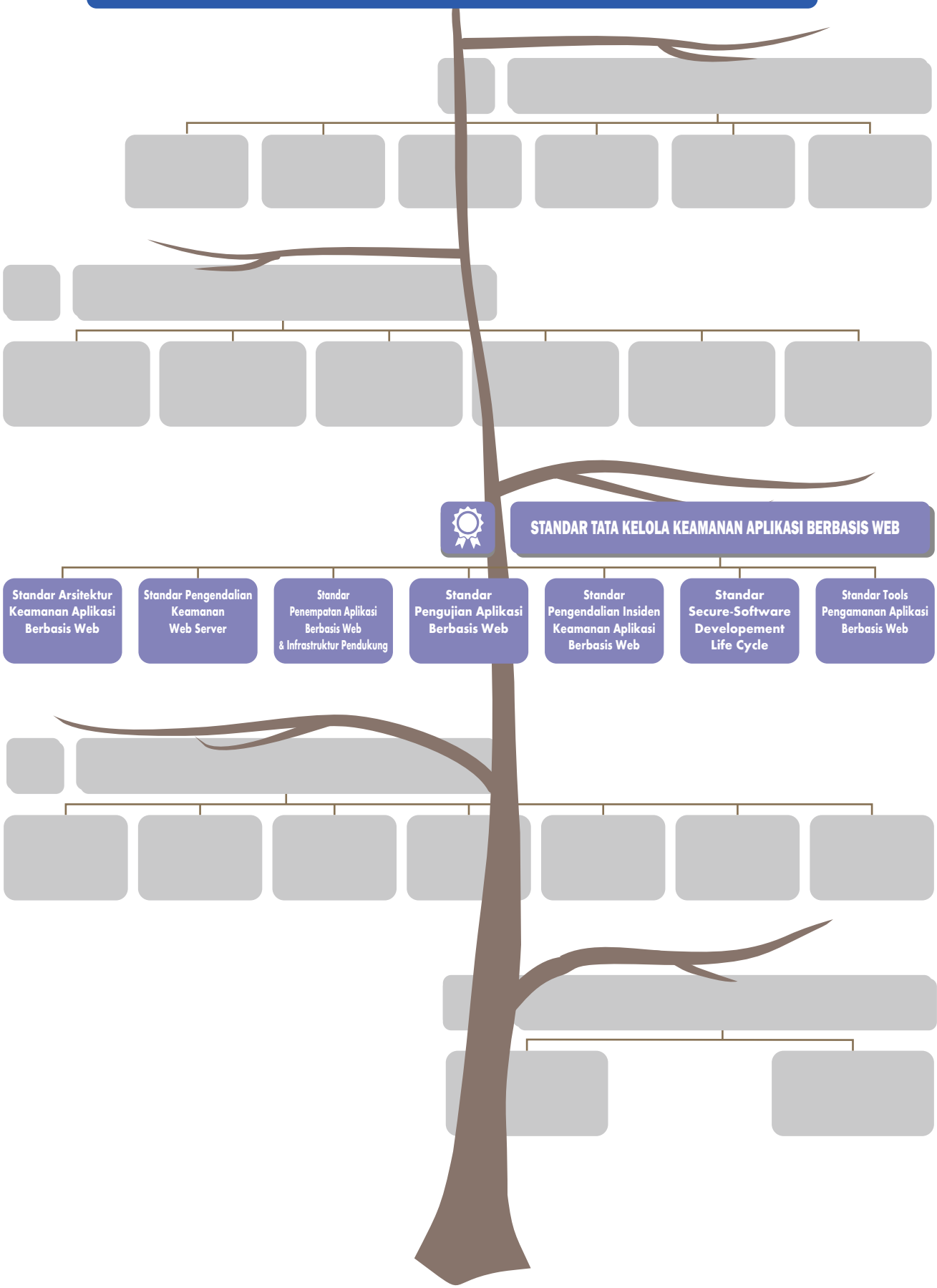
II.6.5.4 TANGGUNGJAWAB

1. Mengimplementasikan program keamanan siber berdasarkan rencana mitigasi risiko siber dan tata kelola keamanan siber
2. Melaksanakan kebijakan keamanan informasi dalam sistem elektronik
3. Menerapkan keamanan dalam siklus informasi mulai dari klasifikasi, kategorisasi, dan Penanggungjawab
4. Menjalankan prosedur dan menerapkan standar keamanan informasi yang berlaku

II.6.5.5 PERSYARATAN

1. Memiliki KKNI Level 5
2. Memiliki sertifikasi okupasi *Junior Cyber Security*

PEDOMAN TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB



III. STANDAR TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB

III.1 STANDAR ARSITEKTUR KEAMANAN APLIKASI BERBASIS WEB

STANDAR
ARSITEKTUR KEAMANAN APLIKASI BERBASIS WEB

BADAN SIBER DAN SANDI NEGARA

2019

A. TUJUAN

1. Untuk memberikan arahan/petunjuk dan memastikan bahwa:
    a. Setiap perancangan sistem Aplikasi Berbasis Web memperhatikan kaidah – kaidah aristektur sistem
    b. Adanya pemisahan tiap komponen/layer aplikasi Aplikasi Berbasis Web sehingga:
        i. Dicapai optimalisasi kinerja yang memungkinkan tiap komponen bekerja independen dan memudahkan mitigasi masalah.
        ii. Ketika terjadi insiden, roll back dapat dilakukan dengan cepat.
        iii. Menciptakan *service staging perimeter protection* yang didukung *access control management* agar mencegah pihak yang tidak berhak untuk masuk ke dalam sistem.
2. Untuk memastikan tingkat ketersediaan (*availability*) sistem ketika terjadi gangguan sesuai dengan tingkat kompleksitas Aplikasi Berbasis Web

B. RUANG LINGKUP

1. Arsitektur sistem web untuk: web statis, web dinamis, dan web dinamis dengan aplikasi transaksional
2. Penentuan Kebutuhan Sumber Daya Komputasi Web

C. ISTILAH DAN DEFINISI

1.	Open Web Server	:	Web Server berbasis free (Terbuka) untuk OS Unix, Linux, BSD, dan Windows. Open Web Server merupakan alternatif pilihan disamping webserver yang sudah terkenal semacam apache / httpd. Open Web Server didesain untuk keamanan, kecepatan, fleksibel dan memenuhi Standar international serta bisa diaplikasikan kedalam mesin produksi.
2.	Cache/Accelerator WAF/Proxy	:	Cache adalah proses penyimpanan sementara data atau halaman HTML dan gambar sebuah Aplikasi Berbasis Web untuk mengurangi penggunaan bandwidth dan loading server. Secara sederhana, cache adalah teknologi yang membantu menampilkan halaman Aplikasi Berbasis Web lebih cepat.

Web Accelerator adalah proxy server yang bertujuan untuk mempercepat akses pada Aplikasi Berbasis Web. Web Accelerator bisa diinstall di disisi klien atau disisi server

Teknik yang digunakan pada *web accelerator* adalah

- Cache & Prefect yaitu data yang baru atau sering diakses disimpan pada tempat penyimpanan sementara baik berupa RAM atau disk kemudian disajikan kembali tanpa perlu mengambil dari server utama .
- Kompresi data dimana ukuran data diperkecil agar bisa lebih cepat.
- Optimize yaitu kode dioptimasi sehingga size atau ukuran menjadi lebih kecil.

Proxy adalah suatu sistem yang memungkinkan kita untuk bisa mengakses jaringan internet menggunakan IP yang berbeda dengan yang diterima oleh perangkat.

Reverse proxy adalah salah satu jenis dari proxy, yang digunakan sebagai perantara antara client dengan web server. Reverse proxy dapat menghandle beberapa web server. Adapun cara kerjanya adalah client akan melakukan akses terhadap sebuah URL maka secara otomatis client akan melakukan request terlebih dahulu ke reverse proxy akan tetapi seolah - olah client melakukan request langsung ke web server. Setelah menerima request dari client, maka reverse proxy akan meneruskan request tersebut ke web server yang dituju

3. Instansi : Kementerian/Lembaga, Instansi pusat atau daerah.
4. *Password* : Kata sandi yang digunakan bersamaan dengan *username* (*sign on/sign in/log-on/log-in*) oleh pemilik yang sah sebelum melakukan koneksi/akses ke sistem komputer.

5.	*Patch*	:	Rutin program atau sekumpulan kecil instruksi yang biasanya dibuat sebagai solusi sementara untuk mengatasi atau memperbaiki permasalahan (*bugs*) pada program komputer dan sering dibuat dalam bentuk *'object code'* yang disisipkan ke dalam program yang akan dieksekusi.
6.	Beban kerja suatu sistem		Himpunan semua input yang diterima sistem dari lingkungannya, selama periode waktu tertentu. Atau banyaknya sumber daya yang harus disediakan untuk melayani setiap request (proses) dan akses. Setiap sistem adalah unik dan akan memiliki karakteristik beban kerja yang berbeda
7.	*Content Delivery Network* (CDN)		adalah kumpulan dari server global yang terletak di beberapa data center dan tersebar di berbagai negara. Jaringan ini berfungsi untuk mengirimkan konten dari server ke suatu Aplikasi Berbasis Web. Yang dilakukan oleh CDN adalah meningkatkan kecepatan pengiriman data melalui jaringan server kepada visitor dari lokasi terdekat yang paling memungkinkan
8.	*Content Management System (CMS)*		adalah sebuah perangkat lunak atau sistem yang mengatur konten pada situs web
9.	*Virtual Private Server/Virtual Dedicated Server*		adalah sebuah server yang dibagi menjadi beberapa virtual server yang dapat diinstall OS dan berbagai aplikasi nya sendiri. VPS atau VDS itu sendiri merupakan teknologi yang memungkinkan sebuah komputer (server) dengan kapasitas sumber daya hardware yang sangat besar dapat dibagi-bagi menjadi beberapa virtual komputer yang mandiri. VPS dapat berjalan layaknya sebuah Dedicated Server dan juga dapat diinstall sistem operasi (OS) tersendiri serta dapat mengatur virtual komputernya tanpa mengganggu virtual komputer yang lain

10. *Load Balancer* adalah perangkat/tools yang berfungsi untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, memaksimalkan throughput, memperkecil waktu tanggap dan menghindari overload pada salah satu jalur koneksi. Load balancer digunakan pada saat sebuah server telah memiliki jumlah user yang telah melebihi maksimal kapasitasnya. Load balancer juga mendistribusikan beban kerja secara merata di dua atau lebih komputer, link jaringan, CPU, hard drive, atau sumber daya lainnya, untuk mendapatkan pemanfaatan sumber daya yang optimal.

D. REFERENSI

1. OWASP: Map Application Architecture (OTG-INFO-010)
2. Standar Keamanan Informasi (ISO 27001:2013 – ISMS)

E. STANDAR

1. Arsitektur Aplikasi Web

a. Dalam merancang Arsitektur Web harus melihat jenis web yang akan dibangun yaitu:

   i. Web statis (non user generated content – non UGC)

   ii. Web Dinamik

   iii. Web Dinamik dengan aplikasi transaksi

b. Rancangan Arsitektur Web Statis minimal menggunakan *Virtual Private Server* dengan persyaratan keamanannya sebagai berikut:

   i. Menggunakan Secure Shell (SSH) untuk masuk ke dalam server

   ii. Mengubah port untuk login ke SSH

   iii. Menggunakan password yang kompleks dan mengubahnya secara rutin

   iv. Menonaktifkan akun root

   v. Menjaga update keamanan terbaru

   vi. Menghindari mengunduh perangkat lunak kecuali dari sumber yang terpercaya

vii. Menonaktifkan port network yang tidak terpakai

viii. Menggunakan enkripsi GnuPG

ix. Mengkonfigurasi firewall

x. Menggunakan SFTP di samping FTP

xi. Membuat folder/boot menjadi read-only

xii. Mengaktifkan update otomatis CMS

xiii. Menginstall anti-malware/antivirus

xiv. Memblokir akses anonymous ke FTP

xv. Menginstall rootkit scanner

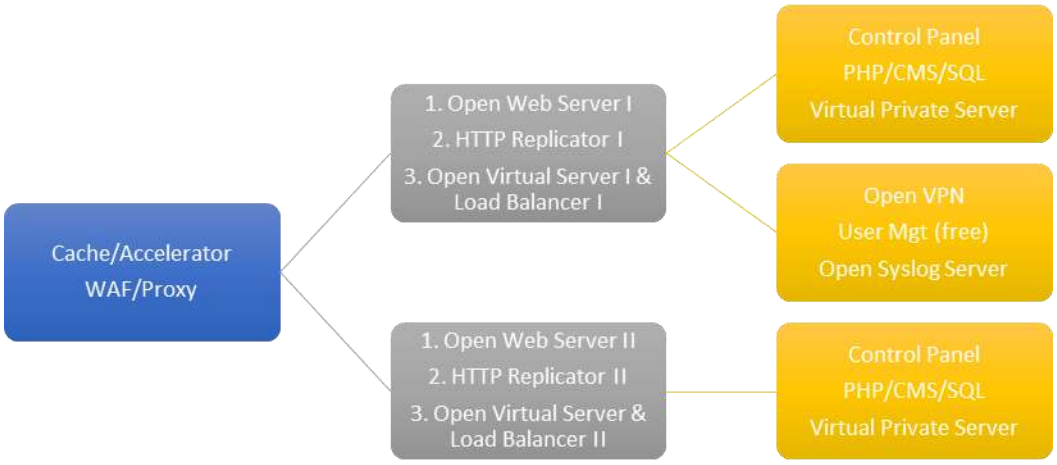
c. Standar arsitektur diatas secara teknis dapat menggunakan tools yang berbasis open source atau komersial.

d. Rancangan Arsitektur Web Statis & Dinamis berbasis open source harus tetap memperhatikan tingkat skalabilitas, fleksibilitas, dan mudah dikembangkan tingkat keamanannya dengan arsitektur minimal sebagai berikut:

i. Redundansi Web Server dan virtualisasi serta load balancer open source (Linux Virtual Server)

ii. Web Server Utama menggunakan User Management Server, Log Server dan Software VPN yang open source

Gambar 3.1 Desain Arsitektur Web Server



f. Rancangan Web dengan Aplikasi Transaksi harus menerapkan model pemisahan tugas di setiap komponen sistem atau fitur yang disediakan yaitu:

   i. Front End – UI/UX,

   ii. Proxy/Accelerator/Load Balancer

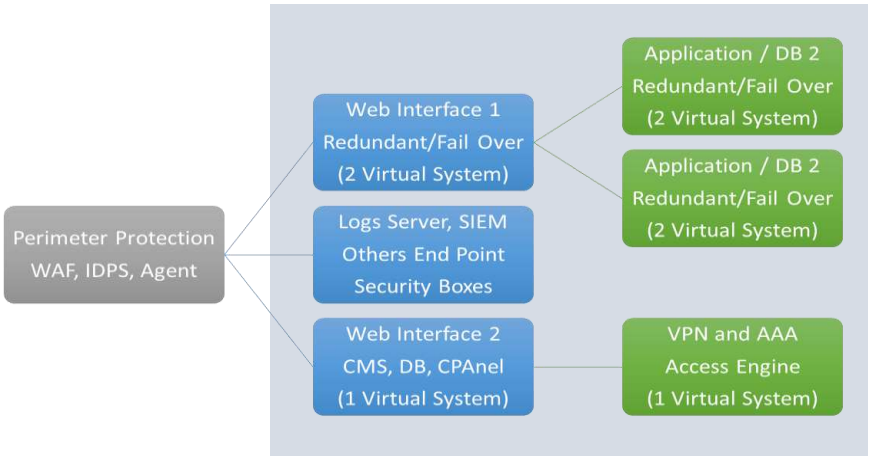
   iii. Web Services,

   iv. Application,

   v. Database.

g. Gambaran rancangan arsitektur Aplikasi Berbasis Web dengan aplikasi dapat digambarkan sebagai berikut:

Gambar 3.2 Desain Arsitektur Layanan Aplikasi Berbasis Web



h. Arsitektur Layanan Aplikasi Berbasis Web meningkatkan kinerja HTTP reverse proxy, selain sebagai penyekat keamanan juga dapat berfungsi untuk mempercepat pengiriman konten Aplikasi Berbasis Web melalui *caching* terintegrasi (*accelerator),* melalui teknik *stream splitting* atau agregasi dengan kerjasama sistem *load balancer* dan kontrol bandwidth.

i. Konfigurasi fine tuning akan meningkatkan layanan Aplikasi Berbasis Web dan keamanannya dengan menerapkan otentikasi pengguna dan SSL tunnel. Jika diperlukan dapat ditambahkan modul kompresi konten Aplikasi Berbasis Websehingga beban kerja server dapat berkurang.

j. Arsitektur Model Anycast dengan menerapkan Content Delivery Network (CDN)

   Bagi instansi-instansi tertentu yang aplikasi Aplikasi Berbasis Webnya dalam satu waktu membutuhkan tingkat ketersediaan yang sangat tinggi (Ditjen Pajak: e-filing di bulan Maret & April, Komisi Pemilihan

Umum (KPU) ketika masa pemilu, BKN ketika penerimaan CPNS, Kemendikbud & Dinas Pendidikan Daerah ketika penerimaan siswa baru dengan zonasi) dapat menerapkan:

k. Desain Arsitektur *Web Anycast/ High Availability Architecture* (HA). Suatu CDN (*Content Delivery Network*) dibangun menggunakan jaringan Anycast. Satu IP yang sama diterapkan pada semua host (mirror) CDN yang berada pada sejumlah network yang berbeda. Kemudian keberadaan IP ini di-broadcast melalui tabel BGP routing di masing-masing upstream network di bawah ASN yang sama. Sehingga IP tersebut akan dapat dikenali aktif berada di sejumlah network yang berbeda sekaligus.

   i. Setiap pengguna di wilayah yang berbeda memungkinkan dilayani oleh server CDN terdekat melalui jalur rute yang terbaik. Misalnya, pengguna di AS atau UE atau Asia akan dilayani oleh server CDN terdekat atau yang paling cepat.

   ii. Jika salah satu server CDN tidak tersedia atau penuh – pengguna akan diarahkan secara otomatis (fail over) ke CDN lain yang terdekat dan masih tersedia (kapasitasnya), melalui perhitungan jalur rute yang terbaik. Sehingga Aplikasi Berbasis Webtersebut akan selalu aktif dan dapat diakses. Secara tidak langsung akan meningkatkan redundansi, kapasitas layanan dan kecepatan akses.

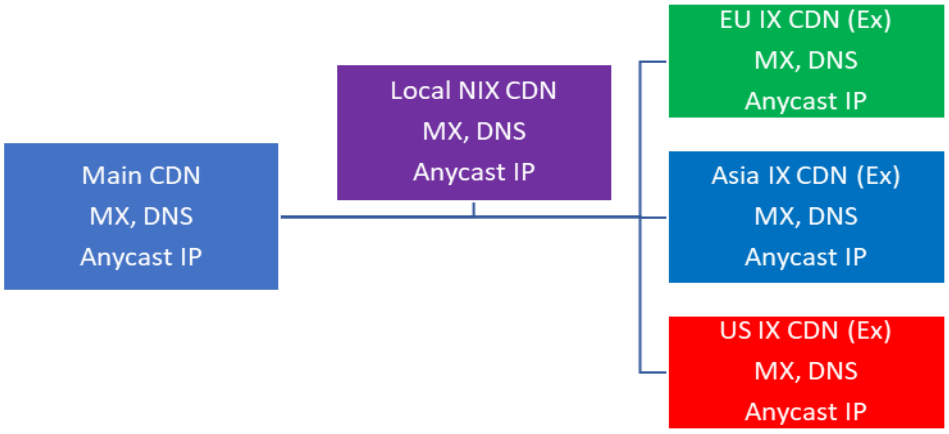
   iii. Pada saat peak, CDN akan mendistribusikan akses secara merata ke banyak host CDN dengan teknik atau menggunakan modul load balancer, sehingga akan lebih efisien. Model Ini juga bermanfaat untuk DDOS karena setiap paket data dipecah ke tempat yang berbeda.

l. Arsitektur ini juga harus mudah untuk direplikasi, sehingga ketika diperlukan pengembangan –menerapkan cluster atau kapabilitas redundansi berbasis anycast, dapat dilakukan dengan cepat. Model Web Anycast memungkinkan efisiensi skenario redundansi dengan menempatkan duplikat sistem (*mirror*) yang identik di sejumlah network yang berbeda sekaligus sebagai pembagi beban (*load balancer*) sehingga dapat menghindari masalah *single point of error/failure (SPOF).*

m. Instansi dapat menggunakan CDN gratis atau berbayar.

Berikut gambaran arsitektur model Web Anycast:

Gambar 3.3. Desain Arsitektur Web Anycast



2. Penentuan Kebutuhan Sumber Daya Komputasi Aplikasi Berbasis Web

a. Suatu layanan Aplikasi Berbasis Web harus direncanakan sesuai dengan kebutuhan sumber daya komputasi agar menjamin kualitas proses dan melayani akses di saat beban terus meningkat atau ketika menghadapi berbagai kondisi beban rendah (*low*), tinggi (*high*), puncak (*peak*), berlebih (*over load*).

b. Dalam menentukan kebutuhan sumber daya komputasi, instansi harus menyusun perencanaan kapasitas, berupa:

   i. Model biaya yang digunakan untuk menentukan biaya konfigurasi server termasuk peripheral, akses, lisensi, perawatan, personel dan pelatihan.

   ii. Model beban kerja, yang membagi beban kerja berdasarkan jenis konten, menghitung intensitas tiap request.

   iii. Model kinerja digunakan untuk menghitung metrik kinerja seperti *response time, error rate dan throughput serta availability*.

   iv. Dan gabungan dari ketiga model ini menghasilkan rasio *cost/performance.*

c. Tahapan dalam perencanaan kapasitas meliputi:

   i. Memahami Lingkungan:

      - Menentukan jenis situs web yang dijalankan
      - Penggunaan Aplikasi Berbasis Web(untuk konsumsi internal atau publik)
      - Jenis server yang digunakan (Web, Database, Application, User Management (AAA))
      - Jenis konektivitas (dedicated, shared, Metronet, VSAT, dll)
      - ISP beserta SLA yang diberikan
      - Sistem Operasi yang digunakan server
      - Aplikasi – aplikasi pendukung yang diperlukan
      - Tingkat layanan (SLA) yang dijanjikan/diberikan kepada pengguna
      - Ketersediaan fasilitas back up
      - Ketersediaan roll back
      - Ketersediaan SOP keamanan (terutama penanganan insiden)

ii. Karakterisasi Beban Kerja Untuk Aplikasi Berbasis Web:

- Karakteristik Situs berita statis akan lebih banyak membutuhkan sumber daya komputasi untuk melayani jumlah akses yang banyak sekaligus pada satu saat dan kapasitas jaringan yang dedicated.
- Karakteristik situs transaksi online bisa membutuhkan sumber daya penyimpanan dan prosesor (CPU) dalam jumlah yang banyak.
- Beban Kerja Server: semua transaksi (pencarian, jelajah, dan pembaruan) yang diproses oleh server dalam satu periode.

  Misalnya server memproses selama 30 menit (atau 1800 detik) dengan 72.000 transaksi (beban kerja) diselesaikan. Karakteristik beban kerja – diwakili intensitas beban (access rate = 72.000 / 1800 = 40 transaksi per detik), jenisnya (query, browsing, dll.) dan permintaan sumber daya dari masing-masing 72.000 transaksi = waktu CPU rata-rata per transaksi dan jumlah rerata operasi Input / Output per transaksi.
- Beban Kerja Jaringan (Network): kebutuhan kapasitas jaringan, dihitung menggunakan kalkulasi sederhana, seperti perkalian rerata jumlah request per detik dengan rerata ukuran file/dokumen yang diakses. Masih ditambahkan beban overhead protocol sebesar 20 persen. Sementara latency, tidak diperhitungkan. Sehingga bila ada 100 permintaan per detik untuk mengakses file sebesar 10 KB, dibutuhkan bandwidth 100 x (10 x 1024 x 8 bit) = 8.192.000 bps atau 8,192 Mbps.
- Tingkat latency harus diperhitungkan di dalam analisis karakteristik dan beban kerja. Semakin kompleks interaksi antara client dengan server, pada dasarnya akan menambah parameter perhitungan. Misalnya ketika beban kerja meningkat dan terjadi antrian,
- Setiap pengembang Aplikasi Berbasis Webharus dapat menentukan berapa ukuran – misalnya file – yang paling efisien untuk diproses dengan menggunakan sumber daya yang paling minimal.
- Untuk meningkatkan kapasitas komputasi dapat berupa kapsitas server yang perlu ditambah menambahkan mirror load balancing.

d. Metrik yang digunakan untuk mengukur kinerja antara lain *response time throughput, error rate, availability*. Misalnya, waktu respons untuk halaman statis tidak boleh melebihi 0,5 detik dan waktu query database tidak melebihi 1 detik per 6 HTTP request. Jika beban berlipat gKita menjadi 12 HTTP request per detik, kinerja tidak turun di bawah 30 persen dan maksimal penolakan request hanya 1 persen saat peak, utilisasi harus di bawah 80%.

e. Untuk mengukur kinerja komputasi dapat menggunakan metodologi alternatif yaitu benchmarking, membandingkan kinerja 2 sistem yang berbeda. Matriks tolok ukur yang populer adalah WebStone dan SPECweb. Kedua aplikasi pengukuran kinerja ini mensimulasikan browser dan melakukan sejumlah request ke server sesuai karakteristik beban kerja yang ditentukan, menerima respons dari server, dan akan mencatat hasil pengukuran (response time, query, latency).

f. Model untuk Kinerja Aplikasi Berbasis Web:

   i. Untuk membangun model kinerja Aplikasi Berbasis Web, adai dua opsi: model simulasi atau analitik. Model simulasi lebih memakan waktu untuk dijalankan dan memerlukan lebih banyak parameter daripada model analitik. Khusus untuk tujuan perencanaan kapasitas, biasanya lebih baik menggunakan model yang lebih cepat dan tingkat yang lebih tinggi, seperti model analitik.

   ii. Model kinerja Aplikasi Berbasis Web, pada dasrnya menggunakan teori antrian yang diterapkan pada CPU, disk, LAN internal, router, dan jaringan. Hasil pengukuran ini bisa digunakan untuk prediksi kinerja setelah memperkirakan evolusi beban kerja. Misalnya, menggunakan model kinerja untuk memprediksi response time ketika request rate meningkat hingga sebesar 60 persen.

   iii. Dalam teori antrian proses komputasi ini, salah satu faktor yang menentukan adalah waktu tunggu proses atau yang disebut dengan latency. Yaitu total waktu yang dihabiskan selama menunggu proses di CPU, RAM, disk, bus, LAN, router dan pengiriman data berlangsung. Contoh, satu request file gambar memerlukan waktu proses 320 ms (0,320 detik). Tingkat request adalah 1,5 per detik. Maka Udisk = 1,5 x 0,320 = 0,48 = 48 persen adalah tingkat efisiensi sumber daya harddisk, sedangkan latencynya adalah 0,32 / (1 - 0,48) = 0,615 detik.

   iv. Dengan kata lain, proses request file gambar menghabiskan waktu 0,615 detik di disk drive. Karena 0,320 detik dihabiskan untuk proses di disk drive, sisanya 0,295 detik (0,615 - 0,320) dihabiskan untuk antrian disk atau latency. Dengan kata lain, pemanfaatan yang tinggi akan menyebabkan waktu tunggu yang tinggi juga. Namun pertumbuhan latency tidak selalu linier dengan tingkat pemanfaatannya. Misalnya, jika tingkat request meningkat 60 persen menjadi 2,4 request per detik, pemanfaatan disk drive tumbuh menjadi 0,768 (2,4 x 0,320) dan latency menjadi 0,320 / (1 - 0,768) = 1,38 detik, terjadi peningkatan 224 persen.

g. Memprediksi Beban Kerja Aplikasi Berbasis Web melalui:

   i. Memperkirakan jumlah pengunjung Aplikasi Berbasis Web untuk merencanakan kapasitas yang lebih memadai

   ii. Beban kerja yang diharapkan selama musim liburan

   iii. Beban kerja ketika ada penambahan fitur dan layanan baru serta peningkatan volume transaksi.

iv. Metoda prediksi dapat kuantitatif atau kualitatif. Metoda kuantitatif sangat bergantung pada keberadaan data historis untuk memperkirakan nilai parameter beban kerja di masa depan. Sedangkan pendekatan kualitatif adalah proses subyektif, berdasarkan penilaian, intuisi, pendapat ahli, analogi sejarah, pengetahuan komersial, dan informasi terkait lainnya. Strategi peramalan harus menggabungkan teknik kuantitatif dan kualitatif.

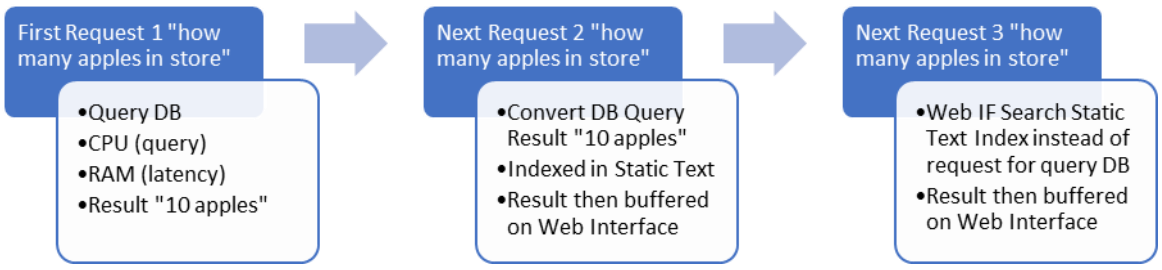
h. Basis metrik tingkat utilisasi yaitu request per second/minute (RPS/RPM) dimana 100 adalah peak dan 80 adalah maximum treshold (high) dan diantara 20-60 adalah (normal), sedangkan di bawah 20 dianggap beban rendah (low). Apabila kondisinya sering berada di atas maximum treshold (high), maka terjadi overload.

i. Jika di bawah beban terendah (low), maka terjadi inefisiensi maka perencanaan kapasitas harus dikoreksi.

j. Memaksimalkan konfigurasi Database untuk fitur akselerasi proses seperti tools indexing dan cache, sekalipun konsekuensinya, diperlukan tambahan RAM untuk menempatkan tabel index dan cache untuk mengurangi latency– teorinya, transfer rate I/O RAM lebih tinggi (cepat) dibanding bila tabel index dan cache ditempatkan di media penyimpan (disk) walaupun sudah SSD.

k. Untuk menjaga kinerja dalam menghadapi latency, pengembang aplikasi Aplikasi Berbasis Web harus menggunakan metoda mengubah hasil query ke database – misalnya proses search data tertentu ke dalam bentuk teks statik dan dijadikan buffer. Ketika ada request ke obyek sama berulang, web interface mengarahkan langsung ke buffer bukan melakukan query ke DB. Sehingga, penggunaan CPU, RAM dapat dihemat. Berikut gambarannya:

Gambar 3.4. Model Query Statik



l. Untuk menjamin tabel Static Index tetap up to date, maka Web Interface harus dikonfigurasi melakukan request query langsung ke database secara periodik dengan memanfaatkan waktu off peak.

III.2 STANDAR PENGENDALIAN KEAMANAN WEB SERVER

STANDAR
PENGENDALIAN KEAMANAN WEB SERVER

BADAN SIBER DAN SANDI NEGARA

2019

A. TUJUAN

1. Untuk memberikan arahan bagi instansi dalam melakukan pengendalian keamanan Web Server terutama pada proses instalasi dan konfigurasi
2. Memastikan Web Server yang dimiliki oleh instansi terjaga keamanannya, baik dari aspek Confidentiality, Integrity, maupun Availibility.

B. RUANG LINGKUP

1. Pengendalian Dalam Instalasi & Konfigurasi Sistem Operasi
2. Pengendalian Dalam Instalasi & Konfigurasi Web Server
3. Pengendalian Web Bots/Crawler/Spider

C. ISTILAH DAN DEFINISI

1.	*LAN (virtual LAN)*	:	adalah suatu model jaringan yang membagi jaringan secara logikal ke dalam beberapa lan yang berbeda. VLAN tidak terbatas pada kondisi fisik jaringan seperti pada LAN, vlan dikonfigurasi secara virtual tanpa harus melihat kondisi peralatan. Oleh sebab itu, VLAN memiliki fleksibilitas di dalam pengaturan jaringan dan memudahkan administrator jaringan dalam membagi jaringannya sesuai dengan fungsi dan kebutuhan keamanan jaringan tersebut
2.	*Lightweight Directory Access Protocol [LDAP]*	:	Adalah suatu protocol untuk mengakses directory secara ringan. Disebut ringan karena LDAP menggunakan jaringan internet yang penggunaan paket-paketnya sangat ringan. LDAP ini merupakan bagian dari Internet Protocol. LDAP ini digunakan untuk mengakses suatu directory misalnya directory telepon, directory email suatu organisasi. Pada LDAP ini tidak hanya membaca informasi, tetapi juga bisa menambah dan mengupdate informasi yang ada directory tersebut. LDAP juga sudah dilengkapi SASL (Simple Authentication and Security Layer) untuk memerika dan memastikan apakah suatu user berhak dan diperbolehkan masuk atau tidak. Karena itulah LDAP juga banyak digunakan untuk 'single sign on', yaitu dengan sekali sign-on, user dapat mengakses berbagai aplikasi yang telah disediakan

3.	*Karberos*		Adalah protokol otentikasi jaringan komputer yang bekerja berdasarkan tiket untuk memungkinkan node berkomunikasi melalui jaringan yang tidak aman untuk membuktikan identitas mereka satu sama lain secara aman.
4.	*Secure Shell*	:	Adalah sebuah protokol jaringan kriptografi untuk komunikasi data yang aman dan diberikan akses untuk melakukan perintah dari jarak jauh (remote) antara dua jaringan komputer. SSH ini berfungsi untuk mengakses ke dalam server dari perangkat apa saja yang mendukung penggunaan SSH ini.
5.	Proxy	:	adalah suatu sistem yang memungkinkan kita untuk bisa mengakses jaringan internet menggunakan IP yang berbeda dengan yang diterima oleh perangkat.
6.	Instansi	:	Kementerian/Lembaga, Instansi pusat atau daerah.
7.	*Password*	:	Kata sandi yang digunakan bersamaan dengan *username* (*sign on/sign in/log-on/log-in*) oleh pemilik yang sah sebelum melakukan koneksi/akses ke sistem komputer.

D. REFERENSI

1. Pedoman Keamanan Web Server: Direktorat Jenderal Aptika - Kemkominfo

E. STANDAR

1. Pengendalian Dalam Instalasi & Konfigurasi Sistem Operasi

a. *Patch* dan *Upgrade* Sistem Operasi Web Server

i. Menjaga server tidak terhubung dengan jaringan lain tapi dengan suatu jaringan terisolasi hingga seluruh *patch* telah dipindahkan ke server melalui alat *out-of-band media* (dalam bentuk CD/DVD)

ii. Menempatkan server dalam suatu *virtual local area network* (VLAN) atau segmen jaringan lain yang sangat membatasi tindakan apa yang dapat dijalankan oleh host pada server tersebut dan komunikasi apa yang dapat mencapai host,yaitu hanya memperbolehkan melakukan patch dan mengkonfigurasi host. Instansi yang berencana menggunakan VLAN harus memastikan bahwa VLAN dikonfigurasi dengan tepat dan setiap perubahan konfigurasi apapun diverifikasi dengan hati-hati.

iii. Melakukan pengujian terhadap patch yang akan diimplementasikan ke Web server dalam suatu sistem lain yang identik dan sudah dikonfigurasi.

b. Menonaktifkan Layanan dan Aplikasi Yang Tidak Perlu

i. Web server harus berada pada suatu host satu fungsi (*single purpose host*) yang sudah ditetapkan.

ii. Menginstalasi konfigurasi web server pada pilihan "instalasi minimum" dengan menghilangkan layanan-layanan yang tidak perlu diantaranya:

- Layanan sharing *file* dan printer sharing, misalnya Windows Network berbasis Input/Output System [NetBIOS] sharing *file* dan printer, Network *File* System [NFS], *File*Transfer Protocol [FTP]
- Program remote control dan remote access, khususnya bagi yang tidak mengenkripsi jalur komunikasinya (misalnya, Telnet). Jika suatu remote control atau remote acces benar-benar dibutuhkan dan komunikasinya tidak dienkripsi dengan kuat, harus diberi tunnel (saluran) melalui protokol yang menyediakan enkripsi, seperti secure shell (SSH) atau IP Security (IPSec).
- Layanan direktori,misalnya Lightweight Directory Access Protocol [LDAP], Kerberos, Network Information System [NIS]
- Layanan email (misalnya Simple Mail Transfer Protocol [SMTP])
- Language compilers dan libraries
- Tools pengembang sistem
- Tool dan utilities manajemen sistem dan jaringan, termasuk Simple Network Management Protocol (SNMP).

c. Konfigurasi Otentikasi Pengguna Sistem Operasi

i. Menghapus atau menonaktifkan akun dan grup default yang tidak dibutuhkan atau noninteraktif

Konfigurasi default OS seringkali termasuk akun guest (dengan atau tanpa password), akun level administrator atau root,dan akun yang berkaitan dengan layanan-layanan lokal dan jaringan dimana nama dan password untuk akun-akun tersebut sangat populer. Sangat penting untuk menghilangkan atau menonaktifkan akun yang tidak perlu untuk meminimalisasi penggunaannya oleh penyerang, termasuk guest akun pada komputer yang mengandung informasi sensitif. Jika tidak ada kebutuhan untuk mempertahankan suatu guest atau grup akun, batasi akses dengan ketat dengan mengubah password.. Untuk sistem Unix/Linux, menonaktifkan login shell atau menyediakan suatu login shell dengan fungsi NULL (misalnya /bin/false).

ii. Membuat grup-grup pengguna (cocok untuk jumlah pengguna yang besar) dengan membagi para pengguna kedalam grup-grup yang tepat. Kemudian menetapkan hak untuk grup, sebagaimana yang didokumentasikan dalam rencana distribusi.

iii. Membuat Akun Pengguna—Merencanakan distribusi dengan mengidentifikasikan siapa yang akan diberi wewenang untuk menggunakan setiap komputer dan layanan-layanannya.

iv. Mengecek Kebijakan Password Organisasi—Atur password akunsecara tepat. Kebijakan ini harus mencakup hal-hal berikut:

- Ukuran—Panjang minimum suatu password. Menetapkan panjang minimum paling sedikit delapan karakter.
- Kompleksitas—Disyaratkan campuran karakter. Password disyaratkan yang mengandung huruf besar dan huruf kecil dan paling sedikit suatu karakter non-alphabet, dan tidak merupakan kata dari kamus.
- Durasi Waktu—Seberapa lama suatu passwordboleh tetap tidak berubah. Para pengguna disyaratkan untuk mengubah passwordmereka secara periodik. Password level administrator atau root sebaiknya diubah tiap 30 hingga 120 hari. Periode untuk password level pengguna sebaiknya ditentukan oleh kombinasi panjang dan kompleksitas passworddengan tingkat sensitivitas informasi yang dilindungi.
- Penggunaan Kembali— Suatu password dimungkinkan untuk digunakan kembali dengan tidak mengizinkan hanya menambah karakter pada password yang telah digunakan, misalnya, password yang asli adalah ”password” diubah menjadi ”Xpassword” atau ”Passwordx”.
- Otoritas—Pihak yang diijinkan untuk mengubah atau mengatur kembali password dan pembuktian seperti apa yang diperlukan sebelum memulai suatu perubahan.
- Keamanan Password—Password harus diamankan, seperti halnya tidak menyimpan password yang tidak terenkripsi pada mail server, dan mensyaratkan para administrator untuk menggunakan password yang berbeda untuk akun administrasi email mereka dari akun adminstrasi mereka lainnya.

v. Mengkonfigurasi komputer untuk mencegah untuk menebak kata sandi (*Password Guessing*)

- Merupakan hal yang relatif mudah bagi seorang pengguna yang tidak sah untuk mencoba mendapatkan akses ke suatu komputer dengan menggunakan perangkat lunak tool otomatis yang mencoba semua password.
- Mengkonfigurasi OS untuk meningkatkan periode waktu antara percobaan login dari setiap percobaan yang tidak berhasil.
- Alternatif lainnya adalah menolak login setelah sejumlah percobaan gagal (misal, tiga). Atau akun dikunci untuk suatu periode waktu tertentu (contohnya 30 menit) atau hingga seorang pengguna dengan otoritas yang tepat mengaktifkannya kembali.

vi. Menginstal dan mengkonfigurasikan mekanisme keamanan lain untuk menguatkan otentikasi

Mempertimbangkan untuk menggunakan mekanisme otentikasi lain seperti *biometric, smart card,* sertifikat *client/server*, atau *sistem one-time password*. Mekanisme ini mungkin lebih sulit dan mahal untuk diimplementasikan, namun mungkin dapat dibenarkan dalam beberapa keadaan. Saat mekanisme otentikasi dan peralatan seperti itu digunakan, kebijakan organisasi harus menyesuaikan. Dimana beberapa kebijakan instansi mungkin perlu mensyaratkan pengggunaan mekanisme otentikasi yang kuat.

d. Instalasi dan konfigurasi kontrol keamanan tambahan

i. OS seringkali tidak menyertakan seluruh kontrol keamanan yang dibutuhkan untuk mengamankan OS, layanan-layanan, dan aplikasi-aplikasi.

ii. Dalam kasus seperti itu, para administrator perlu memilih, menginstal, dan mengkonfigurasikan perangkat lunak tambahan untuk menyediakan kontrol yang tidak ada.

iii. Kontrol-kontrol yang dibutuhkan secara umum adalah sebagai berikut:

- Perangkat lunak Anti-malware, seperti perangkat lunak anti-virus, perangkat lunak anti-spyware, dan rootkit detector, untuk melindungi OS lokal terhadap malware, mendeteksi dan memberantas munculnya suatu infeksi.
- Perangkat lunak pendeteksi dan pencegah penyerangan berbasis host, untuk mendeteksi serangan-serangan yang dijalankan terhadap Web server, termasuk serangan DoS.
- Firewall berbasis host, untuk melindungi server dari akses yang tidak sah.
- Perangkat lunak manajemen patch untuk memastikan bahwa kerentanan diatasi secara tepat. Perangkat lunak manajemen patch hanya dapat digunakan untuk mengaplikasikan patch atau juga untuk mengidentifikasi kerentanan baru dalam OS, layanan-layanan, dan aplikasi-aplikasi Web Server.

e. Konfigurasi *Permission* terhadap file/direktori

i. OS Web server harus membatasi file-file yang dapat diakses terkait proses-proses layanan Web server dengan memberikan akses read-only (hanya baca) terhadap file-file yang diperlukan untuk melakukan layanan dan tidak memiliki akses terhadap file-file lainnya, seperti file log server.

ii. Memastikan direktori dan file-file diluar struktur file konten web tidak dapat diakses, sekalipun jika para pengguna melakukan browsing langsung dengan cara mengakses URL dari file-file tersebut atau melalui *directory traversal attacks* (serangan lintas direktori) terhadap proses Web server.

2. Pengendalian Dalam Instalasi & Konfigurasi Software Web Server

a. Pengendalian Instalasi & Konfigurasi

i. Prinsip dasar instalasi software web server adalah menginstal layanan-layanan yang dibutuhkan Web server dan mengurangi kerentanan apapun yang diketahui.

ii. Aplikasi, layanan, maupun script yang tidak perlu harus dihilangkan segera setelah proses instalasi selesai.

iii. Menginstal perangkat lunak Web server pada dedicated host atau pada Sistem Operasi dedicated guest jika menggunakan virtualisasi.

iv. Menginstal patch atau upgrade perangkat lunak Web Server untuk mengatasi kerentanan yang telah diketahui.

v. Menyediakan media penyimpanan secara fisik atau partisi secara logic untuk konten Web, yang terpisah dari OS dan aplikasi Web server.

vi. Menghapus atau menonaktifkan layanan-layanan yang diinstall oleh aplikasi Web server namun tidak dibutuhkan, misal gropher, FTP, administrasi remote.

vii. Menghapus atau menonaktifkan semua akun login default yang tidak dibutuhkan, yang tercipta pada saat instalasi Web server.

viii. Menghapus semua dokumentasi manufaktur dari server.

ix. Menghapus semua file contoh atau tes dari server, termasuk script dan executable code.

x. Menerapkan template keamanan yang sesuai atau script untuk memperkuat keamanan ke server.

xi. Mengkonfigurasikan kembali HTTP service banner (dan yang dibutuhkan lainnya untuk tidak mempublikasikan atau membuat pemberitahuan mengenai tipe dan versi Web server serta OS).

xii. Sebaiknya memberikan nama direktori, lokasi direktori dan nama file yang tidak Standar (umum). Karena banyak tools serangan dan worm yang menjadikan Web server sebagai targetnya, hanya dengan mencari file dan direktori dalam lokasi default.

b. Pengendalian Akses File/Direktori

i. Menetapkan access previleges bagi masing-masing individu untuk mengakses file, perangkat dan sumber daya lainnya.

ii. File-file yang secara umum dapat diterapkan kontrol aksesnya, yaitu:

- File perangkat lunak aplikasi dan konfigurasi
- File yang berkaitan langsung dengan mekanisme pengamanan:

- File yang berisi nilai hash password dan file lainnya yang digunakan dalam otentikasi
- File yang berisi informasi otorisasi yang digunakan untuk mengendalikan akses
- Material kunci kriptografis yang digunakan dalam layanan konfidensialitas, integritas dan non-repudiasi
- File log server dan audit sistem
- File perangkat lunak sistem dan konfigurasi
- File konten Web

iii. Untuk pengendalian akses file konten web melalui langkah – langkah:

- Menentukan suatu hard drive atau partisi logik tunggal yang diperuntukkan bagi konten Web dengan membuat subdirektori terkait khusus untuk file konten web, termasuk grafik. Namun tidak memuat script dan program-program lain.
- Menetapkan suatu pohon direktori tunggal khusus untuk seluruh script atau program eksternal yang dieksekusi sebagai bagian dari konten Web (misalnya, CGI, Active Server Page [ASP], PHP, ..NET, .JSP).
- Menonaktifkan eksekusi script yang tidak secara khusus berada dibawah kontrol dari akun administratif. Tindakan ini dilakukan dengan pembuatan dan pengontrolan akses terhadap suatu direktori terpisah yang dimaksudkan untuk berisikan script yang sah.
- Menonaktifkan penggunaan *hard links* atau *symbolic links.*
- Mendefinisikan suatu matriks akses konten web yang lengkap dengan mengidentifikasi folder dan file dalam dokumen Web server yang harus dibatasi dan oleh siapa.

iv. Pengendalian URI dan Cookies

- Dilarang menyembunyikan data sensitif seperti username dan password atau sumber daya yang tersembunyi dari server lainnya didalam URI.
- Cookie tidak mengandung data yang dapat digunakan secara langsung oleh seorang penyerang (misal, nama pengguna, password).
- Cookie dapat digunakan jika suatu keadaan yang memaksa untuk mengumpulkan data pada situs, dan harus melalui persetujuan, pemberitahuan, dan proteksi keamanan yang baik.

3. Pengendalian Web Bots/Crawler/Spider

Web *bots* (juga dikenal sebagai *crawler* atau *spider*) merupakan aplikasi software yang digunakan untuk mengumpulkan, menganalisis, dan meng-indeks konten Web. Web bots digunakan oleh berbagai organisasi untuk berbagai tujuan.

a. Jenis - jenis Web Bots diantaranya:

   i. MSNBot, Slurp, dan Googlebot dapat menganalisis, meng-indeks, dan mencatat situs Web untuk mesin pencari (search engines) Web seperti halnya Windows Live Search, Yahoo! dan Google.

   ii. Mediabot digunakan oleh Google untuk mengAnalisis konten yang disajikan oleh suatu halaman AdSense sehingga iklan - iklan yang relevan secara konteks akan disediakan.

   iii. Hyperlink “validator” digunakan oleh Webmaster untuk memvalidasi hyperlink secara otomatis pada situs Web mereka.

   iv. Email Siphon dan Cherry Picker merupakan bots yang didesain secara khusus untuk bergerak dengan pelan pada situs Web guna mendapatkan alamat electronic mail(e-mail) untuk ditambahkan pada spam mailing list.

   v. Beberapa spam bots bergerak dengan pelan dalam situs Web untuk mencari formulir login yang digunakan untuk menciptakan alamat email gratis yang merupakan tempat asal pengiriman spam atau untuk spam blog, guestbook, wikis, dan forum - forumuntuk mendorong urutan search engine dari suatu situs Web tertentu.

   vi. Screen scrapers mendapatkan kembali konten dari situs Web untuk meletakkan suatu copy pada server lain. Copy-copy tersebut dapat digunakan untuk pishing atau untuk berusaha menghasilkan ad revenue (bayaran dari iklan) dengan membuat para pengguna mengunjungi copy tersebut.

b. Penanganan Web Bots berupa:

   i. Membuat suatu file teks terang yang bernama ”robots.txt”. File harus selalu memiliki nama ini, dan harus berada dalam direktori dokumen root Web server. Hanya satu file yang diperbolehkan per situs Web. Karena filerobots.txt merupakan salah suatu Standar yang digunakan oleh para pemrogram bot, sehingga bot tak dikenal (seperti EmailSiphon dan Cherry Picker) seringkali mengabaikan file ini. Filerobots.txt merupakan suatu file teks sederhana yang mengandung beberapa kata kunci dan spesifikasi file. Setiap baris file dapat saja kosong atau berisi kata kunci tunggal dan informasi yang terkait. Kata kunci digunakan untuk memberitahu robot bagian mana dari situs Web yang tidak disertakan. Kata kunci yang diperbolehkan antara lain User-agent dan disallow, misalkan untuk disallow suatu bot tertentu (dalam hal ini Googlebot) dari pemeriksaan suatu halaman Web tertentu:

   User-agent:GoogleBot

   Disallow:tempindex.htm

ii. Perlindungan password merupakan jalan satu-satunya yang dapat diKitalkan untuk meniadakan noncompliant bots atau para pengguna yang ingin tahu. Seringkali, spambots mengabaikan robots.txt dan mencari alamat email pada situs Web dan/atau format dimana mereka dapat menambahkan konten yang berkaitan dengan spam.

c. Teknik penanganan untuk mengurangi jumlah spam, yaitu:

i. Memblokir formulir pengajuan yang menggunakan kata kunci yang berkaitan dengan spam.

ii. Menggunakan kata kunci rel=”nofollow”dalam semua link yang diajukan, yang akan mengakibatkan search engineuntuk menghilangkan link pada algoritma pageranking-nya, yang secara langsung mempengaruhi tujuan dari suatu spambot.

iii. Mensyaratkan pengusul untuk memecahkan suatu Automated Public Turing Test To Tell Computers And Humans Apart (CAPTCHA) sebelum diijinkan mengumpulkan konten.

III.3 STANDAR PENEMPATAN APLIKASI BERBASIS WEB & INFRASTRUKTUR PENDUKUNG

STANDAR

PENEMPATAN APLIKASI BERBASIS WEB & INFRASTRUKTUR PENDUKUNG

BADAN SIBER DAN SANDI NEGARA

2019

A. TUJUAN

1. Untuk memberikan arahan bagi instansi – instansi yang memiliki sistem web statis, web dinamis, web dinamis dengan aplikasi transaksi, serta web dinamis dengan aplikasi transaksi yang membutuhkan tingkat availibility tinggi, sehingga memiliki standar minimum dalam penyediaan infrastruktur pengamanan Aplikasi Berbasis Web.
2. Untuk memastikan aspek *Confidentiality, Integrity, dan Availibility* aplikasi Aplikasi Berbasis Webinstansi terjaga, baik dalam konteks dikelola sendiri maupun pihak ketiga.

B. RUANG LINGKUP

Penempatan dengan model :

1. Dalam Data Center yang dimiliki sendiri secara penuh (*Dedicated DC*)
2. Dalam Data Center Bersama (*Shared Ruang DC*)
3. Dalam Jaringan Awan (*Cloud Network*)
4. Dalam Data Center yang dimiliki sendiri secara penuh disertai Jaringan Awan

C. ISTILAH DAN DEFINISI

1. Web Statis :
   - Lebih cenderung bersifat informative dimana pengguna/pengunjung hanya dapat melihat – lihat informasi di Aplikasi Berbasis Webtersebut, tidak bisa mengisi data.
   - Interaksi yang terjadi antara pengguna dan server hanyalah seputar pemrosesan link saja.
   - Halaman-halaman Aplikasi Berbasis Webtidak memliki database, data dan informasi tidak berubah-ubah kecuali diubah sintaksnya. Dokumen Aplikasi Berbasis Webyang dikirim kepada client akan sama isinya dengan yang ada di web server.
   - Untuk menambah halaman harus menambah file baru.
2. Web Dinamis :
   - Pengguna dapat mengupdate informasi Aplikasi Berbasis Weblangsung dari Aplikasi Berbasis Webnya.
   - Mengubah tampilan Aplikasi Berbasis Webmelalui Content Management System (CMS)
   - Menggunakan database yang digunakan untuk menampung banyaknya data, sehingga Aplikasi Berbasis Webtinggal mengambil data dari database.

3.	Web Dinamis dengan Aplikasi Transaksional		• Adanya fasilitas akses ke suatu aplikasi untuk konsumsi publik atau internal (internet atau intranet atau menggunakan VPN) • Umumnya terkait dengan data – data yang konfidensial dan sensitif
4.	Web Dinamis dengan Aplikasi Transaksional dan tingkat availibility	:	Web Dinamis dengan Aplikasi Transaksional dimana dalam momen tertentu atau secara rutin aplikasi itu diakses oleh banyak orang dan sangat intensif, sehingga diminimalisasi terjadinya downtime.
5.	Proxy	:	Adalah suatu sistem yang memungkinkan kita untuk bisa mengakses jaringan internet menggunakan IP yang berbeda dengan yang diterima oleh perangkat.
6.	Instansi	:	Kementerian/Lembaga, Instansi pusat atau daerah.
7.	*Password*	:	Kata sandi yang digunakan bersamaan dengan *username* (*sign on/sign in/log-on/log-in*) oleh pemilik yang sah sebelum melakukan koneksi/akses ke sistem komputer.
8.	*Cloud computing*	:	Adalah penggabungan teknologi komputerisasi dan internet, dimana data mulai dari skala kecil hingga besar tersimpan di server internet, yang memungkinkan kita dapat mengakses data kita dari berbagai lokasi dan melalui berbagai platform.
9.	Software as a Service		Salah satu layanan cloud yang menyediakan lisensi software dan delivery model dimana pengguna dapat memanfaatkan software tersebut dimanapun dengan menggunakan device apapun melalui koneksi internet. Pengguna tidak perlu lagi melakukan install, update, atau menangani masalah pada software yang digunakan karena semua hal tersebut telah dikelola oleh vendor, pengguna hanya tinggal menggunakan layanan yang disediakan. Contoh: Dropbox, Google Apps, MTarget, Salesforce, Cisco WebEx.
10.	Platform as a Service		Salah satu layanan Cloud yang menyediakan platform untuk dimanfaatkan pengguna dalam membuat aplikasi diatasnya. Pengguna dapat membangun aplikasi, upload aplikasi, testing, dan mengatur konfigurasi. Contoh: Amazon Web Service, Microsoft Azure, Facebook, dll.

11.	Infrastructure as a Service	Salah satu layanan Cloud yang paling fleksibel karena pengguna memiliki kendali penuh terhadap infrastruktur yang digunakan, mulai dari server cloud, jaringan, sistem operasi, hingga penyimpanan/storage. Dalam IaaS, pengguna juga dapat membuat "pusat data virtual" di cloud dan memiliki akses ke seluruh data tanpa harus memiliki hardware sendiri. DigitalOcean, Linode, Rackspace, Amazon Web Services (AWS), Cisco Metapod, Microsoft Azure, Google Compute Engine (GCE)

D. REFERENSI

1. OWASP: Application Security Verification Standard 4.0

E. STANDAR

1. Gambaran Umum

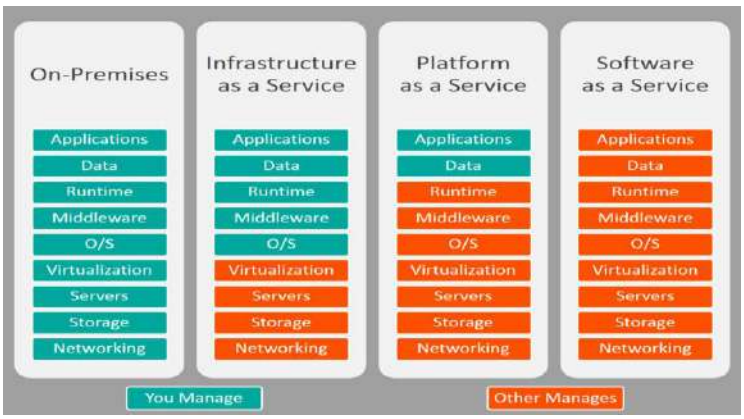
a. Instansi dapat menempatkan Server dan Infrastruktur Pendukungnya dengan berbagai model diantaranya:

i. Server Web & Infrastruktur Pendukungnya dimiliki sendiri secara penuh (*on - premises*) & termasuk pengelolaannya.

ii. Komputasi Awan berupa:

- Infrastruktur sebagai Service (*Infrastructures as a Service)*
- Platform sebagai Service *(Platform as a Service)*
- Perangkat Lunak sebagai Service *(Software as a Service)*

Gambar 3.5 Beberapa Model Penempatan Server & Infrastruktur Pendukungnya



b. Untuk keperluan aplikasi Aplikasi Berbasis Web, tidak direkomendasikan Instansi menggunakan *Software as a Service*, kecuali untuk kondisi yang sangat mendesak. Dan segera mengubahnya kembali ke model yang lain.

2. Server Web & Infrastruktur dimiliki sendiri secara penuh (*on – premises*)

a. Bentuk penempatan: Dalam Data Center yang dimiliki sendiri secara penuh (*Dedicated DC*)

i. Peruntukan: Web Dinamis dengan Aplikasi Transaksi non availibility

1. Persyaratan fisik minimal

- Menggunakan sistem akses fisik dan prasyaratnya

2. Persyaratan perangkat minimal

- Menggunakan UTM/NGFW dan prasyaratnya
- Menggunakan Sistem Proteksi 0-hari dan persyaratannya
- Menggunakan FIM dan prasyaratnya
- Menggunakan SIEM dan prasyaratnya
- Menggunakan NBA dan prasyaratnya
- Menggunakan koneksi-aman untuk pihak ketiga
- Bila menggunakan sistem server virtual harus menggunakan firewall virtual dan prasyaratnya
- Menggunakan koneksi-aman/proxi-aman untuk akses lebih dari laman awal

ii. Peruntukan: Web Dinamis & Statik

1. Persyaratan fisik minimal:

- Menggunakan sistem akses fisik dan prasyaratnya

2. Persyaratan perangkat minimal:

- Menggunakan UTM/NGFW dan prasyaratnya
- Menggunakan FIM dan prasyaratnya
- Menggunakan koneksi aman untuk pihak-ketiga

b. Bentuk penempatan: Sewa Rak dalam Data Center Bersama

i. Peruntukan: Web Dinamik dengan Aplikasi Transaksi Non Availibility

1. Persyaratan fisik minimal:
   - Data Center bersama menggunakan sistem akses fisik dan prasyaratnya
2. Persyaratan perangkat minimal:
   - Menggunakan UTM/NGFW dan prasyaratnya
   - Menggunakan Sistem Proteksi 0-hari dan persyaratannya
   - Menggunakan FIM dan prasyaratnya
   - Menggunakan SIEM dan prasyaratnya
   - Menggunakan NBA dan prasyaratnya
   - Menggunakan koneksi-aman untuk pihak ketiga
   - Bila menggunakan sistem server virtual harus menggunakan firewall virtual dan prasyaratnya
   - Menggunakan koneksi-aman/proxi-aman untuk akses lebih dari laman awal

ii. Peruntukan: Web Statik dan Web Dinamis

1. Persyaratan fisik minimal:
   - Data Center bersama menggunakan sistem akses fisik dan prasyaratnya
2. Persyaratan perangkat minimal:
   - Menggunakan UTM/NGFW dan prasyaratnya
   - Menggunakan FIM dan prasyaratnya
   - Menggunakan koneksi-aman untuk pihak ketiga

c. Bentuk penempatan: Kolokasi server dalam Data Center Bersama

i. Peruntukan: Web Dinamis dengan Transaksi Aplikasi

1. Persyaratan fisik minimal:
   - Data Center bersama menggunakan sistem akses fisik dan prasyaratnya
2. Persyaratan perangkat minimal:
   - Bila menggunakan sistem server-virtual harus menggunakan firewall-virtual dan prasyaratnya
   - Menggunakan FIM dan prasyaratnya
   - Menggunakan Sistem Pengecoh dengan prasyaratnya
   - Menggunakan koneksi-aman/proxi-aman untuk akses

ii. Peruntukan: Web Statik dan Web Dinamis

1. Persyaratan fisik minimal:

- Datacenter bersama menggunakan sistem akses fisik dan prasyaratnya

2. Persyaratan perangkat minimal:

- Bila menggunakan sistem server-virtual harus menggunakan firewall-virtual dan prasyaratnya
- Menggunakan eksternal FIM dan prasyaratnya
- Menggunakan koneksi-aman/proxi-aman untuk akses

3. Server Web Berbasis Awan (Cloud Based)

a. Bentuk Servis: Hosting/Platform sebagai Servis (*Platform as a Service/PaaS*)

i. Peruntukan: Web dengan Transaksi Aplikasi

i. Persyaratan servis minimal:

- Menyewa firewall-virtual dan prasyaratnya
- Menggunakan eksternal FIM dan prasyaratnya
- Menggunakan Sistem Pengecoh dengan prasyaratnya
- Menggunakan koneksi-aman/proxi-aman untuk akses

ii. Peruntukan: Web Statik dan Web Dinamis

i. Persyaratan servis minimal:

- Menyewa firewall-virtual dan prasyaratnya
- Menggunakan eksternal FIM dan prasyaratnya
- Menggunakan koneksi-aman/proxi-aman untuk akses

b. Bentuk servis: Platform sebagai Servis (Platform as a Service/PaaS)

i. Peruntukan: Tidak disarankan untuk digunakan Web dengan transaksi ataupun Web Dinamis

ii. Peruntukan: Web Statik

i. Persyaratan servis minimal:

- Menggunakan eksternal FIM dan prasyaratnya
- Menggunakan koneksi-aman/proxi-aman untuk akses

c. Bentuk servis: Perangkat Lunak sebagai Servis (*Software as a Service/SaaS*)

i. Peruntukan: Tidak disarankan untuk digunakan Web dengan transaksi, Web Dinamis, ataupun Web Statik

4. Kombinasi Server Web yang dimiliki sendiri secara penuh + redundansi atau servis Server Web berbasis Awan

a. Bagian penempatan: Dalam Data Center yang dimiliki sendiri secara penuh

i. Peruntukan: Web dengan transaksi

i. Persyaratan fisik minimal:

- Menggunakan sistem akses fisik dan prasyaratnya

ii. Persyaratan perangkat minimal:

- Menggunakan UTM/NGFW dan prasyaratnya
- Menggunakan Sistem Proteksi 0-hari dan persyaratannya
- Menggunakan FIM dan prasyaratnya
- Menggunakan SIEM dan prasyaratnya
- Menggunakan NBA dan prasyaratnya
- Menggunakan koneksi-aman untuk pihak-ketiga
- Bila menggunakan sistem server-virtual harus menggunakan firewall-virtual dan prasyaratnya
- Menggunakan koneksi-aman/proxi-aman untuk akses lebih dari laman-awal

ii. Peruntukan: Web Statik dan Web Dinamis

i. Persyaratan fisik minimal:

- Menggunakan sistem akses fisik dan prasyaratnya

ii. Persyaratan perangkat minimal:

- Menggunakan UTM/NGFW dan prasyaratnya
- Menggunakan FIM dan prasyaratnya
- Menggunakan koneksi-aman untuk pihak-ketiga

b. Bagian penempatan: Sewa Rak dalam Data Center Bersama

i. Peruntukan: Web dengan transaksi

1. Persyaratan fisik minimal:

- Datacenter bersama menggunakan sistem akses fisik dan prasyaratnya

2. Persyaratan perangkat minimal:

- Menggunakan UTM/NGFW dan prasyaratnya
- Menggunakan Sistem Proteksi 0-hari dan persyaratannya
- Menggunakan FIM dan prasyaratnya
- Menggunakan SIEM dan prasyaratnya

- Menggunakan NBA dan prasyaratnya
- Menggunakan koneksi-aman untuk pihak-ketiga
- Bila menggunakan sistem server-virtual harus menggunakan firewall-virtual dan prasyaratnya
- Menggunakan koneksi-aman/proxi-aman untuk akses lebih dari laman-awal

ii. Peruntukan: Web Statik dan Web Dinamis

1. Persyaratan fisik minimal:
   - Datacenter bersama menggunakan sistem akses fisik dan prasyaratnya
2. Persyaratan perangkat minimal:
   - Menggunakan UTM/NGFW dan prasyaratnya
   - Menggunakan FIM dan prasyaratnya
   - Menggunakan koneksi-aman untuk pihak-ketiga

c. Bagian penempatan: Kolokasi server dalam Data Center Bersama

i. Peruntukan: Web dengan transaksi

1. Persyaratan fisik minimal:
   - Datacenter bersama menggunakan sistem akses fisik dan prasyaratnya
2. Persyaratan perangkat minimal:
   - Bila menggunakan sistem server-virtual harus menggunakan firewall-virtual dan prasyaratnya
   - Menggunakan FIM dan prasyaratnya
   - Menggunakan Sistem Pengecoh dengan prasyaratnya
   - Menggunakan koneksi-aman/proxi-aman untuk akses

ii. Peruntukan: Web Statik dan Web Dinamis

1. Persyaratan fisik minimal:
   - Datacenter bersama menggunakan sistem akses fisik dan prasyaratnya
2. Persyaratan perangkat minimal:
   - Bila menggunakan sistem server-virtual harus menggunakan firewall-virtual dan prasyaratnya
   - Menggunakan eksternal FIM dan prasyaratnya
   - Menggunakan koneksi-aman/proxi-aman untuk akses

d. Bagian servis: Infrastruktur sebagai Servis (*Infrastructure as a Service/IaaS*)

   i. Peruntukan: Web dengan transaksi aplikasi

      1. Persyaratan servis minimal:

         - Menyewa firewall-virtual dan prasyaratnya
         - Menggunakan eksternal FIM dan prasyaratnya
         - Menggunakan Sistem Pengecoh dengan prasyaratnya
         - Menggunakan koneksi-aman/proxi-aman untuk akses

   ii. Peruntukan: Web Statik dan Web Dinamis

      1. Persyaratan servis minimal:

         - Menyewa firewall-virtual dan prasyaratnya
         - Menggunakan eksternal FIM dan prasyaratnya
         - Menggunakan koneksi-aman/proxi-aman untuk akses

e. Bagian Servis: Platform sebagai Servis (*Platform as a Service/PaaS*)

   i. Peruntukan: Tidak disarankan untuk digunakan sebagai redundansi Web dengan transaksi ataupun Web Dinamis

   ii. Peruntukan: Web Statik

      1. Persyaratan servis minimal:

         - Menggunakan eksternal FIM dan prasyaratnya
         - Menggunakan koneksi-aman/proxi-aman untuk akses

f. Bagian servis: Perangkat Lunak sebagai Servis (*Software as a Service/SaaS*)

   i. Peruntukan: Perangkat Lunak sebagai Servis (*Software as a Service/SaaS)* tidak bisa digunakan sebagai redundansi

III.4 STANDAR PENGUJIAN APLIKASI BERBASIS WEB

STANDAR
PENGUJIAN KERENTANAN APLIKASI BERBASIS WEB

BADAN SIBER DAN SANDI NEGARA

2019

A. TUJUAN

a. Instansi dapat mengidentifikasi kelemahan keamanan suatu aplikasi, sistem komputer, atau suatu jaringan.
b. Jika celah kelemahan ditemukan dan dapat dibuktikan dengan beberapa Analisis resikonya, maka instansi dapat segera melakukan perbaikan sistem.

B. RUANG LINGKUP

a. Vulnerability Assessment
b. Penetration Testing

C. ISTILAH DAN DEFINISI

1. Penetration testing	:	Suatu kegiatan pengujian keamanan sebuah sistem, aplikasi, atau jaringan untuk mengetahui keamanan yang terdapat pada sistem atau aplikasi mempunyai celah keamanan sehingga dapat segera diperbaiki dengan melakukan patch.. Hal ini dilakukan agar keamanan yang terdapat pada suatu sistem atau aplikasi yang diuji menjadi semakin kuat. Selain melakukan pengujian, juga mendokumentasi tingkat keamanan dari sistem atau aplikasi yang diuji untuk selanjutnya dibuatkan laporan kepada lembaga/pimpinan
2. Instansi	:	Kementerian/Lembaga, Instansi Pusat dan Daerah
3. *Intranet*	:	Jaringan pribadi yang menggunakan protokol komunikasi internet protocol, dapat terhubung ke internet (tidak selalu), dan hanya dapat digunakan dalam lingkungan terbatas.
4. *Password*	:	Kata sandi yang digunakan bersamaan dengan username (sign on/sign in/log-on/log-in) oleh pemilik yang sah sebelum melakukan koneksi/akses ke sistem komputer.
5. *Patch*	:	Rutin program atau sekumpulan kecil instruksi yang biasanya dibuat sebagai solusi sementara untuk mengatasi atau memperbaiki permasalahan (bugs) pada program komputer dan sering dibuat dalam bentuk ’object code’ yang disisipkan ke dalam program yang akan dieksekusi.
6. Directory Traversal		Jenis eksploitasi HTTP yang digunakan oleh penyerang untuk mendapatkan akses tidak sah ke file dan direktori terbatas. Selain itu serangan lintasan direktori menggunakan perangkat lunak server web untuk mengeksploitasi mekanisme keamanan yang tidak memadai dan mengakses file dan direktori yang disimpan di luar folder root Aplikasi Berbasis Web.

7. Vulnerability Assessment	Adalah proses identifikasi dan kuantifikasi kerentanan keamananan pada suatu lingkungan keamanan sistem infomasi. Dapat diartikan juga sebagai suatu evaluasi mendalam terhadap keamanan sistem infomasi yang aktif digunakan.
8. Logic Bomb	Merupakan salah satu program jahat yang ditempelkan pada computer agar mameriksa suatu kumpulan kondisi di system. Ketikaa kondisi-kondisi yang dimaksud ditemui, logic mengeksekusi suatu fungsi yang menghasilkan aksi-aksi tak diotorisasi. Logic bomb menempel pada suatu program resmi yang diset meledak ketika kondisi-kondisi tertentu dipenuhi. Contoh kondisi-kondisi untuk memicu logic bomb adalah ada atau tidaknya file tertentu, hari tertentu dari minggu atau tanggal, atau pemakai menjalankan aplikasi tertentu. Begitu terpicu, bomb mengubah atau menghapus data atau seluruh file, menyebabkan mesin berhenti, atau menyebabkan kerusakan lain.

D. REFERENSI

1. OWASP Testing Guides Release 4.0

E. STANDAR

1. Penilaian Kerentanan Sistem (*Vulnerability Assessment*)

a. Kegiatan ini dilakukan untuk menghadapi, menghindari, membagi, dan mengurangi kerentanan sistem sampai batas toleransi risiko yang dapat diterima.

b. Instansi dalam periode tertentu harus melakukan tahapan penilaian kerentanan sistem berupa:\

. Penyusunan Katalog asset sistem informasi dan sumber daya lainnya mencakup seluruh sistem yang digunakan pada proses bisnis instansi

i. Menentukan nilai dan tingkat kepentingan terhadap aset. Hasil dalam bentuk daftar aset berdasarkan urutan yang dinilai paling tinggi.

ii. Identifikasi kerentanan keamanan dan potensi ancaman (dengan dampak terburuk) yang dapat terjadi pada setiap aset. Hasil dalam bentuk daftar ancaman berdasarkan tingkat dampak yang dinilai paling tinggi

iii. Mitigasi atau eliminasi ancaman dilakukan (atau direncanakan) terhadap ancaman yang memilik tingkat dampak dan aset bernilai paling tinggi.

2. Pengujian Aplikasi Berbasis Web (*Penetration Test*)

a. Penentuan Tujuan (*Identifying Objectives*)

Dalam tahapan ini, instansi yang meminta pengujian harus menyampaikan:

- tujuan dan ruang lingkup pengujian;
- skenario pengujian;
- batasan pengujian;
- teknik dan tools yang digunakan dalam pengujian

b. Pengumpulan/Pemindaian Informasi (*Gathering & Scanning Information*)

Mengumpulkan informasi sebanyak mungkin dari suatu Aplikasi Berbasis Web melalui mesin pencari atau melalui pendekatan secara aktif dengan perangkat yang dirancang untuk memetakan / mengumpulkan informasi dari sebuah Aplikasi Berbasis Web. Sebuah aplikasi sebagai alat teknis untuk mengumpulkan berbagai data lanjutan pada target yang telah kita tentukan. Pada tahapan ini data yang dicari lebih umum, yaitu mengenai sistem yang dimiliki.

c. Memperoleh Akses (*Gaining Access*)

Mendapatkan akses untuk mengambil alih kendali dari satu atau lebih perangkat jaringan/sistem untuk selanjutnya mengekstrak data dari target, untuk selanjutnya menggunakan perangkat tersebut untuk meluncurkan serangan pada target lainnya.

d. Mempertahankan Akses (*Maintaining Access*)

Membuat beberapa langkah-langkah yang diperlukan agar tetap berada di lingkungan target dengan tujuan untuk mengumpulkan data sebanyak mungkin. Pada fase ini, penyerang harus tetap dalam kondisi diam agar tidak dapat tertangkap ketika sedang menggunakan lingkungan host.

e. Penutupan Jejak (*Covering Track*)

Menutupi jejak terhadap setiap perubahan yang telah dilakukan, otorisasi yang telah ditingkatkan dan lain-lain. Semuanya harus kembali dalam kondisi non-recognition (tidak diakui) oleh seorang host administrator jaringan/sistem

f. Objek pengujian penetrasi meliputi:

- Layanan yang menggunakan koneksi internet (Aplikasi Berbasis Web, VPN endpoint, infrastruktur e-mail, extranet, dan lain-lain).
- Sistem internal atau servis yang terdapat di dalam jaringan (Active Directory, Exchange, dan lain-lain).
- Aplikasi mobile (iOs & Android), web, dan desktop.
- Jaringan internal.

g. JIka penetration test menggunakan pihak ketiga, sebelum melakukan penetration testing, ada disepakati kontrak antara pentester dengan instansi yang aplikasi atau sistemnya akan diuji, sehingga dalam konteks hukum bahwa kegiatan pentest merupakan kegiatan yang legal.

h. Jenis – jenis pentest yang dapat dilakukan diantaranya:

- *Black Box Testing*

  Pengujian dilakukan berdasarkan detail aplikasi, seperti tampilan aplikasi, fungsi-fungsi yang terdapat pada aplikasi, serta penyesuaian alur fungsi pada aplikasi dengan bisnis yang diinginkan oleh pelanggan. Pengujian ini dilakukan tanpa melihat dan menguji source code program yang ada pada aplikasi.

- *White Box Testing*

  Pengujian dilakukan berdasarkan detail prosedur serta alur logika dari sebuah kode program. Pada metoda ini, tester akan melihat keseluruhan source code sebuah program untuk menemukan bugs dari kode program tersebut.

- *Grey Box Testing*

  Pengujian merupakan kombinasi dari *Black Box* dan *White Box*, dimana pentester melakukan pengujian aplikasi berdasarkan spesifikasi namun menggunakan cara kerja dari dalam aplikasi tersebut alias source code program.

i. Dari jenis – jenis pentest diatas, item – item yang diuji secara umum adalah:

1. Pemindaian kerentanan/Pengujian server yang mengidentifikasi / melaporkan:

- Perangkat lunak tidak resmi
- Port server yang dibuka secara tidak tepat, protokol dan layanan yang diaktifkan
- Layanan yang salah konfigurasi atau diaktifkan secara tidak tepat, misalnya, ftp dan telnet
- Kredensial yang tersimpan tidak sesuai terkait dengan pekerjaan batch, skrip, atau file teks biasa
- Akun lokal yang tidak sesuai dengan aturan kata sandi yang tidak kadaluwarsa
- Metoda dan level enkripsi yang digunakan tidak memadai
- Kemampuan untuk mendapatkan akses tidak sah ke kunci enkripsi
- Kata sandi server yang lemah (seperti dapat ditentukan melalui pemecahan kata sandi)
- Patch yang hilang

2. Pemindaian kerentanan/Pengujian Aplikasi Aplikasi Berbasis Webyang mengidentifikasi / melaporkan:
    - Panggilan atau tanggapan API tidak aman
    - Antarmuka aplikasi lintas tidak aman
    - Pengodean kode kustom yang tidak aman dalam produk COTS atau kode yang menggunakan API COTS
    - Pengodean tidak aman dan fungsionalitas fungsi aplikasi sensitif atau antarmuka akses istimewa seperti layar administrator aplikasi
    - Kemampuan untuk mengeksekusi perintah atau menyuntikkan kode (mis. Perintah OS, injeksi SQL, Cross-site Scripting, injeksi LDAP)
    - Kontrol manajemen sesi yang tidak memadai
    - Kemampuan untuk melakukan serangan jalur URL path
    - Kemampuan untuk menyebabkan kondisi overflow (mis. Parameter overflow dan buffer overflow)
    - Kemampuan untuk melakukan serangan encoding karakter
    - Kemampuan untuk berkompromi dengan aplikasi dengan memberikan nilai input yang tidak sesuai (mis. Pengujian fuzz)
    - Kelemahan keamanan dalam Layanan Aplikasi Berbasis Web(berbasis REST dan SOAP) yang digunakan oleh aplikasi
3. Analisis kode statis untuk mengidentifikasi / melaporkan:
    - Keberadaan Bom Logika (*Logic Bomb*) atau *Backdoors*
    - Fitur debug yang diaktifkan
    - Kredensial disimpan secara tidak benar dalam kode
4. Pemindaian/Pengujian *Middleware* untuk mengidentifikasi / melaporkan:
    - Pengaturan konfigurasi yang tidak memadai
    - Direktori tidak resmi yang dapat diakses atau ditampilkan, mis. Pencacahan direktori
    - File aplikasi sisi server yang tidak sah yang dapat diakses untuk diunduh atau diperiksa oleh klien (mis. Melihat konten file php, jsp, atau asp)
    - Informasi produk yang tidak perlu ditampilkan (mis. Modul yang dipasang)
    - Akun atau fitur yang seharusnya tidak perlu diaktifkan
    - Patch yang hilang
    - Kata sandi yang masih disetting default

5. Pemindaian/Pengujian Database untuk mengidentifikasi / melaporkan:
    - Pengaturan konfigurasi yang tidak memadai
    - Akun atau fitur yang tidak perlu diaktifkan
    - Hak istimewa yang berlebihan diberikan untuk objek basis data atau ke file OS basis data
    - Akun lokal yang tidak memadai dengan kata sandi yang tidak kedaluwarsa
    - Kredensial disimpan secara tidak benar dalam pekerjaan batch atau skrip
    - Pemisahan tugas yang tidak memadai
    - Adanya utilitas khusus atau fitur debug yang diaktifkan di lingkungan produksi
    - Ketidakcukupan metoda enkripsi dan level yang digunakan
    - Kemampuan untuk mendapatkan akses tidak sah ke kunci enkripsi
    - Kata sandi basis data yang lemah (mis. melalui pemecahan kata sandi)
    - Replikasi database melalui saluran yang tidak aman
    - Kemampuan untuk membaca, memodifikasi, menyalin, atau menghapus data konfigurasi, log, dan informasi kontrol akses
    - Kecukupan kontrol untuk semua titik masuk dan keluar dari suatu aplikasi
    - *Patch* yang hilang
    - Kata sandi yang masih disetting default
6. Pemindaian/Pengujian Aplikasi Aplikasi Berbasis Web untuk mengidentifikasi / melaporkan:
    - Konfigurasi yang tidak memadai
    - Akun atau fitur yang seharusnya tidak perlu diaktifkan
    - Hak Akses yang berlebihan
    - Kemampuan untuk memotong jalur akses aplikasi yang normal
    - Kendali manajemen sesi yang tidak memadai
    - Pemisahan infrastruktur yang tidak memadai
    - Kata sandi aplikasi yang lemah (mis. Melalui pemecahan kata sandi)
    - Pemisahan aset yang tidak memadai berdasarkan tujuan dan lingkungan
    - Akses istimewa tidak melalui jalur administratif

- Kemampuan untuk meningkatkan hak istimewa
- Kemampuan untuk menyusup data
- Kemampuan untuk menyimpan konten berbahaya
- Kemampuan untuk mendapatkan akses tidak sah ke data dan ke file produk yang diinstal
- Kemampuan untuk mendapatkan akses tidak sah ke kunci enkripsi
- Kemampuan untuk mendapatkan akses tanpa izin ke antarmuka dan alat administratif
- Kemampuan untuk membaca, memodifikasi, menyalin, atau menghapus data konfigurasi, log, dan informasi kontrol akses
- Utilitas yang diistimewakan atau mengaktifkan fitur debugging di lingkungan produksi
- Metoda dan level enkripsi yang digunakan tidak memadai
- Kemampuan untuk mendapatkan akses tidak sah ke kunci enkripsi
- Kerentanan terhadap serangan umum seperti DDoS, dan replay sesi
- Akses yang tidak terotorisasi atau fungsionalitas aplikasi yang tidak dibatasi berdasarkan akses yang diberikan sesuai dengan peran pengguna
- Kecukupan kendali untuk semua titik masuk dan keluar dari suatu aplikasi
- Kata sandi yang masih disetting default
- Kecukupan dokumentasi aplikasi

3. Tools Pendukung Pengujian

a. Dalam melakukan pengujian selain metoda yang digunakan maka diperlukan tools pendukung yang beredar di pasaran, mulai dari aplikasi dalam bentuk satuan atau dalam bentuk sistem yang utuh. Tools yang disediakan juga ada yang bersifat berbayar atau lisensi.

b. Dengan semakin berkembangnya teknologi alat bantu pengujian, maka instansi tidak harus selalu menggunakan alat/software yang terdapat dalam daftar ini.

Berikut ini beberapa tools komersial dan distro yang gratis (terbuka) untuk digunakan dalam membantu pengujian keamanan.

Tabel 3.1 Tools Untuk Melakukan Pengujian Keamanan Web

Name	Owner	Licence	Platforms
Abbey Scan	MisterScanner	Free	SaaS
Acunetix WVS	Acunetix	Commercial/Free (Limited Capability)	Windows
Application Security on Cloud	IBM	Commercial	SaaS
AppScan	IBM	Commercial	Windows
App Scanner	Trustwave	Commercial	Windows
AppSpider	Rapid7	Commercial	Windows
AppTrana Aplikasi Berbasis WebSecurity Scan	AppTrana	Free	SaaS
Arachni	Arachni	Free for most use cases	Most platforms supported
AVDS	Beyond Security	Commercial/Free (Limited Capability)	SaaS
BlueClosure BC Detect	BlueClosure	Commercial, 2 weeks trial	Most platforms supported
BREACHLOCK Dynamic Application Security Testing	Breachlock	Commercial	SaaS
Burp Suite	PortSwiger	Commercial/Free (Limited Capability)	Most platforms supported
Contrast	Contrast Security	Commercial/Free (Full featured for 1 App)	SaaS or On-Premises
Detectify	Detectify	Commercial	SaaS
Digifort- Inspect	Digifort	Commercial	SaaS
edgescan	edgescan	Commercial	SaaS
GamaScan	GamaSec	Commercial	Windows
Grabber	Romain Gaucher	Open Source	Python 2.4, BeautifulSoup and PyXML
Gravityscan	Defiant, Inc.	Commercial/Free (Limited Capability)	SaaS
Grendel-Scan	David Byrne	Open Source	Windows, Linux and Macintosh

Name	Owner	Licence	Platforms
GoLismero	GoLismero Team	GPLv2.0	Windows, Linux and Macintosh
IKare	ITrust	Commercial	N/A
ImmuniWeb	High-Tech Bridge	Commercial/Free (Limited Capability)	SaaS
InsightVM		Commercial with Free Trial	SaaS
Indusface Web Application Scanning	Indusface	Commercial/Free Trial	SaaS
N-Stealth	N-Stalker	Commercial	Windows
Nessus	Tenable	Commercial	Windows
Netsparker	MavitunaSecurity	Commercial	Windows
Nexpose	Rapid7	Commercial/Free (Limited Capability)	Windows/Linux
Nikto	CIRT	Open Source	Unix/Linux
Probely	Probely	Commercial/Free (Limited Capability)	SaaS
Proxy.app	Websecurify	Commercial	Macintosh
QualysGuard	Qualys	Commercial	N/A
Retina	BeyondTrust	Commercial	Windows
Securus	Orvant, Inc	Commercial	N/A
Sentinel	WhiteHat Security	Commercial	N/A
SOATest	Parasoft	Commercial	Windows/Linux/Solaris
Tinfoil Security	Tinfoil Security, Inc.	Commercial/Free (Limited Capability)	SaaS or On-Premises
Trustkeeper Scanner	Trustwave SpiderLabs	Commercial	SaaS
Vega	Subgraph	Open Source	Windows, Linux and Macintosh
Vex	UBsecure	Commercial	Windows

Wapiti	Informática Gesfor	Open Source	Windows, Unix/Linux and Macintosh
Web Security Scanner	DefenseCode	Commercial	On-Premises
WebApp360	TripWire	Commercial	Windows
WebCookies	WebCookies	Free	SaaS
WebInspect	Micro Focus	Commercial	Windows
WebReaver	Websecurify	Commercial	Macintosh
WebScanService	German Web Security	Commercial	N/A
Websecurify Suite	Websecurify	Commercial/Free (Limited Capability)	Windows, Linux, Macintosh
Wikto	Sensepost	Open Source	Windows
w3af	w3af.org	GPLv2.0	Linux and Mac
Zed Attack Proxy	OWASP	Open Source	Windows Unix/Linux and Macintosh

4. Item Pengujian berbasis OWASP

Beberapa pengujian dasar berbasis OWASP sebagai berikut:

Tabel 3.2 Item Pengujian Berbasis OWASP

Item Pengujian	Tujuan	Teknik
Testing for Credentials Transported over an Encrypted Channel (OTG-AUTHN-001)	Memverifikasi bahwa otentikasi data pengguna ditransfer melalui saluran terenkripsi	Analisis paket header
Testing for default credentials (OTG-AUTHN-002)	Memverifikasi adakah penggunaan *default password* (*weak password*)	*Brute force password*
Testing for Weak lock out mechanism (OTG-AUTHN-003)	Memverifikasi apakah ada mekanisme penguncian akun	Mencoba *invalid login* beberapa kali
Testing for bypassing authentication schema (OTG-AUTHN-004)	Memverifikasi *direct page request* tanpa proses login	*Direct page request*
Test remember password functionality (OTG-AUTHN-005)	Memverifikasi bahwa *password* tidak disimpan dalam bentuk teks, tetapi hash	Analisis *cookies*
Testing for Browser cache weakness (OTG-AUTHN-006)	Memeriksa apakah aplikasi menginstruksikan browser untuk tidak ingat data sensitif.	Analisis browser *history* dan browser *cache*
Testing Directory traversal/file include (OTG-AUTHZ-001)	Menguji apakah aplikasi tahan terhadap *malicious string*	Memasukkan malicious string pada address bar
Testing for bypassing authorization schema (OTG-AUTHZ-002)	Percobaan akses ke dalam fungsi administrasi tanpa *login*	Manipulasi *http request header*
Testing for Privilege Escalation (OTG-AUTHZ-003)	Mencoba mendapatkan akses admin dari *user*	Manipulasi *http request header*
Testing for Insecure Direct Object References (OTG-AUTHZ-004)	Mencoba mengakses file tanpa login	Scanning
Testing for Bypassing Session Management Schema (OTG-SESS-001)	Memeriksa apakah token pada cookies dan sesi lainnya dibuat dalam cara yang aman dan *unpredictable*	Analisis *cookies*
Testing for Cookies attributes (OTG-SESS-002)	Memeriksa apakah *cookies* menyimpan informasi masa *expired* di hard *drive client*	Analisis *Cookies*

Item Pengujian	Tujuan	Teknik
Testing for Session Fixation (OTG-SESS-003)	Memeriksa apakah session yang diberikan pada client selalu diperbarui setelah proses authentifikasi	Analisis *Cookies*
Testing for Exposed Session Variables (OTG-SESS-004)	Memeriksa pada cookies apakah *Cookie, SessionID, Hidden Field* jelas terlihat	Analisis *Cookies*
Testing for Cross Site Request Forgery (CSRF) (OTG-SESS-005)	Memeriksa POST dan GET request pada halaman login apakah terimplementasi dengan baik	Analisis *Cookies*

III.5 STANDAR PENGENDALIAN INSIDEN KEAMANAN APLIKASI BERBASIS WEB

STANDAR
PENGENDALIAN INSIDEN KEAMANAN APLIKASI BERBASIS WEB

BADAN SIBER DAN SANDI NEGARA

2019

A. TUJUAN

1. Untuk memastikan bahwa:
   a. Pengendalian/respon insiden keamanan Aplikasi Berbasis Webdapat diatasi secepat mungkin.
   b. Adanya *knowledge management* dan *continuous improvement* dari setiap insiden yang terjadi.
2. Memberikan petunjuk dalam melakukan pengendalian insiden keamanan Aplikasi Berbasis Web.

B. RUANG LINGKUP

1. Tahapan Pengendalian Insiden Keamanan
2. Pencegahan Insiden
3. Penerapan Tahapan Pengendalian Insiden Keamanan berdasarkan jenis insiden

C. ISTILAH DAN DEFINISI

1.	*Suspicious files/Malicious Code*	:	Perangkat perusak, perangkat lunak berbahaya atau perangkat lunak jahat adalah perangkat lunak yang diciptakan untuk menyusup atau merusak sistem komputer, server atau jejaring komputer tanpa izin dari pemilik.
2.	Instansi	:	Kementerian/Lembaga, Instansi pusat atau daerah.
3.	*Password*	:	Kata sandi yang digunakan bersamaan dengan *username* (*sign on/sign in/log-on/log-in*) oleh pemilik yang sah sebelum melakukan koneksi/akses ke sistem komputer.
4.	*Patch*	:	Rutin program atau sekumpulan kecil instruksi yang biasanya dibuat sebagai solusi sementara untuk mengatasi atau memperbaiki permasalahan (*bugs*) pada program komputer dan sering dibuat dalam bentuk *'object code'* yang disisipkan ke dalam program yang akan dieksekusi.

D. REFERENSI

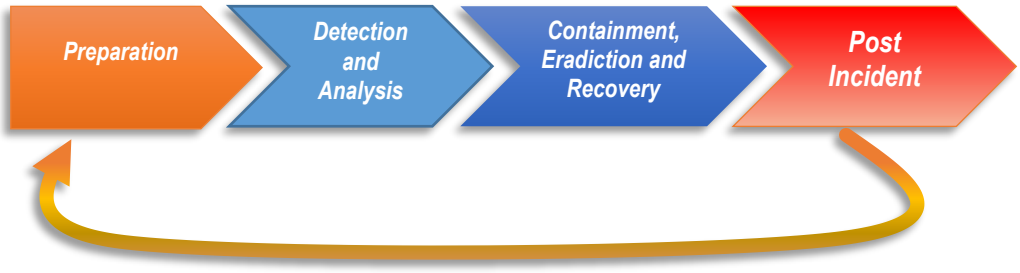
1. NIST Special Publication 800-61 Rev2: Computer Security Incident Handling Guide

E. STANDAR

1. Tahapan Pengendalian Insiden

Standar Tahapan Pengendalian Insiden meliputi:

Gambar 3.6 Tahapan Pengendalian Insiden



a. Persiapan *(Preparation)*

- Membangun dan melatih tim respon insiden, mempersiapkan aplikasi analisis yang akan digunakan dan sumber daya yang diperlukan.
- Tim insiden menyiapkan untuk menganalisis insiden, tim respon insiden menentukan insiden yang paling mungkin terjadi di dalam organisasi. Analisis risiko sangat penting untuk menentukan insiden-insiden yang mungkin terjadi pada suatu aset informasi di instansi. Dengan pimpinan tim insiden respon melakukan identifikasi risiko-risiko yang terjadi, apa yang harus dilakukan dan bagaimana mencari solusi dari permasalahan tersebut.
- Pembatasan jumlah insiden yang akan terjadi dengan memilih dan menerapkan satu set kontrol berdasarkan hasil penilaian risiko

b. Deteksi & Analisis (*Detection and Analysis*)

- Ada beberapa sumber untuk pendeteksian insiden, seperti pemantaun traffic IDS, laporan dari internal. Dari laporan pendeteksian insiden maka tim respon insiden melakukan analisis insiden, dan laporan analisis tersebut disajikan bukan berdasarkan spekulasi dan asumsi, tetapi berdasarkan hasil analisis yang sudah dilakukan dengan menggunakan aplikasi – aplikasi analisis khusus disesuaikan berdasarkan laporan insiden tersebut.
- Pendeteksian pelanggaran keamanan diperlukan sebagai bahan analisis Tim Insiden, untuk mengetahui kapan insiden tersebut terjadi. Disesuaikan dengan tingkatan setiap insiden tersebut dapat mengurangi dampak dari insiden sehingga proses pemulihan dapat terselesaikan dengan cepat. Selama fase ini , aktivitas siklus kembali ke deteksi dan analisis - misalnya, untuk melihat apakah tambahan host yang terinfeksi oleh malware sementara pemberantasan insiden malware

c. Penahanan, Pemberantasan & Pemulihan (*Containment, Eradiction and Recovery*)

Setelah proses analisis, maka proses selanjutnya bagaimana menentukan strategi untuk proses *Containment, Eradiction dan Recovery*. Kompatibilitas dalam perancangan strategi sangat penting untuk mempertimbangkan hal – hal:

- Kemungkinan potensi kerusakan pada sumber daya
- Efektifitas dan durasi dalam strategi pemberian solusi dari setiap penanganan insiden

Sebelum memulai proses penanganan, pihak pimpinan tim insiden respon sudah megetahui dan menyetujui laporan hasil analisis insiden tersebut, karena laporan hasil analisis insiden tersebut akan menjadi pendukung utama dalam memulai proses penanganan insiden tersebut.

d. Aktivitas Paska Insiden (*Post Incident Activity*)

- Tim Insiden menerbitkan sebuah laporan yang merinci penyebab dan biaya insiden tersebut dan langkah-langkah organisasi untuk melakukan pencegahan insiden di masa depan
- Setelah melakukan proses containment, eradiction dan recovery, maka tahap selanjutnya melakukan pertemuan antara pimpinan dan steakholder untuk menyusun langkah – langkah kedepan-nya apabila terjadi insiden tersebut terjadi lagi di masa depan.

2. Pencegahan Insiden

a. Blokir *suspicious files*

- Konfigurasi email server dan client untuk memblokir *attachment* dengan ekstensi file yang berkaitan dengan *malicious code* (misalnya .pif, .vbs, bisa juga .rar atau .zip agar lebih ketat) dan kombinasi ekstensi file yang mencurigakan (misalnya .txt.vbs., .Htm.exe.). Namun cara ini mungkin juga memblokir aktifitas yang sah. Beberapa instansi mengubah ekstensi file terlebih dahulu sebelum dikirim kemudian penerima harus men-*save* dan me- *rename* file terlebih dahulu sebelum dijalankan.
- Mengurangi sharing file pada OS karena banyak worm yang menyebar melalui sharing file pada host yang menjalankan OS. Sebuah infeksi tunggal dapat dengan cepat menyebar ke ratusan atau ribuan host melalui unsecured share.

b. Menetapkan prosedur kepada semua user dari sistem, aplikasi, domain, sampai workstation untuk mengubah password mereka secara periodik. Hal ini merupakan cara mencegah akses yang tidak terotorisasi.

c. Meningkatkan Pengetahuan Keamanan Informasi (*Security Awareness*)

- Unit IT Security harus mengadakan pelatihan security awareness yang dilakukan satu kali dalam setahun yang diikuti oleh seluruh pegawai. Hal ini dimaksudkan agar setiap karyawan menyadari aturan perilaku dan tanggungjawab mereka dalam melindungi misi Instansi.
- Untuk meningkatkan *security awareness* tidak hanya melalui pelatihan saja satu kali setahun saja, tetapi dapat dilakukan dalam sesi-sesi kecil perdivisi atau grup dengan materi yang menyeluruh mengenai keamanan informasi. Misalnya masalah attachment pada email, pergantian password, serta sharing file. Selain itu juga dapat dilakukan workshop, melalui situs Web, dan stiker.

3. Deteksi Insiden

a. Manajemen *Patch*

Secara rutin unit IT Security melakukan *vulnerability assessment* secara periodik. Hal ini dilakukan untuk mengurangi potensi terjadinya insiden keamanan. Dari hasil penilaian kerentanan yang telah diinformasikan kepada Penanggungjawab masing - masing perangkat seharusnya dilakukan eskalasi terhadap kerentanan yang ada. *Manajemen patch* sangat penting untuk mengurangi kerentanan yang ada pada aplikasi. Unit IT Security juga bertanggungjawab untuk melakukan manajemen *patch*, misalnya dengan memperoleh, menguji, dan mendistribusikan *patch* untuk para administrator yang sesuai dan pengguna diseluruh Instansi. Terkadang manajemen *patch* sering dibutuhkan saat mencoba untuk melakukan pemulihan dari insiden dengan skala besar.

b. Penggunaan SIEM (*Security Incident Event Management*) lebih ditingkatkan.

Terkadang sulit untuk mengidentifikasi suatu event sebagai insiden yang potensial. Semua yang mengeluarkan log harus dimonitor dengan baik. Mulai dari log aplikasi, perangkat (firewall, IDPS, appliance), sistem operasi, maupun database. SIEM bekerja untuk meningkatkan keamanan informasi pada infrastruktur jaringan instansi dengan efisien dan dapat mendeteksi kejadian yang mencurigakan. SIEM bertugas untuk memberikan deteksi awal dan peringatan terhadap event keamanan.

4. Deteksi & Analisis Berdasarkan Jenis Insiden

a. Insiden Secara Umum

Tabel 3.3 Deteksi & Analisis Insiden Secara Umum

Tahap	Kontrol
Deteksi dan Analisis	- Mengetahui insiden yang telah terjadi - Menganalisis sesuatu yang akan datang berdasarkan indikasi yang ada - Mencari informasi yang saling berhubungan - Melakukan pengkajian - Mendokumentasikan semua proses investigasi dan hasil pengumpulan bukti yang ada. - Mengklasifikasi insiden menggunakan beberapa kategori yang ada

b. Insiden *Distributed Denial of Service (DDoS)*

Tabel 3.4 Deteksi & Analisis DDoS

Tahap	Kontrol
Deteksi dan analisis	- Memprioritaskan penanganan insiden berdasarkan pengaruh bisnis. - Mengidentifikasi sistem/resource mana yang terkena dan memprediksikan sistem/resource mana yang akan terkena selanjutnya. - Melakukan estimasi terhadap pengaruh secara teknis saat ini dan yang akan berpotensi terkena dari insiden tersebut. - Melaporkan insiden kepada internal staf insiden respon yang ditunjuk dan juga kepada organisasi yang diluar.
Penahanan, Penghapusan, dan	- Memperoleh, mempertahankan, mengamankan, dan mendokumentasi bukti yang ada.

Pemulihan	- Menahan insiden memberhentikan serangan DoS apabila serangan belum berhenti. - Mengidentifikasi dan mengurangi semua kerentanan yang digunakan. - Apabila belum tertahankan, jalankan filter berdasarkan sifat dari serangan jika memungkinkan. - Apabila belum tertahankan lakukan relokasi target. - Mengembalikan sistem yang terkena seperti sistem yang dapat dioperasikan sediakala. - Mengkonfirmasi sistem yang terkena dapat berfungsi secara normal. - Apabila diperlukan, jalankan monitoring tamba- han untuk melihat kemungkinan terjadinya aktifitas yang sama yang akan terjadi dimasa depan
Kegiatan Pasca Insiden	- Membuat laporan lanjutan - Mengadakan meeting yang membahas insiden tersebut

c. Insiden *Unauthorized Access*

Tabel 3.5 Deteksi & Analisis Insiden *Unauthorized Access*

Tahap	Kontrol
Deteksi dan analisis	- Memprioritaskan penanganan insiden berdasarkan pengaruh bisnis. - Mengidentifikasi sistem/resource mana yang terkena dan memprediksikan sistem/resource mana yang akan terkena selanjutnya. - Melakukan estimasi terhadap pengaruh secara tehnis saat ini dan yang akan berpotensi terkena dari insiden tersebut. - Melaporkan insiden kepada internal staf insiden yang ditunjuk dan juga kepada organisasi yang diluar.

Penahanan, Penghapusan, dan Pemulihan	- Melakukan penahanan pada awal kejadian - Memperoleh, mempertahankan, mengamankan, dan mendokumentasi bukti yang ada. - Mengkonfirmasi penahanan kejadian - Melakukan analisis lebih lanjut terhadap insiden tersebut dan menentukan apakah penahanan sudah cukup (termasuk memeriksa sistem lain untuk tKita-tKita intrusi) - Melaksanakan tindakan penahanan tambahan jika perlu - Mengidentifikasi dan mitigasi semua kerentanan yang telah dieksploitasi. - Menghapus semua komponen insiden dari system - Mengembalikan sistem yang terkena ke keadaan operasional semula - Menkonfirmasikan bahwa sistem yang terkena te- lah berfungsi normal - Apabila diperlukan, jalankan monitoring tamba- han untuk melihat kemungkinan terjadinya aktiti- tas yang sama yang akan terjadi dimasa depan
Kegiatan Pasca Insiden	- Membuat laporan lanjutan - Mengadakan meeting yang membahas insiden tersebut

d. Insiden Virus/Worm/Trojan

Tabel 3.6 Deteksi & Analisis Insiden Virus/Worm/Trojan

Tahap	Kontrol
Deteksi dan analisis	- Memprioritaskan penanganan insiden berdasarkan pengaruh terhadap bisnis instansi. - Mengidentifikasi sistem/resource mana yang terkena dan memprediksikan sistem/*resource* mana yang akan terkena selanjutnya. - Melakukan estimasi terhadap pengaruh secara teknis saat ini dan yang akan berpotensi terkena dari insiden

	tersebut. - Melaporkan insiden kepada internal staf insiden respon yang ditunjuk dan juga kepada organisasi yang diluar.
Penahanan, Penghapusan, dan Pemulihan	- Mengidentifikasi sistem yang terinfeksi - Disconnect semua sistem yang terinfeksi dari jaringan - Mitigasi kerentanan yang dimanfaatkan oleh malicious code - Apabila diperlukan, halangi mekanisme transmisi untuk malicious code - Disinfeksi, mengkarantina, menghapus, dan mengganti file yang terinfeksi - Mengurangi kerentanan dieksploitasi untuk host lain dalam organisasi - Mengkonfirmasikan bahwa sistem yang terkena telah berfungsi normal - Apabila diperlukan, jalankan monitoring tambahan untuk melihat kemungkinan terjadinya aktiti- tas yang sama yang akan terjadi dimasa depan.
Kegiatan Pasca Insiden	- Membuat laporan lanjutan - Mengadakan meeting yang membahas insiden tersebut

5. Mitigasi Ancaman

Tabel 3.7 Tindakan Mitigasi berdasarkan Jenis Ancaman

Nama Ancaman	Gambaran Ancaman	Mitigasi
DDoS	Jenis serangan terhadap sebuah komputer atau server di dalam jaringan internet dengan cara menghabiskan sumber (resource) yang dimiliki oleh komputer tersebut sampai komputer tersebut tidak dapat menjalankan fungsinya dengan benar sehingga secara tidak langsung mencegah pengguna lain untuk memperoleh akses layanan dari komputer yang diserang. Dalam sebuah serangan Denial of Service, si penyerang akan mencoba untuk mencegah akses seorang pengguna terhadap sistem atau jaringan dengan menggunakan beberapa cara, yakni sebagai berikut: a. Membanjiri lalu lintas jaringan dengan banyak data sehingga lalu lintas jaringan yang datang dari pengguna yang terdaftar menjadi tidak dapat masuk ke dalam sistem jaringan. Teknik ini disebut sebagai traffic flooding. b. Membanjiri jaringan dengan banyak request terhadap sebuah layanan jaringan yang disedakan oleh sebuah host sehingga request yang datang dari pengguna terdaftar tidak dapat dilayani oleh layanan tersebut. Teknik ini disebut sebagai request flooding.	- Indentifikasi aktifitas service apakah loadnya besar - Capturing aktifitas DDOS - Identifikasi Port serangan yang dilakakun DDOS - Melakukan pacthing terhadap bug – bug aplikasi yang memungkin bisa terserang DDOS - Melakukan monitoring IDS - Melakukan Blocking terhadap IP yang melakukan DDOS - Melakukan filtering dengan firewall dan dikombinasikan dengan perangkat monitoring IDS

	c. Mengganggu komunikasi antara sebuah host dan kliennya yang terdaftar dengan menggunakan banyak cara, termasuk dengan mengubah informasi konfigurasi sistem atau bahkan perusakan fisik terhadap komponen dan server	
Malware	Adalah perangkat lunak yang di ciptakan untuk atau merusak sistem komputer atau jaringan komputer tanpa izin dari pemilik	- Identifikasi adanya port dan koneksi yang aktif - Identifikasi process yang mencurigakan - Identifikasi start up - Identifikasi service yang sedang berjalan - Identifikasi file dan folder yang mencurigakan - Melakukan monitoring IDS terhadapa anomaly malware - Menerapkan security policy terkait aturan sharing file dan browsing - Melakukan scanning menggunakan antivirus
Aplikasi Berbasis Web defacement	Adalah bentuk serangan untuk merubah tampilan Aplikasi Berbasis Web, hacker memanfaatkan vulnerability dari penggunaan cms open source yang tidak dimanajemen dengan baik ataupun karena password yang lemah.	- Backup log aktifitas dari webserver - Scanning dengan aplikasi Rootkit scanner - Backup source tools hacking - Selalu update CMS - Audit struktur pemrograman - Lakukan patching pada sistem operasi dan aplikasi

		- Dedicated hosting - Manajemen password - Manajemen user - Memasang FIM (File Integrity Monitoring)
Phishing	Adalah tindakan memperoleh informasi pribadi seperti User ID, Password dan data-data sensitif lainnya dengan menyamar sebagai orang atau organisasi yang berwenang melalui sebuah email.	- Pengecekan pola dari aktifitas phishing tersebut apakah melalui email atau link Aplikasi Berbasis Web - Pengecekan email header apakah IP tersebut legitimate - Pengecekan file folder yang mencurigakan di sisi server Aplikasi Berbasis Web - Backup log aktifitas Phishing - Menghapus file folder yang mencurigakan apabila phishing tersebut disisipkan di Aplikasi Berbasis Web - Menginformasikan IP dan Domain yang legitimate - Menginformasikan untuk analisis email header

III.6 STANDAR SECURE-SOFTWARE DEVELOPMENT LIFE CYCLE

STANDAR
SECURE-SOFTWARE DEVELOPMENT LIFE CYCLE (S-SDLC)

BADAN SIBER DAN SANDI NEGARA

2019

A. TUJUAN

Memastikan bahwa pembangunan atau pengembangan sistem informasi (*Software Development Life Cycle*) berbasis web selalu memperhatikan aspek – aspek keamanan dalam setiap tahapan.

B. RUANG LINGKUP

1. Tahapan umum SDLC
2. Aspek keamanan yang diperhatikan dalam setiap tahapan SDLC

C. ISTILAH DAN DEFINISI

1. Secure Software Development Life Cycle	:	Serangkaian kegiatan yang dilakukan untuk mengembangkan, memelihara, dan memberikan solusi perangkat lunak yang aman. Kegiatan mungkin tidak harus berurutan; mereka bisa bersamaan atau berulang.
2. Whitelist	:	Daftar sekumpulan web domain atau alamat email, dan URL yang terindikasi "aman" sehingga secara otomatis akan diterima oleh komputer maupun jaringan agar dapat diakses. Teknologi whitelist banyak digunakan karena menjadi kontrol akses sehingga komputer atau jaringan tertentu dapat mengakses domain atau menerima email yang diperbolehkan atau tidak diperbolehkan oleh organisasi atau individu.
3. Instansi	:	Kementerian/Lembaga, Instansi pusat atau daerah.
4. Session ID	:	Nomor unik yang diberikan server situs Web untuk pengguna tertentu selama durasi kunjungan (session). ID sesi dapat disimpan sebagai cookie, form field, atau URL (Uniform Resource Locator). Beberapa server Web menghasilkan ID sesi dengan hanya menambah angka statis. Namun, sebagian besar server menggunakan algoritma yang melibatkan metoda yang lebih kompleks, seperti memfaktorkan dalam tanggal dan waktu kunjungan bersama dengan variables yang ditentukan oleh administrator server

5. *Password* : Kata sandi yang digunakan bersamaan dengan *username* (*sign on/sign in/log-on/log-in*) oleh pemilik yang sah sebelum melakukan koneksi/akses ke sistem komputer.
6. *Patch* : Rutin program atau sekumpulan kecil instruksi yang biasanya dibuat sebagai solusi sementara untuk mengatasi atau memperbaiki permasalahan (*bugs*) pada program komputer dan sering dibuat dalam bentuk *'object code'* yang disisipkan ke dalam program yang akan dieksekusi.
7. *Cookie* : Bagian kecil data yang dikirim dari situs web dan disimpan di komputer pengguna oleh browser web pengguna saat pengguna sedang menjelajah.
8. *Session* : Pertukaran informasi sementara dan interaktif antara dua atau lebih perangkat komunikasi atau antara komputer dan pengguna

D. REFERENSI

1. OWASP Secure Software Development Life Cycle Project (S-SDLC)

E. STANDAR

1. Tahapan S-SDLC

1.1. Pembuatan Kerangka Acuan Kerja & Perencanaan Proyek

a. Dokumen Kerangka Acuan Kerja harus berisikan bagian yang secara jelas menyebutkan:

- Kebutuhan fungsional keamanan aplikasi
- Kebutuhan jaminan keamanan aplikasi
- Kebutuhan dokumentasi terkait keamanan aplikasi
- Deskripsi lingkungan pengembangan, pengujian, pra-produksi/staging dan produksi/operasional.
- Syarat uji terima keamanan aplikasi

b. Dokumen Perencanaan proyek harus berisikan bagian yang secara jelas menyebutkan:

- Tahapan peninjauan terkait keamanan aplikasi melalui pemeriksaan dokumen formal SDLC maupun pelaksanaan skenario ujicoba khusus untuk keamanan aplikasi.
- Ada pihak yang secara jelas ditunjuk sebagai Security Officer yang bertugas melakukan peninjauan dan audit atas pelaksanaan kegiatan diatas.
- Bagian yang memastikan bahwa rencana pekerjaan yang berhubungan dengan pemenuhan kebutuhan maupun standar pedoman keamanan aplikasi memiliki sumber daya tenaga ahli, waktu dan dana yang cukup dan memadai.

1.2. Identifikasi Kebutuhan Bisnis

a. Dokumen kebutuhan bisnis (*Business Requirement Spesification*, BRS) wajib mencantumkan persyaratan keamanan aplikasi dari sisi bisnis pengguna.

b. Pembagian peran (*application role*) antar pengguna harus didefinisikan dengan jelas dengan menggunakan kaidah pemisahan tugas fungsi (*segregation of duty*).

c. Mendefinisikan sumber daya yang dibutuhkan untuk melaksanakan Standar keamanan yang diperlukan oleh aplikasi dan juga yang dibutuhkan dalam proses pengembangannya.

d. Menerapkan manajemen risiko untuk mengidentifikasikan ancaman keamanan, kemungkinan dan dampak yang berkaitan dengan penggunaan aplikasi.

1.3. Identifikasi Kebutuhan Fungsional

a. Dokumen kebutuhan fungsional (*Functional Requirement Specification*, FRS) wajib mencantumkan bagian khusus mengenai persyaratan keamanan aplikasi.

b. Pembagian hak akses, fungsi antar menu berbasis peran pengguna harus didefinisikan dengan jelas dengan menggunakan kaidah pemisahan fungsi tugas (segregation of duty).

c. Mendefinisikan persyaratan ujicoba fungsional khusus untuk kebutuhan keamanan aplikasi, termasuk di dalamnya standar tool yang digunakan untuk pengembangan, untuk melakukan ujicoba keamanan, mekanisme, skenario dan standar lolos ujicoba

d. Mendefinisikan sumber daya yang dibutuhkan untuk melaksanakan standar keamanan yang diperlukan oleh aplikasi dan juga yang dibutuhkan dalam proses pengembangannya.

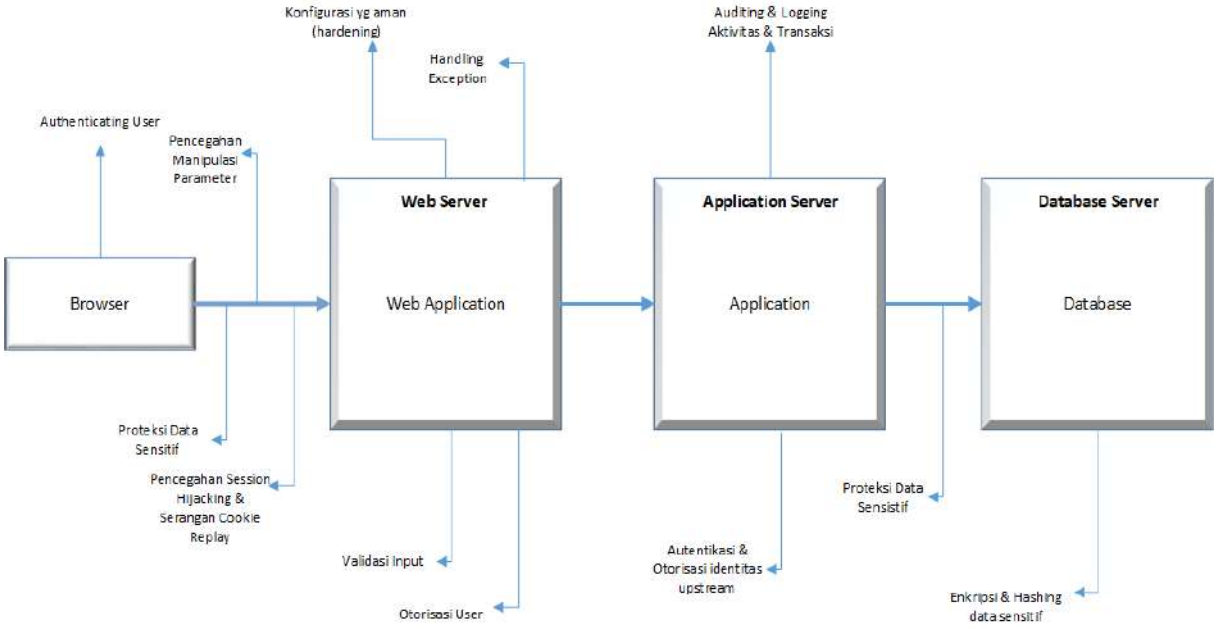
e. Menerapkan manajemen risiko untuk mengidentifikasikan ancaman keamanan, kemungkinan dan dampak yang berkaitan dengan penggunaan aplikasi dari sudut pandang teknis.

1.4. Desain Aplikasi

a. Desain aplikasi wajib mengakomodasi kebutuhan keamanan yang didefinisikan dalam dokumen BRS dan FRS. Disain aplikasi ditinjau oleh unit IT Operation untuk memastikan kesesuaian dengan kebutuhan persyaratan keamanan dalam BRS dan FRS.

b. Desain sistem aplikasi dibuat harus dengan memperhatikan ancaman-ancaman serangan yang mungkin terjadi pada titik-titik yang digambarkan dalam arsitektur berikut ini:

Gambar 3.7 Aspek Keamanan dalam Desain Aplikasi



1.5. Konstruksi Aplikasi

a. Pengembang harus mematuhi Standar keamanan aplikasi dan *Standar Quality Control* dalam melakukan konstruksi aplikasi.

b. Pengembangan aplikasi harus mematuhi persyaratan keamanan yang telah ditentukan dalam dokumen disain aplikasi.

c. Pengembang harus menggunakan Standar tool pemrograman yang telah disahkan oleh IT Development sebagai tools yang telah memiliki fungsi validasi keamanan Standar.

d. Hasil konstruksi harus ditinjau oleh fungsi audit untuk memastikan pekerjaan yang dilakukan telah sesuai dengan persyaratan keamanan aplikasi.

1.6. Pengujian Aplikasi

a. Pengembang harus membuat skenario ujicoba khusus yang mengevaluasi persyaratan keamanan yang ditentukan dalam dokumen BRS dan FRS.

b. Skenario ujicoba yang mengevaluasi hasil konstruksi aplikasi berdasarkan pedoman keamanan aplikasi harus dibuat dan dilaksanakan oleh fungsi audit.

c. Skenario ujicoba berisi Standar yang harus dipenuhi untuk setiap skenario.

d. Hasil ujicoba didokumentasikan dan ditindaklanjuti jika masih ditemukan ketidaksesuaian dengan standar yang telah ditentukan.

1.7. Sosialisasi & Pelatihan Keamanan Aplikasi

a. Pengembang aplikasi baik internal maupun rekanan wajib memperoleh pelatihan dan sosialisasi mengenai kebijakan keamanan informasi, pedoman keamanan pengembangan aplikasi, dan standar SDLC. Pelatihan dan sosialisasi yang dilakukan adalah:

- Sosialisasi / Pelatihan awal untuk karyawan dan rekanan baru.
- Sosialisasi / Pelatihan pengulangan berkala 1 tahun sekali.

b. Pengguna aplikasi selain diberikan pelatihan mengenai penggunaan aplikasi juga diberikan pelatihan dasar mengenai keamanan aplikasi terutama yang berkaitan dengan hak akses/perlindungan user & password.

c. Setiap kegiatan pelatihan keamanan pengembangan aplikasi maupun keamanan dasar aplikasi didokumentasikan dalam laporan kegiatan pelatihan.

d. Pelatihan awal harus diberikan kepada pengembang internal ataupun rekanan sebelum melakukan pekerjaannya.

e. Pelatihan bagi pengguna harus diberikan kepada pengguna sebelum hak akses kepada aplikasi digunakan oleh pengguna tersebut.

f. Pelatihan harus dilakukan kembali ketika ada perubahan sistem yang mengubah cara penggunaan atau terjadi revisi dalam dokumen pedoman ini.

2. Pedoman Teknis Penerapan Keamanan Pada Tahapan S-SDLC

2.1. Memvalidasi data input

a. Mengasumsikan seluruh masukan adalah malicious.
b. Menggunakan validasi input tersentralisasi.
c. Tidak tergantung pada validasi di sisi client tetapi juga menggunakan validasi di sisi server.
d. Memvalidasi tipe, panjang, format dan rentang dari data.
e. Menyusun sebuah rutin/kode program validasi secara terpusat.
f. Menentukan karakter set yang tepat untuk semua sumber. Contohnya UTF-8
g. Menyandikan data ke set karakter umum sebelum divalidasi
h. Semua kegagalan validasi harus dicatatat pada input yang ditolak
i. Bilamana sistem mendukung set karakter UTF-8 yang dikayakan, validasi dilakukan setelah proses decoding UTF-8 selesai.
j. Melakukan validasi semua data yang disediakan client sebelum diolah, termasuk semua parameter, URL dan konten header HTTP (Seperti nama dan nilai cookie). Memastikan untuk menyertakan umpan balik otomatis dari *java script, flash* dan kode lain yang ditanam.
k. Memverifikasi nilai-nilai header mengandung hanya karakter ASCII pada permintaan dan respon permintaan terkait.
l. Melakukan verifikasi data dari redirection
m. Memvalidasi semua input dengan daftar karakter yang diijinkan, ketika kondisi tersebut memungkinkan.
n. Dalam hal terdapat karakter yang berpotensi membahayakan pada input yang harus diterima, perlu dipastikan terdapat pengendali tambahan, seperti encoding output, pengamanan proses API-API khusus dan menghitung utilitas data dalam sistem. Contoh umum karakter berbahaya seperti < > " ' % ( ) & + \ \' \".
o. Bila rutin/kode sumber Standar validasi tidak dapat menangani input yang masuk, input terkait harus diperiksa secara terpisah.
p. Memeriksa semua byte bernilai null.
q. Memeriksa semua karakter baris baru (%0d, %0a, \r, \n).
r. Memeriksa semua “titik titik garis miring karakter perubahan jalur (../ atau ..\). Dalam kondisi dimana UTF-8 yang karakternya dikayakan digunakan, alamat jalur di representasikan seperti %c0%ae%c0%ae/.

2.2. Otentikasi

a. Mengelompokan halaman situs berdasarkan akses anonim, akses teridentifikasi, dan akses terotentikasi.

b. Menggunakan aturan password sesuai kebijakan keamanan TI.

c. Menggunakan masa berlaku password dan penonaktifan akun user.

d. Tidak menyimpan credentials user di database tetapi dapat menggunakan domain controller/SSO. Enkripsi saluran komunikasi ketika mengirimkan informasi username dan password menggunakan https.

2.3. Otorisasi

a. Menerapkan *whitelist* untuk *web server* dan *application server* yang bisa mengakses sql request ke database server.

b. Halaman yang diidentifikasikan untuk akses terotentikasi selalu memvalidasi user session di server dan hak aksesnya terhadap halaman tersebut, bukan sekedar masa berlaku sesinya saja kecuali jika memang ditujukan untuk publik.

c. Membatasi akses aplikasi ke *system level resources server* seperti files, folders, registry keys, Active Directory objects, database objects, event logs dan sebagainya.

d. Menggunakan Windows Access List untuk membatasi user apa yang bisa mengakses resource dan operasi apa yang bisa dilakukannya.

2.4. Data Sensitif/Rahasia

a. Tidak menyimpan dan mengirimkan data user dan password database dalam plain text.

b. Menggunakan secure communication (SSL) ketika mengirimkan user dan password database.

2.5. Manajemen Sesi

a. Membatasi masa hidup / berlaku session.

b. Enkripsi isi dari cookies untuk otentikasi.

c. Melindungi session state dari akses yang tak terotorisasi atau pencurian sesi.

d. Memastikan bahwa sesi ditutup ketika pengguna log out

e. Memastikan bahwa sesi mengalami timeout setelah tidak aktif selama jangka waktu tertentu.

f. Memastikan bahwa semua halaman yang membutuhkan otentikasi untuk diakses, memiliki tautan untuk logout.

g. Memastikan bahwa nomor sesi tidak pernah diungkapkan selain di header cookie, khususnya di URL, pesan error, atau log. Ini termasuk memastikan bahwa aplikasi tidak memperbolehkan penulisan ulang session cookies melalui URL.

h. Memastikan bahwa nomor sesi tidak pernah diungkapkan selain di header cookie, khususnya di URL, pesan error, atau log. Ini termasuk memastikan bahwa aplikasi tidak memperbolehkan penulisan ulang session cookies melalui URL.

i. Memastikan bahwa ID sesi diubah setiap kali login dan setiap otentikasi ulang.

j. Memastikan bahwa sesi id berubah atau dibersihkan pada saat logout.

k. Memastikan bahwa id sesi yang diakui sebagai sah oleh aplikasi adalah hanya id yang dihasilkan oleh framework aplikasi.

l. Memastikan bahwa token sesi yang telah terotentikasi cukup panjang dan acak untuk bertahan dari jenis serangan yang sering terjadi berdasarkan ancaman ancaman di lingkungan.

2.6. Manipulasi Parameter

a. Tidak mempercayai field yang dapat dimanipulasi oleh client (query strings, form fields, cookies, atau HTTP headers).

b. Memvalidasi semua nilai parameter yang dikirim oleh client.

2.7. Exception Handling

a. Menggunakan exception handling yang terstruktur untuk memudahkan penambahan maupun pencarian masalah.

b. Tidak menampilkan pesan kesalahan secara detil dari server.

c. Tidak mencatat log data pribadi seperti password.

2.8. Log dan Audit

a. Aplikasi harus memiliki fungsi audit dan log minimal pada fungsi login dan akses halaman web.

b. Aplikasi harus dapat mencatat setiap login aplikasi yang sukses dan yang gagal, kapan, siapa usernya, dari IP berapa dan nama komputernya.

c. Aplikasi harus bisa mencatat user, kapan, IP dan nama computer dari setiap sesi user yang mencoba mengakses halaman yang tidak diberikan hak akses untuknya.

2.9. Konfigurasi Web Server

a. Menghapus atau menonaktifkan akun yang tak digunakan

b. Menonaktifkan Akun Guest

c. Mengganti nama akun administrator

d. Menonaktifkan akun IUSR (default akun internet anonim)

e. Membuat akun Aplikasi Berbasis Webanonim sendiri secara custom.

f. Menggunakan kebijakan password yang kuat

g. Menonaktifkan anonymous logons.

h. Membatasi remote logons.

i. Menghapus virtual direktori berikut yang terinstal sebagai contoh: IISSamples, IISAdmin, IISHelp,dan Scripts.

j. Menonaktifkan setelan Parent Path untuk menghindari serangan direktori traversal.

k. Tidak boleh memasang protokol yang plain text dan berbahaya secara bawaan seperti: Telnet, Post Office Protocol (POP3), Simple Mail Transfer Protocol (SMTP), and File Transfer Protocol (FTP)

2.10. Proses Persetujuan (Approval) melalui tautan email

a. Penggunaan token unik terenkripsi untuk setiap URL permintaan persetujuan yang dikirimkan melalui suatu email yang kode parameternya tidak mudah dipahami user.

b. Adanya konfirmasi kembali dari sistem atas data yang berhasil / gagal dicatat oleh sistem atas tindakan yang dilakukan oleh pengguna

c. Memastikan pencatatan kontrol keamanan menyediakan kemampuan untuk mencatat peristiwa keberhasilan dan kegagalan yang diidentifikasi sebagai yang terkait keamanan.

d. Permintaan email persetujuan tidak boleh menggunakan alamat email pemohon sebagai pengirim, harus menggunakan alamat email akun dari sistem aplikasi

III.7 STANDAR TOOLS PENGAMANAN APLIKASI BERBASIS WEB

STANDAR
PERANGKAT/TOOLS PENGAMANAN APLIKASI BERBASIS WEB

BADAN SIBER DAN SANDI NEGARA

2019

A. TUJUAN

1. Memberikan petunjuk bagi instansi – instansi yang memiliki sistem web statis, web dinamis, web dinamis dengan aplikasi transaksi, serta web dinamis dengan aplikasi transaksi yang membutuhkan tingkat availibility tinggi, agar dalam mengadakan/menyediakan tools pengamanan web mengacu/memiliki Standar minimum.
2. Untuk memastikan bahwa dalam pengadaan perangkat pengamanan tidak terlalu berlebihan (*overkill*) ataupun terlalu minim.

B. RUANG LINGKUP

Standar minimum untuk:

1. Intrusion Detection & Prevention System (IDPS)
2. Web Application Firewall (Firewall Aplikasi Web)
3. Security Information and Event Management (SIEM)
4. Network Behavioral Analysis/NBA (Analisis Perilaku Jaringan)
5. File Integrity Monitoring (Pemantauan Integritas File)
6. IP Reputation Engine (Mesin Reputasi IP)
7. Synchronization System (Sistem Sinkronisasi)
8. Network Analytic System (Sistem Analitik Jaringan)

C. ISTILAH DAN DEFINISI

1. Zero Day Vulnerability	:	informasi kelemahan keamanan dari suatu aplikasi dimana informasi ini belum diumumkan ke publik.
2. *Zero Day Attack*	:	serangan yang dilakukan penyerang sebelum informasi kelemahan keamanan dari aplikasi tersebut diumumkan ke publik.
3. *Zero Day Protection*		kemampuan aplikasi menangkal *Zero Day Attack*/ kemampuan menangkal malware bahkan pada saat malware tersebut belum diumumkan ke publik.

4. Korelasi		kemampuan untuk melakukan berbagai teknik korelasi untuk mengintegrasikan berbagai sumber, untuk mengubah data menjadi informasi yang bermanfaat. Korelasi biasanya merupakan fungsi dari bagian Manajemen Peristiwa Keamanan dari solusi SIEM lengkap
5. Proxy	:	suatu sistem yang memungkinkan kita untuk bisa mengakses jaringan internet menggunakan IP yang berbeda dengan yang diterima oleh perangkat.
6. Instansi	:	Kementerian/Lembaga, Instansi pusat atau daerah.
7. *Password*	:	Kata sandi yang digunakan bersamaan dengan *username* (*sign on/sign in/log-on/log-in*) oleh pemilik yang sah sebelum melakukan koneksi/akses ke sistem komputer.
8. *Content Delivery Network* (CDN)	:	adalah kumpulan dari server global yang terletak di beberapa data center dan tersebar di berbagai negara. Jaringan ini berfungsi untuk mengirimkan konten dari server ke suatu Aplikasi Berbasis Web. Yang dilakukan oleh CDN adalah meningkatkan kecepatan pengiriman data melalui jaringan server kepada visitor dari lokasi terdekat yang paling memungkinkan

D. REFERENSI

1. OWASP: Application Security Verification Standard 4.0
2. Kebijakan Keamanan Informasi .

E. STANDAR

I. Intrusion Detection & Prevention System (IDPS)

PENJELASAN :

- IPS adalah suatu bentuk sistem keamanan jaringan yang berfungsi untuk mendeteksi dan mencegah ancaman yang teridentifikasi.
- IPS memantau jaringan secara terus-menerus, mencari kemungkinan insiden berbahaya dan menangkap informasinya.

- IPS melaporkan peristiwa ini ke administrator sistem dan mengambil tindakan pencegahan, seperti menutup titik akses dan mengonfigurasi firewall untuk mencegah serangan di masa depan.
- Solusi IPS juga dapat digunakan untuk mengidentifikasi masalah dengan kebijakan keamanan, menghalangi karyawan dan tamu jaringan melanggar aturan yang terkandung dalam kebijakan ini.

PRASYARAT :

- Next Generation Firewall System (Sistem Firewall Generasi Lanjut)
- Unified Threat Management System (Sistem Manajemen Ancaman Terpadu)

KOLABORASI :

- Next Generation Firewall System (Sistem Firewall Generasi Lanjut)
- Unified Threat Management System (Sistem Manajemen Ancaman Terpadu)
- 0-Day Protection System (Sistem Proteksi 0-Hari)
- Network Analytics System (Sistem Analitik Jaringan)

METODA :

- Signature-based (Berbasis-pratKita)
- Anomaly-based (Berbasis-anomali)
- Policy-based (Berbasis-kebijakan)

SYARAT UMUM :

- Mampu mendeteksi dan mencegah ancaman berbasis-pratKita dengan berkolaborasi dengan sistem firewall generasi lanjut dan/atau sistem manajemen ancaman terpadu.
- Mampu mendeteksi dan mencegah ancaman berbasis-anomali dengan berkolaborasi dengan sistem proteksi 0-hari dan/atau sistem analitik jaringan.
- Dapat mendeteksi dan mencegah pelanggaran kebijakan jaringan dengan berkolaborasi dengan sistem firewall.

CONTOH PRODUK :
- OSSEC (Gratis)
- Sagan (Gratis)
- Open WIPS-NG (Gratis, hanya untuk Jaringan Nirkabel)
- Fail2Ban (Gratis)
- Bro NSM (Gratis)

Untuk IDPS berbayar, saat ini hampir semuanya telah terintegrasi dengan perangkat lainnya

II. Web Application Firewall (Firewall Aplikasi Web)

PENJELASAN :
- WAF adalah sebuah sistem firewall yang dikhususkan untuk menyaring, memantau, dan menghalangi trafik HTTP/S dari dan oleh aplikasi web.

PRASYARAT :
- Next Generation Firewall System (Sistem Firewall Generasi Lanjut)
- Unified Threat Management System (Sistem Manajemen Ancaman Terpadu)

KOLABORASI :
- Next Generation Firewall System (Sistem Firewall Generasi Lanjut)
- Unified Threat Management System (Sistem Manajemen Ancaman Terpadu)

METODA :
- Signature-based (Berbasis-prasyarat)

SYARAT UMUM :
- Mampu mendeteksi dan mencegah ancaman berbasis-pra prasyarat baik dengan kemampuan sendiri ataupun dengan berkolaborasi dengan sistem firewall generasi lanjut dan/atau sistem manajemen ancaman terpadu.
- Mampu bertindak sebagai dekriptor akses berbasis SSL sebelum dienkripsi lagi ke server web.
- Mampu mendeteksi dan mencegah 10 ancaman utama aplikasi web; yaitu:

> Hidden Field Manipulation (Manipulasi field-tersembunyi)

Merupakan salah satu peretasan dengan cara mengubah input pada field-tersembunyi. Serangan ini terutama berfokus pada situs web e-niaga.

> Cookie Poisoning

Merupakan tindakan memanipulasi atau memalsukan cookie (sepotong kecil data yang dibuat dan disimpan di browser/peramban pengguna yang melacak informasi penting mengenai informasi sesi untuk situs tertentu) untuk tujuan melewati langkah-langkah pengamanan atau mengirim informasi palsu ke server.

> Parameter Tampering (Pengubahan Parameter)

Serupa dengan HFM, merupakan salah satu peretasan dengan cara mengubah input pada parameter-parameter web dalam URL, untuk mendapatkan akses otorisasi tanpa seijin pengguna.

> Buffer Overflow Attacks (Serangan Buffer Overflow)

Merupakan peretasan dengan menggunakan input-cacat untuk menyebabkan kondisi dimana ektra informasi tidak dapat ditampung pada buffer dan meng-overwrite ruang memori bersebelahan yang dapat mengakibatkan sistem-terhenti, bocornya informasi, teraksesnya kode tertentu, atau memasukkan kode pada aplikasi.

> Cross Site Scripting (XSS)

Serangan XSS adalah peretasan dengan injeksi, di mana skrip diinjeksikan ke situs web yang tepercaya. Serangan XSS terjadi ketika penyerang menggunakan aplikasi web untuk mengirim kode berbahaya, umumnya dalam bentuk skrip sisi peramban, ke pengguna-akhir yang berbeda.

> Backdoor or Debug options (Opsi debug)

Merupakan peretasan dengan memanfaatkan backdoor atau opsi debug tersedia yang dibuat secara sengaja oleh

developer/pemrogram untuk memeriksa aplikasi, sehingga peretas dapat mengakses informasi sensitif secara mudah.

> Stealth Commanding (Perintah Senyap)

Merupakan peretasan dengan memanfaatkan kelemahan inheren pada eksekusi perintah yang terdapat pada web server.

> Forced Browsing (Jelajah Paksa)

Jelajah paksa adalah serangan untuk mengakses sumber daya yang tidak dirujuk oleh aplikasi, tetapi masih dapat diakses. Peretasan manual ini biasanya menggunakan prediksi bila indeks direktori aplikasi berbasis pada generator angka atau nilai yang terprediksi.

> Third Party Misconfigurations (Miskonfigurasi Pihak Ketiga)

Merupakan kondisi yang sangat umum, dimana kontrol sekuriti untuk aplikasi web dan/atau server gagal diimplementasikan dengan baik atau salah diimplementasikan.

> Known Vulnerabilities (Kerentanan yang diketahui)

Merupakan kondisi dimana pembuatan dan/atau implementasi sebuah aplikasi web menggunakan kode yang secara umum diketahui memiliki celah keamanan, menggunakan kode yang diambil dari orang lain tanpa verifikasi terlebih dahulu, atau menggunakan kode yang diambil dari sumber yang tingkat keamanannya rendah.

CONTOH PRODUK :

- Imperva (Berbayar)
- Akamai (Berbayar)
- Radware (Berbayar)
- Signal Sciences (Berbayar)
- NAXSI (Gratis)
- Nemesida (Gratis)

III. Security Information and Event Management (SIEM)

PENJELASAN :

- SIEM adalah sebuah produk yang menyatukan beberapa fungsi pemantauan; terutama:
  - Log Management (Manajemen Log)
  - Security Information System (Sistem Informasi Keamanan)
  - Security Event System (Sistem Peristiwa Keamanan)
- Dalam praktiknya banyak produk mempromosikan terminologi mereka sendiri.
- Produk-produk komersial memberikan kombinasi yang berbeda dari fungsi-fungsi ini dan tumpang tindih pada praktiknya.
- SIEM dalam penggunaannya memerlukan kesatuan waktu agar tKita siaga yang tercipta dari korelasi informasi dan/atau peristiwa keamanan dapat menjadi sebuah tKita siaga yang tepat dan tidak terancukan oleh informasi dan/atau peristiwa keamanan yang tercipta baik sebelum ataupun sesudahnya.

PRASYARAT :

- Synchronization System (Sistem Sinkronisasi)

KOLABORASI :

- Network Management System (Sistem Manajemen Jaringan)
- Server and Application Monitoring System (Sistem Pemantauan Server dan Aplikasi)
- Unifed Endpoint Management (Manajemen Endpoint Terpadu)
- Performance Monitoring System (Sistem Pemantauan Performa)
- Big Data Engine (Enjin Mahadata)
- IT Service Management (Manajemen Servis TI)
- File Integrity Monitoring (Pemantauan Integritas File)

METODA :
- Data Aggregation (Agregasi Data)
- Correlation (Korelasi)
- Alerting (TKita Siaga)

SYARAT UMUM :
- SIEM minimal harus dapat memenuhi ketiga metoda utama SIEM.
- Log/SNMP Perangkat yang teragregasi ke SIEM harus menggunakan timestamp (stempel waktu) yang tersinkronisasi.

CONTOH PRODUK :
- Splunk (Berbayar)
- LogRhythm (Berbayar)
- Dell RSA (Berbayar)
- Exabeam (Berbayar)
- McAfee (Berbayar)
- Securonix (Berbayar)
- Snort (Gratis)
- OSSEC (Gratis)
- Elasticsearch (Gratis)

IV. Network Behavioral Analysis/NBA (Analisis Perilaku Jaringan)

PENJELASAN :
- NBA adalah cara untuk meningkatkan keamanan jaringan dengan memantau trafik dan mencatat tindakan atau penyimpangan yang tidak biasa dari operasi normal.
- Solusi NBA memantau apa yang terjadi di dalam jaringan, mengumpulkan data dari banyak titik untuk mendukung analisis luring (offline).
- Setelah menetapkan tolok ukur untuk trafik normal, program NBA secara pasif memantau aktivitas jaringan dan menKitai pola yang tidak diketahui, baru atau tidak biasa yang mungkin mengindikasikan adanya ancaman.

- NBA juga dapat memantau dan mencatat tren dalam penggunaan lebar-pita (bandwidth) dan protokol.
- NBA sangat baik untuk menemukan malware baru dan eksploitasi 0-hari.

PRASYARAT :
- Network Analytics System (Sistem Analitik Jaringan)

KOLABORASI :
- 0-Day Protection System (Sistem Proteksi 0-Hari)
- Network Analytics System (Sistem Analitik Jaringan)

METODA :
- Data Aggregation (Agregasi Data)
- Benchmarking (Penetapan Tolok Ukur)
- Traffic Passive Monitoring (Pemantauan Pasif Trafik)
- Flow and Route Analysis (Analisis Aliran dan Rute)

SYARAT UMUM :
- Perangkat NBA minimal dapat memenuhi empat metoda utama di atas.
- Perangkat NBA minimal dapat berfungsi sebagai transparent bridging (jembatan transparan) dalam implementasinya.

CONTOH PRODUK :
- NetScout (Berbayar)
- Riverbed (Berbayar)
- VIAVI (Berbayar)
- Broadcom (Berbayar)
- Colasoft (berbayar)
- LiveAction (Berbayar)
- SevOne (Berbayar)
- FlowMatrix (Gratis)

V. File Integrity Monitoring (Pemantauan Integritas File)

PENJELASAN	:	• FIM merupakan sebuah sistem yang memantau integritas dari sebuah file pada server secara berkala dan terus menerus. • Bila terjadi perubahan pada file yang dipantau, FIM akan memberikan tKita-siaga dan memberikan opsi apakah file tersebut akan dikembalikan sesuai dengan duplikat file yang ada pada FIM atau diotorisasikan untuk diubah dan FIM akan memperbarui duplikat yang ada pada FIM. • FIM dalam bahasa umum biasa disebut dengan Web Deface Monitoring
PRASYARAT	:	-
KOLABORASI	:	• Storage System (Sistem Penyimpanan)
METODA	:	• On premises Agent-based (Di tempat Berbasis-agen) • On premise Agentless (Di tempat Tanpa-agen) • Cloud -based Agent-based (Berbasis-awan Berbasis-agen) • Cloud -based Agentless (Berbasis-awan Tanpa-agen)
SYARAT UMUM	:	• Baik sistem on premise atau cloud-based harus mampu mendeteksi perubahan file secara berkala sesuai dengan waktu yang ditetapkan oleh pengguna. • Untuk sistem berbasis-agen, harus menggunakan sumber daya yang minimal (< 5%) agar tidak mengganggu fungsi utama server.
CONTOH PRODUK	:	• ELock (Berbayar) • TrustWave (Berbayar) • Qualys (Berbayar) • Tripwire (Berbayar) • OSSEC (Gratis)

VI. IP Reputation Engine (Mesin Reputasi IP)

PENJELASAN	:	• IP RE adalah sebuah sistem yang berbasiskan kepercayaan melalui reputasi sebagai sarana mengidentifikasi kemungkinan ancaman keamanan siber. • IP RE dapat dibentuk oleh para pengguna dengan saling menilai dalam komunitas daring (online) untuk membangun kepercayaan melalui reputasi. • Atau IP RE dapat dilakukan oleh sebuah institusi yang kemudian akan membagikannya kepada pengguna melalui perangkat.
PRASYARAT	:	-
KOLABORASI	:	• Next Generation Firewall System (Sistem Firewall Generasi Lanjut) • Unified Threat Management System (Sistem Manajemen Ancaman Terpadu) • Global Reputation Source (Sumber Reputasi Global) • Big Data Engine (Enjin Magadata)
METODA	:	• Active (Aktif) • Passive (Pasive)
SYARAT UMUM	:	• Pada metoda aktif, IP RE minimal harus dapat melakukan pengecekan dan korelasi kepada institusi yang terkait. • Pada metoda pasif, IP RE minimal harus dapat menerima informasi dari sumber kepercayaan reputasi yang terkait.
CONTOH PRODUK	:	• Secucloud (Berbayar) • Akamai (Berbayar) • DenyAll (Berbayar) • Cisco (Berbayar) • McAfee (Berbayar) • DNSBL (Gratis) • Spamhaus (Gratis)

VII. Synchronization System (Sistem Sinkronisasi)

PENJELASAN :

- SS merupakan sebuah sistem sinkronisasi stempel-waktu yang detil dan tepat.
- SS memungkinkan perangkat-perangkat dan/atau sistem-sistem yang terhubung dapat memiliki satu sinkronisasi waktu sehingga memungkinkan sebuah peristiwa terdeteksi secara tepat saat terjadinya.
- Dengan ketepatan sinkronisasi waktu, sebuah peristiwa keamanan dapat terinci kronologinya, pengambilan keputusan juga dapat menjadi lebih tepat, serta memperkecil terjadinya putus jaringan untuk perangkat-perangkat yang sensitif terhadap waktu.

PRASYARAT : -

KOLABORASI :

- Global Positioning System (Sistem Penentuan Posisi Global)
- Master Timing System (Sistem Master Pengaturan Waktu)
- Network Time Protocol Server (Server Protokol Waktu Jaringan)

METODA :

- Master Timing (Master Pengaturan Waktu)
- Grandmaster Timing (Grandmaster Pengaturan Waktu)

SYARAT UMUM :

- MT harus dapat menerima signal waktu secara langsung (Stratum-1) pada 1 (satu) Global Navigation Satellite System (GNSS/Sistem Satelit Navigasi Global) utama yaitu A-GPS, Galileo, BeiDou, dan GLONASS.
- GMT harus dapat menerima signal waktu secara langsung (Stratum-1) pada lebih dari 1 (satu) Global Navigation Satellite System (GNSS/Sistem Satelit Navigasi Global) utama yaitu A-GPS, Galileo, BeiDou, dan GLONASS.
- GMT harus memiliki opsi penambahan atomic clock (jam atom) untuk cadangan penghitungan waktu bilamana terputus secara total dari GNSS.

- MT dan/atau GMT harus dapat berfungsi sebagai NTP Server.

CONTOH PRODUK :
- Bodet MT (Berbayar)
- Fibrolan GMT (Berbayar)
- Orolia MT (Berbayar)

VIII. Network Analytic System (Sistem Analitik Jaringan)

PENJELASAN :
- NAS merupakan sebuah sistem berbasis packet analyzer (penganalisisan paket) yang digunakan sebagai pengumpul data jaringan menjadi sebuah mahadata untuk mengetahui performa jaringan, memprediksi tren trafik jaringan, dan/atau memantau reliabilitas jaringan yang ada.
- Hasil dari NAS dapat diolah lebih lanjut untuk NBA, dan saat ini banyak perangkat NAS juga dapat langsung berfungsi sebagai perangkat NBA.

PRASYARAT : -

KOLABORASI :
- Intrusion Prevention System (Sistem Pencegahan Intrusi)
- 0-Day Protection System (Sistem Proteksi 0-Hari)

METODA :
- Manual
- Otomatis

SYARAT UMUM :
- NAS yang terhubung dengan NBA dan IPS atau berfungsi gKita sebagai NBA dan IPS harus dapat memutuskan dan/atau membelokkan alur jaringan pada kejadian DDoS.

CONTOH PRODUK :
- Radware (Berbayar)
- Netscout (Berbayar)
- Cisco (Berbayar)
- Riverbed (Berbayar)
- Akamai (Berbayar)

IX. Secure Proxy (Proksi Aman)

PENJELASAN	:	• SP merupakan sebuah cara mengalihkan jalur koneksi-logik dari langsung menjadi melewati SP terlebih dahulu. • SP akan menyaring koneksi ke domain sebenarnya dan meneruskan koneksi yang telah tersaring ke domain sebenarnya.
PRASYARAT	:	-
KOLABORASI	:	Security Information and Event Management (Manajemen Informasi dan Peristiwa Keamanan)
METODA	:	• On premises (Di tempat) • Cloud -based (Berbasis-awan)
SYARAT UMUM	:	Baik sistem on premise atau cloud-based harus mampu mendeteksi gangguan dan ancaman sebelum meneruskan ke koneksi sebenarnya.
CONTOH PRODUK	:	• BitGlass (Berbayar) • Symantec (Berbayar) • Microsoft (Berbayar) • Netskope (Berbayar) • McAfee (Berbayar) • CipherCloud (Berbayar)

X. Deception System (Sistem Pengecoh)

PENJELASAN	:	• DS merupakan pengembangan dari Honeypot System (Sistem Pot-Madu) yaitu sebuah sistem yang tidak hanya membuat sebuah Pot-Madu untuk mengecoh peretas melainkan membuat keseluruhan sistem palsu baik perangkat keras server, perangkat keras jaringan, perangkat keras pengguna, serta perangkat lunaknya. • DS juga kan membuat keseluruhan kredensial yang nantinya akan digunakan untuk membangun labirin-sistem dimana sistem asli akan tersembunyi.

- Bila peretas berhasil menembus pertahanan utama sistem keamanan (dalam hal ini UTM/NGFW yang kerap digunakan), DS akan menjadikan sistem palsu sebagai target empuk yang mudah diretas, kemudian DS akan menyuplai peretas dengan berbagai macam data, file, dan akses ke perangkat palsu lainnya untuk memperlambat proses peretasan hingga pihak terkait dapat mengisolasi peretasan.

PRASYARAT : -

KOLABORASI : Security Information and Event Management (Manajemen Informasi dan Peristiwa Keamanan)

METODA :
- Appliance based with agent and Software based for Cloud (Berbasis perangkat disertai agen dan perangkat lunak untuk Sistem-awan)

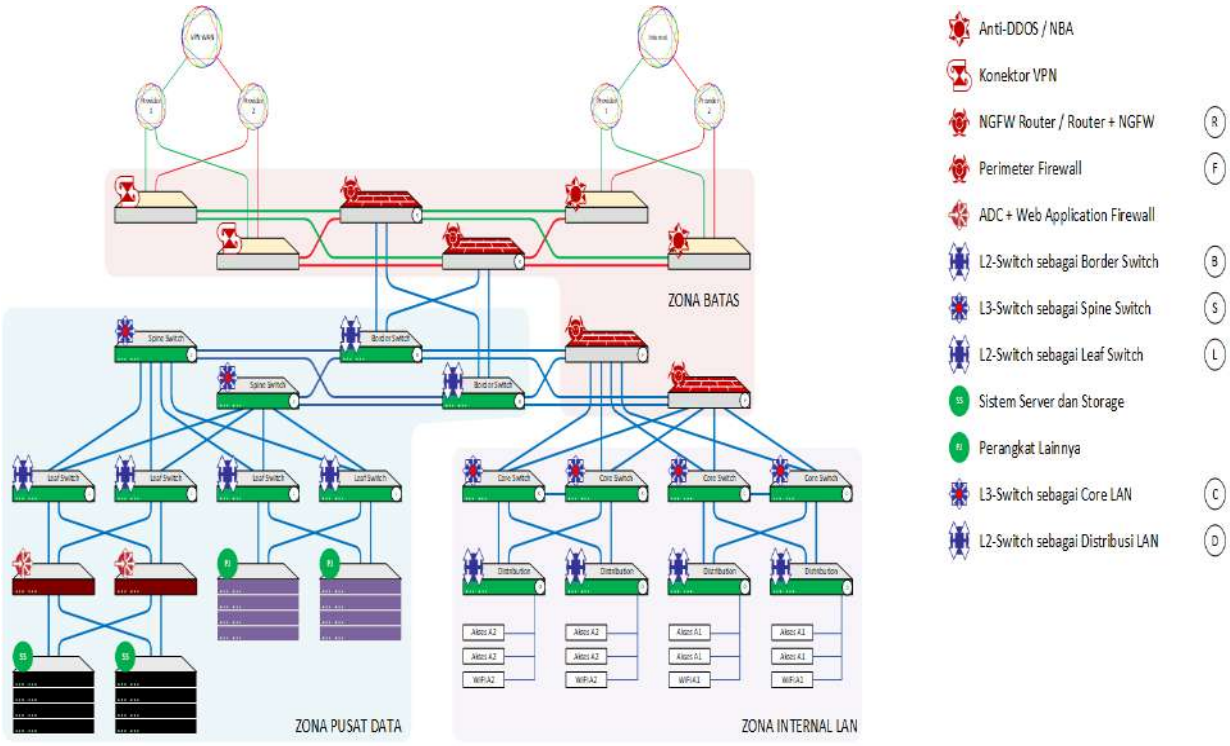
SYARAT UMUM :
- Perangkat utama DS harus mampu membuat keseluruhan sistem palsu untuk mengecoh peretas baik dari eksternal ataupun internal.
- Agen DS harus mampu membuat keseluruhan kredensial palsu untuk masing-masing perangkat server dan perangkat pengguna untuk mengecoh peretas.
- Perangkat lunak DS harus mampu bertindak sebagai redundan untuk Server berbasis sistem-awan.

CONTOH PRODUK :
- Attivo (Berbayar)
- Acalvio (Berbayar)
- TrapX (Berbayar)
- Illusive (Berbayar)
- Fidelis (Berbayar)

XI. STANDAR DESAIN NETWORK & PERANGKAT KEAMANAN (TOPOLOGI SPINE – LEAF)

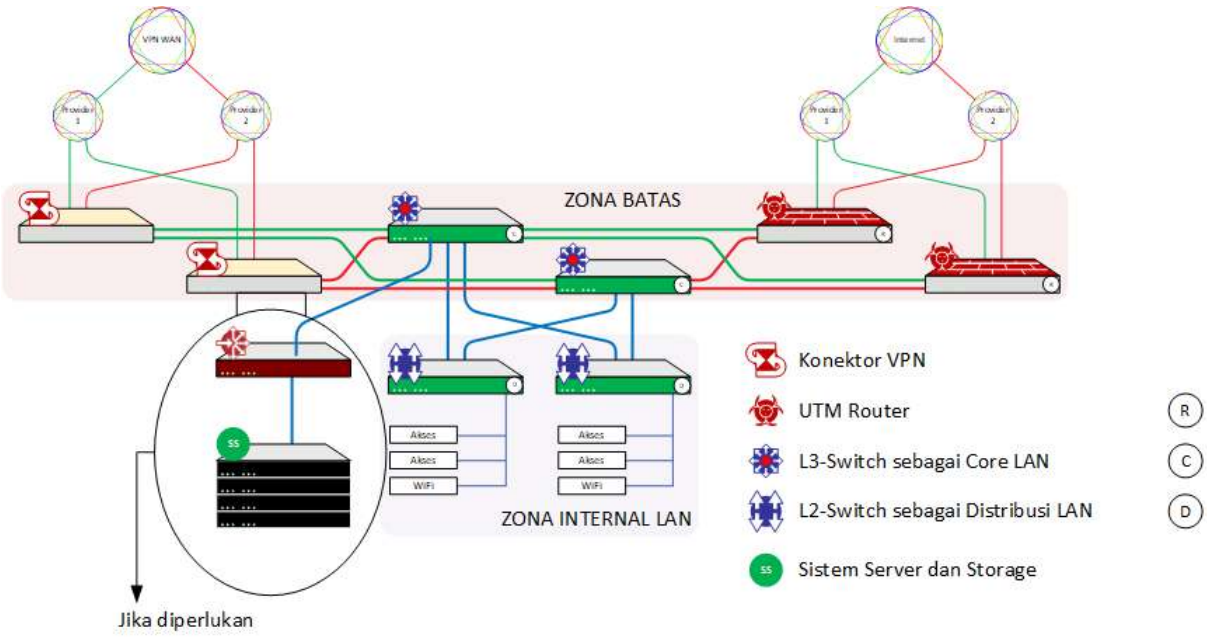
1. Kantor Pusat

Gambar 3.8 Standar Desain Network & Perangkat Keamanan Kantor Pusat



2. Kantor Wilayah/Cabang/Regional

Gambar 3.9 Standar Desain Network & Perangkat Keamanan Kantor Wilayah/Cabang



Gambaran Umum:

1. Topologi Spine and Leaf adalah topologi jaringan pusat data (Data Center) dua lapis yang berguna untuk pusat data (Data Center) yang mengalami lebih banyak lalu lintas jaringan timur-barat (horizontal antar server; contohnya server aplikasi, server web, server transaksi, dll) daripada lalu lintas utara-selatan (vertikal dari server ke luar datacenter; contohnya mainframe, koneksi klien-server, thin-client, terminal server, dll).
2. Topologi terdiri dari switch-leaf (yang terhubung dengan server dan penyimpanan/storage) dan switch-spine (terhubung dengan leaf-switch).
3. Switch-leaf terhubung secara mesh ke switch-spine, membentuk lapis-akses yang memberikan titik koneksi jaringan untuk server.

Karakteristik

1. Memiliki dua lapis switch yaitu:
    - Switch-Spine sebagai core
    - Switch-Leaf sebagai:
        - Switch Akses ke server dan penyimpanan/storage
        - Pembatas (Border Switch) ke jaringan lainnya di luar jarigan pusat data (datacenter)
2. Tidak terdapat koneksi antar Switch-Spine.
3. Menggunakan Protokol-Routing untuk meningkatkan penggunaan lebar-pita (bandwidth) secara penuh untuk koneksi utara-selatan (vertikal dari switch-leaf ke switch-spine).
4. Tidak menggunakan STP (*Spanning Tree Protocol*) untuk menghindari looping melainkan menggunakan protokol-routing seperti OSPF (*Open Shortest Path First*) atau BGP (*Border Gateway Protocol*)
5. Seluruh rute dikonfigurasi aktif serta menggunakan ECMP (*Equal-Cost Multipathing*)

Keuntungan

1. Stabilitas
    - Penambahan switch dan/atau perawatan/maintenance switch tidak memerlukan downtime.
    - Kerusakan baik switch-spine ataupun switch-leaf tidak menghasilkan downtime.
    - Tidak memerlukan jeda waktu untuk penentuan koneksi karena tidak menggunakan STP melainkan protocol-routing .

2. Skalabilitas

    - Penambahan kapasitas dapat dilakukan dengan mudah yaitu dengan menambahkan Switch-Spine dan/atau Switch-Leaf.

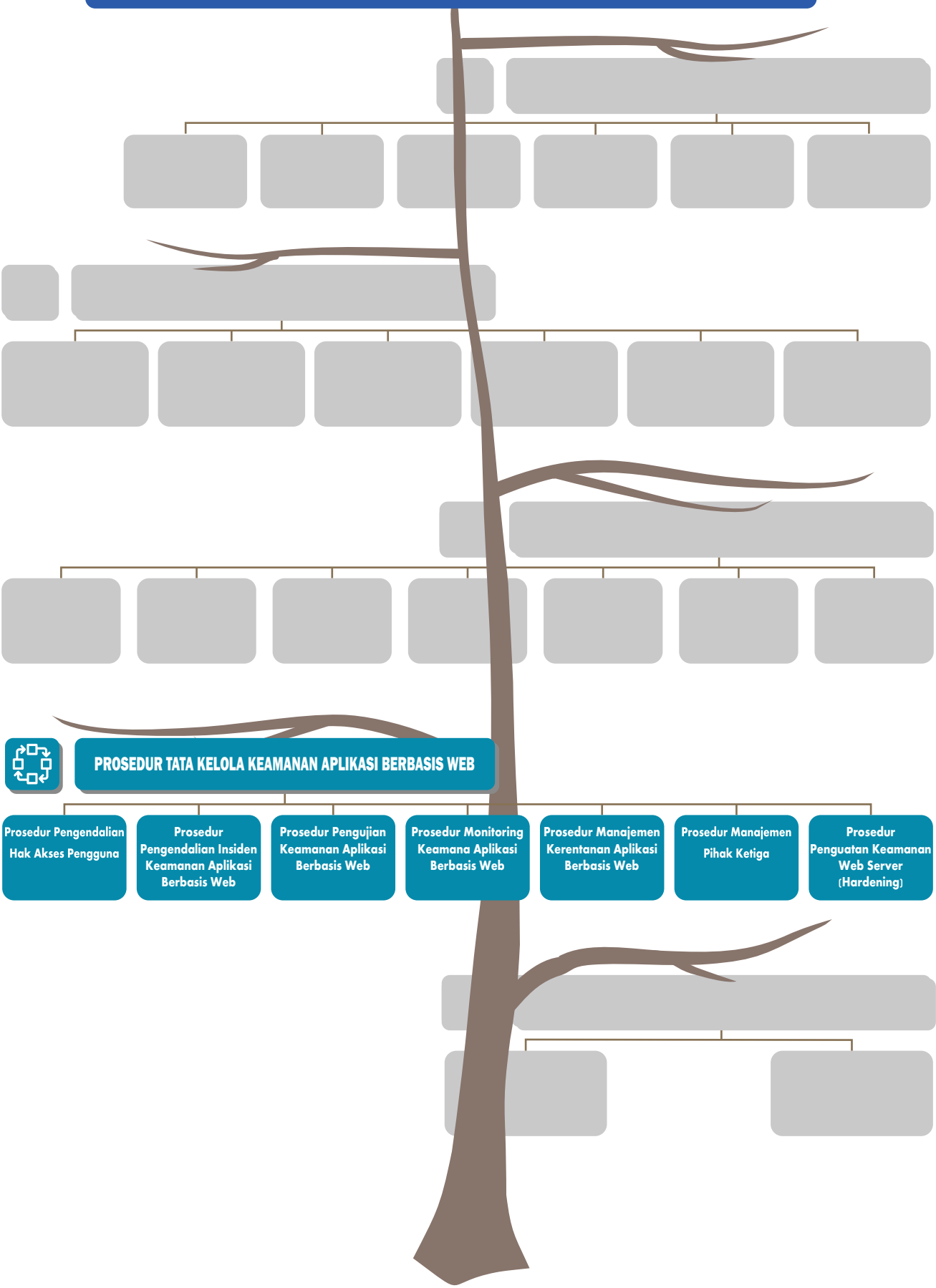
3. Pengurangan latensi

    - Hanya menggunakan dua lapis switch dibandingkan dengan topologi terdahulu yang menggunakan tiga lapis switch.

4. Pengurangan biaya

    - Topologi Spine and Leaf tidak selalu memerlukan switch yang sangat besar ataupun padat densitas portnya, sehingga mengurangi biaya pembelian dan biaya penggunaan listrik tiap bulannya.

PEDOMAN TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB



IV. PROSEDUR TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB

IV.1 PROSEDUR PENGENDALIAN HAK AKSES PENGGUNA

PROSEDUR PENGENDALIAN/PEMBAGIAN HAK AKSES PENGGUNA

A. TUJUAN:

1. Memastikan bahwa hanya pengguna yang berwenang yang berhak akses ke sumber daya informasi .
2. Mengatur tata cara dan tahapan dalam melakukan pengendalian terhadap akses sumber daya informasi.

B. RUANG LINGKUP:

1. Prosedur ini berlaku untuk akses terhadap:
    a. Fasilitas email dan internet
    b. Aplikasi Aplikasi Berbasis Web, basis data, sistem operasi

C. REFERENSI:

1. ISO 27001:2005 (Access Control)

D. DEFINISI/SINGKATAN:

1. Pemohon adalah pengguna TIK yang mengirim permintaan penanganan insiden dan/atau layanan TIK.
2. TIK (Teknologi Informasi dan Komunikasi) adalah meliputi perangkat keras, perangkat lunak dan sistem operasi PC (*Personal Computer*) dan *Server*, Jaringan Komunikasi Data dan Suara, Aplikasi Sistem Informasi dan *Database* Sistem Informasi.
3. User adalah akun yang digunakan untuk mengakses suatu aplikasi atau sistem
4. Hak akses adalah hak akses suatu akun terhadap suatu aplikasi atau sistem (read, update, write, upload, download, dll)
5. Struktural dibawah Penanggungjawab TI (Teknologi Informasi) adalah:
    a. Dukungan Teknis (Dukti)
    b. Pengembangan Aplikasi (Bangsi)
    c. Pengembangan Infrastruktur (Bangtur)
    d. Operasional TI (Opti)
    e. Layanan TI (Yanti)

E. PENANGGUNGJAWAB:

1. Penanggungjawab TI bertindak dan bertanggungjawab atas pengelolaan TI secara keseluruhan.
2. Penanggungjawab TI bertindak sebagai penanggungjawab atas Proses Pengendalian Hak Akses
3. Pengembangan Aplikasi (Bangsi) bertindak sebagai pelaksana Pengendalian Hak Akses aplikasi dan database.
4. Pengembangan Infrastruktur (Bangtur) bertindak sebagai pelaksana Pengendalian Hak Akses infrastruktur ke jaringan intranet.
5. Operasional TI (Opti) bertindak sebagai pelaksana Pengendalian Hak Akses ke jaringan internet, wide area network, administrasi database dan administrasi sistem operasi server.

F. DOKUMEN PENDUKUNG:

1. F-PY, Form Permintaan Layanan Hak Akses Aplikasi dan Data
2. L-LP,Laporan Layanan Pengguna
3. L-HA, Laporan Permintan Hak Akses
4. D-UV, Daftar pengguna Valid

G. URAIAN PROSEDUR:

1. Pemohon:
    a. Mengirim permintaan layanan TIK terkait Hak Akses.
    b. Menerima Laporan Permintaan Hak Akses
    c. Menerima dan memproses terkait konfirmasi validitas pengguna
    d. Mengirim pengguna yang valid

2. Layanan TI (Yanti):
    a. Menerima dan memeriksa permintaan layanan TIK terkait Hak Akses dari Pemohon
    b. Menerima permintaan generate ND.PPR terkait dengan permintaan Hak Akses
    c. Membuat ND.PPR untuk permintaan Hak Akses
    d. Melakukan pemeriksaan up-to-date CMDB bekerjasama dengan Struktural dibawah Penanggungjawab TI

e. Membuat L-LP dan kirim ke Penanggungjawab TI.

f. Membuat L-HA dan kirim ke Pemohon.

g. Menerima Nota Dinas.Konfirmasi validitas pengguna

h. Mengirim daftar pengguna ke Pemohon

i. Menerima daftar pengguna yang valid dari Pemohon

3. Penanggungjawab TI:

   a. Menerima L-LP dari Layanan TI (Yanti)

4. Struktural dibawah Penanggungjawab TI (Bangsi, Bangtur, Opti):

   a. Menyusun daftar pengguna yang akses ke:

      - Fasilitas email dan Internet
      - Aplikasi, basis data, sistem operasi

   b. Mengidentifikasi dan mengkategorisasi pengguna yang valid dan yang tidak valid

   c. Mengirim Nota Dinas. daftar pengguna dan konfirmasi validitas pengguna

   d. Melakukan update pengguna (hak akses)

H. PERFORMA INDIKATOR:

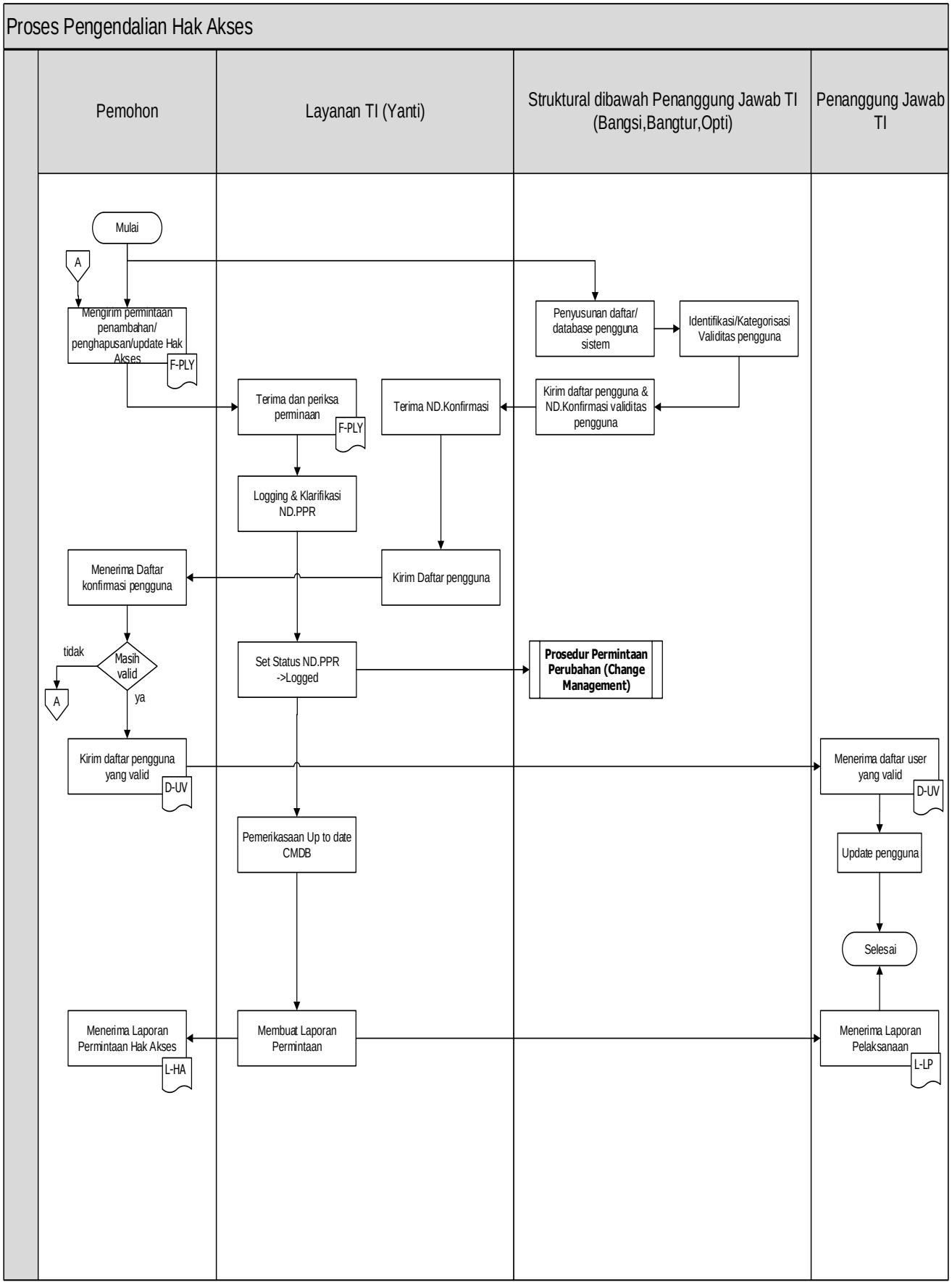
1. Jumlah permintaan registrasi dan penghapusan hak akses yang diselesaikan
2. Jumlah peninjauan hak akses yang diselesaikan

I. LAMPIRAN

1. Tidak ada

Disposisi	Nama	Jabatan	Paraf
Disiapkan oleh			
Diperiksa oleh			
Disahkan oleh			

Gambar 4.1 Prosedur Pengendalian Hak Akses Pengguna



Formulir	:	F-PY. 01
Judul	:	Formulir Permintaan Hak Akses Aplikasi dan Data

Formulir Permintaan Hak Akses Aplikasi dan Data

Dasar Permintaan

Surat Nomor :

Tanggal : ☐☐ - ☐☐ - ☐☐☐☐

tanggal bulan tahun

Komponen Aplikasi

Nama Aplikasi :

Nama File Eksekusi :

Platform : ☐ *Windows* ☐ *Unix* ☐ Lainnya, sebutkan

:

Jenis Aplikasi : ☐ *Web Based* ☐ *Client Server* ☐ Lainnya, sebutkan

Bhs. Pemrograman :

Engine Aplikasi : ☐ IIS ☐ *Apache* ☐ Lainnya, sebutkan

Alokasi *Server*	
Alokasi IP	

Komponen Data

Nama Data :

Platform : ☐ *Windows* ☐ *Unix* ☐ Lainnya, sebutkan

:

Database : ☐ SQL *Server* ☐ *Oracle* ☐ Lainnya, sebutkan

Volume Inisialisasi : ☐ kB

Est. Pertumbuhan : ☐ % per tahun

Dukungan *Back-up* : ☐ Ya ☐ Tidak

Penanggungjawab Struktural

I Penanggungjawab Teknis 1

1. Nama/NIP : ...../.....
2. Unit Kerja : .....
3. Telephone : .....
4. Email : .....

II Penanggungjawab Teknis 2

1. Nama/NIP : ...../.....
2. Unit Kerja : .....
3. Telephone : .....
4. Email : .....

Mengetahui, Jakarta, 20.....

Kepala Unit TI Pemohon,

ID No. ID No.

Formulir	:	D-UV 01
Judul	:	Daftar Pengguna User Sistem/Aplikasi

Daftar Pengguna User Sistem/Aplikasi

Nama Sistem/ Aplikasi :

No	Nama Pengguna	NIP	Username	Jenis Akses (Admin/Super User/User)

Formulir	:	L-LP 01
Judul	:	Laporan Layanan Pengguna

Laporan Layanan Pengguna

Tiket ID	Nama Permintaan Hak Akses	Tanggal Permintaan	Tanggal Pelaksanaan	Status (Selesai/Tidak)	Update CMDB	Keterangan

Mengetahui, Jakarta, 20.....

Kepala Unit TI Pemohon

ID No. ID No.

Formulir	:	L-HA 01
Judul	:	Laporan Hak Akses

Laporan Hak Akses

Tiket ID :

Tanggal Permintaan :

Tanggal Pelaksanaan :

Status : ☐ Selesai

☐ Tidak Selesai

Daftar Pengguna yang diberi Akses

Nama	NIP	Jenis Akses	Keterangan

Penjelasan Umum Teknis

Mengetahui, Jakarta, 20,.....

Kepala Unit TI Pemohon

ID No. ID No.

IV.2 PROSEDUR PENGENDALIAN INSIDEN KEAMANAN APLIKASI BERBASIS WEB

PROSEDUR PENGENDALIAN INSIDEN KEAMANAN APLIKASI BERBASIS WEB

A. TUJUAN

1. Memitigasi risiko insiden keamanan Aplikasi Berbasis Web
2. Memastikan bahwa setiap petugas/tim keamanan informasi dan Aplikasi Berbasis Web, tanggap terhadap insiden keamanan informasi/Aplikasi Berbasis Webyang terjadi.
3. Memastikan penanganan insiden keamanan Aplikasi Berbasis Webdengan cepat dan tepat.
4. Memastikan adanya knowledge management dan repository yang berkelanjutan dalam menangani insiden keamanan Aplikasi Berbasis Web

B. RUANG LINGKUP

Prosedur ini hanya berlaku untuk pengendalian insiden yang dikarenakan oleh:

a. DoS/DDoS
b. Malware
c. Web defacement
d. Phishing

C. REFERENSI

1. NIST Special Publication 800-61 Revision 2: Computer Security Incident Handling Guide

D. DEFINISI/SINGKATAN

1. Pemohon adalah pengguna TIK yang mengirim permintaan pengendalian insiden dan/atau layanan TIK.
2. TIK (Teknologi Informasi dan Komunikasi) adalah meliputi perangkat keras, perangkat lunak dan sistem operasi PC (*Personal Computer*) dan *Server*, Jaringan Komunikasi Data dan Suara, Aplikasi Sistem Informasi dan *Database* Sistem Informasi.
3. User adalah akun yang digunakan untuk mengakses suatu aplikasi atau sistem
4. Hak akses adalah hak akses suatu akun terhadap suatu aplikasi atau sistem (read, update, write, upload, download, dll)

E. PENANGGUNGJAWAB

1. Tim Security Incident Respon bertindak dan bertanggungjawab atas penanganan insiden keamanan Aplikasi Berbasis Webdan mencari breakthrough agar kondisi setelah insiden tersebut dapat kembali ke kondisi normal secepat - cepatnya
2. Tim Teknis bertindak dan bertanggungjawab atas proses perbaikan karena insiden sehingga sistem dapat beroperasi secara nomal kembali.

F. LANGKAH – LANGKAH

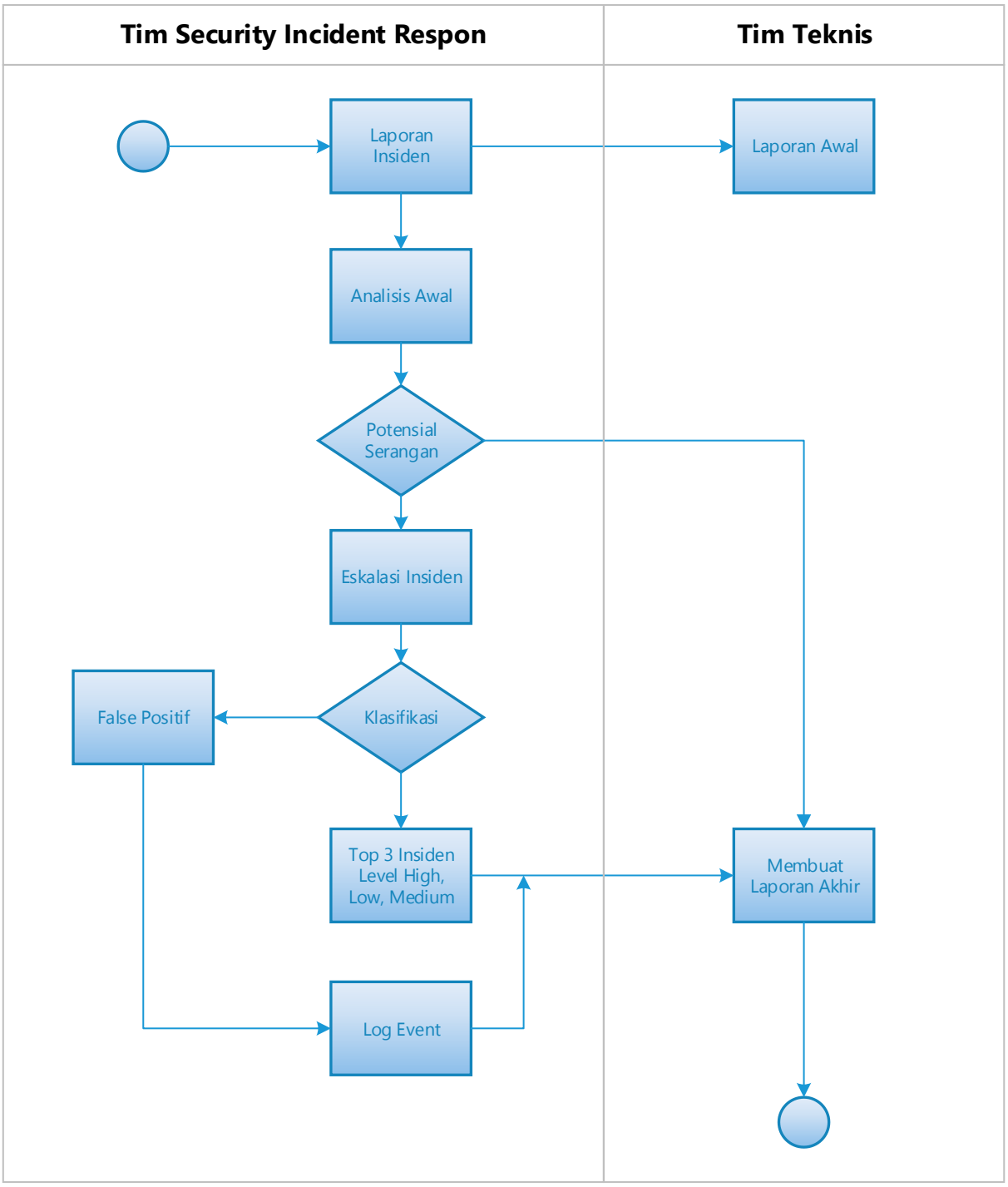
a. Aktivitas Incident Respon

Tabel 4.1 Aktivitas Incident Respon

LANGKAH	AKTIVITAS	AKTOR
1	Tim Security Incident Respon menemukan insiden dan melaporkan kepada Tim Teknis	Tim Security Incident Respon
2	Tim Security Incident Respon melakukan analisis Awal dengan mencari informasi dan data – data yang terkait insiden	Tim Security Incident Respon
3	Berdasarkan hasil analisis kemudian menentukan seberapa besar potensial serangan tersebut. Jika false positive maka akan dibuatkan laporan akhir	Tim Security Incident Respon
4	Apabila insiden berpotensi serangan maka melakukan eskalasi lebih dalam dan melakukan pengkajian	Tim Security Incident Respon
5	Mengklasifikasikan kategori Insiden dari jenis serangan, jika bukan termasuk ke top 3 maka akan dimasukan ke log events	Tim Security Incident Respon
6	Kategori insiden termasuk Top 3 , maka tim analisis Security Incident Respon membuat laporan penanganan	Tim Security Incident Respon
7	Tim analisis Security Incident Respon membuat laporan akhir	Tim Security Incident Respon

b. Diagram Alir Incident Respon

Gambar 4.2 Diagram Alir Incident Respons



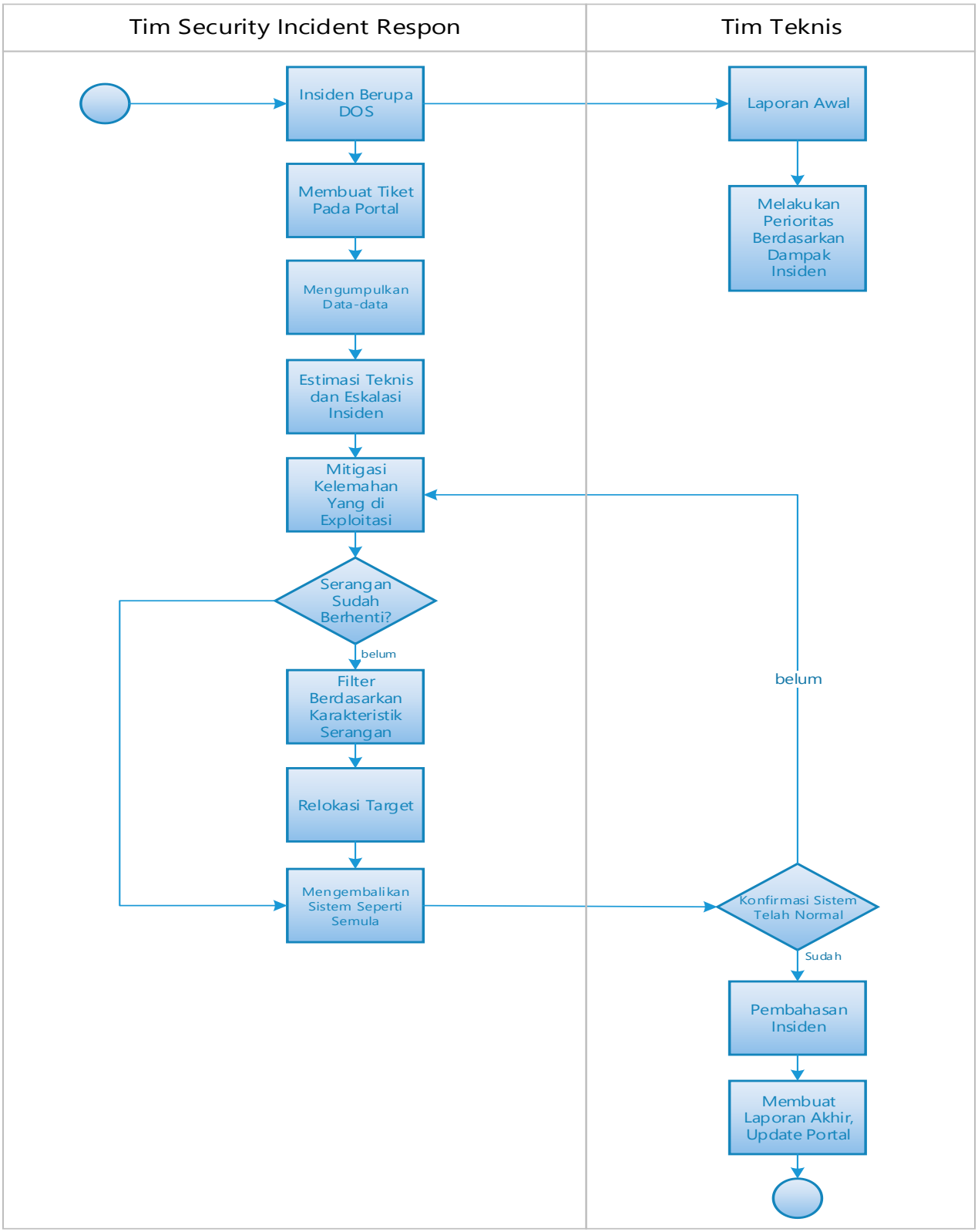
c. Proses Penanganan Insiden DoS/DdoS

Tabel 4.2 Proses Penanganan Insiden DoS/DDoS

LANGKAH	AKTIVITAS	Aktor
1	Tim Security Incident Respon menemukan insiden DoS dan melaporkan kepada Tim Teknis	Tim Security Incident Respon
1.2	Membuat laporan awal terkait insiden DoS	Tim Teknis
1.3	Melakukan prioritas untuk penanganan	
2	Tim Security Incident Respon membuat tiket insiden pada portal serta melaporkan ke tim teknis	Tim Security Incident Respon
3	Melakukan pengumpulan data – data terkait Insiden DoS (Log)	Tim Security Incident Respon
4	Apa pengaruh dari dampak serangan insiden DoS terhadap jaringan di Kemhan, dan menutup service apache yang bermasalah	Tim Security Incident Respon
5	Melakukan identifikasi serta mitigasi kerentanan Service Apache yang dieksploitasi dari serangan DoS	
6	Apakah serangan belum berhenti ?	Tim Security Incident Respon
6.1	Melakukan filtering berdasarkan karateristik serangan DoS	
6.2	Melakukan mitigasi ulang terhadap kerentanan Service Apache	
7	Apabila serangan berhenti	Tim Security Incident Respon
7.1	Service Apache yang sudah di Patch dioperasikan kembali	Tim Security Incident Respon
8	Konfirmasi ke user apakah masih terjadi serangan	Tim Teknis
8.1	Apabila belum maka proses mitigasi diulangi kembali	Tim Security Incident Respon
9	Melakukan pembahasan insiden DoS	Tim Teknis
10	Membuat laporan hasil Analisis Serangan DoS terhadap Service Apache dan penangulangan terhadap insiden tersebut.	Tim Teknis

d. Diagram Alir Penanganan Insiden DoS/DDoS

Gambar 4.3 Diagram Alir Penanganan Insiden DoS/DDoS



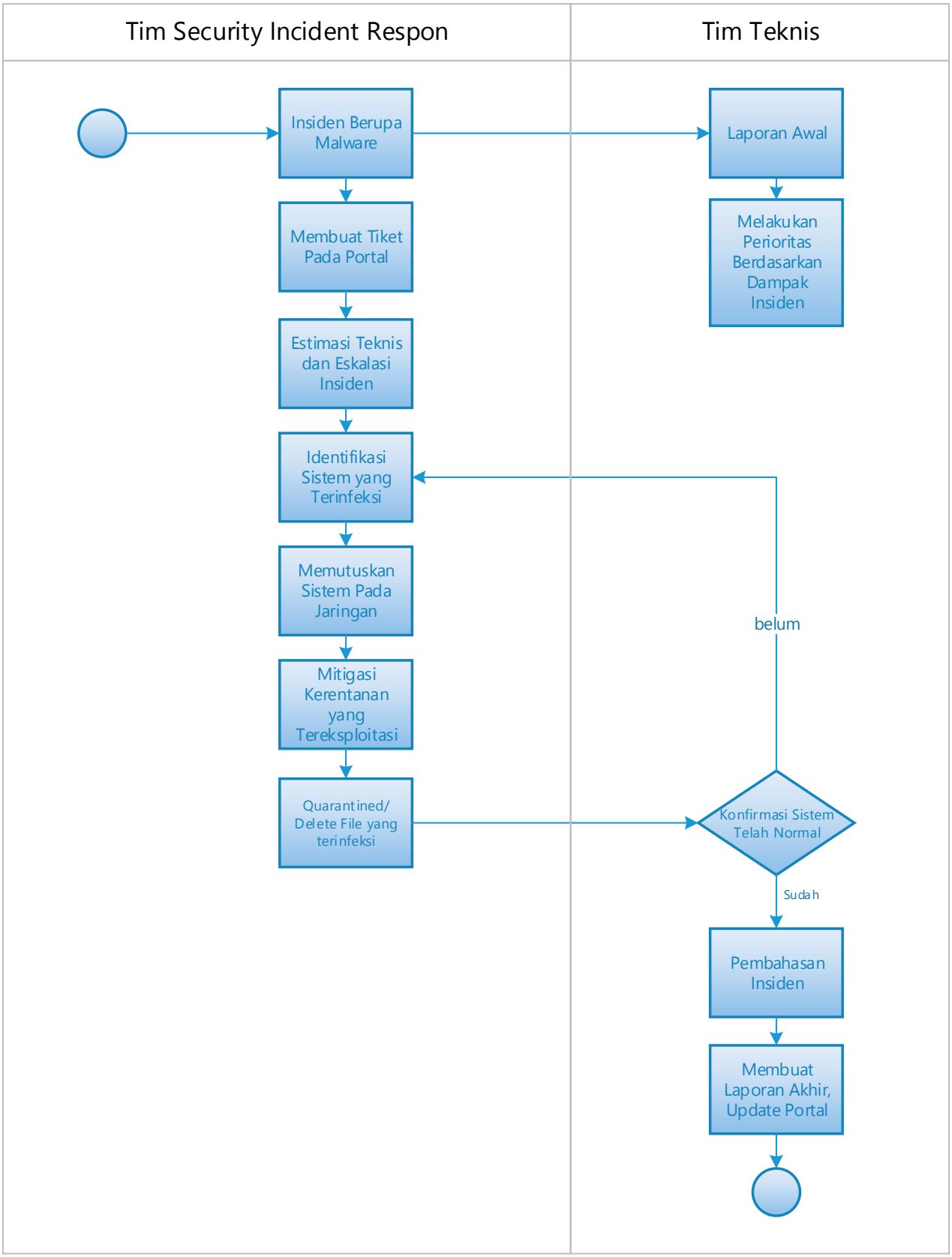
e. Aktivitas Penanganan Insiden Malware

Tabel 4.3 Aktivitas Penanganan Insiden Malware

LANGKAH	AKTIVITAS	PERSONIL
1	Tim Security Incident Respon menemukan insiden Malware dan melaporkan kepada Tim Teknis	Tim Security Incident Respon
1.2	Membuat laporan awal terkait insiden Malware	Tim Teknis
1.3	Melakukan prioritas untuk penanganan	Tim Teknis
2	Tim Security Incident Respon membuat tiket insiden pada portal serta melaporkan ke tim teknis	Tim Security Incident Respon
3	Melakukan identifikasi serta mitigasi sistem yang terinfeksi	
4	Apabila malware tersebut sangat berbahaya maka dilakukan pemutusan jaringan terhadap sistem yang terinfeksi	Tim Security Incident Respon
5	Melakukan Mitigasi terhadap kerentanan yang dieksploitasi oleh malware	Tim Security Incident Respon
6	Tim Security Incident Respon memberikan arahan kepada tim teknis untuk membersihkan, mengkarantina atau mendelete file yang terinfeksi	Tim Security Incident Respon
7	Konfirmasi ke Tim Teknis apakah masih terjadi serangan	Tim Teknis
7.1	Apabila belum maka proses mitigasi diulangi kembali	Tim Security Incident Respon
8	Melakukan pembahasan insiden Malware	Tim Teknis
9	Membuat laporan hasil Analisis Serangan Malware terhadap Sistem dan penangulangan terhadap insiden tersebut.	Tim Teknis

f. Diagram Alir Penanganan Insiden Malware

Gambar 4.4 Diagram Alir Penanganan Insiden Malware



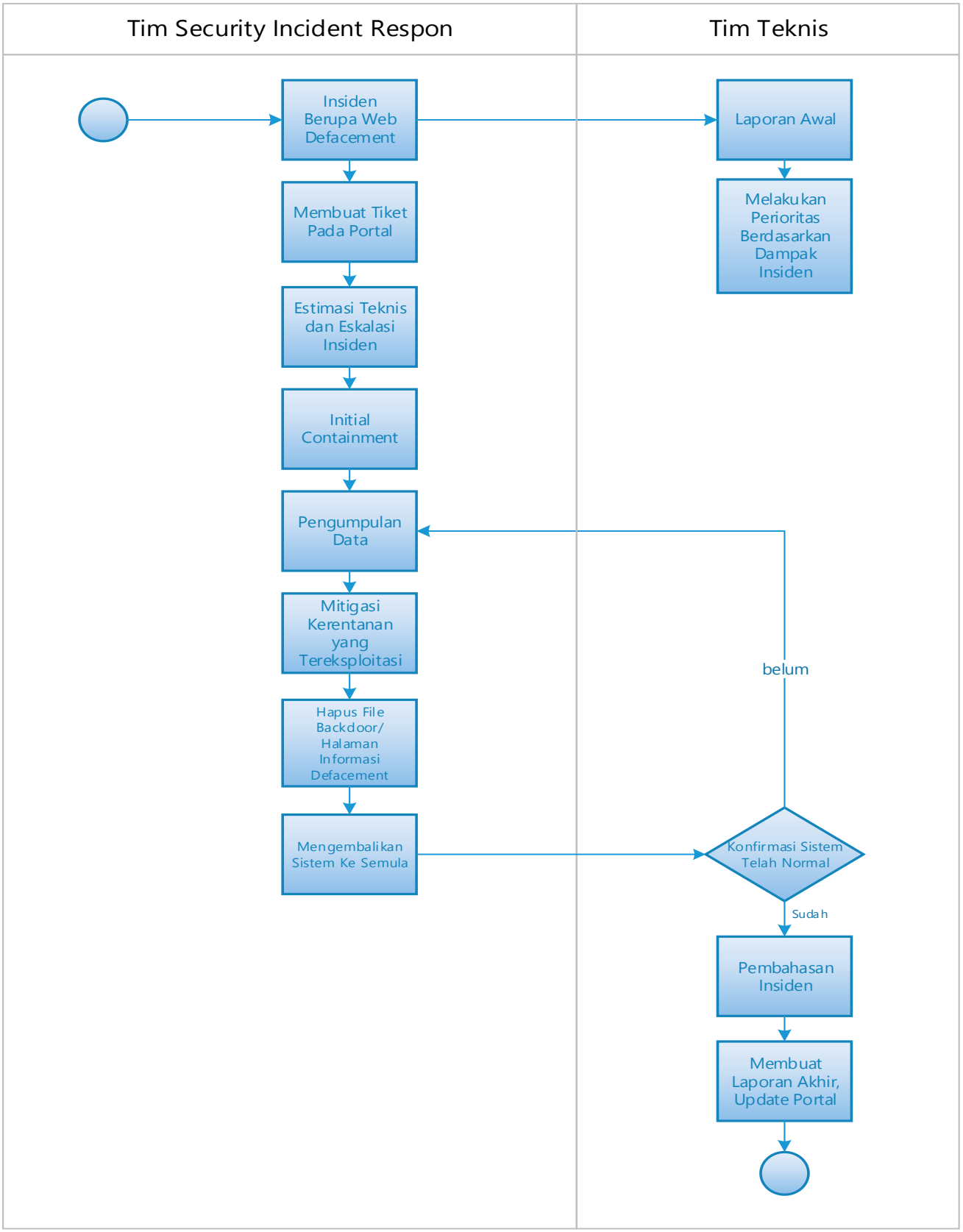
g. Aktivitas Penanganan Insiden Web Defacement

Tabel 4.4 Aktivitas Penanganan Insiden Web Defacement

LANGKAH	AKTIVITAS	PERSONIL
1	Tim Security Incident Respon menemukan insiden Web defacement dan melaporkan kepada Tim Teknis	Tim Security Incident Respon
1.2	Membuat laporan awal terkait insiden Web Defacement	Tim Teknis
1.3	Melakukan prioritas untuk penanganan	Tim Teknis
2	Tim Security Incident Respon membuat tiket insiden pada portal serta melaporkan ke tim teknis	Tim Security Incident Respon
3	Apa pengaruh atau dampak dari serangan defacement tersebut.	Tim Security Incident Respon
4	Memberikan arahan pengantian halaman web yang terdefacement diganti dengan halaman informasi “sedang dalam perbaikan”	Tim Security Incident Respon
5	Melakukan pengumpulan data – data terkait temuan – temuan aplikasi hacking seperti backdoor dan file halaman informasi defacement	Tim Security Incident Respon
6	Melakukan indentifikasi serta mitigasi kerentanan pada web yang ter-deface	Tim Security Incident Respon
6	Hapus file backdoor dan file halaman informasi deface	Tim Security Incident Respon
7	Mengembalikan halaman web seperti semula, berdasarkan data web yang pernah di backup	
8	Konfirmasi ke Tim Teknis apakah masih terjadi serangan	Tim Teknis
8.1	Apa bila belum maka proses mitigasi diulangi kembali	Tim Security Incident Respon
8	Melakukan pembahasan insiden Web defacement	Tim Teknis
9	Membuat laporan hasil Analisis Web defacement terhadap Sistem dan penangulangan terhadap insiden tersebut.	Tim Teknis

h. Diagram Alir Penanganan Insiden Web Defacement

Gambar 4.5 Diagram Alir Penanganan Insiden Web Defacement



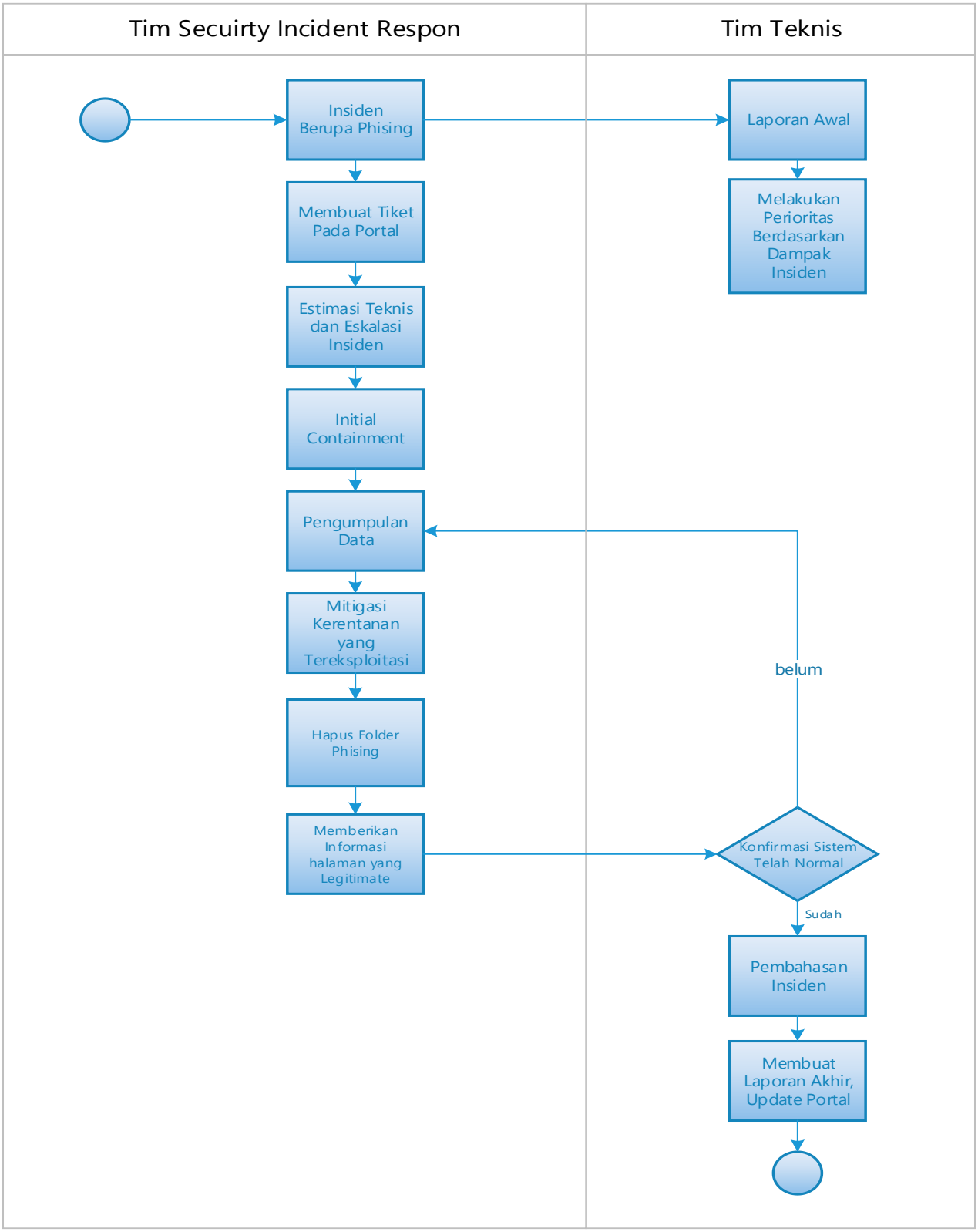
i. Aktivitas Penanganan Insiden Phising

Tabel 4.5 Aktivitas Penanganan Insiden Phising

LANGKAH	AKTIVITAS	PERSONIL
1	Tim Security Incident Respon menemukan insiden Phishing dan melaporkan kepada Tim Teknis	Tim Security Incident Respon
1.2	Membuat laporan awal terkait insiden Web Defacement	Tim Teknis
1.3	Melakukan prioritas untuk penanganan	Tim Teknis
2	Tim Security Incident Respon membuat tiket insiden pada portal serta melaporkan ke tim teknis	Tim Security Incident Respon
3	Apa pengaruh atau dampak dari Phishing tersebut apakah pencurian account (data user dan password)	Tim Security Incident Respon
4	Memberikan arahan mendisable halaman web Phishing	Tim Security Incident Respon
5	Melakukan pengumpulan data – data terkait temuan – temuan aplikasi hacking seperti backdoor dan file Phishing	Tim Security Incident Respon
6	Melakukan indentifikasi serta mitigasi kerentanan pada web yang tersisipi halaman Phishing	Tim Security Incident Respon
6	Hapus file backdoor dan file/Folder Phishing	Tim Security Incident Respon
7	Memberikan informasi halaman Aplikasi Berbasis Webyagn legitimate	Tim Security Incident Respon
8	Konfirmasi ke Tim Teknis apakah masih terjadi serangan	Tim Teknis
8.1	Apabila belum maka proses mitigasi diulangi kembali	Tim Security Incident Respon
8	Melakukan pembahasan insiden Phishing site	Tim Teknis
9	Membuat laporan hasil Analisis Phishing dan penangulangan terhadap insiden tersebut.	Tim Teknis

j. Diagram Alir Penanganan Insiden Phising

Gambar 4.6 Diagram Alir Penanganan Insiden Phising



IV.3 PROSEDUR PENGUJIAN KEAMANAN APLIKASI BERBASIS WEB

PROSEDUR
PENGUJIAN KEAMANAN APLIKASI BERBASIS WEB

A. TUJUAN:

1. Memberikan referensi panduan pengujian keamanan Aplikasi Berbasis Web yang secara best practice dimaksudkan untuk dapat menemukan risiko keamanan dalam suatu sistem Aplikasi Berbasis Web.
2. Memastikan bahwa setiap petugas/tim keamanan informasi dan Aplikasi Berbasis Web, tanggap dalam melakukan langkah untuk menghilangkan risiko keamanan yang terdapat dalam suatu sistem Aplikasi Berbasis Web.
3. Memastikan Aplikasi Berbasis Web beserta datanya aman dari kemungkinan berbagai tindakan yang tidak sah (*unauthorized action*).
4. Memastikan kegiatan pengujian keamanan Aplikasi Berbasis Web berjalan dengan baik dan efektif.
5. Memastikan adanya knowledge management dan repository yang berkelanjutan dalam aktivitas pengujian keamanan Aplikasi Berbasis Web.

B. RUANG LINGKUP:

1. Prosedur ini mencakup langkah-langkah pengujian keamanan Aplikasi Berbasis Webterhadap suatu risiko keamanan.
2. Prosedur ini tidak secara rinci mencakup risiko keamanan terhadap Aplikasi Berbasis Webyang berkaitan dengan manajemen atau operasional yang mungkin terjadi, sehingga perlu dilakukan kajian lanjutan yang diantaranya meliputi: pengujian terhadap manusia (people), kebijakan (policy), dan proses (process) di dalam organisasi.

C. REFERENSI:

1. OWASP Testing Guide version 4.0
2. OWASP Top 10 - 2017: The Ten Most Critical Web Application Security Risks

D. DEFINISI/ SINGKATAN:

1. TIK (Teknologi Informasi dan Komunikasi) adalah istilah dalam sistem komputer yang meliputi perangkat keras (hardware), perangkat lunak (software), jaringan data dan suara, aplikasi sistem informasi dan database, dan lain sebagainya.
2. Pengguna (User) adalah akun yang digunakan untuk mengakses suatu aplikasi atau sistem TIK.
3. Hak akses adalah hak akses suatu akun Pengguna terhadap suatu aplikasi atau sistem TIK (misal: read, write, delete, upload, download, dll).
4. Vulnerability (Kerentanan) adalah suatu kelemahan berkaitan dengan keamanan sistem komputer (Aplikasi Berbasis Web), yang dapat dieksploitasi oleh suatu sumber ancaman (threat source), untuk melakukan suatu tindakan yang tidak sah (unauthorized action) dalam sistem komputer (Aplikasi Berbasis Web) tersebut.

5. Exploit (Eksploitasi) adalah cara di mana suatu kerentanan dapat dimanfaatkan untuk melakukan aktivitas yang berbahaya (malicious activity) oleh peretas. Eksploitasi adalah langkah selanjutnya dari si penyerang setelah menemukan sebuah kerentanan dalam sistem komputer (Aplikasi Berbasis Web).
6. Vulnerability Scanning adalah pemindaian terhadap suatu sistem komputer berbasis jaringan (misal: Aplikasi Berbasis Web) dengan menggunakan suatu program komputer yang dirancang untuk mencari berbagai kelemahan atau kerentanan yang terdapat dalam sistem komputer tersebut.
7. Penetration Testing adalah simulasi serangan siber yang secara resmi dilakukan pada suatu sistem komputer (Aplikasi Berbasis Web) untuk mengevaluasi keamanan sistemnya. Pengujian ini dilakukan dengan mengeksploitasi kelemahan atau kerentanan sistem komputer yang teridentifikasi, khususnya terhadap potensi pihak yang tidak berwenang untuk bisa mendapatkan akses ke fitur dan data sistem komputer.

E. PENANGGUNGJAWAB:

1. Penguji Keamanan Aplikasi Berbasis Web adalah unit kerja dalam organisasi atau pihak ketiga yang bertindak sebagai pelaksana atas kegiatan pengujian keamanan pada sistem Aplikasi Berbasis Web.
2. Penanggungjawab Pengujian adalah tim yang dibentuk dan bertanggungjawab atas kegiatan pengujian keamanan pada sistem Aplikasi Berbasis Web, yang umumnya terdiri dari:
    a. Fungsi Pengaturan Pengguna dan Hak Akses TIK bertindak sebagai Penanggungjawab atas pengaturan pengguna dan pengendalian hak akses pada sistem aplikasi dan database.
    b. Fungsi Pengembang Aplikasi TIK bertindak sebagai Penanggungjawab atas pengembangan dan pemeliharaan sistem aplikasi dan database.
    c. Fungsi Pengembangan Infrastruktur dan Jaringan TIK bertindak sebagai Penanggungjawab atas pengembangan dan pemeliharaan perangkat keras, LAN, WAN, jaringan internet, administrasi database dan administrasi sistem operasi server.
    d. Fungsi Pengendali Keamanan TIK bertindak sebagai pelaksana pengendalian keamanan sistem infrastruktur dan sistem aplikasi.
    e. Fungsi Operasional TIK bertindak sebagai pelaksana operasional harian atas sistem infrastruktur dan sistem aplikasi.

F. LANGKAH – LANGKAH:

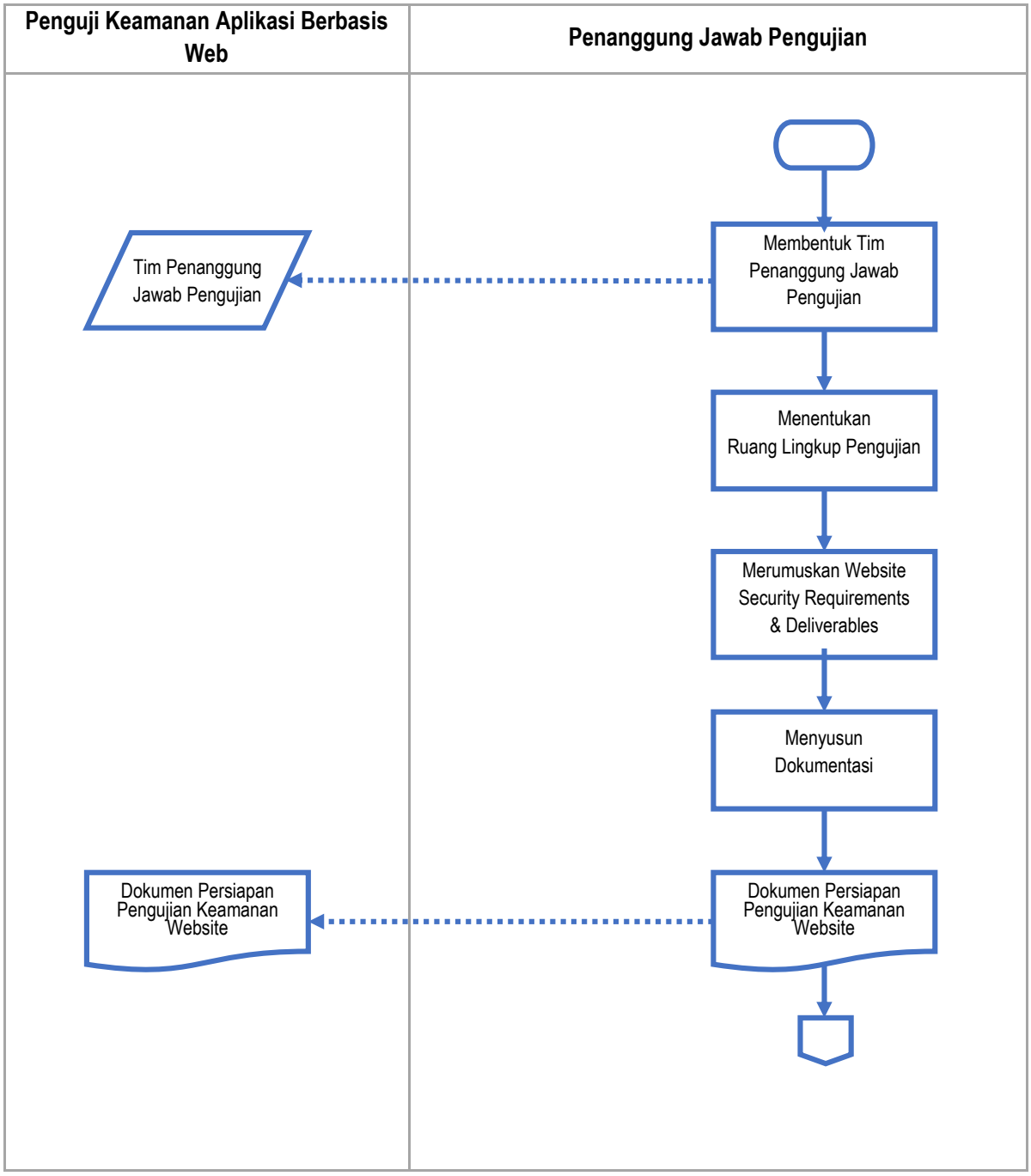
a. Aktivitas Persiapan

Tabel 4.6 Langkah Persiapan Prosedur Pengujian Keamanan Aplikasi Berbasis Web

LANGKAH	AKTIVITAS	AKTOR
1	Membentuk Tim Penanggungjawab Pengujian yang akan terlibat dalam proses pengujian keamanan Aplikasi Berbasis Web, berikut para personilnya yang umumnya berasal dari fungsi antara lain: • Fungsi Pengaturan Pengguna dan Hak Akses TIK • Fungsi Pengembang Aplikasi TIK • Fungsi Pengembangan Infrastruktur dan Jaringan TIK • Fungsi Pengendali Keamanan TIK • Fungsi Operasional TIK	Penanggungjawab Pengujian
1.1	Menentukan metode koordinasi dan komunikasi antar personil dalam Tim Penanggungjawab Pengujian, dan komunikasi dengan pihak lain yang terlibat dalam pengujian	Penanggungjawab Pengujian
1.2	Menyiapkan berbagai sumber daya yang mungkin akan dipergunakan oleh Tim Penanggungjawab Pengujian dalam proses pengujian Aplikasi Berbasis Web	Penanggungjawab Pengujian
2	Menentukan ruang lingkup pekerjaan pengujian keamanan Aplikasi Berbasis Web	Penanggungjawab Pengujian
3	Merumuskan persyaratan keamanan Aplikasi Berbasis Web (security requirements) yang ingin dicapai dan menentukan hasil keluaran (deliverables) pengujian keamanan Aplikasi Berbasis Web	Penanggungjawab Pengujian
4	Menyusun dokumentasi terkait persiapan pengujian Aplikasi Berbasis Websebagai acuan pelaksanaan kegiatan bagi Tim Penanggungjawab Pengujian dan pelaksana Penguji Keamanan Aplikasi Berbasis Web	Penanggungjawab Pengujian

b. Diagram Alir Aktivitas Persiapan

Gambar 4.7 Diagram Alir Penanganan Insiden Phising



c. Aktivitas Pra-Pengujian

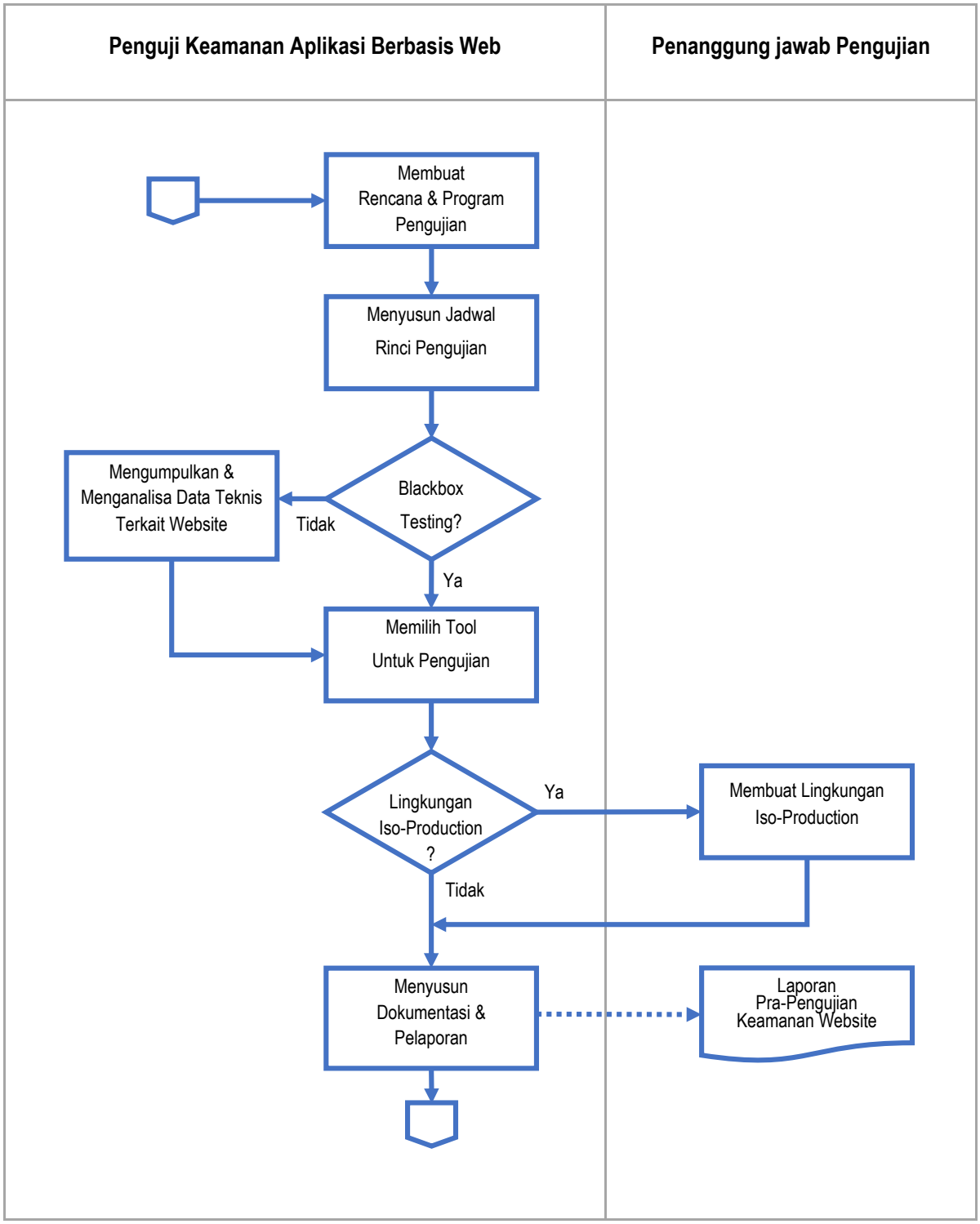
Tabel 4.7 Langkah Pra-Pengujuan Keamanan Aplikasi Berbasis Web

LANGKAH	AKTIVITAS	AKTOR
1	Membuat rencana dan program pengujian, menentukan strategi pelaksanaan berdasarkan ruang lingkup dan persyaratan keamanan. Rincian item-item yang diuji mengacu pada dokumen *"Standar Pengujian Kerentanan Aplikasi Web, BSSN – 2019"*, diantaranya berkaitan dengan komponen penting Aplikasi Berbasis Web berikut: server, aplikasi web, kode statis, middleware, dan database	Penguji Keamanan Aplikasi Berbasis Web
2	Membuat rincian jadwal pengujian dan alokasi sumber daya yang diperlukan dalam kegiatan pengujian keamanan Aplikasi Berbasis Web	Penguji Keamanan Aplikasi Berbasis Web
3	Jika pengujian menggunakan metode whitebox/ greybox, maka: • Mengumpulkan dan mengAnalisis informasi terkait Aplikasi Berbasis Webyang akan diuji keamanannya, serta bila dimungkinkan juga data-data teknis pendukung, seperti infrastruktur jaringan dan perangkat keamanan terkait lainnya	Penguji Keamanan Aplikasi Berbasis Web
4	Menentukan jenis tool pengujian yang akan dipergunakan sesuai dengan rencana dan program pengujian, serta strategi pelaksanaan pengujian	Penguji Keamanan Aplikasi Berbasis Web
5	Jika sistem Aplikasi Berbasis Webyang akan diuji dianggap kritikal dan berisiko untuk dilakukan pengujian secara langsung agar pengujian keamanan Aplikasi Berbasis Webbisa dilakukan tanpa mengganggu kinerja sistem yang live:	Penanggung jawab Pengujian

	• Membuat web server backup di lingkungan "iso-production" yang benar-benar identik dengan sistem di lingkungan "production".	
6	Menyusun dokumentasi dan pelaporan terkait rencana dan program pengujian sebagai acuan pelaksanaan kegiatan pengujian keamanan Aplikasi Berbasis Web. Menyerahkan dokumen laporan pra-pengujian keamanan Aplikasi Berbasis Web kepada Penanggungjawab Pengujian	Penguji Keamanan Aplikasi Berbasis Web

d. Diagram Alir Aktivitas Pra-Pengujian

Gambar 4.8 Diagram Alir Pra-Pengujian



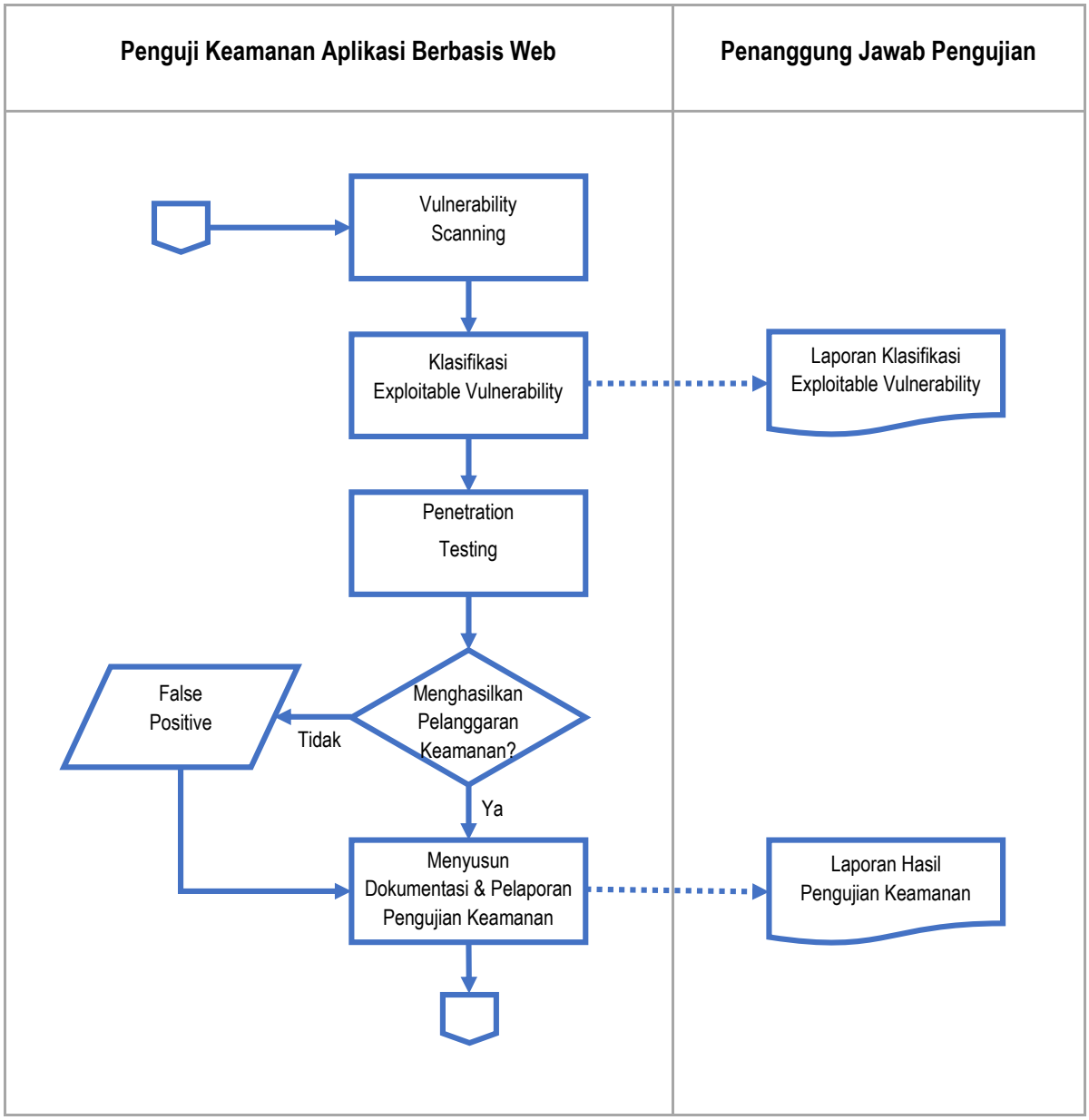
e. Aktivitas Pengujian

Tabel 4.8 Langkah Pengujian Keamanan Aplikasi Berbasis Web

LANGKAH	AKTIVITAS	AKTOR
1	Melakukan vulnerability scanning terhadap sistem Aplikasi Berbasis Websesuai dengan rencana dan program pengujian	Penguji Keamanan Aplikasi Berbasis Web
2	Melakukan klasifikasi terhadap hasil vulnerability scanning, terutama yang berkaitan dengan exploitable vulnerabity	Penguji Keamanan Aplikasi Berbasis Web
3	Menyusun prioritas pengujian lanjutan berdasarkan potensi risiko keamanan aplikasi web, misalnya berdasarkan OWASP Top 10	Penguji Keamanan Aplikasi Berbasis Web
4	Melakukan pengujian lanjutan, penetration testing berdasarkan klasifikasi dan prioritas yang ditetapkan	Penguji Keamanan Aplikasi Berbasis Web
5	Jika penetration testing untuk suatu vulnerability memungkinkan untuk menghasilkan pelanggaran keamanan, maka: • Menyusun tingkat vulnerability berdasarkan risiko keamanan dan kemungkinan untuk menghasilkan pelanggaran keamanan	Penguji Keamanan Aplikasi Berbasis Web
6	Membuat dokumentasi dan laporan hasil pengujian keamanan Aplikasi Berbasis Web, diantaranya: • Meliputi vulnerability yang ditemukan dan bagaimana vulnerability tersebut dapat dieksploitasi oleh penyerang, serta rekomendasi untuk memperbaiki. • Dokumentasi tersebut perlu ditulis secara rinci dan sistematis sebagai bahan untuk ditindaklanjuti oleh PENANGGUNGJAWAB Pengujian	Penguji Keamanan Aplikasi Berbasis Web

f. Diagram Alir Aktivitas Pengujian

Gambar 4.9 Diagram Alir Aktivitas Pengujian



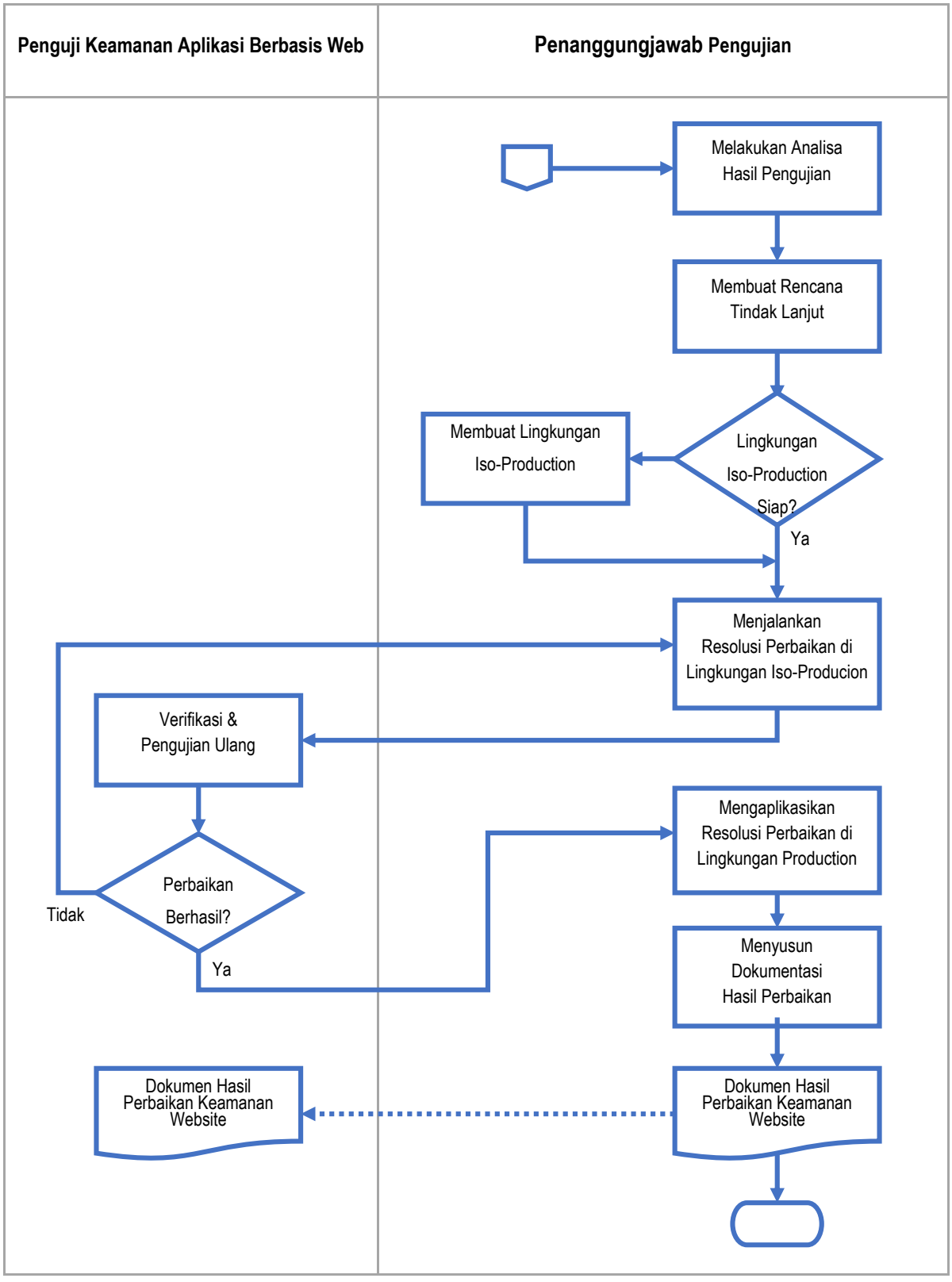
g. Aktivitas Tindak Lanjut Hasil Pengujian

Tabel 4.9 Tindak Lanjut Hasil Pengujian Keamanan Aplikasi Berbasis Web

LANGKAH	AKTIVITAS	AKTOR
1	Menganalisis hasil pengujian untuk menilai tingkat urgensi terhadap risiko keamanan Aplikasi Berbasis Webdan menentukan prioritas penanganan serta tindak lanjutnya	Penanggungjawab Pengujian
2	Menyusun rencana tindak lanjut dan prioritasnya terkait resolusi perbaikan dari kerentanan-kerentanan Aplikasi Berbasis Webyang didapatkan dari hasil pelaksanaan pengujian	Penanggungjawab Pengujian
3	Jika sebelumnya pengujian keamanan Aplikasi Berbasis Webdilakukan langsung pada sistem yang live, maka: • Membuat web server backup di lingkungan "iso-production" yang benar-benar identik dengan sistem yang live • Sedapat mungkin resolusi perbaikan dilakukan di lingkungan "iso-production" sebelum menerapkannya di sistem yang live	Penanggungjawab Pengujian
4	Menjalankan resolusi perbaikan di lingkungan "iso-production" sesuai dengan rencana tindak lanjut yang telah dibuat	Penanggungjawab Pengujian
5	Melakukan verifikasi dan pengujian ulang setelah dilakukan resolusi perbaikan terhadap kerentanan Aplikasi Berbasis Web, untuk mengetahui apakah perbaikan yang dilakukan telah berhasil menghilangkan kerentanan yang ada.	Penguji Keamanan Aplikasi Berbasis Web(didampingi oleh Penanggungjawab Pengujian)
6	Jika hasil pengujian ulang berhasil menghilangkan kerentanan yang ada, maka: • Memastikan bahwa tidak muncul kerentanan baru yang disebabkan oleh perbaikan yang dilakukan sebelumnya • Menerapkan resolusi perbaikan tersebut pada lingkungan "production" (live)	Penanggungjawab Pengujian
6	Membuat dokumentasi dan laporan hasil resolusi perbaikan yang telah selesai dilakukan, agar dapat dipakai sebagai catatan histori dan referensi	Penanggungjawab Pengujian

h. Diagram Alir Aktivitas Tindak Lanjut Hasil Pengujian

Gambar 4.10 Diagram Alir Aktivitas Tindak Lanjut Hasil Pengujian



IV.4 PROSEDUR MONITORING KEAMANAN APLIKASI BERBASIS WEB

PROSEDUR
MONITORING KEAMANAN APLIKASI BERBASIS WEB

A. TUJUAN:

1. Memberikan referensi panduan kegiatan monitoring keamanan Aplikasi Berbasis Web yang secara berkelanjutan memantau suatu sistem Aplikasi Berbasis Web dari kemungkinan adanya potensi risiko keamanan atau terjadinya suatu serangan siber.
2. Memastikan bahwa setiap petugas/tim keamanan informasi dan Aplikasi Berbasis Web, tanggap dalam melakukan langkah untuk mengurangi risiko keamanan yang terdeteksi dalam kegiata monitoring keamanan suatu sistem Aplikasi Berbasis Web.
3. Memastikan Aplikasi Berbasis Web beserta datanya aman dari kemungkinan berbagai tindakan yang tidak sah (unauthorized action).
4. Memastikan kegiatan monitoring keamanan Aplikasi Berbasis Web berjalan dengan baik dan efektif.
5. Memastikan adanya pencatatan semua kejadian dalam aktivitas monitoring keamanan Aplikasi Berbasis Web.

B. RUANG LINGKUP:

Prosedur ini mencakup langkah-langkah monitoring keamanan suatu sistem Aplikasi Berbasis Web atas risiko keamanan dengan adanya potensi masalah keamanan ataupun serangan siber yang terjadi.

C. REFERENSI:

1. OWASP Testing Guide version 4.0
2. OWASP Top 10 - 2017: The Ten Most Critical Web Application Security Risks

D. DEFINISI/ SINGKATAN:

1. TIK (Teknologi Informasi dan Komunikasi) adalah istilah dalam sistem komputer yang meliputi perangkat keras (hardware), perangkat lunak (software), jaringan data dan suara, aplikasi sistem informasi dan database, dan lain sebagainya.
2. Pengguna (User) adalah akun yang digunakan untuk mengakses suatu aplikasi atau sistem TIK.
3. Hak akses adalah hak akses suatu akun Pengguna terhadap suatu aplikasi atau sistem TIK (misal: read, write, delete, upload, download, dll).
4. *Vulnerability* (Kerentanan) adalah suatu kelemahan berkaitan dengan keamanan sistem komputer (Aplikasi Berbasis Web), yang dapat dieksploitasi oleh suatu sumber ancaman (threat source), untuk melakukan suatu tindakan yang tidak sah (unauthorized action) dalam sistem komputer (Aplikasi Berbasis Web) tersebut.
5. *Exploit* (Eksploitasi) adalah cara di mana suatu kerentanan dapat dimanfaatkan untuk melakukan aktivitas yang berbahaya (malicious activity) oleh peretas. Eksploitasi adalah langkah selanjutnya dari si penyerang setelah menemukan sebuah kerentanan dalam sistem komputer (Aplikasi Berbasis Web).

6. *Vulnerability Scanning* adalah pemindaian terhadap suatu sistem komputer berbasis jaringan (misal: Aplikasi Berbasis Web) dengan menggunakan suatu program komputer yang dirancang untuk mencari berbagai kelemahan atau kerentanan yang terdapat dalam sistem komputer tersebut.
7. *Penetration Testing* adalah simulasi serangan siber yang secara resmi dilakukan pada suatu sistem komputer (Aplikasi Berbasis Web) untuk mengevaluasi keamanan sistemnya. Pengujian ini dilakukan dengan mengeksploitasi kelemahan atau kerentanan sistem komputer yang teridentifikasi, khususnya terhadap potensi pihak yang tidak berwenang untuk bisa mendapatkan akses ke fitur dan data sistem komputer.

B. PENANGGUNGJAWAB:

1. Penanggungjawab Monitoring Keamanan Aplikasi Berbasis Web, adalah unit kerja dalam organisasi atau pihak ketiga yang bertindak sebagai Penanggungjawab atas pelaksanaan kegiatan monitoring keamanan pada sistem Aplikasi Berbasis Web.
2. Tim Teknis adalah Penanggungjawab atas semua hal teknis yang berkaitan dengan sistem Aplikasi Berbasis Web, yang umumnya terdiri dari:
    a. Fungsi Pengaturan Pengguna dan Hak Akses TIK bertindak sebagai Penanggungjawab atas pengaturan pengguna dan pengendalian hak akses pada sistem aplikasi dan database.
    b. Fungsi Pengembang Aplikasi TIK bertindak sebagai Penanggungjawab atas pengembangan dan pemeliharaan sistem aplikasi dan database.
    c. Fungsi Pengembangan Infrastruktur dan Jaringan TIK bertindak sebagai Penanggungjawab atas pengembangan dan pemeliharaan perangkat keras, LAN, WAN, jaringan internet, administrasi database dan administrasi sistem operasi server.
    d. Fungsi Pengendali Keamanan TIK bertindak sebagai pelaksana pengendalian keamanan sistem infrastruktur dan sistem aplikasi.
    e. Fungsi Operasional TIK bertindak sebagai pelaksana operasional harian atas sistem infrastruktur dan sistem aplikasi.

C. LANGKAH – LANGKAH:

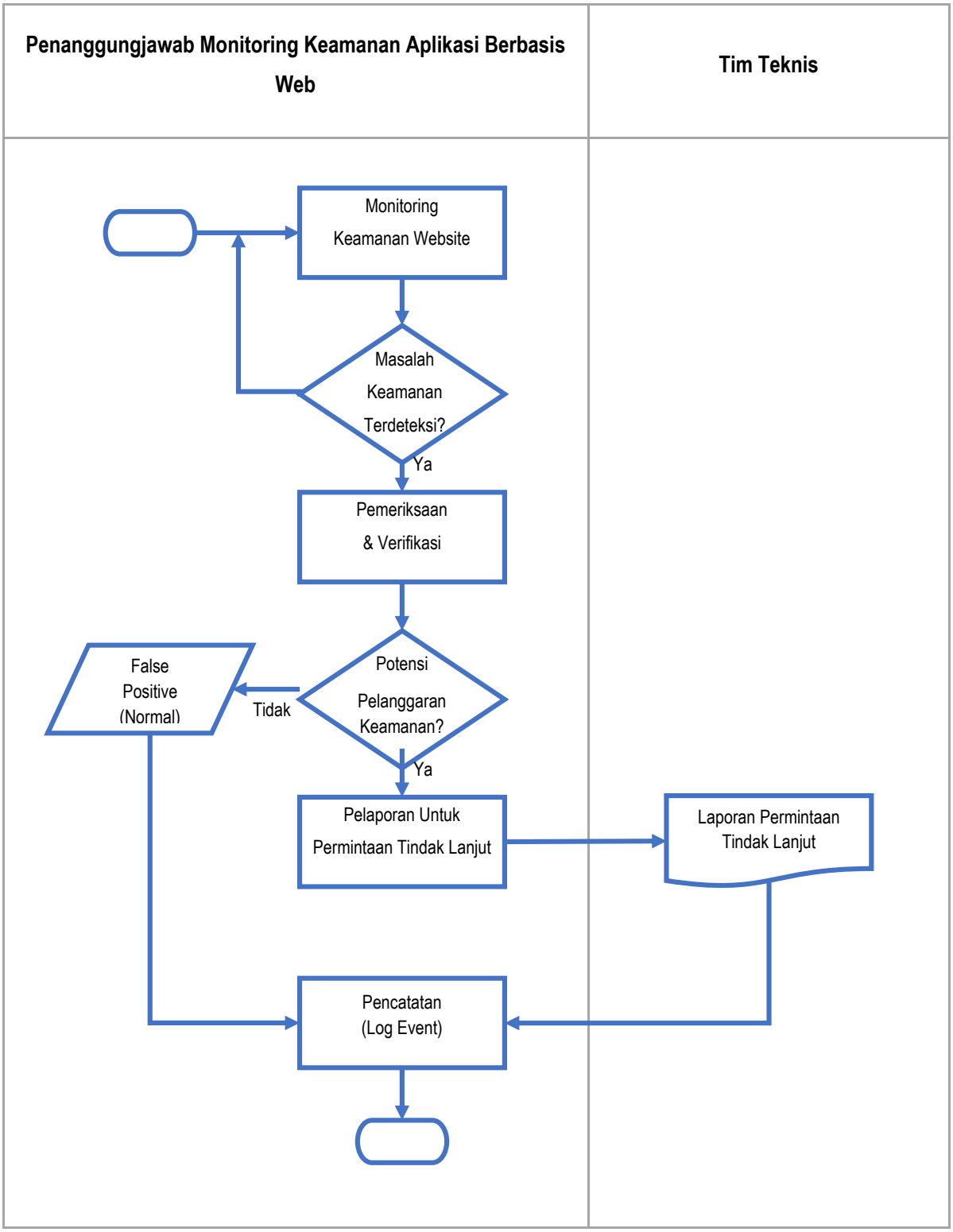
a. Aktivitas Monitoring Keamanan Aplikasi Berbasis Web

Tabel 4.10 Aktivitas Monitoring Keamanan Aplikasi Berbasis Web

LANGKAH	AKTIVITAS	AKTOR
1	Melaksanakan kegiatan monitoring keamanan terhadap suatu Aplikasi Berbasis Websecara berkesinambungan. Berbagai aktivitas monitoring tersebut dapat mencakup, antara lain: • Monitoring kerentanan (vulnerability) • Monitoring sertifikat SSL/TLS • Monitoring lama hidup server/ service (up-time) • Monitoring masa berlaku domain • Monitoring Aplikasi Berbasis Webdari daftar blacklist • Monitoring DNS terkait Aplikasi Berbasis Web	Penanggungjawab Monitoring Keamanan Aplikasi Berbasis Web
2	Jika dalam kegiatan monitoring terdeteksi suatu masalah keamanan, maka: • Melakukan pemeriksaan dan validasi terhadap masalah keamanan yang terdeteksi untuk mem-validasi masalah keamanan tersebut, dan apakah berpotensi untuk terjadinya suatu pelanggaran keamanan terhadap Aplikasi Berbasis Webyang dimonitor	Penanggungjawab Monitoring Keamanan Aplikasi Berbasis Web
3	Melakukan penilaian tingkat potensi masalah keamanan yang terdeteksi dalam salah satu tingkatan (level) sebagai acuan prioritas penanganan, diantaranya: • Normal, jika hasil pemeriksaaan dan validasi menunjukkan (false positive) sehingga dapat diabaikan. • Rendah (low) • Medium (med) • Tinggi (high) • Parah (severe)	Penanggungjawab Monitoring Keamanan Aplikasi Berbasis Web
4	Untuk tingkatan low, med, high, dan severe, melakukan pelaporan kepada “Tim Teknis” terkait untuk permintaan tindak lanjut atas masalah keamanan Aplikasi Berbasis Webtersebut	Penanggungjawab Monitoring Keamanan Aplikasi Berbasis Web
5	Melakukan pencatatan kejadian (log event) secara terstruktur dan rinci atas semua masalah keamanan Aplikasi Berbasis Webyang terdeteksi	Penanggungjawab Monitoring Keamanan Aplikasi Berbasis Web

b. Diagram Alir Aktivitas Monitoring Keamanan Aplikasi Berbasis Web

Gambar 4.11 Diagram Alir Aktivitas Monitoring Keamanan Aplikasi Berbasis Web



IV.5 PROSEDUR MANAJEMEN KERENTANAN APLIKASI BERBASIS WEB

PROSEDUR
MANAJEMEN KERENTANAN KEAMANAN APLIKASI BERBASIS WEB

A. TUJUAN:

1. Memberikan referensi panduan bagi aktivitas manajemen kerentanan Aplikasi Berbasis Web yang secara bertahap dengan skala prioritas dan berkelanjutan, untuk mengurangi ancaman keamanan yang mungkin terjadi akibat adanya kerentanan dalam sebuah Aplikasi Berbasis Web.
2. Memastikan bahwa Penanggungjawab manajemen kerentanan Aplikasi Berbasis Web, tanggap dalam melakukan langkah-langkah untuk meminimalkan risiko keamanan atas suatu sistem Aplikasi Berbasis Web.
3. Memastikan Aplikasi Berbasis Web beserta datanya menjadi lebih aman dari kemungkinan berbagai tindakan yang tidak sah (unauthorized action).
4. Memastikan kegiatan manajemen kerentanan Aplikasi Berbasis Web berjalan dengan baik dan efektif.
5. Memastikan adanya dokumentasi dari setiap kegiatan dalam aktivitas manajemen kerentanan Aplikasi Berbasis Web.

B. RUANG LINGKUP:

Prosedur ini mencakup langkah-langkah secara umum untuk meminimalkan ancaman keamanan yang mungkin terjadi akibat terdapatnya kerentanan-kerentanan dalam suatu sistem Aplikasi Berbasis Web, melalui kegiatan manajemen kerentanan Aplikasi Berbasis Web.

C. REFERENSI:

1. OWASP Testing Guide version 4.0
2. OWASP Top 10 - 2017: The Ten Most Critical Web Application Security Risks

D. DEFINISI/ SINGKATAN:

1. TIK (Teknologi Informasi dan Komunikasi) adalah istilah dalam sistem komputer yang meliputi perangkat keras (hardware), perangkat lunak (software), jaringan data dan suara, aplikasi sistem informasi dan database, dan lain sebagainya.
2. Pengguna (User) adalah akun yang digunakan untuk mengakses suatu aplikasi atau sistem TIK.
3. Hak akses adalah hak akses suatu akun Pengguna terhadap suatu aplikasi atau sistem TIK (misal: read, write, delete, upload, download, dll).
4. Vulnerability (Kerentanan) adalah suatu kelemahan berkaitan dengan keamanan sistem komputer (Aplikasi Berbasis Web), yang dapat dieksploitasi oleh suatu sumber ancaman (threat source), untuk melakukan suatu tindakan yang tidak sah (unauthorized action) dalam sistem komputer (Aplikasi Berbasis Web) tersebut.
5. Exploit (Eksploitasi) adalah cara di mana suatu kerentanan dapat dimanfaatkan untuk melakukan aktivitas yang berbahaya (malicious activity) oleh peretas. Eksploitasi adalah langkah selanjutnya dari si penyerang setelah menemukan sebuah kerentanan dalam sistem komputer (Aplikasi Berbasis Web).

6. Vulnerability Scanning adalah pemindaian terhadap suatu sistem komputer berbasis jaringan (misal: Aplikasi Berbasis Web) dengan menggunakan suatu program komputer yang dirancang untuk mencari berbagai kelemahan atau kerentanan yang terdapat dalam sistem komputer tersebut.
7. Penetration Testing adalah simulasi serangan siber yang secara resmi dilakukan pada suatu sistem komputer (Aplikasi Berbasis Web) untuk mengevaluasi keamanan sistemnya. Pengujian ini dilakukan dengan mengeksploitasi kelemahan atau kerentanan sistem komputer yang teridentifikasi, khususnya terhadap potensi pihak yang tidak berwenang untuk bisa mendapatkan akses ke fitur dan data sistem komputer.

E. PENANGGUNGJAWAB:

1. Penanggungjawab Manajemen Kerentanan Aplikasi Berbasis Web, adalah unit kerja dalam organisasi atau pihak ketiga yang bertindak sebagai Penanggung jawab atas pelaksanaan kegiatan manajemen kerentanan pada sistem Aplikasi Berbasis Web.
2. Tim Teknis adalah Penanggung jawab atas semua hal teknis yang berkaitan dengan sistem Aplikasi Berbasis Web, yang umumnya terdiri dari:
    a. Fungsi Pengaturan Pengguna dan Hak Akses TIK bertindak sebagai Penanggungjawab atas pengaturan pengguna dan pengendalian hak akses pada sistem aplikasi dan database.
    b. Fungsi Pengembang Aplikasi TIK bertindak sebagai Penanggungjawab atas pengembangan dan pemeliharaan sistem aplikasi dan database.
    c. Fungsi Pengembangan Infrastruktur dan Jaringan TIK bertindak sebagai Penanggungjawab atas pengembangan dan pemeliharaan perangkat keras, LAN, WAN, jaringan internet, administrasi database dan administrasi sistem operasi server.
    d. Fungsi Pengendali Keamanan TIK bertindak sebagai pelaksana pengendalian keamanan sistem infrastruktur dan sistem aplikasi.
    e. Fungsi Operasional TIK bertindak sebagai pelaksana operasional harian atas sistem infrastruktur dan sistem aplikasi.

F. LANGKAH – LANGKAH:

a. Aktivitas Manajemen Kerentanan Aplikasi Berbasis Web

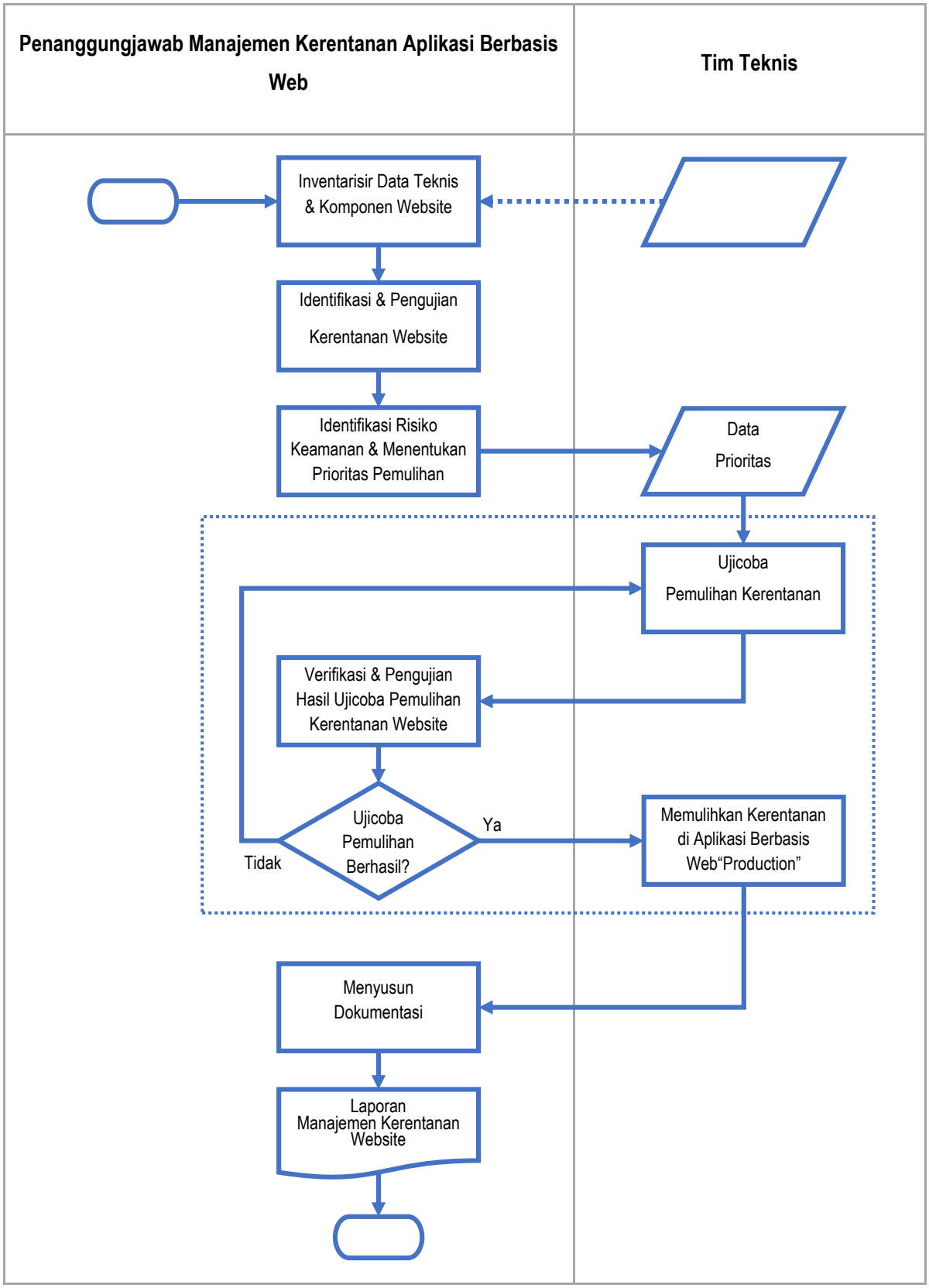
Tabel 4.11 Aktivitas manajemen Kerentanan Aplikasi Berbasis Web

LANGKAH	AKTIVITAS	AKTOR
1	Melaksanakan inventarisir terhadap data-data teknis dalam Aplikasi Berbasis Web, meliputi: alamat IP, DNS, sistem operasi, alamat MAC, port, layanan, software, proses, hardware, event log, dan lainnya	Penanggungjawab Manajemen Kerentanan Aplikasi Berbasis Web
2	Melakukan identifikasi dan pengujian kerentanan Aplikasi Berbasis Web, baik secara manual maupun dengan menggunakan tool. Rincian item-item yang diuji secara umum adalah mengacu pada dokumen *"Standar Pengujian Kerentanan Aplikasi Web, BSSN – 2019"*, diantaranya berkaitan dengan komponen penting Aplikasi Berbasis Webberikut: server, aplikasi web, kode statis, middleware, dan database	Penanggungjawab Manajemen Kerentanan Aplikasi Berbasis Web
3	Melakukan identifikasi risiko berdasarkan potensi ancaman dan risiko keamanan aplikasi web (misalnya: berdasarkan OWASP Top 10), serta menentukan prioritas pemulihan kerentanan sesuai tingkat risikonya. Selanjutnya informasi dan data kerentanan berikut prioritas pemulihannya diserahkan pada "Tim Teknis" terkait untuk ditindaklanjuti.	Penanggungjawab Manajemen Kerentanan Aplikasi Berbasis Web
4	Melakukan langkah pemulihan kerentanan Aplikasi Berbasis Web	
4.1	Melakukan ujicoba pemulihan kerentanan dengan melakukan berbagai langkah untuk bisa menutup suatu celah keamanan.	Tim Teknis

	4.2	Melakukan verifikasi dan melakukan pengujian atas hasil ujicoba pemulihan kerentanan Aplikasi Berbasis Webyang telah dilakukan oleh Tim Teknis	Penanggungjawab Manajemen Kerentanan Aplikasi Berbasis Web
	4.3	Jika ujicoba pemulihan kerentanan Aplikasi Berbasis Webdinyatakan berhasil untuk menutup celah keamanan, maka langkah pemulihan kerentanan diterapkan di Aplikasi Berbasis Web“Production”.	Tim Teknis
5		Melakukan pendokumentasian untuk setiap kegiatan yang berkaitan dengan aktivitas manajemen kerentanan Aplikasi Berbasis Web, sebagai bahan untuk analisis kedepan terkait keamanan Aplikasi Berbasis Webdan knowledge repository. Secara periodik membuat “Laporan Manajemen Kerentanan Aplikasi Berbasis Web”.	Penanggungjawab Manajemen Kerentanan Aplikasi Berbasis Web

b. Diagram Alir Aktivitas Manajemen Kerentanan Aplikasi Berbasis Web

Gambar 4.12 Diagram Alir Aktivitas Manajemen Kerentanan Aplikasi Berbasis Web



IV.6 PROSEDUR MANAJEMEN PIHAK KETIGA

PROSEDUR
MANAJEMEN PIHAK KETIGA

A. TUJUAN:

1. Memberikan referensi dan panduan umum bagi aktivitas pengelolaan pihak ketiga dalam Aplikasi Berbasis Web, agar dapat meminimalkan risiko keamanan yang mungkin terjadi dalam sebuah Aplikasi Berbasis Web.
2. Memastikan bahwa setiap petugas/tim yang terlibat dalam kegiatan ini, tanggap dalam melakukan langkah-langkah untuk meminimalkan risiko keamanan atas suatu sistem Aplikasi Berbasis Web.
3. Memastikan Aplikasi Berbasis We beserta datanya menjadi lebih aman dari kemungkinan berbagai tindakan yang tidak sah (unauthorized action).
4. Memastikan kegiatan pengelolaan pihak ketiga dalam Aplikasi Berbasis Web berjalan dengan baik dan efektif.
5. Memastikan adanya dokumentasi dari setiap kegiatan dalam aktivitas pengelolaan pihak ketiga dalam Aplikasi Berbasis Web.

B. RUANG LINGKUP:

Prosedur ini mencakup pengelolaan Aplikasi Berbasis Web oleh pihak ketiga yang lebih dititikberatkan pada aspek keamanan Aplikasi Berbasis Webdan memberikan batasan umum terhadap tanggungjawab terhadap keamanan Aplikasi Berbasis Webdari pihak-pihak yang terlibat.

C. REFERENSI:

1. OWASP Testing Guide version 4.0
2. Kebijakan Keamanan Informasi
3. OWASP Top 10 - 2017: The Ten Most Critical Web Application Security Risks

D. DEFINISI/ SINGKATAN:

1. Web hosting adalah layanan (service) yang memungkinkan suatu organisasi dan individu untuk menampilkan Aplikasi Berbasis Web atau halaman web, sehingga bisa diakses oleh pengguna dari Internet.
2. TIK (Teknologi Informasi dan Komunikasi) adalah istilah dalam sistem komputer yang meliputi perangkat keras (hardware), perangkat lunak (software), jaringan data dan suara, aplikasi sistem informasi dan database, dan lain sebagainya.
3. Pengguna (User) adalah akun yang digunakan untuk mengakses suatu aplikasi atau sistem TIK.
4. Hak akses adalah hak akses suatu akun Pengguna terhadap suatu aplikasi atau sistem TIK (misal: read, write, delete, upload, download, dll).
5. Vulnerability (Kerentanan) adalah suatu kelemahan berkaitan dengan keamanan sistem komputer (Aplikasi Berbasis Web), yang dapat dieksploitasi oleh suatu sumber ancaman (threat source), untuk melakukan suatu tindakan yang tidak sah (unauthorized action) dalam sistem komputer (Aplikasi Berbasis Web) tersebut.

6. Exploit (Eksploitasi) adalah cara di mana suatu kerentanan dapat dimanfaatkan untuk melakukan aktivitas yang berbahaya (malicious activity) oleh peretas. Eksploitasi adalah langkah selanjutnya dari si penyerang setelah menemukan sebuah kerentanan dalam sistem komputer (Aplikasi Berbasis Web).
7. Vulnerability Scanning adalah pemindaian terhadap suatu sistem komputer berbasis jaringan (misal: Aplikasi Berbasis Web) dengan menggunakan suatu program komputer yang dirancang untuk mencari berbagai kelemahan atau kerentanan yang terdapat dalam sistem komputer tersebut.
8. Penetration Testing adalah simulasi serangan siber yang secara resmi dilakukan pada suatu sistem komputer (Aplikasi Berbasis Web) untuk mengevaluasi keamanan sistemnya. Pengujian ini dilakukan dengan mengeksploitasi kelemahan atau kerentanan sistem komputer yang teridentifikasi, khususnya terhadap potensi pihak yang tidak berwenang untuk bisa mendapatkan akses ke fitur dan data sistem komputer.

E. PENANGGUNGJAWAB:

1. Koordinator Keamanan Aplikasi Berbasis Web, adalah individu dalam organisasi yang bertindak sebagai Penanggungjawab atas koordinator keamanan pada sistem Aplikasi Berbasis Web di organisasi.
2. Pengelola Aplikasi Berbasis Web Pihak Ketiga, adalah pihak lain di luar Instansi yang bertindak sebagai Penanggungjawab atas pelaksanaan kegiatan web hosting pada sistem Aplikasi Berbasis Web.
3. Pengembang Aplikasi Web, adalah Penanggungjawab atas pengembangan, pemeliharaan dan semua hal teknis yang berkaitan dengan software aplikasi web dan database.

F. LANGKAH – LANGKAH:

a. Aktivitas Persiapan

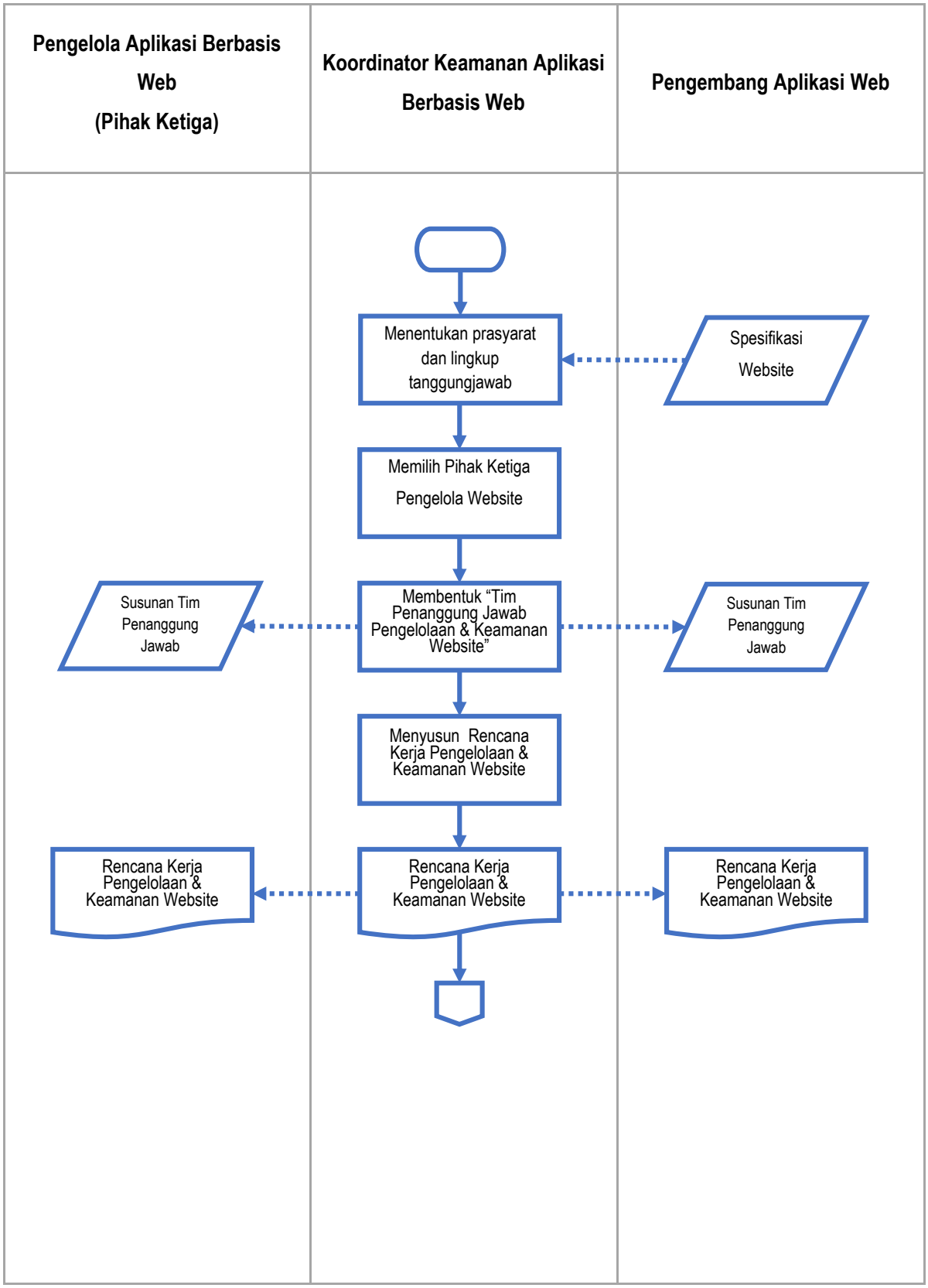
Tabel 4.12 Aktivitas Persiapan Manajemen Pihak Ketiga

LANGKAH	AKTIVITAS	AKTOR
1	Menentukan prasyarat keamanan untuk Aplikasi Berbasis Webyang akan dikelola dan lingkup tanggungjawab dalam keamanan pengelolaan Aplikasi Berbasis Web • Berdasarkan jenis aplikasi web, tingkat kritikalitas dan sensitivitas data, yang mengacu pada "Kebijakan Keamanan Informasi ". • Berdasarkan skema web hosting yang akan digunakan. ▪ Shared Hosting ▪ Managed Hosting (VPS, Dedicated Server, Cloud)	Koordinator Keamanan Aplikasi Berbasis Web
2	Memilih Pihak Ketiga Pengelola Aplikasi Berbasis Web, dalam hal ini adalah perusahaan penyedia layanan web hosting (web hosting provider): • Sesuai dengan skema web hosting dan yang memenuhi prasyarat keamanan Aplikasi Berbasis Web • Membuat kesepakatan layanan dan lingkup tanggungjawab terkait keamanan Aplikasi Berbasis Web	Koordinator Keamanan Aplikasi Berbasis Web
3	Membentuk dan mengkoordinasikan "Tim Penanggungjawab Pengelolaan dan Keamanan Aplikasi Berbasis Web", terdiri dari: • Pihak Instansi: ▪ Koordinator Keamanan Aplikasi Berbasis Web ▪ Pengembang Aplikasi Web • Pihak Ketiga: ▪ Pengelola Aplikasi Berbasis Web Serta menentukan metode komunikasi, koordinasi, dan pelaporan	Koordinator Keamanan Aplikasi Berbasis Web

4	Menyusun rencana kerja pengelolaan Aplikasi Berbasis Webdalam hal keamanan sesuai lingkup tanggungjawab setiap pihak. • Instansi: bertanggungjawab atas keamanan Aplikasi Berbasis Web, diantaranya yang berkaitan dengan: software aplikasi web, kode program, struktur file ownership, file permissions, middleware, database, dan sebagainya.. • Pihak Ketiga: bertanggungjawab atas keamanan Aplikasi Berbasis Web, diantaranya yang berkaitan dengan: akses fisik, hardware (fisik atau virtual), sistem operasi, infrastruktur jaringan, dan sebagainya.	Tim Penanggungjawab Pengelolaan dan Keamanan Aplikasi Berbasis Web

b. Diagram Alir Aktivitas Persiapan

Gambar 4.13 Diagram Alir Aktivitas Persiapan Manajemen Pihak Ketiga



c. Aktivitas Pelaksanaan

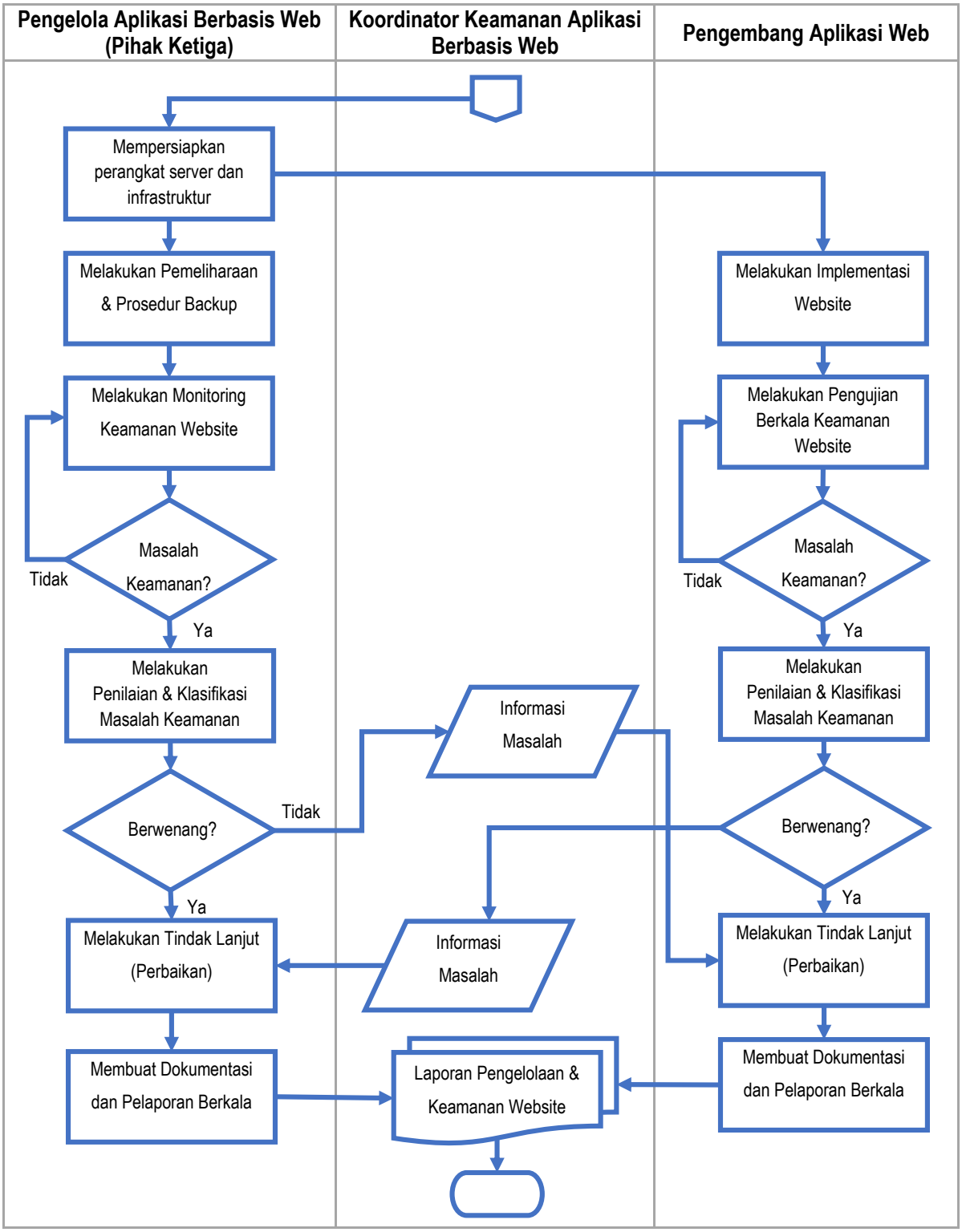
Tabel 4.13 Aktivitas Pelaksanaan Manajemen Pihak Ketiga

LANGKAH	AKTIVITAS	AKTOR
1	Mempersiapkan hardware dan infrastruktur untuk pengelolaan Aplikasi Berbasis Web. Pastikan bahwa prasyarat keamanan pengelolaan Aplikasi Berbasis Webtelah dipenuhi dan diimplementasikan.	Pengelola Aplikasi Berbasis Web (Pihak Ketiga)
2	Melakukan implementasi Aplikasi Berbasis Web pada perangkat server yang telah dipersiapkan. Pastikan bahwa semua hal teknis yang menyangkut keamanan Aplikasi Berbasis Webtelah dipenuhi dan dikerjakan.	Tim Teknis (Web Developer)
3	Melakukan pemeliharaan hardware dan infrastruktur, sekaligus menjalankan prosedur back-up secara rutin.	Pengelola Aplikasi Berbasis Web (Pihak Ketiga)
4	Melaksanakan kegiatan monitoring atau pengujian secara berkala terhadap keamanan Aplikasi Berbasis Websesuai tanggungjawabnya, namun tidak selalu terbatas pada lingkup tanggungjawab yang telah ditetapkan.	• Pengelola Aplikasi Berbasis Web (Pihak Ketiga) • Pengembang Aplikasi Web
5	Jika dalam kegiatan monitoring atau pengujian terdeteksi suatu masalah keamanan, maka melakukan pemeriksaan dan validasi terhadap masalah keamanan yang terdeteksi untuk menentukan langkah selanjutnya,	• Pengelola Aplikasi Berbasis Web (Pihak Ketiga) • Pengembang Aplikasi Web
5.1	Memilah dan mengklasifikasikan masalah keamanan tersebut terkait dengan tingkat potensi ancaman (risiko keamanan),	
5.2	Melaporkan masalah keamanan tersebut kepada “Koordinator Keamanan Aplikasi Berbasis Web”, bila masalah keamanan tersebut berada di luar wewenangnya untuk bisa langsung melakukan tindak lanjut.	

	5.3	"Koordinator Keamanan Aplikasi Berbasis Web" selanjutnya menyampaikan masalah keamanan tersebut ke pihak yang berwenang untuk ditindaklanjuti.	Koordinator Keamanan Aplikasi Berbasis Web
6		Melakukan langkah tindak lanjut (perbaikan) terhadap masalah keamanan sesuai dengan tanggungjawab dan wewenangnya.	• Pengelola Aplikasi Berbasis Web (Pihak Ketiga) • Pengembang Aplikasi Web
7		Membuat dokumentasi dan membuat laporan kegiatan pengelolaan Aplikasi Berbasis Web(dalam hal keamanan) secara periodik untuk diserahkan kepada "Koordinator Keamanan Aplikasi Berbasis Web"	• Pengelola Aplikasi Berbasis Web (Pihak Ketiga) • Pengembang Aplikasi Web

d. Diagram Alir Aktivitas Pelaksanaan

Gambar 4.14 Diagram Alir Aktivitas Pelaksanaan Manajemen Pihak Ketiga



IV.7 PROSEDUR PENGUATAN KEAMANAN WEB SERVER (HARDENING)

IV.7.1 HARDENING WEB SERVER NGINX

HARDENING WEB SERVER

NGINX

A. TUJUAN

Memberikan petunjuk/*guidance* dalam melakukan hardening pada web server NGINX

B. RUANG LINGKUP

Tahapan hardening pada web server NGINX

C. REFERENSI

1. OWASP Secure Configuration Guide Project
   https://www.owasp.org/index.php/OWASP_Secure_Configuration_Guide

D. LANGKAH - LANGKAH

1. Mengecek konfigurasi default files & port Nginx sebagai berikut:

   - **/usr/local/nginx/conf/** or **/etc/nginx/**– The nginx server configuration directory and /usr/local/nginx/conf/nginx.conf is main configuration file.
   - **/usr/local/nginx/html/** or **/var/www/html**– The default document location.
   - **/usr/local/nginx/logs/** or **/var/log/nginx** – The default log file location.
   - Nginx **HTTP default port**: TCP 80
   - Nginx **HTTPS default port**: TCP 443

   Setelah itu kita dapat menguji perubahan konfigurasi Nginx dengan perintah sebagai berikut:

   ```
   # /usr/local/nginx/sbin/nginx -t
   ```

   atau

   ```
   # nginx -t
   ```

   Outputs:

```
the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
configuration file /usr/local/nginx/conf/nginx.conf test is successful
```

Untuk perubahan konfigurasi, dengan mengetik:

```
# /usr/local/nginx/sbin/nginx -s reload
```

atau

```
# nginx -s reload
```

Atau untuk menghentikan layanan server Nginx, dengan mengetik:

```
# /usr/local/nginx/sbin/nginx -s stop
```

atau

```
# nginx -s stop
```

2. Melakukan Hardening

#1: Mengaktifkan SELinux

Security-Enhanced Linux (SELinux) is adalah fitur kernel Linux yang menyediakan mekanisme untuk mendukung keamanan kendali akses sehingga dapat menghentikan banyak serangan kedalam sistem linux.

Do Boolean Lockdown

Jalankan perintah getsebool -a dan lockdown system:

```
getsebool -a | less
getsebool -a | grep off
getsebool -a | grep on
```

Untuk mengamankan mesin, lihat pengaturan perintah getsebool ke 'on' atau ke 'off'. Atur boolean SE Linux yang benar untuk menjaga fungsionalitas dan perlindungan. Perlu dicatat bahwa SELinux menambahkan overhead 2-8% untuk instalasi RHEL atau CentOS.

#2: Meminimalisasi previllege

File halaman web ; /html /php harus melalui partisi yang terpisah. Sebagai contoh, buat partisi bernama /dev// sda5 dan mount di /nginx. Memastikan /nginx di-mount dengan izin noexec, nodev dan nosetuid. Berikut entri /etc /fstab untuk instalasi nginx:

```
LABEL=/nginx/nginx  ext3 defaults,nosuid,noexec,nodev 1 2
```

#3: Hardening Linux /etc/sysctl.conf

Kita dapat mengontrol dan mengkonfigurasi kernel linux di /etc/sysctl.conf.

```
# Avoid a smurf attack
net.ipv4.icmp_echo_ignore_broadcasts = 1
 # Turn on protection for bad icmp error messages
net.ipv4.icmp_ignore_bogus_error_responses = 1
 # Turn on syncookies for SYN flood attack protection
net.ipv4.tcp_syncookies = 1
 # Turn on and log spoofed, source routed, and redirect packets
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.default.log_martians = 1
 # No source routed packets here
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0
 # Turn on reverse path filtering
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1
 # Make sure no one can alter the routing tables
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.default.secure_redirects = 0
 # Don't act as a router
net.ipv4.ip_forward = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0
```

```
# Turn on execshild
kernel.exec-shield = 1
kernel.randomize_va_space = 1
 # Tuen IPv6
net.ipv6.conf.default.router_solicitations = 0
net.ipv6.conf.default.accept_ra_rtr_pref = 0
net.ipv6.conf.default.accept_ra_pinfo = 0
net.ipv6.conf.default.accept_ra_defrtr = 0
net.ipv6.conf.default.autoconf = 0
net.ipv6.conf.default.dad_transmits = 0
net.ipv6.conf.default.max_addresses = 1
 # Optimization for port usefor LBs
# Increase system file descriptor limit
fs.file-max = 65535
 # Allow for more PIDs (to reduce rollover problems); may break some programs 32768
kernel.pid_max = 65536
 # Increase system IP port limits
net.ipv4.ip_local_port_range = 2000 65000
 # Increase TCP max buffer size setable using setsockopt()
net.ipv4.tcp_rmem = 4096 87380 8388608
net.ipv4.tcp_wmem = 4096 87380 8388608
 # Increase Linux auto tuning TCP buffer limits
# min, default, and max number of bytes to use
# set max to at least 4MB, or higher if you use very high BDP paths
# Tcp Windows etc
net.core.rmem_max = 8388608
net.core.wmem_max = 8388608
net.core.netdev_max_backlog = 5000
net.ipv4.tcp_window_scaling = 1
```

#4: Menghapus semua module Nginx yang tidak diperlukan

Mengkonfigurasi dan menginstal nginx hanya menggunakan modul yang diperlukan. Misalnya, menonaktifkan modul SSI dan autoindex dengan mengetik:

```
# ./configure --without-http_autoindex_module --without-http_ssi_module
# make
# make install
```

Type the following command to see which modules can be turn on or off while compiling nginx server:

```
# ./configure --help | less
```

Disable modul Nginx yang tidak diperlukan

(Optional) Mengubah Nginx Version Header

Edit src/http/ngx_http_header_filter_module.c, enter:

```
# vi +48 src/http/ngx_http_header_filter_module.c
```

Keluar baris – baris sebagai berikut:

```
static char ngx_http_server_string[] = "Server: nginx" CRLF;
static char ngx_http_server_full_string[] = "Server: " NGINX_VER CRLF;
```

Mengubahnya dengan :

```
static char ngx_http_server_string[] = "Server: Ninja Web Server" CRLF;
static char ngx_http_server_full_string[] = "Server: Ninja Web Server" CRLF;
```

Save and close file tersebut. Lakukan compile di Server. Menambahkan yang berikut ini di nginx.conf untuk mematikan nomor versi nginx yang ditampilkan di semua halaman kesalahan yang dihasilkan secara otomatis:

```
server_tokens off
```

#5: Use mod_security (Hanya untuk backend Server Apache)

Mod_security menyediakan firewall level aplikasi untuk Apache. Instal mod_security untuk semua server web Apache backend. Ini akan menghentikan banyak serangan injeksi.

#6: Install SELinux Policy To Harden The Nginx Webserver

Secara default SELinux tidak akan melindungi server web nginx. Namun, dapat menginstal dan menyusun perlindungan sebagai berikut. Pertama, instal dukungan waktu kompilasi SELinux yang diperlukan:

```
# yum -y install selinux-policy-targeted selinux-policy-devel
```

Unduh kebijakan SELinux yang ditargetkan untuk meningkatkan keamanan server web nginx di server Linux

```
# cd /opt
# wget 'http://downloads.sourceforge.net/project/selinuxnginx/se-ngix_1_0_10.tar.gz?use_mirror=nchc'
```

Untar the file berikut:

```
# tar -zxvf se-ngix_1_0_10.tar.gz
```

Compile file yang sama

```
# cd se-ngix_1_0_10/nginx
# make
```

Outputnya akan keluar sebagai berikut:

```
Compiling targeted nginx module
/usr/bin/checkmodule: loading policy configuration from tmp/nginx.tmp
/usr/bin/checkmodule: policy configuration loaded
/usr/bin/checkmodule: writing binary representation (version 6) to tmp/nginx.mod
Creating targeted nginx.pp policy package
rm tmp/nginx.mod.fc tmp/nginx.mod
```

Instal modul SELinux nginx.pp yang dihasilkan:

```
# /usr/sbin/semodule -i nginx.pp
```

#7: Firewall Berbasis Iptables Terbatas

Script berikut akan memblokir semua port kecuali

- Incoming HTTP (TCP port 80) requests
- Incoming ICMP ping requests
- Outgoing ntp (port 123) requests
- Outgoing smtp (TCP port 25) requests

```
#!/bin/bash
IPT="/sbin/iptables"
#### IPS ######
# Get server public ip
SERVER_IP=$(ifconfig eth0 | grep 'inet addr:' | awk -F'inet addr:' '{ print $2}' | awk '{ print $1}')
LB1_IP="204.54.1.1"
LB2_IP="204.54.1.2"
# Do some smart logic so that we can use damm script on LB2 too
OTHER_LB=""
SERVER_IP=""
[[ "$SERVER_IP" == "$LB1_IP" ]] && OTHER_LB="$LB2_IP" || OTHER_LB="$LB1_IP"
[[ "$OTHER_LB" == "$LB2_IP" ]] && OPP_LB="$LB1_IP" || OPP_LB="$LB2_IP"
### IPs ###
PUB_SSH_ONLY="122.xx.yy.zz/29"

#### FILES #####
BLOCKED_IP_TDB=/root/.fw/blocked.ip.txt
SPOOFIP="127.0.0.0/8 192.168.0.0/16 172.16.0.0/12 10.0.0.0/8 169.254.0.0/16 0.0.0.0/8 240.0.0.0/4
255.255.255.255/32 168.254.0.0/16 224.0.0.0/4 240.0.0.0/5 248.0.0.0/5 192.0.2.0/24"
BADIPS=$( [[ -f ${BLOCKED_IP_TDB} ]] && egrep -v "^#|^$" ${BLOCKED_IP_TDB})

### Interfaces ###
PUB_IF="eth0" # public interface
LO_IF="lo" # loopback
VPN_IF="eth1" # vpn / private net
### start firewall ###
```

```
echo "Setting LB1 $(hostname) Firewall..."
 # DROP and close everything
$IPT -P INPUT DROP
$IPT -P OUTPUT DROP
$IPT -P FORWARD DROP
 # Unlimited lo access
$IPT -A INPUT -i ${LO_IF} -j ACCEPT
$IPT -A OUTPUT -o ${LO_IF} -j ACCEPT
 # Unlimited vpn / pnet access
$IPT -A INPUT -i ${VPN_IF} -j ACCEPT
$IPT -A OUTPUT -o ${VPN_IF} -j ACCEPT
 # Drop sync
$IPT -A INPUT -i ${PUB_IF} -p tcp ! --syn -m state --state NEW -j DROP
 # Drop Fragments
$IPT -A INPUT -i ${PUB_IF} -f -j DROP
 $IPT -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP
$IPT -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL ALL -j DROP


# Drop NULL packets
$IPT -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL NONE -m limit --limit 5/m --limit-burst 7 -j LOG --log-prefix "
NULL Packets "
$IPT -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL NONE -j DROP
 $IPT -A INPUT -i ${PUB_IF} -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
 # Drop XMAS
$IPT -A INPUT -i ${PUB_IF} -p tcp --tcp-flags SYN,FIN SYN,FIN -m limit --limit 5/m --limit-burst 7 -j LOG --log-
prefix " XMAS Packets "
$IPT -A INPUT -i ${PUB_IF} -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP
```

```
# Drop FIN packet scans
$IPT -A INPUT -i ${PUB_IF} -p tcp --tcp-flags FIN,ACK FIN -m limit --limit 5/m --limit-burst 7 -j LOG --log-prefix " Fin Packets Scan "
$IPT -A INPUT -i ${PUB_IF} -p tcp --tcp-flags FIN,ACK FIN -j DROP
 $IPT -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROP
 # Log and get rid of broadcast / multicast and invalid
$IPT -A INPUT -i ${PUB_IF} -m pkttype --pkt-type broadcast -j LOG --log-prefix " Broadcast "
$IPT -A INPUT -i ${PUB_IF} -m pkttype --pkt-type broadcast -j DROP
 $IPT -A INPUT -i ${PUB_IF} -m pkttype --pkt-type multicast -j LOG --log-prefix " Multicast "
$IPT -A INPUT -i ${PUB_IF} -m pkttype --pkt-type multicast -j DROP
 $IPT -A INPUT -i ${PUB_IF} -m state --state INVALID -j LOG --log-prefix " Invalid "
$IPT -A INPUT -i ${PUB_IF} -m state --state INVALID -j DROP

# Log and block spoofed ips
$IPT -N spooflist
for ipblock in $SPOOFIP
do
  $IPT -A spooflist -i ${PUB_IF} -s $ipblock -j LOG --log-prefix " SPOOF List Block "
  $IPT -A spooflist -i ${PUB_IF} -s $ipblock -j DROP
done
$IPT -I INPUT -j spooflist
$IPT -I OUTPUT -j spooflist
$IPT -I FORWARD -j spooflist

# Allow ssh only from selected public ips
for ip in ${PUB_SSH_ONLY}
```

```
do
 $IPT -A INPUT -i ${PUB_IF} -s ${ip} -p tcp -d ${SERVER_IP} --destination-port 22 -j ACCEPT
 $IPT -A OUTPUT -o ${PUB_IF} -d ${ip} -p tcp -s ${SERVER_IP} --sport 22 -j ACCEPT
done


# allow incoming ICMP ping pong stuff
$IPT -A INPUT -i ${PUB_IF} -p icmp --icmp-type 8 -s 0/0 -m state --state NEW,ESTABLISHED,RELATED -m limit --limit 30/sec -j ACCEPT
$IPT -A OUTPUT -o ${PUB_IF} -p icmp --icmp-type 0 -d 0/0 -m state --state ESTABLISHED,RELATED -j ACCEPT
 # allow incoming HTTP port 80
$IPT -A INPUT -i ${PUB_IF} -p tcp -s 0/0 --sport 1024:65535 --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
$IPT -A OUTPUT -o ${PUB_IF} -p tcp --sport 80 -d 0/0 --dport 1024:65535 -m state --state ESTABLISHED -j ACCEPT
 # allow outgoing ntp
$IPT -A OUTPUT -o ${PUB_IF} -p udp --dport 123 -m state --state NEW,ESTABLISHED -j ACCEPT
$IPT -A INPUT -i ${PUB_IF} -p udp --sport 123 -m state --state ESTABLISHED -j ACCEPT


# allow outgoing smtp
$IPT -A OUTPUT -o ${PUB_IF} -p tcp --dport 25 -m state --state NEW,ESTABLISHED -j ACCEPT
$IPT -A INPUT -i ${PUB_IF} -p tcp --sport 25 -m state --state ESTABLISHED -j ACCEPT


### add your other rules here ####
 ########################
# drop and log everything else
$IPT -A INPUT -m limit --limit 5/m --limit-burst 7 -j LOG --log-prefix " DEFAULT DROP "
$IPT -A INPUT -j DROP
 exit 0
```

#8: Mengendalikan Serangan Buffer Overflow

Mengedit nginx.conf dan menset batas ukuran buffer untuk semua klien.

vi /usr/local/nginx/conf/nginx.conf

```
## Start: Size Limits & Buffer Overflows ##
client_body_buffer_size 1K;
client_header_buffer_size 1k;
client_max_body_size 1k;
large_client_header_buffers 2 1k;
## END: Size Limits & Buffer Overflows ##
```

dimana,

1. **client_body_buffer_size 1k** – (default is 8k or 16k): menentukan ukuran tubuh (body) buffer
2. **client_header_buffer_size 1k** – menentukan ukuran header buffer untuk permintaan dari klien. Untuk sebagian besar permintaan header, ukuran buffer 1K sudah mencukupi. Dapat tingkatkan jika ada tren peningkatan dari cookies (mis., klien WAP).
3. **client_max_body_size 1k**– menentukan ukuran tubuh maksimum yang diterima dari permintaan klien, ditunjukkan oleh baris-Panjang Konten di header permintaan. Jika ukuran lebih besar dari yang diberikan, maka klien mendapatkan notifikasi error " Request Entity Too Large" (413). Dapat ditingkatkan ketika mendapatkan unggahan file melalui metoda POST.
4. **large_client_header_buffers 2 1k** – menentukan jumlah dan ukuran buffer maksimum untuk header besar dari permintaan klien. Secara default ukuran satu buffer sama dengan ukuran halaman, tergantung pada platform ini baik 4K atau 8K, jika pada akhir koneksi permintaan koneksi dikonversikan ke status tetap-hidup, maka buffer ini dibebaskan. 2x1k akan menerima URI data 2kB. Ini juga akan membantu memerangi bot buruk dan serangan DoS.

   Kita juga dapat mengendalikan timeout untuk meningkatkan performa server dan memotong klien dengan mengedit sebagai berikut:

```
## Start: Timeouts ##
client_body_timeout 10;
client_header_timeout 10;
keepalive_timeout 5 5;
send_timeout  10;
## End: Timeouts ##
```

1. **client_body_timeout 10;** – menentukan batas waktu baca untuk tubuh terhadap permintaan dari klien. Batas waktu ditetapkan hanya jika tubuh tidak masuk dalam satu langkah baca. Jika setelah waktu ini klien mengirim sesuatu, nginx menginformasikan error " Request time out" (408).
2. **client_header_timeout 10;** – memberikan batas waktu dengan membaca judul permintaan klien. Batas waktu ditetapkan hanya jika header tidak masuk dalam satu langkah baca. Jika setelah waktu ini klien mengirim sesuatu, nginx menginformasikan error " Request time out" (408).
3. **keepalive_timeout 5 5;** – Parameter pertama menetapkan batas waktu untuk koneksi tetap-hidup dengan klien. Server akan menutup koneksi setelah waktu ini. Parameter kedua opsional menetapkan nilai waktu di header Keep-Alive: timeout = waktu respons. Header ini dapat meyakinkan beberapa browser untuk menutup koneksi, sehingga server tidak perlu melakukannya. Tanpa parameter ini, nginx tidak mengirim header Keep-Alive (meskipun ini bukan yang membuat koneksi “tetap hidup”). Instruksi menetapkan batas waktu dengan membaca judul permintaan klien. Batas waktu ditetapkan hanya jika header tidak masuk dalam satu langkah baca. Jika setelah waktu ini klien mengirim sesuatu, nginx menginformasikan error " Request time out” (408).
4. **send_timeout 10;** – memberikan batas waktu respons kepada klien. Timeout dibuat bukan pada seluruh transfer jawaban, tetapi hanya antara dua operasi pembacaan, jika setelah waktu ini klien tidak mengambil sesuatu, maka nginx mematikan koneksi.

#9: Mengendalikan Koneksi Secara Simultan

Kita dapat menggunakan modul NginxHttpLimitZone untuk membatasi jumlah koneksi simultan untuk sesi yang ditugaskan dari satu alamat IP. Dengan mengedit nginx.conf sebagai berikut:

```
### Directive describes the zone, in which the session states are stored i.e. store in slimits. ###
### 1m can handle 32000 sessions with 32 bytes/session, set to 5m x 32000 session ###
 limit_zone slimits $binary_remote_addr 5m;
 ### Control maximum number of simultaneous connections for one session i.e. ###
### restricts the amount of connections from a single ip address ###
 limit_conn slimits 5;
```

Pernyataan diatas membatasi jumlah koneksi klien hanya untuk 5 klien secara bersamaan

#10: Memperbolehkan akses hanya untuk ke domain saja

Jika bot hanya melakukan pemindaian server acak untuk semua domain, cukup melakukan penolakan saja. Hanya mengizinkan domain virtual yang dikonfigurasi atau reverse proxy. Tidak menampilkan permintaan menggunakan alamat IP:

```
## Only requests to our Host are allowed i.e. nixcraft.in, images.nixcraft.in and www.nixcraft.in
 if ($host !~ ^(nixcraft.in|www.nixcraft.in|images.nixcraft.in)$ ) {
  return 444;
 }
##
```

#11: Membatasi Metoda yang tersedia

GET dan POST adalah metoda yang paling umum di Internet. Jika server web tidak memerlukan penerapan semua metoda yang tersedia, metoda – metoda tersebut harus dinonaktifkan. Berikut ini cara memfilter dan hanya memperbolehkan metoda GET, HEAD, dan POST:

```
## Only allow these request methods ##
 if ($request_method !~ ^(GET|HEAD|POST)$ ) {
  return 444;
 }
## Do not accept DELETE, SEARCH and other methods ##
```

Penjelasan:

- Metoda GET digunakan untuk meminta dokumen seperti https://www.bssn.go.id/index.php.
- Metoda HEAD identik dengan GET kecuali jika server tidak harus mengembalikan tubuh pesan dalam responnya.
- Metoda POST dapat melibatkan apa saja, seperti menyimpan atau memperbarui data, atau memesan produk, atau mengirim E-mail dengan mengirimkan formulir. Ini biasanya diproses menggunakan skrip sisi server seperti PHP, PERL, Python dan sebagainya. Kita harus menggunakan ini jika akan mengunggah file dan memproses formulir di server.

#12: Menangkal User Agent Tertentu

Memblokir user agent seperti pemindai, bot, dan spammer yang mungkin menyalahgunakan web server kita.

```
## Block download agents ##
 if ($http_user_agent ~* LWP::Simple|BBBike|wget) {
  return 403;
 }
##
```

Memblokir robot seperti msnbot dan scrapbot (contoh):

```
## Block some robots ##
 if ($http_user_agent ~* msnbot|scrapbot) {
  return 403;
 }
```

#13: Memblokir Spam Rujukan

Spam Rujukan sangat berbahaya. Kita dapat memblokir akses ke spammer rujukan dengan garis-garis ini**.**

```
## Deny certain Referers ###
 if ( $http_referer ~* (babes|forsale|girl|jewelry|love|nudit|organic|poker|porn|sex|teen) )
 {
  # return 404;
  return 403;
 }
##
```

#13: Menghentikan Image Hotlinking

Tautan gambar atau HTML biasa terjadi dimana seseorang membuat tautan ke situs kita atau salah satu gambar di situs kita, dengan menampilkannya di situs mereka sendiri. Dampaknya kita akan membayar tagihan bandwidth dan membuat konten terlihat seperti bagian dari situs pembajak. Ini biasanya dilakukan di forum atau di blog. Berikut cara memblokir dan menghentikan hotlinking gambar di web server:

```
# Stop deep linking or hot linking
location /images/ {
 valid_referers none blocked www.contoh.com contoh.com;
 if ($invalid_referer) {
 return 403;
 }
}
```

Contoh lain: Menulis dan menampilkan kembali gambar

Another example with link to banned image:

```
valid_referers blocked www.contoh.com contoh.com;
 if ($invalid_referer) {
 rewrite ^/images/uploads.*\.(gif|jpg|jpeg|png)$ http://www.examples.com/banned.jpg last
 }
```

#14: Membatasi Akses Direktori

Kita dapat mengatur kontrol akses direktori yang sudah ditentukan. Semua direktori web harus dikonfigurasi berdasarkan kebutuhan/kepentingan tertentu.

Membatasi Akses berdasarkan IP Address

Kita dapat membatasi akses terhadap suatu direktori berbasis IP Address untuk direktori /docs/ directory:

```
location /docs/ {
 ## block one workstation
 deny 192.168.1.1;

 ## allow anyone in 192.168.1.0/24
 allow 192.168.1.0/24;

 ## drop rest of the world
 deny all;
}
```

Penggunaan Password Untuk Memproteksi Direktori

Membuat file password dan menambah user: dani:

```
# mkdir /usr/local/nginx/conf/.htpasswd/
# htpasswd -c /usr/local/nginx/conf/.htpasswd/passwd dani
```

Mengedit file nginx.conf dan proteksi direktori yang dibutuhkan sebagai berikut:

```
### Password Protect /personal-images/ and /delta/ directories ###
location ~ /(personal-images/.*|delta/.*) {
 auth_basic "Restricted";
 auth_basic_user_file /usr/local/nginx/conf/.htpasswd/passwd;
}
```

Ketika file password digenerate user dapat ditambahkan didalam direktori usr/local/nginx/conf/.htpasswd/passwd dengan perintah sebagai berikut:

```
# htpasswd -s /usr/local/nginx/conf/.htpasswd/passwd userName
```

#15: Mengkonfigurasi SSL Nginx

Konfigurasi SSL

Langkah dasar dan pertama dalam keamanan web adalah menerapkan SSL agar dalam mengakses aplikasi web sudah menggunakan https dan menambahkan lapisan enkripsi dalam komunikasi.

- Menggunakan Open SSL untuk mengenerate CSR (Certificate Signing Request) dengan 2048 bit dan sha-2 melalui perintah sebagai berikut:

```
OpenSSL req -nodes -new -sha256 -newkey rsa:2048 -keyout bestflare.key -out bestflare.csr
```

- Dari perintah diatas akan mengenerate CSR dan file key dan langsung aktif. Setelahnya untuk tidak lupa mengganti nama file .csr dan key tersebut .
- Setelah mendapatkan CSR yang ditKitatangani oleh otoritas sertifikat dan memiliki sertifikat yang ditKitatangani, kita dapat menerapkannya di Nginx seperti di bawah ini.
    - Masuk ke server nginx
    - Buka folder conf di mana Kita memiliki file ssl.conf.

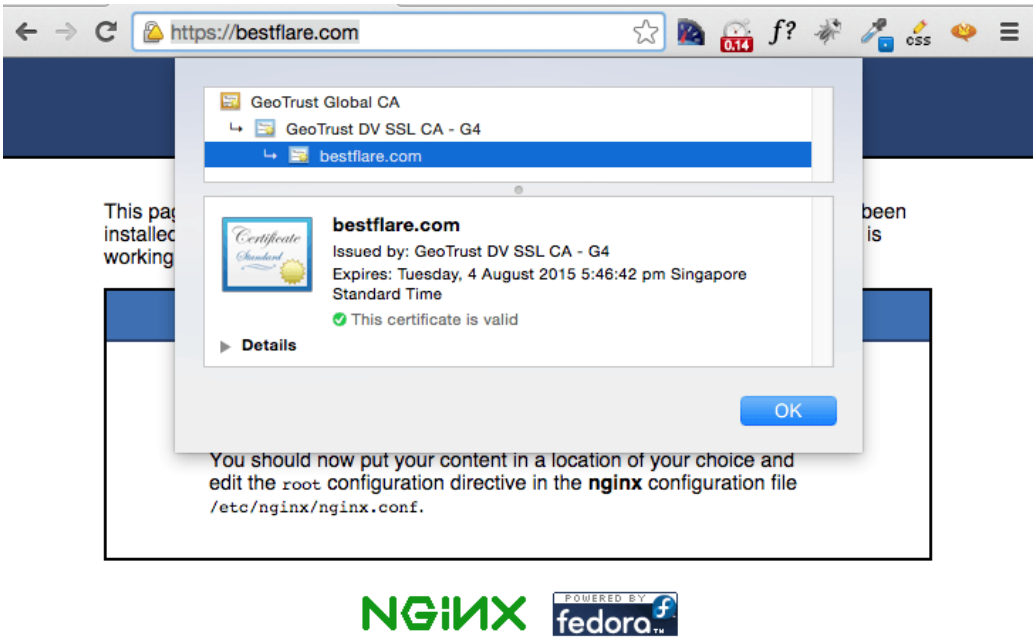
Note: dan default instalasi di Linux akan kita dapatkan di **/etc/nginx/conf.d.**

- Mengedit file dan menambahkan ssl certificate, yang akan mengaktifkan Nginx pada port 443

```
server {
listen 443 ssl;
 server_name bestflare.com;
 ssl   on;
 ssl_certificate /opt/cert/bestflare.pem;
 ssl_certificate_key /opt/cert/bestflare.key;
}
```

Note: Tidak lupa untuk mengubah path sertifikat dan file key.

- Save konfigurasi dan restart Nginx. Maka SSL cert is berhasil diimplementasikan.



Optimalisasi SSL/TLS

Setelah mendapatkan SSL bukan berarti web secara penuh aman, tetapi perlu penerapan konfigurasi untuk mengamankan web server.

Menonaktifkan SSL 3 dan tetap mempertahankan TLS

Secara umum SSL 3 rentan, dan disarankan menggunakan protokol TLS yang kuat.

- Mengedit file ssl.conf dan tambahkan di bawah ini dalam blok server

```
ssl_protocols TLSv1.2;
```

- Save file ssl.conf dan restart Nginx

Menonaktifkan weak cipher suites

- Tambahkan statement berikut pada server block didalam file ssl.conf:

```
ssl_ciphers "EECDH+ECDSA+AESGCM EECDH+aRSA+AESGCM EECDH+ECDSA+SHA384
EECDH+ECDSA+SHA256 EECDH+aRSA+SHA384 EECDH+aRSA+SHA256 EECDH+aRSA+RC4 EECDH
EDH+aRSA HIGH !RC4 !aNULL !eNULL !LOW !3DES !MD5 !EXP !PSK !SRP !DSS";
```

- Save file ssl.conf dan restart Nginx

Install Chain Certificate

Kita perlu mendapatkan sertifikat berantai dari lembaga otoritas. Sebagian besar Anda akan menemukan di situs web mereka.

- Tambahkan konten sertifikasi dalam sertifikat situs web tersut seperti dibawah ini. Dalam contoh ini adalah /opt/cert/bestflare.pem

```
[root@Chandan conf.d]# cat /opt/cert/bestflare.pem
-----BEGIN CERTIFICATE-----
MIIE0TCCA7mgAwIBAgIDAa0NMA0GCSqGSIb3DQEBCwUAMGYxCzAJBgNVBAYTAlVT
MRYwFAYDVQQKEw1HZW9UcnVzdCBJbmMuMR0wGwYDVQQLExREb21haW4gVmFsaWRh
dGVkIFNTTDEgMB4GA1UEAxMXR2VvVHJ1c3QgRFYgU1NMIENBIC0gRzQwHhcNMTUw
NzA0MTQwMjQ1WhcNMTUwODA0MDk0NjQyWjCBkTETMBEGA1UECxMKR1QyMDAyNzIz
OTEnMCUGA1UECxMed3d3Lmdlb3RydXN0LmNvbS9yZXNvdXJjZXMvY3BzMTkwNwYD
VQQLEzBEb21haW4gQ29udHJvbCBWYWxpZGF0ZWQgLSBHZW9UcnVzdCh5KSBTU0wg
VHJpYWwxFjAUBgNVBAMTDWJlc3RmbGFyZS5jb20wggEiMA0GCSqGSIb3DQEBAQUA
A4IBDwAwggEKAoIBAQCpwB1cxv8dbkiU8NdWNCOjKxH7zyvH/Xj6enwCcQBmIzG/
CitlpbtbLl6dpxBcFk2ihGCNfwt72y0szCAk9m1XvtsHXtLUKZ83TgeodVtlt+D+
Oyj59uhaGRHq8hcpVeMqlGGsZ8WXziUhMTQB83n4CRLwO+CTWS5+1mNsMdok0Dwc
QnHsoiAaVSadLJNZfL+7LCIc11vHbBU1fMfLlVcFn/NfTVz4bhHH7ed77qmbzGqF
zxIc/QPKvHz2S9XVhr8YVPJLIsv9iHpf2DbdLcuouQ+EZTfJ37wPyjF2h0jMo4RJ
OyIstngDa92/ddfHwqDmxrwrlw1ljVTZSXsw6zXDAgMBAAGjggFaMIIBVjAfBgNV
HSMEGDAWgBQLUOx37yqb/+wDoQr/rcbkKhjHPjBXBggrBgEFBQcBAQRLMEkwHwYI
KwYBBQUHMAGGE2h0dHA6Ly9ndS5zeW1jZC5jb20wJgYIKwYBBQUHMAKGGmh0dHA6
Ly9ndS5zeW1jYi5jb20vZ3UuY3J0MA4GA1UdDwEB/wQEAwIFoDAdBgNVHSUEFjAU
BggrBgEFBQcDAQYIKwYBBQUHAwIwGAYDVR0RBBEwD4INYmVzdGZsYXJlLmNvbTAr
BgNVHR8EJDAiMCCgHqAchhpodHRwOi8vZ3Uuc3ltY2IuY29tL2d1LmNybDAMBgNV
HRMBAf8EAjAAMFYGA1UdIARPME0wSwYGZ4EMAQIBMEEwPwYIKwYBBQUHAgEWM2h0
dHBzOi8vd3d3Lmdlb3RydXN0LmNvbS9yZXNvdXJjZXMvcmVwb3NpdG9yeS9sZWdh
bDANBgkqhkiG9w0BAQsFAAOCAQEAPWtccsp7xEWbnB4EDGgqk1oCBDZ7EE7raZ8n
Ou1GmPJvfcmz4l32Lo9nq0arF7hak37Fs7n33w+QWARppXVwkiZNTf5BCUmekxfv
VXX16bU/XkjL/deR45vNxh+71sY0utf6HCFvcKmPAv9T87xwDr74Ku4EnrNg5D/V
bjK+vQrwuCJwl1fg4l5GMlc5qzKPmwWaBkxpLhwKywnoV9rWTVrlHG+v+RX+h4Ed
rQiQnBoKw+9eIVoDouRMxZnBoHqKAFegF2qgj5yMkaCHtJ8ahkzmGbwpdxVH954y
/qHWJwd51wwCq974pSmFMwJ8NHtT2kZY1VROMkvi8TG1p+yQCw==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIERDCCAyygAwIBAgIDAjp4MA0GCSqGSIb3DQEBCwUAMEIxCzAJBgNVBAYTAlVT
MRYwFAYDVQQKEw1HZW9UcnVzdCBJbmMuMRswGQYDVQQDExJHZW9UcnVzdCBHbG9i
YWwgQ0EwHhcNMTQwODI5MjIyNDU4WhcNMjIwNTIwMjIyNDU4WjBmMQswCQYDVQQG
EwJVUzEWMBQGA1UEChMNR2VvVHJ1c3QgSW5jLjEdMBsGA1UECxMURG9tYWluIFZh
bGlkYXRlZCBTU0wxIDAeBgNVBAMTF0dlb1RydXN0IERWIFNTTCBDQSAtIEc0MIIB
IjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA30GUetr35DFDtuoBG1zOY+r6
baPZau4tmnX51ZxbvTTf2BzJbdgEiNputbe18DCuQNZd+sRTwdQinQR0EaaV1UV8
QQVY4Ezd+e5VvV9G3K0TCJ0s5PeC5gcrng6MNKHOxKHggXCGAAY/Lep8myiuGyiL
0QnT5/BFpLG6EWeQVXuP3u04XKHh44PEw3KRT5juHMKAqmSlPoNiHMzgnvhawBMS
faKni6PnnyrXm8rL7ZcBnCiEUQRQQby0/HjpG88U6h8P/C4BMo22NcsKGDvsWj48
G9OZQx4v973zWxK5B17tPtGph8x3cifU2XWiY0uTNr3lXNe/X3kNszKnC7JjIwID
AQABo4IBHTCCARkwHwYDVR0jBBgwFoAUwHqYaI2J+6sFZAwRfap9ZbjKzE4wHQYD
VR0OBBYEFAtQ7HfvKpv/7AOhCv+txuQqGMc+MBIGA1UdEwEB/wQIMAYBAf8CAQAw
DgYDVR0PAQH/BAQDAgEGMDUGA1UdHwQuMCwwKqAooCaGJGh0dHA6Ly9nLnN5bWNi
LmNvbS9jcmxzL2d0Z2xvYmFsLmNybDAuBggrBgEFBQcBAQQiMCAwHgYIKwYBBQUH
MAGGEmh0dHA6Ly9nLnN5bWNkLmNvbTBMBgNVHSAERTBDMEEGCmCGSAGG+EUBBzYw
MzAxBggrBgEFBQcCARYlaHR0cDovL3d3dy5nZW90cnVzdC5jb20vcmVzb3VyY2Vz
L2NwczANBgkqhkiG9w0BAQsFAAOCAQEAMyTVkKopDDW5L8PHQpPAxhBLAwh2hBCi
4OdTEifyCtp/Otz9XHlajxd0Q1Ox1dFdWbmmhGTK8ToKWZYQv6mBV4tch9x/4+S7
BXqgMgkTThCBKB+cA2K89AG1KYNGB7nnuF3I6dHdrTv4NNvB0ZWpkRjtPCw3EU3M
/lM+UEP5w1ZBrFObbAWymuLgWVcwMrYmThMlzfpIcA91VWAR9TvVXlo8i1sPD2JC
SGGFixD0wYi/f1+KwtfNK5RcHzRKCK/rromoSHVVlR27wJoBufQDIj7U5lIwDWe5
wJH9LUwwjr2MpQSRu6Srfw/Yb/BmAMmjXPWwj4PmnFrmtrnFvL7kAg==
-----END CERTIFICATE-----
[root@Chandan conf.d]#
```

- Save file dan restart Nginx

Secure Diffie-Hellman untuk TLS

Salah satu praktik terbaik yang akhir-akhir ini ditambahkan dalam daftar adalah untuk mengamankan diffie-hellman.

Membuat DH Group unik dan menambahkan ssl_dhparam di file ssl.conf

- Generate Unique DH Group dengan menggunakan OpenSSL

```
OpenSSL dhparam -out dhparams.pem 4096
```

Ini akan memakan waktu beberapa menit dan akan menghasilkan file dhparams.pem pada direktori kerja saat ini

- Salin dhparams.pem ke folder cert
- Ubah ssl.conf dan tambahkan berikut dalam blok server

```
ssl_dhparam /opt/cert/dhparams.pem;
```

- Save file ssl.conf dan restart Nginx

Hasil lengkap dari optimasi ssl.conf

```
# HTTPS server configuration
server {
 listen 443 ssl;
 server_name bestflare.com;
 ssl   on;
 ssl_certificate /opt/cert/bestflare.pem;
 ssl_certificate_key /opt/cert/bestflare.key;
 ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
 ssl_prefer_server_ciphers on;
 ssl_ciphers "EECDH+ECDSA+AESGCM EECDH+aRSA+AESGCM EECDH+ECDSA+SHA384
EECDH+ECDSA+SHA256 EECDH+aRSA+SHA384 EECDH+aRSA+SHA256 EECDH+aRSA+RC4 EECDH
EDH+aRSA HIGH !RC4 !aNULL !eNULL !LOW !3DES !MD5 !EXP !PSK !SRP !DSS";
```

```
 ssl_dhparam /opt/cert/dhparams.pem;
}
```

Menonaktifkan Metoda Http yang tidak diperlukan

Mengizinkan modul GET, HEAD & POST Http tetapi tidak untuk TRACE atau DELETE karena berisiko mendapat serangan Pelacakan Lintas Situs (Cross-Site Tracking) dan berpotensi peretas mencuri informasi cookie.

- Salin dhparams.pem ke folder cert
- Ubah ssl.conf dan tambahkan berikut ini dalam blok server

```
if ($request_method !~ ^(GET|HEAD|POST)$ )
{
 return 405;
}
```

Save file tersebut and restart Nginx. Disini kita akan menampilkan notifikasi 405 (Not Allowed) jika seseorang mencoba menggunakan TRACE, DELETE, PUT, OPTIONS.

```
ChKitans-iMac:~ chKitan$ telnet bestflare.com 80
Trying 128.199.100.162...
Connected to bestflare.com.
Escape character is '^]'.
TRACE / HTTP/1.1
Host: testing
HTTP/1.1 405 Not Allowed
Server: nginx
Date: Sat, 11 Jul 2015 06:04:34 GMT
Content-Type: text/html
Content-Length: 166
```

```
Connection: close
```

Memproteksi dari serangan Clickjacking

Kita dapat memasukkan X-FRAME-OPTIONS dalam HTTP Header untuk mencegah serangan clickjacking. Hal ini dapat dilakukan dengan menambahkan statement tersebut dalam file Nginx.conf

```
add_header X-Frame-Options "SAMEORIGIN";
```

Header di atas akan menginstruksikan browser hanya untuk memuat sumber daya dari asal yang sama.

Proteksi dengan X-XSS

Memasukkan proteksi X-XSS HTTP Header untuk memitigasi serangan Cross-Site Scripting

- Ubah file default.conf atau ssl.conf dengan menambahkan statement sebagai berikut:

```
add_header X-XSS-Protection "1; mode=block";
```

  - Save konfigurasi file diatas dan restart Nginx.

#16: Pengamanan Nginx dan PHP

PHP adalah salah satu bahasa scripting sisi server yang populer. Untuk pengamanannya dapat mengedit /etc/php.ini sebagai berikut:

```
# Disallow dangerous functions
disable_functions = phpinfo, system, mail, exec
 ## Try to limit resources ##
 # Maximum execution time of each script, in seconds
max_execution_time = 30
 # Maximum amount of time each script may spend parsing request data
```

```
max_input_time = 60
 # Maximum amount of memory a script may consume (8MB)
memory_limit = 8M
 # Maximum size of POST data that PHP will accept.
post_max_size = 8M
 # Whether to allow HTTP file uploads.
file_uploads = Off
 # Maximum allowed size for uploaded files.
upload_max_filesize = 2M
 # Do not expose PHP error messages to external users
display_errors = Off
 # Turn on safe mode
safe_mode = On
 # Only allow access to executables in isolated directory
safe_mode_exec_dir = php-required-executables-path
 # Limit external access to PHP environment
safe_mode_allowed_env_vars = PHP_
 # Restrict PHP information leakage
expose_php = Off
 # Log all errors
log_errors = On
 # Do not register globals for input data
register_globals = Off
 # Minimize allowable PHP post size
post_max_size = 1K
 # Ensure PHP redirects appropriately
cgi.force_redirect = 0
```

```
# Disallow uploading unless necessary
file_uploads = Off
# Enable SQL safe mode
sql.safe_mode = On
# Avoid Opening remote files
allow_url_fopen = Off
```

#17: Menjalankan Nginx dalam suatu Chroot Jail (Containers)

Menempatkan nginx di chroot jail akan meminimalkan kerusakan karena potensi pembobolan dengan mengisolasi server web ke bagian kecil dari sistem file. Anda dapat menggunakan jenis pengaturan chroot tradisional dengan nginx. Jika dimungkin menggunakan Chroot Jail FreeBSD, XEN, Debian / Ubuntu, LXD pada Fedora, atau virtualisasi OpenVZ.

#18: Membatasi Koneksi Input Per IP Address pada level firewall

Server web harus mengawasi koneksi dan membatasi koneksi per detik. PF Firewall dan Iptables firewall dapat membatasi pengguna akhir sebelum mengakses server nginx.

Linux Iptables: Membatasi Koneksi Input Nginx Per Detik

Contoh berikut adalah upaya mendrop koneksi yang masuk jika adanya upaya melakukan koneksi lebih dari 15 kali dan melebihi waktu 60 detik ke port 80:

```
/sbin/iptables -A INPUT -p tcp --dport 80 -i eth0 -m state --state NEW -m recent --set
/sbin/iptables -A INPUT -p tcp --dport 80 -i eth0 -m state --state NEW -m recent --update --seconds 60 --hitcount 15 -j DROP
service iptables save
```

BSD PF Firewall: Membatasi Koneksi Input Nginx Per Detik

- Mengedit file konfigurasi: /etc/pf.conf dan memperbarunya sebagai berikut.

```
webserver_ip="202.54.1.1"
table <abusive_ips> persist
block in quick from <abusive_ips>
pass in on $ext_if proto tcp to $webserver_ip port www flags S/SA keep state (max-src-conn 100, max-src-conn-rate 15/5, overload <abusive_ips> flush)
```

- Statement diatas akan membatasi jumlah maksimum koneksi per sumber hingga 100. 15/5 menentukan jumlah koneksi per detik atau rentang detik yaitu tingkat membatasi jumlah koneksi hingga 15 dalam rentang 5 detik. Jika ada yang melanggar aturan maka akan diblokir.

#19: Mengkonfigurasi Operating System untuk memproteksi Web Server

File dalam DocumentRoot (/ nginx atau / usr / local / nginx / html) tidak boleh dimiliki atau ditulis oleh pengguna tertentu (contoh: nginx)

```
# find /nginx -user nginx
# find /usr/local/nginx/html -user nginx
```

Mengubah kepemilikan file menjadi root atau pengguna lain. Seperangkat izin khusus / usr / local / nginx / html /

```
# ls -l /usr/local/nginx/html/
```

Contoh outputs:

```
-rw-r--r-- 1 root root 925 Jan 3 00:50 error4xx.html
-rw-r--r-- 1 root root 52 Jan 3 10:00 error5xx.html
-rw-r--r-- 1 root root 134 Jan 3 00:52 index.html
```

Kita harus menghapus file cadangan yang tidak diperlukan yang dibuat oleh vi atau editor teks lainnya:

```
# find /nginx -name '.?*' -not -name .ht* -or -name '*~' -or -name '*.bak*' -or -name '*.old*'
# find /usr/local/nginx/html/ -name '.?*' -not -name .ht* -or -name '*~' -or -name '*.bak*' -or -name '*.old*'
```

#20: Membatasi Koneksi Keluar Nginx

Hacker akan mengunduh file secara lokal di server kita dengan menggunakan alat seperti wget. Menggunakan iptables untuk memblokir koneksi keluar dari pengguna Nginx. Modul ipt_owner berupaya untuk mencocokkan berbagai karakteristik pembuat paket, untuk paket yang dibuat secara lokal. Ini hanya valid dalam rantai OUTPUT. Dalam contoh ini, mengizinkan pengguna yang bernama dani untuk terhubung ke luar menggunakan port 80:

```
/sbin/iptables -A OUTPUT -o eth0 -m owner --uid-owner dani -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
/sbin/iptables -A OUTPUT -o eth0 -m owner --uid-owner dani -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j DENY
```

Dapat juga menambahkan aturan di atas ke skrip shell berbasis iptables untuk tidak mengizinkan pengguna server web nginx terhubung di luar.

#21: Memelihara Software agar tetap up to date

Secara regular harus selalu memperbarui perangkat lunak dan kernel Linux kita. Menerapkan patches sesuai versi atau distro Linux yang digunakan. Jika menggunakan Linux Debian / Ubuntu dapat menggunakan perintah apt-get / perintah apt untuk menerapkan patches:

`$ sudo apt-get update`

`$ sudo apt-get upgrade`

Jika menggunakan Linux RHEL/CentOS/Oracle/Scientific Linux, dapat menggunakan yum command:

`$ sudo yum update`

Jika menggunakan Linux Linux dapat menggunakan apk command:

`# apk update`

`# apk upgrade`

#22: Proteksi dari clickjacking

Menambahkan statement berikut didalam file nginx.conf atau virtual domain untuk menghindari clickjacking:

`add_header X-Frame-Options SAMEORIGIN;`

#23: Menonaktifikan content-type sniffing pada beberapa web browser

Menambahkan statement berikut didalam file nginx.conf atau virtual domain:

```
add_header X-Content-Type-Options nosniff;
```

#24: Mengaktifikan filter terhadap Cross-site scripting (XSS)

Menambahkan statement berikut didalam file nginx.conf atau virtual domain:

```
add_header X-XSS-Protection "1; mode=block";
```

#25: Memonitor Nginx

Memeriksa secara rutin file Log. Log ini akan memberi informasi tentang serangan terhadap server dan memastikan tingkat keamanan yang diperlukan.

```
# grep "/login.php??" /usr/local/nginx/logs/access_log
# grep "...etc/passwd" /usr/local/nginx/logs/access_log
# egrep -i "denied|error|warn" /usr/local/nginx/logs/error_log
```

IV.7.2 HARDENING WEB SERVER APACHE

HARDENING WEB SERVER

APACHE

A. TUJUAN

Memberikan petunjuk/*guidance* dalam melakukan hardening pada web server Apache

B. RUANG LINGKUP

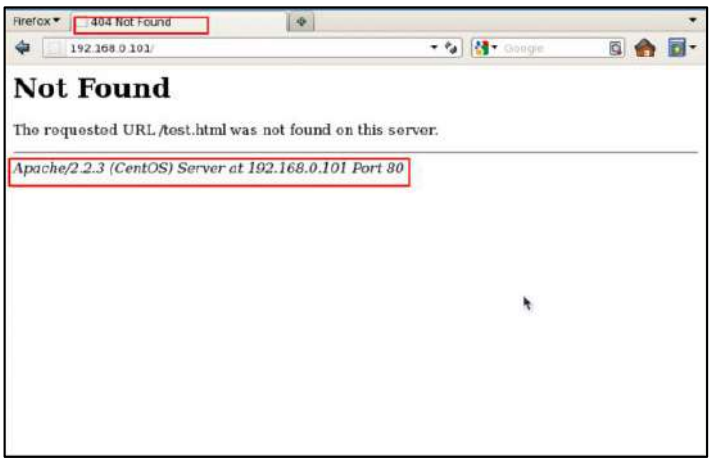
Tahapan hardening pada web server Apache

C. REFERENSI

1. Security Tips - Apache HTTP Server Version 2.4

D. LANGKAH - LANGKAH

1. Sebelum melakukan rekonfigurasi di web server Apache ini, sebaiknya perlu diketahui konfigurasi default dari Apache tersebut, yaitu:
   a. Direktori root untuk web server ada di: /var/www/html or /var/www
   b. File untuk konfigurasi utamanya ada di: /etc/httpd/conf/httpd.conf (RHEL/CentOS/Fedora) and /etc/apache2/apache2.conf (Debian/Ubuntu).
   c. Default HTTP Port: 80 TCP
   d. Default HTTPS Port: 443 TCP
   e. Menguji hasil setting file konfigurasi menggunakan sintaks: httpd -t
   f. Akses ke log file web server ada di: /var/log/httpd/access_log
   g. Akses ke file error log web server ada di: /var/log/httpd/error_log
2. Menyembunyikan identitas versi OS dan Apache ketika terjadi error
   a. Default sebelum dilakukan rekonfigurasi:
      Salah satu ancaman keamanan yang besar bagi suatu web server, jika identitas versi OS dan Apache ini ditampilkan



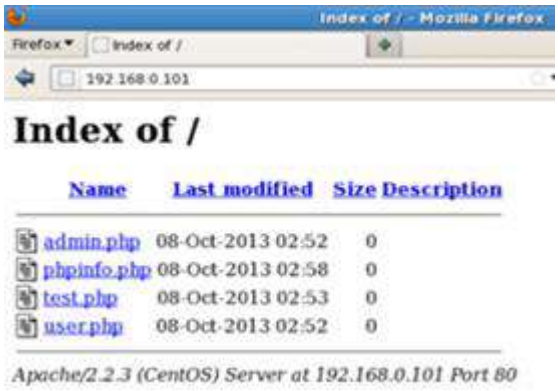
b. Rekonfigurasi untuk menyembunyikan identitas versi OS dan Apache

Buka file konfigurasi dengan editor vim dan cari "ServerSignature", yang secara default “On” menjadi “Off”, dan mematikan tanda tangan server ini pada baris kedua yaitu "ServerTokens Prod".

```
# vim /etc/httpd/conf/httpd.conf (RHEL/CentOS/Fedora)
# vim /etc/apache2/apache2.conf (Debian/Ubuntu)
ServerSignature Off
ServerTokens Prod
# service httpd restart (RHEL/CentOS/Fedora)
# service apache2 restart (Debian/Ubuntu)
```

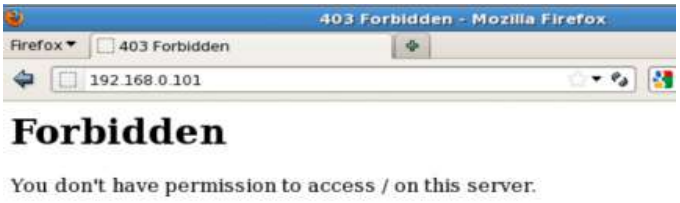
3. Menonaktifkan Directory Listing

Secara default web server Apache akan memperlihatkan semua daftar konten dari direktori root, seperti gambar berikut ini:



Kita dapat mematikan daftar direktori dengan menggunakan direktif options dalam file konfigurasi terhadap direktori tertentu. Untuk itu kita perlu membuat entri di file httpd.conf atau apache2.conf.

```
<Directory /var/www/html>
 Options - Indexes
</Directory>
```



4. Mengupdate Apache secara rutin

   Komunitas pengembang Apache terus berupaya mengatasi masalah keamanan dan merilis versi terbarunya dengan opsi keamanan baru. Sangat disarankan untuk menggunakan versi terbaru dari Apache sebagai server web.

   Untuk memeriksa versi Apache: Kita dapat memeriksa versi Apache saat ini dengan perintah httpd -v.

   ```
   # httpd -v
   Server version: Apache/2.2.15 (Unix)
   Server built: Aug 13 2013 17:29:28
   ```

   Kita dapat memperbarui versi Apache dengan perintah berikut.

   ```
   # yum update httpd
   # apt-get install apache2
   ```

   disarankan juga untuk terus memperbarui Kernel OS Linux ke rilis yang lebih stabil dan terbaru.

5. Menonaktifkan Modul – modul yang tidak dibutuhkan

   Berikut adalah modul – modul default yang ada didalam OS Linux. Untuk menonaktifkan modul tinggal memasukkan tanda # disetiap baris pernyataan modul. Beberapa modul yang umum sering dinonaktifkan yaitu: mod_imap, mod_include, mod_info, mod_userdir, dan mod_autoindex.

   ```
   # grep LoadModule /etc/httpd/conf/httpd.conf
   # have to place corresponding `LoadModule' lines at this location so the
   # LoadModule foo_module modules/mod_foo.so
   LoadModule auth_basic_module modules/mod_auth_basic.so
   LoadModule auth_digest_module modules/mod_auth_digest.so
   LoadModule authn_file_module modules/mod_authn_file.so
   LoadModule authn_alias_module modules/mod_authn_alias.so
   LoadModule authn_anon_module modules/mod_authn_anon.so
   LoadModule authn_dbm_module modules/mod_authn_dbm.so
   LoadModule authn_default_module modules/mod_authn_default.so
   LoadModule authz_host_module modules/mod_authz_host.so
   LoadModule authz_user_module modules/mod_authz_user.so
   LoadModule authz_owner_module modules/mod_authz_owner.so
   LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
   LoadModule authz_dbm_module modules/mod_authz_dbm.so
   LoadModule authz_default_module modules/mod_authz_default.so
   LoadModule ldap_module modules/mod_ldap.so
   ```

```
LoadModule authnz_ldap_module modules/mod_authnz_ldap.so
LoadModule include_module modules/mod_include.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule logio_module modules/mod_logio.so
LoadModule env_module modules/mod_env.so
LoadModule ext_filter_module modules/mod_ext_filter.so
....
```

6. Menjalankan Apache dengan User & Group Yang terpisah

   Dalam instalasi default, Apache menjalankan prosesnya dengan tidak menggunakan user atau daemon. Maka untuk alasan keamanan, sangat disarankan untuk menjalankan Apache di akun yang tidak istimewa. Jadi dibuat akun yang tidak mudah ditebak. Contoh: http-web.

   Membuat User & Group Apache

   ```
   # groupadd http-web
   # useradd -d /var/www/ -g http-web -s /bin/nologin http-web
   ```

   Kemudian menginformasikan kepada Apache untuk menjalankan dengan user baru tersebut. Dan untuk melakukannya, kita perlu mengentri di /etc/httpd/conf/httpd.conf dan restart layanan yaitu:

   Buka /etc/httpd/conf/httpd.conf dengan editor vim dan cari kata kunci "Pengguna" dan "Grup" dan selanjutnya menentukan nama pengguna dan nama grup yang akan digunakan (contoh: http-web).

   ```
   User http-web
   Group http-web
   ```

7. Penggunaan Allow and Deny untuk membatasi akses direktori

   Kita dapat membatasi akses ke direktori dengan opsi "Allow" dan "Deny" dalam file httpd.conf. Dalam contoh ini, kita akan mengamankan direktori root, dengan mengentri opsi "Allow" dan "Deny" berikut ini di file httpd.conf.

   ```
   <Directory />
    Options None
    Order deny,allow
    Deny from all
   </Directory>
   ```

   a. Opsi Options “None” – Opsi ini tidak mengizinkan pengguna untuk mengaktifkan fitur opsi apapun.
   b. Order deny, allow – Proses diurutkan mulai "Deny" setelah itu "Allow".
   c. Deny from all – Ini akan menolak permintaan dari semua orang ke direktori root, tidak ada yang akan dapat mengakses direktori root.

8. Penggunaan Modul mod_security dan mod_evasive untuk mengamankan Apache

Dua modul ini “mod_security” dan “mod_evasive” adalah modul Apache terkait keamanan yang sangat populer.

Mod_Security

Mod_security berfungsi sebagai firewall untuk aplikasi web yang memungkinkan untuk memantau lalu lintas secara real time. Selain itu, mampu melindungi situs web atau server web dari serangan brute force. Cukup dengan menginstal mod_security di server.

Instalasi mod_security pada LinuxUbuntu/Debian

```
$ sudo apt-get install libapache2-modsecurity
$ sudo a2enmod mod-security
$ sudo /etc/init.d/apache2 force-reload
```

Instalasi mod_security pada Linux RHEL/CentOS/Fedora/

```
# yum install mod_security
# /etc/init.d/httpd restart
```

Mod_Evasive

Mod_evasive bekerja sangat efisien dalam mencegah serangan DoS, DDoS, dan brute force HTTP. Modul ini mendeteksi serangan dengan tiga metoda, yaitu:

a. Jika begitu banyak permintaan datang ke halaman yang sama dalam beberapa kali per detik.

b. Jika ada proses anak mencoba untuk membuat lebih dari 50 permintaan bersamaan.

c. Jika ada IP yang masih mencoba untuk membuat permintaan baru ketika itu sementara daftar hitam.

Mod_evasive dapat diinstal langsung dari sumbernya.

9. Menonaktifikan Simbolic Link Apache

Secara default Apache mengikuti symlinks dan dapat mematikan fitur ini dengan FollowSymLinks dengan direktif Opsi. Dan untuk melakukannya kita perlu membuat entri berikut di file konfigurasi utama.

```
Options -FollowSymLinks
```

Dan, jika ada pengguna atau situs web tertentu yang perlu mengaktifkan FollowSymLinks, kita cukup menulis aturan dalam file “.htaccess” dari situs web itu**.**

```
# Enable symbolic links
Options +FollowSymLinks
```

Catatan: Untuk mengaktifkan aturan penulisan ulang di dalam file ".htaccess" "AllowOverride All" harus ada dalam konfigurasi utama secara global**.**

10. Mematikan Server Side Includes dan CGI Execution

Kita dapat mematikan server side includes (mod_include) dan CGI execution jika tidaj diperlukan dengan memodifikasi file konfigurasi sebagai berikut:

```
Options -Includes
Options -ExecCGI
```

Kita dapat juga melakukan ini untuk direktori tertentu dengan tag direktori. Dalam contoh ini, kita mematikan eksekusi file Includes dan Cgi pada direktori "/ var / www / html / web1".

```
<Directory "/var/www/html/web1">
Options -Includes -ExecCGI
</Directory>
```

Berikut adalah beberapa nilai lain yang dapat "on" atau "off" dengan direktif opsi yaitu:

a. Opsi All – opsi ini berarti mengaktifkan semua opsi sekaligus dan ini adalah nilai default. Jika tidak ingin menentukan nilai apa pun secara eksplisit dalam file conf Apache atau .htaccess.

b. Opsi IncludeNOEXEC - Opsi ini memungkinkan sisi server termasuk tanpa izin eksekusi ke file perintah atau cgi.

c. Opsi MultiViews – Opsi ini memungkinkan konten yang dinegosiasikan multiview dengan modul mod_negotiation.

d. Opsi SymLinksIfOwnerMatch - Mirip dengan FollowSymLinks. Tapi hanya akan mengikuti ketika pemiliknya sama antara link dan direktori asli yang dilinkan.

11. Membatasi Ukuran Request

Secara default, Apache tidak memiliki batasan terhadap ukuran total permintaan HTTP tetapi ketika mengizinkan permintaan besar pada server web, ada kemungkinan menjadi korban serangan Denial of Service. Kita dapat membatasi ukuran permintaan arahan Apache "LimitRequestBody" dengan tag direktori.

Anda dapat mengatur nilai dalam byte dari 0 (tidak terbatas) hingga 2147483647 (2GB) yang diizinkan. Kita dapat menetapkan batas ini sesuai dengan kebutuhan situs. Misalkan kita mengizinkan unggahan dan Anda ingin membatasi ukuran unggahan untuk direktori tertentu.

Dalam contoh ini, user_uploads adalah direktori yang berisi file yang diunggah oleh pengguna, dan dibatasi sampai 500 ribu:

```
<Directory "/var/www/myweb1/user_uploads">
 LimitRequestBody 512000
</Directory>
```

12. Memproteksi Serangan DDOS dan Hardening

Berikut adalah beberapa opsi direktif yang dapat membantu untuk mengendalikannya.

a. TimeOut: opsi direktif ini memungkinkan untuk mengatur waktu server akan menunggu suatu event tertentu selesai sebelum gagal. Nilai standar/defaultnya adalah 300 detik. Menjaga nilai ini tetap rendah di situs-situs yang terkena serangan DDOS.

b. MaxClients: opsi direktif ini memungkinkan kita untuk menetapkan batas koneksi yang akan dilayani secara bersamaan. Setiap koneksi baru akan diantrikan setelah batas ini.. Nilai standarnya adalah 256.

c. KeepAliveTimeout: Jumlah waktu server akan menunggu permintaan berikutnya sebelum menutup koneksi. Nilai default adalah 5 detik.

d. LimitRequestFields: Ini membantu kita untuk menetapkan batas pada jumlah header permintaan http yang akan diterima dari klien. Nilai default-nya adalah 100. Disarankan untuk menurunkan nilai ini jika serangan DDos terjadi sebagai akibat dari begitu banyak header permintaan http.

e. LimitRequestFieldSize: Ini membantu kita untuk menetapkan batas ukuran pada header permintaan http.

13. Mengaktifkan Logging Apache

Apache memungkinkan kita untuk logging secara independen terpisah dengan logging OS. Logging Apache menyediakan lebih banyak informasi, seperti perintah yang dimasukkan oleh pengguna yang telah berinteraksi dengan server web. Untuk melakukannya, perlu memasukkan modul mod_log_config.

Ada tiga arahan terkait logging yang tersedia pada Apache.

a. TransferLog: Membuat file log.

b. LogFormat: Menentukan format khusus.

c. CustomLog: Membuat dan memformat file log.

Logging Apache ini juga dapat menggunakannya untuk hosting virtual. Sebagai contoh, di sini adalah konfigurasi host virtual situs web dengan logging diaktifkan.

```
<VirtualHost *:80>
DocumentRoot /var/www/html/contoh.com/
ServerName www.contoh.com
```

```
DirectoryIndex index.htm index.html index.php
ServerAlias contoh.com
ErrorDocument 404 /story.php
ErrorLog /var/log/httpd/contoh.com_error_log
CustomLog /var/log/httpd/contoh.com_access_log combined
</VirtualHost>
```

14. Pengamanan Apache dengan Sertifikat SSL

Apache menggunakan modul mod_ssl untuk mendukung sertifikat SSL.

```
# OpenSSL genrsa -des3 -out contoh.com.key 1024
# OpenSSL req -new -key contoh.com.key -out exmaple.csr
# OpenSSL x509 -req -days 365 -in contoh.com.com.csr -signkey contoh.com.com.key -out contoh.com.com.crt
```

Setelah sertifikat Anda dibuat dan ditandatangani. Sekarang Anda perlu menambahkan ini dalam konfigurasi Apache. Buka file konfigurasi utama dengan editor vim dan tambahkan baris berikut dan restart layanan.

```
<VirtualHost 172.16.25.125:443>
SSLEngine on
SSLCertificateFile /etc/pki/tls/certs/contoh.com.crt
SSLCertificateKeyFile /etc/pki/tls/certs/contoh.com.key
SSLCertificateChainFile /etc/pki/tls/certs/sf_bundle.crt
ServerAdmin ravi.saive@contoh.com
ServerName contoh.com
DocumentRoot /var/www/html/example/
ErrorLog /var/log/httpd/contoh.com-error_log
CustomLog /var/log/httpd/contoh.com-access_log common
</VirtualHost>
```

Dan jika buka browser dan ketik https://contoh.com, maka akan melihat sertifikat yang ditandatangani.

IV.7.3 HARDENING WEB SERVER MICROSOFT IIS 8

HARDENING WEB SERVER

MICROSOFT IIS 8

A. TUJUAN

Memberikan petunjuk/*guidance* dalam melakukan hardening pada web server Microsoft IIS 8

B. RUANG LINGKUP

Tahapan hardening pada web server Microsoft IIS 8

C. REFERENSI

1. Hardening IIS - OWASP

D. LANGKAH - LANGKAH

1. Menonaktifkan Konfigurasi Default Web Server IIS

Cara terbaik untuk menghindari lubang keamanan potensial ini adalah dengan menempatkan semua konten di folder \ wwwroot baru di luar \ inetpub.

a. Membuka IIS Manager

b. Pada the Connections pane expand the Sites node dan pilih Default Web Site

c. Pada Actions pane klik stop pada Manage Aplikasi Berbasis Web

d. Kemudian klik Application Pool list dan pilih “DefaultAppPool”

e. Pada Actions pane klik stop pada Application Pool Tasks

f. Restart IIS.

2. Menonakatifkan Akses Direktori Web Content and Script

Izin berlebihan untuk akun pengguna web yang anonim berkontribusi pada kompromi server web.

a. Browse ke web content pada C:\inetpub\wwwroot\

b. Copy atau cut content ke folder web khusus dan terbatas pada drive non-sistem seperti D:\webroot\

c. Mengubah pemetaan aplikasi atau Direktori Virtual apapun untuk mencerminkan lokasi baru

3. Host Headers

Serangan rebinding DNS dapat membahayakan atau menyalahgunakan data atau fungsi situs.

a. Membuka IIS Manager

b. Pada Connections pane expand the Sites node dan pilih Default Web Site

c. Pada the Actions pane klik Bindings

d. Pada Dialog Box Site Bindings, pilih binding yang header hostnya akan dikonfigurasi, dalam contoh ini port 80

e. Klik Edit

f. Dibawah host name, masukan FQDN Aplikasi Berbasis Web, sebagai contoh <www.examplesite.com>

g. Klik OK, kemudian Close

4. Directory Browsing

Penyerang dapat mengeksploitasi fitur Directory Browsing untuk mengakses file yang tidak sah melalui traversal direktori.

a. Buka IIS Manager
b. In the Connections pane expand the Sites node and select Web Site
c. Click Directory Browsing icon in IIS (Feature View)
d. In the Actions pane click Disable to disable Directory Browsing

5. Application Pool Identity

Membuat identitas khusus untuk setiap kumpulan aplikasi sehingga akan lebih baik melacak masalah yang terjadi dalam setiap situs web.

a. Buka IIS Manager
b. Dalam the Connections pane, expand the Server node dan klik Application Pools
c. Highlight an Application Pool to review and in the Connection pane click Advanced Setting
d. Scroll down to the Process Model section and set the value for Identity to ApplicationPoolIdentity, Network Service or a custom identity with rights and privileges equal to or less than the built-in-security principal.
e. Restart IIS.

6. Application Pools

Menetapkan aplikasi intensif sumber daya ke kumpulan aplikasi mereka sendiri meningkatkan kinerja server dan aplikasi.

a. Buka IIS Manager
b. Buka the Sites node underneath the machine node
c. Pilih the Site to be changed
d. Pada Actions pane, pilih Advanced Settings
e. Klik Select… box next to the Application Pool text box
f. Pilih Application Pool yang diinginkan
g. Kemudian pilih, dan klik OK

7. Memastikan Application Pools di Bawah Identitas yang Unik

Mengatur Application Pools dengan identitas yang unik akan mengurangi potensi kerusakan identitas sehingga aplikasi tidak akan mudah dikompromikan.

a. Buka IIS Manager
b. Buka the Application Pools node underneath the machine node
c. Create new and then select Application Pool that have been created
d. Klik kanan Application Pool dan pilih Advanced Settings.
e. Di bawah bagian Process Model, temukan opsi Identity dan pastikan ApplicationPoolIdentity diatur

8. Identitas Pengguna Anonymous

Mengkonfigurasi identitas pengguna anonim dalam identitas *Application Pools* akan membantu isolasi situs.

a. Buka the IIS Manager GUI dan telusuri server, site, atau application yang diinginkan
b. Pada Features View, double klik icon Authentication
c. Pilih opsi Anonymous Authentication dan pada the Actions pane pilih Edit
d. Pilih Application pool identity in the modal window dan tekan tombol OK

9. Webserver TLS Version

Penyerang dapat mencegat informasi clear text yang sensitif melalui jaringan.

a. Buka IIS Manager
b. Buka the Sites node underneath the machine node
c. Double click the SSL icon
d. Klik the Require SSL and Require SSL 128-Bit check boxes.

Aktifkan protokol TLS 1.2 pada R2, pastikan kunci berikut disetel ke 0.

HKLM/System/CurrentControlSet/Control\SecurityProviders\SCHANNEL\Protocols\TLS1.2\Server\DisabledByDefault

10. Mengaktifkan Pembatasan IP Address dan Domain

Pemfilteran Alamat IP dan Pembatasan Domain memungkinkan administrator untuk mengonfigurasi server dengan memblokir akses admin dan hanya pendaftaran IP.

a. Buka IIS Manager
b. Buka the IP Address dan Domain Restrictions feature
c. Buka Feature atau Double click IP Address dan Setting Domain Restriction

d. Klik Edit and Feature settings in Action Pane

e. Piliht Deny in Access untuk klien yang tidak ditentukan sehingga tidak dapat mengakses panel admin

f. Kemudian klik Add Allow entry for memungkinkan akses untuk alamat IP atau Rentang alamat (Pengguna Internal atau Eksternal)

11. Menghapus entri yang tidak diperlukan dari dokumen default

Jika tidak ada dokumen default di direktori, klien akan menerima kesalahan " file not found" atau " directory browsing denied". Termasuk jika kita menggunakan satu dokumen standar atau menggunakan dokumen pertama dalam daftar, hal ini mempercepat waktu permintaan.

a. Buka IIS Manager

b. Buka the Sites node underneath the machine node

c. Pada Features View, find and double-click the Default Document

d. Klik dokumen yang ingin Anda hapus, dan kemudian klik Remove in Action Pane

e. Klik Add in Action pane and type the name of Default Document in the box that you want to add

f. Kemudian klik OK

Note: Tidak memberikan nama umum (seperti: default.aspx, index.html, dll.) ke halaman awal / beranda situs

12. Enkripsi String Koneksi DB

Menyimpan semua informasi sensitif dalam file teks biasa atau string koneksi DB akan berdampak pada kompromi keamanan.

a. Buka Command Prompt dengan Administrator privileges

b. Pada Command Prompt, masukkan:

- Software aspnet_regiis.exe terletak di% systemroot% \ Microsoft.NET \ Framework \ versionNumberfolder.
- Jika konfigurasi web Anda berada di jalur direktori "/ SampleApplication /", masukkan yang berikut untuk mengenkripsi ConnectionString. Gunakan Aspnet_regiis.exe dengan opsi -pef dan tentukan jalur aplikasi seperti yang ditunjukkan di bawah ini.
  aspnet_regiis -pef "connectionStrings" -app "/SampleApplication/Web.config"
- Cukup melakukan perintah berikut untuk mendekripsi elemen connectionStrings di file Web.config.
  aspnet_regiis -pdf "connectionStrings" -app "/SampleApplication/Web.config"
- Catatan: Parameter "ConnectionStrings" case sensitive

13. Otentikasi Formulir SSL

Menerapkan SSL untuk Formulir Otentikasi akan melindungi kerahasiaan kredensial selama proses login dan membantu mengurangi risiko pencurian informasi pengguna.

a. Buka IIS Manager dan telusuri untuk tier yang sesuai
b. Pada Features View, double-click Authentication
c. Pada Authentication page, pilihForms Authentication
d. Pada Actions pane, klik Edit
e. Cek checkbox dari Requires SSL pada cookie settings section, klik OK

14. Form Authentikasi

Penyerang dapat melakukan hi-jack sesi pengguna untuk mendapatkan akses tidak sah ke aplikasi.

a. Buka IIS Manager dan arahkan ke tingkat di mana Formulir Otentikasi diaktifkan
b. Pada Features View, double-click Authentication
c. Pada Authentication page, pilih Forms Authentication
d. Pada Actions pane, klik Edit
e. Pada bagian Cookie settings, pilih Use cookies dari Mode dropdown

15. Mode Proteksi Cookie

Penyerang dapat melakukan hi-jack cookie untuk mendapatkan akses tidak sah ke aplikasi.

a. Buka IIS Manager dan Forms Authentication is enabled
b. Pada Features View, double-click Authentication
c. Pada Authentication page, pilih Forms Authentication
d. Pada Actions pane, klik Edit
e. Pada Cookie settings section, verify the drop-down for Protection mode is set for Encryption and validation

16. Memastikan Elemen Kredensial Format password Tidak Clear Text

Penyerang dapat mencegat kredensial otentikasi yang dikirim dalam bentuk clear text.

a. Temukan dan buka file konfigurasi tempat kredensial disimpan
b. Temukan elemen <credentials>
c. Jika ada, pastikan kata sandi tidak diposisi clear text
d. Ubah Format password ke SHA1 atau MD5

17. Mengkonfigurasikan SSL untuk Otentikasi Dasar

Kredensial yang dikirim dalam bentuk clear text dapat dengan mudah dicegat oleh kode jahat atau penyerang. Menerapkan Secure Sockets Layer akan membantu mengurangi kemungkinan kredensial yang dibajak.

a. Buka IIS Manager
b. Dalam Connections pane on the left, pilih server yang akan dikonfigurasi
c. Pada the Connections pane, expand the server, then expand Sites dan pilih site yang kan dikonfigurasi
d. Pada the Actions pane, click Bindings; the Site Bindings dialog appears
e. Jika suatu HTTPS binding tersedia, klik Close dan lihat"To require SSL"
f. Jika tidak ada HTTPS binding, lakukan langkah - langkah berikut:
    1. Pada Site Bindings dialog, klik Add; maka akan muncul Add Site Binding dialog
    2. Dibawah Type, pilih https
    3. Dibawah SSL certificate, pilih an SSL certificate
    4. klik OK, dan close
    5. Ubah password Format ke SHA1 atau MD5

18. Mengaktifkan Logging IIS Lanjut

Korelasi log dan penetapan waktu untuk setiap aktivitas berbahaya yang terdeteksi tidak dapat dilakukan secara akurat.

a. Buka Internet Information Services (IIS) Manager
b. Klik the server in the Connections pane
c. Double-click the Advanced Logging icon pada the Home page
d. Klik Enable Advanced Logging in the Actions pane

Catatan: Berikut field yang harus dilog:

a. date
b. time
c. s-ip
d. cs-method
e. cs-uri-stem
f. cs-uri-query
g. s-port
h. c-ip
i. cs(User-Agent)
j. cs(Referer)
k. sc-status
l. sc-bytes

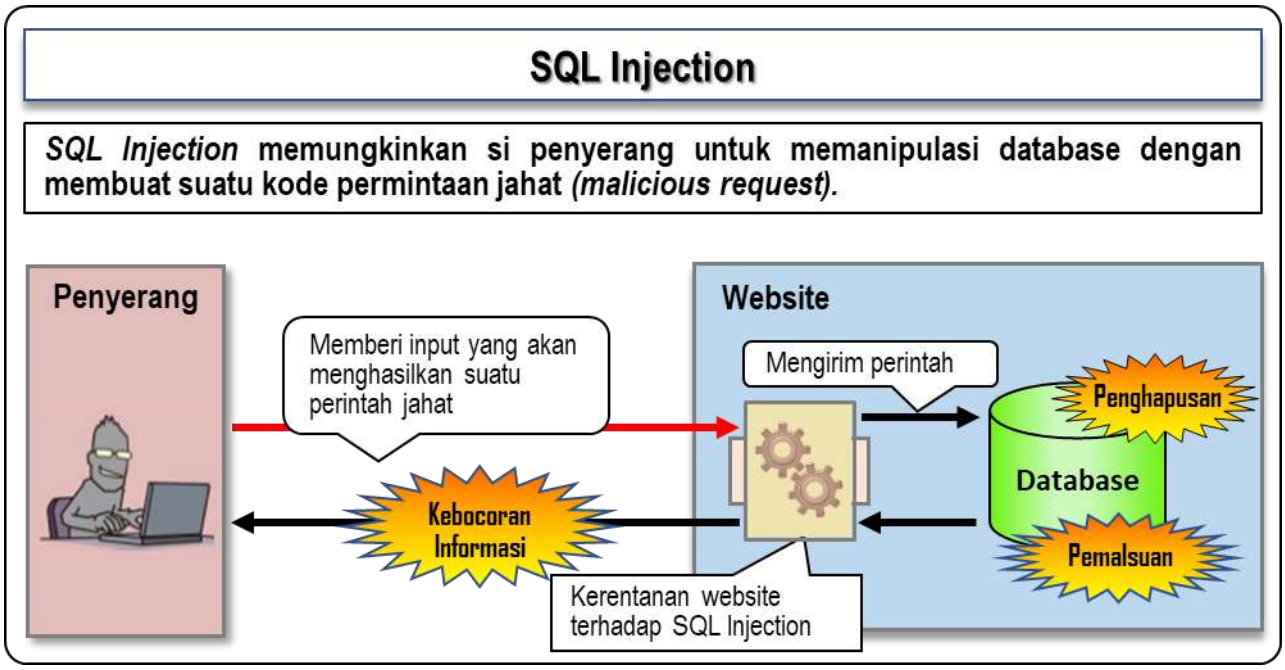
IV.7.4 HARDENING BERBASIS OWASP

HARDENING BERBASIS OWASP

IV.7.4.1 SQL INJECTON

Sebagian besar aplikasi web pasti menggunakan pernyataan SQL untuk akses/operasi database (perintah untuk mengoperasikan basis data) berdasarkan input pengguna. Ini berarti jika proses pembuatan pernyataan SQL tidak dijaga dengan aman, akan sangat dimungkinkan adanya serangan dan manipulasi terhadap basis data. Hal ini biasa disebut dengan " kerentanan SQL Injection "dan metode serangan yang mengeksploitasi kerentanan ini disebut dengan" serangan SQL Injection ".

Gambar 4.15 Visualisasi Proses SQL Injection



IV.7.4.1.1 POTENSI ANCAMAN

Ancaman ini memungkinkan penyerang untuk:

- Melihat bahkan mengeksploitasi data sensitif yang disimpan dalam database (mis. pengungkapan informasi pribadi)
- Memalsukan dan / atau menghapus data yang disimpan dalam database (mis. pemalsuan halaman web, perubahan kata sandi, penutupan sistem)
- Membypass otentikasi login

  Semua operasi yang diizinkan di bawah akun hak istimewa ketika masuk aplikasi bisa jadi menjadi tidak sah
- Menjalankan perintah OS menggunakan prosedur tersimpan (mis: pembajakan sistem, menjadikan PC target sebagai bot (titik peluncuran) untuk menyerang sistem lain)

IV.7.4.1.2 SOLUSI

- Membuat semua pernyataan SQL menggunakan placeholder.

  Biasanya, SQL memiliki mekanisme untuk membangun pernyataan SQL menggunakan placeholder. Ini adalah mekanisme untuk meletakkan simbol (placeholder) di tempat variabel dalam template pernyataan SQL dan mengganti dengan nilai data aktual secara mekanis. Dibandingkan dengan metode di mana aplikasi web secara langsung membuat pernyataan SQL melalui gabungan (concatenation), metode yang menggunakan placeholder dapat menghilangkan kerentanan injeksi SQL.

  Proses mengganti placeholder dengan nilai data aktual disebut mengikat. Ada dua metode mengikat: satu adalah metode dimana pernyataan SQL dikompilasi menjaga placeholder di dalamnya dan mesin database menggantinya dengan nilai data aktual terkait (placeholder statis) dan metode lainnya pustaka koneksi database melakukan pelepasan diri dan menggantikan placeholder dengan nilai data aktual terkait (placeholder dinamis)
- Saat membuat pernyataan SQL melalui gabungan, menggunakan API khusus yang ditawarkan oleh mesin basis data untuk melakukan pelepasan diri dan membuat literal dalam pernyataan SQL dengan benar.

  Saat memasukkan nilai sebagai tipe string, kita akan mengurung nilai dalam satu tanda kutip. Dalam hal ini, kita harus melakukan pelepasan diri untuk string literal untuk membersihkan karakter khusus (mis. 'ke' 'dan \ ke \\). Saat memasukkan nilai sebagai tipe numerik, membuatnya diproses sebagai numerik literal (misalnya memasukkannya ke dalam tipe numerik).
- Tidak menulis pernyataan SQL secara langsung pada parameter yang akan diteruskan ke aplikasi web.
  - Menentukan pernyataan SQL dalam parameter aplikasi web secara langsung dapat menyebabkan risiko seseorang memalsukan nilai parameter dan memanipulasi basis data.
- Memberikan hak minimum untuk akun basis data.
  - Jika hak istimewa akun database yang digunakan aplikasi web dalam mengakses database lebih tinggi dari yang diperlukan, maka kerusakan serangan yang ditimbulkan bisa menjadi lebih serius.
  - Periksa perintah dalam berinteraksi dengan database dan memberikan hak akses minimum kepada akun tersebut, cukup untuk menjalankan perintah tersebut
- Membatasi informasi yang ditampilkan dalam pesan kesalahan di web browser.
  - Jika pesan kesalahan berisi informasi tentang nama mesin database atau pernyataan SQL yang dimiliki menyebabkan kesalahan, maka pengguna jahat bisa mendapatkan informasi yang berguna untuk menyerang situs web.
  - Kesalahan pesan dapat dimanfaatkan tidak hanya untuk memberikan tips untuk menyerang tetapi juga untuk menunjukkan hasil serangan. Disarankan untuk tidak menampilkan pesan kesalahan yang terkait dengan operasi database di browser web pengguna.

IV.7.4.1.3 CONTOH

PHP & PostgreSQL

Kondisi Rentan

$ query = "SELECT * FROM usr WHERE uid = ' **$ uid** ' AND pass = ' **$ passh** ';

$ result = pg_query ($ conn, $ query);

Di atas adalah bagian dari kode sumber yang mengimplementasikan otentikasi pengguna. $ uid di baris pertama adalah ID pengguna yang disediakan oleh pengguna. $ passh adalah nilai hash web aplikasi menghitung berdasarkan kata sandi yang dimasukkan pengguna. Di baris pertama, aplikasi web menggunakan variabel-variabel ini untuk menyusun pernyataan SQL dan menetapkannya ke $ query. Fungsi pg_query () di baris kedua adalah fungsi PostgreSQL yang disediakan oleh PHP dan mengeksekusi $ query, sebagai pernyataan SQL yang diatur di baris pertama. Contoh program ini tidak memiliki proses untuk melarikan diri terkait nilai $ uid, yang memungkinkan penyerang melakukan serangan injeksi SQL dengan memasukkan nilai yang dibuat khusus dengan pernyataan SQL yang berbahaya.

Seperti dalam kasus ini, jika aplikasi web tidak melakukan pelolosan untuk nilai-nilai yang dilewatkan oleh eksternal parameter, dapat menyebabkan eksekusi pernyataan SQL yang tidak terduga.

Sebagai contoh, anggaplah pengguna memasukkan "talas '-" sebagai ID pengguna, pernyataan SQL dikirim ke database akan menjadi sebagai berikut:

SELECT FROM * FROM WHERE uid = ' **talas' --** 'AND pass =' eefd5bc2 ... '

Kutipan tunggal (') yang digunakan dalam pernyataan SQL di atas adalah karakter khusus, yang mendefinisikan string literal dengan melampirkan string data dalam sepasang tanda kutip tunggal. Demikian juga, dua tanda hubung berturut-turut () adalah karakter khusus yang memberitahu database untuk mengabaikan semua yang muncul setelahnya sebagai komentar.

Yang berarti basis data akan mengabaikan ['AND pass = eefd5bc2 ..] ketika nilai [talas' -] ditetapkan dalam $ uid.

Akibatnya, pernyataan SQL yang dikirim dan dieksekusi oleh database akan menjadi seperti ini

SELECT * FROM usr WHERE uid = " **talas' --**

Artinya, jika akun pengguna "talas" memang ada dalam database, penyerang bisa masuk tanpa mengetahui kata sandi yang berhubungan dengan talas. Selain itu, tidak hanya melewati otentikasi pengguna tetapi juga dapat memanipulasi basis data secarar yang tidak diinginkan dengan hanya mengubah string untuk memberi input ke $ uid. Hal ini disebabkan karena tidak ada proses melarikan diri untuk nilai elemen yang menyusun pernyataan SQL.

Fungsi pg_query () mampu mengeksekusi beberapa query SQL. Jika fungsi ini rentan untuk injeksi SQL, penyerang bisa memasukkan lebih banyak permintaan selain yang asli, yang akan meningkatkan ancaman.

Di bawah ini adalah contoh yang menggambarkan masalah ini:

```
// Set two SQL queries in $query
$query = "SELECT item FROM shop WHERE id = 1;
SELECT item FROM shop WHERE id = 2;"
$result = pg_query($conn, $query);
```

Tindakan Korektif #1

Menggunakan fungsi pg_prepare () atau fungsi pg_execute () daripada menggunakan fungsi pg_query ().

```
$result = ($conn, "query", 'SELECT * FROM usr WHERE uid= $1 AND pass=$2);
$result = ($conn, "query", array($uid, $passh));
```

Fungsi pg_prepare () dan fungsi pg_execute () adalah fungsi PostgreSQL yang disediakan di PHP 5.1.0 dan yang lebih baru dan hanya didukung oleh PostgreSQL 7.4 dan yang lebih baru.

Fungsi pg_prepare () menghasilkan pernyataan yang disiapkan. Argumen ketiga adalah pernyataan SQL dimana variabel dirujuk menggunakan placeholder (variabel terikat) $ 1, $ 2 ... tanpa nilai aktual.

Fungsi pg_execute () mengeksekusi pernyataan yang disiapkan oleh fungsi pg_prepare () dibuat. Ketika placeholder digunakan dalam pernyataan yang disiapkan, fungsi pg_execute () dikonversi setiap elemen dari argumen ketiga ($ uid dan $ passh dalam kasus ini) menjadi string dan mengaturnya dalam placeholder yang sesuai (disebut "mengikat") dan mengeksekusi pernyataan SQL yang lengkap. Penggunaan placeholder akan menyelamatkan kita dalam melakukan pelepasan diri secara eksplisit.

Tindakan Korektif #2

Menggunakan fungsi pg_query_params() daripada menggunakan fungsi pg_query ().

```
$result = ($conn, 'SELECT * FROM usr WHERE uid = $1
                AND pass = $2', array($uid, $passh));
```

Fungsi pg_query_params () adalah fungsi untuk PostgreSQL yang disediakan oleh PHP 5.1.0 atau yang lebih baru dan hanya didukung oleh PostgreSQL versi 7.4 atau yang lebih baru. Fungsi pg_query_params () tidak membuat pernyataan etapi dilengkapi dengan kemampuan placeholder. Dibutuhkan pernyataan SQL di mana placeholder ($ 1, $ 2, ...) digunakan sebagai argumen kedua dan nilai aktual untuk placeholder sebagai argumen ketiga. Penggunaan placeholder akan menyelamatkan kita dalam melakukan pelepasan diri secara eksplisit.

Tindakan Korektif #3

Menggunakan fungsi pg_escape_string() dan melakukan pelepasan diri (*escaping*) untuk semua elemen dalam pernyataan SQL yang dieksekusi melalui fungsi pg_query ().

```
$query = "SELECT * FROM usr WHERE uid = '".($uid)."'
AND pass = '".($passh)."'";
$result = pg_query($conn, $query);
```

Fungsi pg_escape_string () adalah fungsi PostgreSQL yang tersedia dalam PHP 4.2.0 atau yang lebih baru dan hanya didukung oleh PostgreSQL 7.2 atau yang lebih baru. Hal ini akan meloloskan diri dari karakter khusus yang ditunjuk di PostgreSQL.
Penggunaan pg_escape_string () akan melakukan pelepasan diri secara otomatis. Dalam kode di atas, $ passh melewati proses pelolosan. $ passh adalah nilai hash yang dihitung dari kata sandi dan tidak mungkin dieksploitasi dalam upaya injeksi SQL. Namun demikian, direkomendasikan untuk melakukan pelepasan diri untuk elemen-elemen yang diproses secara internal ini termasuk $ passh.

PHP & MySQL

Kondisi Rentan

```
$query = "SELECT * FROM usr WHERE uid = " AND pass = "";
$result = mysql_query($query);
```

Ini adalah bagian dari kode sumber yang mengimplementasikan otentikasi pengguna. Program ini juga tidak memiliki proses melepaskan diri untuk nilai input $ uid, yang memungkinkan penyerang untuk memulai serangan injeksi SQL dengan memasukkan nilai yang dibuat khusus yang akan berubah menjadi pernyataan SQL yang berbahaya.

Tindakan Korektif #1

Menggunakan fungsi mysqli, seperti mysqli_prepare (), mysqli_stmt_bind_param () dan mysqli_stmt_execute () daripada menggunakan mysql_query.

Fungsi mysqli_prepare (), mysqli_stmt_bind_param () dan mysqli_stmt_execute () adalah fungsi MySQL yang disediakan dalam ekstensi PHP mysqli dan hanya didukung oleh MySQL 4.1.3 atau yang lebih baru.

Fungsi mysqli_prepare () menghasilkan pernyataan yang disiapkan. Argumen kedua adalah pernyataan siap di mana pernyataan SQL diekspresikan menggunakan placeholder "?" tanpa nilai aktual.

Fungsi mysqli_stmt_bind_param () mengikat nilai data aktual (nilai bind) ke placeholder dalam pernyataan yang disiapkan yang dibuat oleh fungsi mysqli_prepare ().

Ketiga dan argumen selanjutnya ($ uid dan $ passh dalam kasus ini) adalah nilai ikat. Argumen kedua "ss" menunjukkan jenis nilai bind (untuk string).

Karena kedua elemen, $ uid dan $ passh, adalah tipe "string", dengan mengatur dua ss

Fungsi mysqli_stmt_execute () mengeksekusi pernyataan yang telah disiapkan

// Create a prepared statement

$stmt = ($conn, "SELECT * FROM usr WHERE uid= ? AND pass = ?");

// Bind $uid and $passh to the SQL statement (corresponding placeholders)

($stmt, "ss", $uid, $passh);

// Execute the SQL statement

($stmt);

Tindakan Korektif #2

Menggunakan fungsi mysql_real_escape_string () untuk melakukan pelepasan diri ke semua elemen yang membentuk pernyataan SQL yang akan dieksekusi oleh fungsi mysql_query ().

Fungsi mysql_real_escape_string () adalah fungsi MySQL yang disediakan oleh PHP 4.3.0 atau yang lebih baru.

$query = "SELECT * FROM usr WHERE uid = '".

($uid)."' AND pass = '".

($passh)."'";

$result = mysql_query($query);
$query = "SELECT * FROM usr WHERE uid = '$uid' AND pass = '$passh'";
$sth =$dbh->prepare($query);
$sth->execute();

PERL

Kondisi Rentan

```
$query = "SELECT * FROM usr WHERE uid = '$uid' AND pass = '$passh'";

$sth =$dbh->prepare($query);

$sth->execute();
```

Ini adalah bagian dari kode sumber yang mengimplementasikan otentikasi pengguna. Contoh ini menggunakan database dengan standar modul antarmuka yang biasa disebut dengan DBI.

Program sampel ini tidak memiliki proses melepaskan diri untuk nilai input untuk $ uid dan memungkinkan penyerang untuk melakukan serangan injeksi SQL dengan memasukkan nilai yang dibuat khusus yang dapat diubah menjadi pernyataan SQL berbahaya, yang biasa digunakan pada Perl.

Sampel ini menunjukkan kesalahan pengkodean yang umum namun berbahaya saat menggunakan Perl DBI.

Metode persiapan () dalam modul DBI menghasilkan pernyataan yang disiapkan dan tidak mendukung placeholder. Demikian juga, metode execute () mengeksekusi pernyataan yang disiapkan yang dibuat oleh menyiapkan () metode dan juga mampu mengikat jika pernyataan yang disiapkan berisi placeholder.

Hal yang rentan dalam contoh program ini adalah bahwa program ini tidak menggunakan placeholder atau melakukan pelepasan diri meskipun pernyataan SQL yang dikomposisikan berisi variabel yang dapat dieksploitasi, yang membuat aplikasi ini rentan terhadap serangan injeksi SQL

Tindakan Korektif #1

Saat membuat pernyataan SQL dalam metode prep () pada modul DBI, harus menggunakan placeholder "?" di tempat variabel. Kemudian, menentukan nilai mengikat yang akan ditetapkan ke placeholder dalam metode mengeksekusi ()

```
$sth =$dbh->prepare("SELECT * FROM usr WHERE uid = AND pass = ");

$sth->execute($uid, $passh);
```

Tindakan Korektif #2

Menggunakan metode quote () dalam modul DBI dan melakukan pelepasan diri untuk variabel.
Metode quote () akan mengambil string yang ditentukan dalam argumennya, keluar dari karakter khusus dalam string dan mengembalikan output setelah melampirkannya dengan tanda kutip ganda.

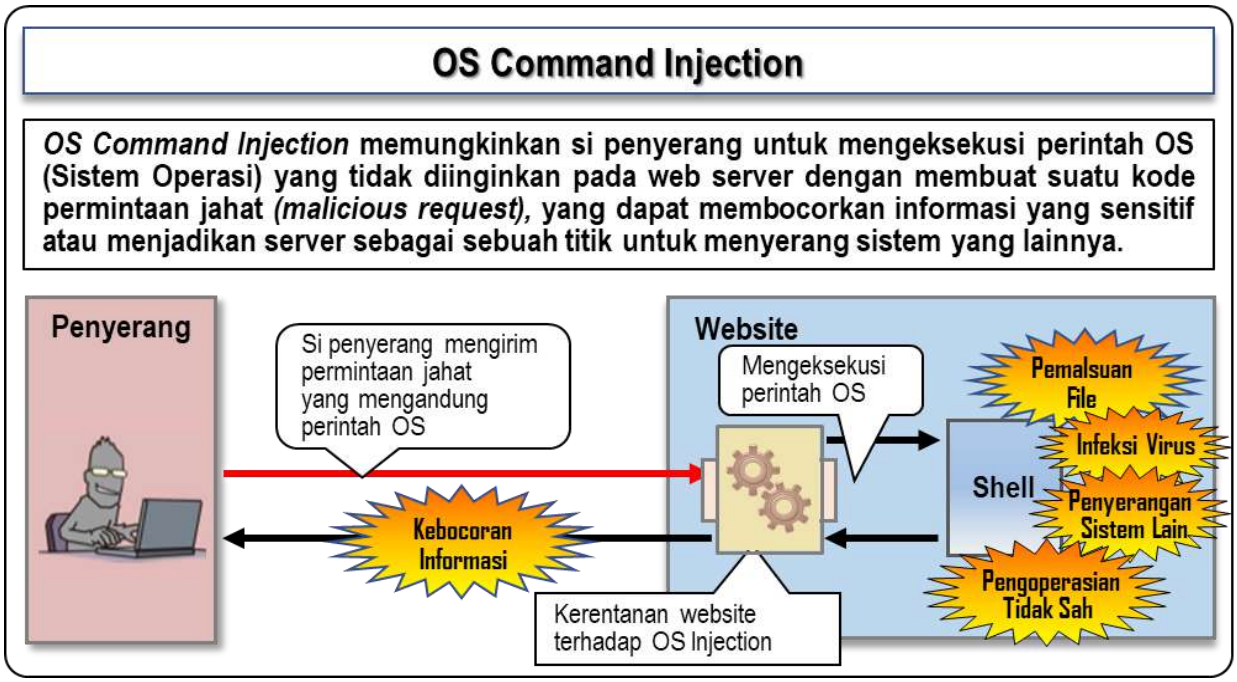
Apa yang dikenali sebagai karakter khusus berbeda dari mesin database ke mesin database dan itu adalah masalah yang harus ditangani ketika melakukan melarikan diri. DBI menyediakan satu set driver, yang disebut DBD (Database Drivers) untuk beradaptasi dengan berbagai mesin basis data. Metode quote () di DBI memungkinkan DBD menangani perbedaan mesin basis data dan menawarkan transparansi pengguna untuk masalah ini.

```
$sth = $dbh->prepare("SELECT * FROM usr
WHERE uid =".$dbh->($uid)." AND
pass =".$dbh->($passh));
$sth->execute();
```

IV.7.4.2 OS COMMAND INJECTON

Penyerang memungkinkan secara remote menjalankan perintah level OS melalui aplikasi tersebut. Masalah ini disebut "kerentanan Perintah Injeksi OS" dan metode menyerang yang mengeksploitasi kerentanan ini disebut "serangan Perintah OS". Injeksi perintah OS memungkinkan penyerang untuk mengeksekusi perintah OS yang tidak diinginkan di server web dengan permintaan yang dibuat dengan niat jahat, sehingga dapat menyebabkan bocornya informasi sensitif atau mengubah server menjadi bot (titik peluncuran) untuk menyerang orang lain.

Gambar 4.16 Visualisasi Proses OS Command Injection



IV.7.4.2.1 POTENSI ANCAMAN

Ancaman ini memungkinkan penyerang untuk:

- Melihat, memalsukan, dan menghapus file yang disimpan di server (mis: pengungkapan informasi sensitif, pemalsuan file konfigurasi)
- Memanipulasi sistem dengan tujuan jahat (mis. Mematikan OS yang tidak diinginkan, menambah / menghapus akun pengguna)
- Mengunduh dan menjalankan program jahat (mis. virus, infeksi worm dan bot, implementasi backdoor)
- Menjadikan sistem sebagai titik peluncuran untuk menyerang sistem lain (serangan Denial of Service, pengintaian dan spamming)

IV.7.4.2.2 SOLUSI

- Beberapa bahasa pemrograman yang digunakan untuk menulis aplikasi web memiliki fungsi yang dapat memanggil perintah - perintah shell, seperti fungsi open () dan fungsi – fungsi lain (Perl: open (), system (), eval () PHP: exec (), passthru (), shell_exec (), system (), popen ()). Fungsi open () misalnya akan mengambil nama file sebagai argumen dan memasukkan tanda "| (pipa) " maka akan memanggil dan menjalankan perintah OS. Itu artinya berbahaya karena memungkinkan input eksternal digunakan sebagai argumennya.
- Hindari menggunakan fungsi yang bisa memanggil perintah shell.
- Saat menggunakan fungsi yang dapat memanggil perintah shell, periksa semua variabel yang membentuk parameter shell dan memastikan yang mengeksekusi hanya mereka yang diberikan untuk dieksekusi.

IV.7.4.2.3 CONTOH

Program Perl yang menjalankan perintah sendmail

Kondisi Rentan

```
$from =~ s/"|;|'|<|>|\|| //ig;
open(MAIL, "|/usr/sbin/sendmail -t -i -f $from");
```

Di atas adalah bagian dari program yang mengirim email dengan alamat email yang dimasukkan oleh formulir web pengguna sebagai pengirim.

Alamat email input disimpan dalam variabel $ from. Baris pertama menghilangkan shell khusus karakter ",;, ', <,>, | dan spasi dari isi $. Baris kedua memanggil OS perintah sendmail untuk memulai proses pengiriman surat dan meneruskan konten $ dari ke baris perintah pilihan.

Meskipun ada sanitasi pada baris pertama, implementasi ini masih rentan terhadap perintah injeksi OS.

Dalam implementasi ini, jika nilai $ from adalah someone@example.jp, perintah berikut adalah dieksekusi:

/usr/sbin/sendmail -t -i -f someone@example.jp

Namun, jika nilai $ from dibuat dengan maksud jahat dan `touch [0x09] / tmp / foo` (di mana [0x09] berarti tabulasi horisontal) dimasukkan, perintah berikut akan dieksekusi dan perintah injeksi OS bisa berhasil dilakukan.

/usr/sbin/sendmail -t -i -f `touch[0x09]/tmp/foo`

Kutipan belakang (`) adalah karakter meta shell yang mengeksekusi perintah jual yang diletakkan di antara bagian belakang mengutip dan mengembalikan output perintah ke baris perintah. Dalam program sampel, kutipan ganda dan kutipan tunggal disanitasi tetapi kutipan belakang tidak tersentuh. Kelalaian ini mengakibatkan memungkinkan penyerang untuk mengeksekusi perintah yang tidak diinginkan.

Selain itu, menghapus ruang di baris pertama dari program sampel dapat memberikan arti yang salah yang akan memberikan jaminan bahwa penyerang tidak dapat secara bebas menentukan opsi baris perintah bahkan jika dia bisa mengeksekusi perintah. Menggunakan tabulasi horizontal [0x09] seperti di atas memungkinkan penyerang untuk menentukan opsi baris perintah. Di sini, tabulasi horizontal berfungsi sebagai pemisah karakter.

Tindakan Korektif #1

Menggunakan Library

Dengan menghentikan menjalankan perintah OS. Fungsionalitas yang saat ini diaktifkan dengan menjalankan perintah OS dapat dilakukan melalui penggunaan library yang ada.

```
use Mail::Sendmail;
%mail = (From => $from, …);
sendmail(%mail);
```

Program sampel ini adalah program untuk mengirim email, dengan menggunakan pengirim surat MAIL : : Sendmail.

Tindakan Korektif #2

Tidak menempatkan nilai parameter di baris perintah

Jika pustaka yang dapat diganti tidak tersedia dan Anda tidak bisa berhenti menggunakan perintah, masih ada kemungkinan Anda dapat menghapus kerentanan perintah injeksi OS dengan mengubah cara melakukan perintah.

Dalam contoh program ini, alamat email pengirim menggunakan opsi baris perintah yang menyebabkan kerentanan.

Dengan cara ini, nilai $ from tidak digunakan dalam baris perintah dan karenanya akan menghilangkan kerentanan karena perintah injeksi OS.

```
$from =~ s/\r|\n//ig;
open(MAIL, '|/usr/sbin/sendmail -t -i');
…
print MAIL "From: $from\n";
```

Tindakan Korektif #3

Jika pustaka yang dapat diganti tidak tersedia dan tidak bisa berhenti menggunakan perintah, masih ada kemungkinan dapat menghapus kerentanan perintah injeksi OS dengan menjalankan perintah tanpa mengakses shell

```
open(MAIL, '|-') || exec '/usr/sbin/sendmail', '-t', '-i', '-f', '$from';
```

IV.7.4.3 UNCHECKED PATH PARAMETER/DIRECTORY TRAVERSAL

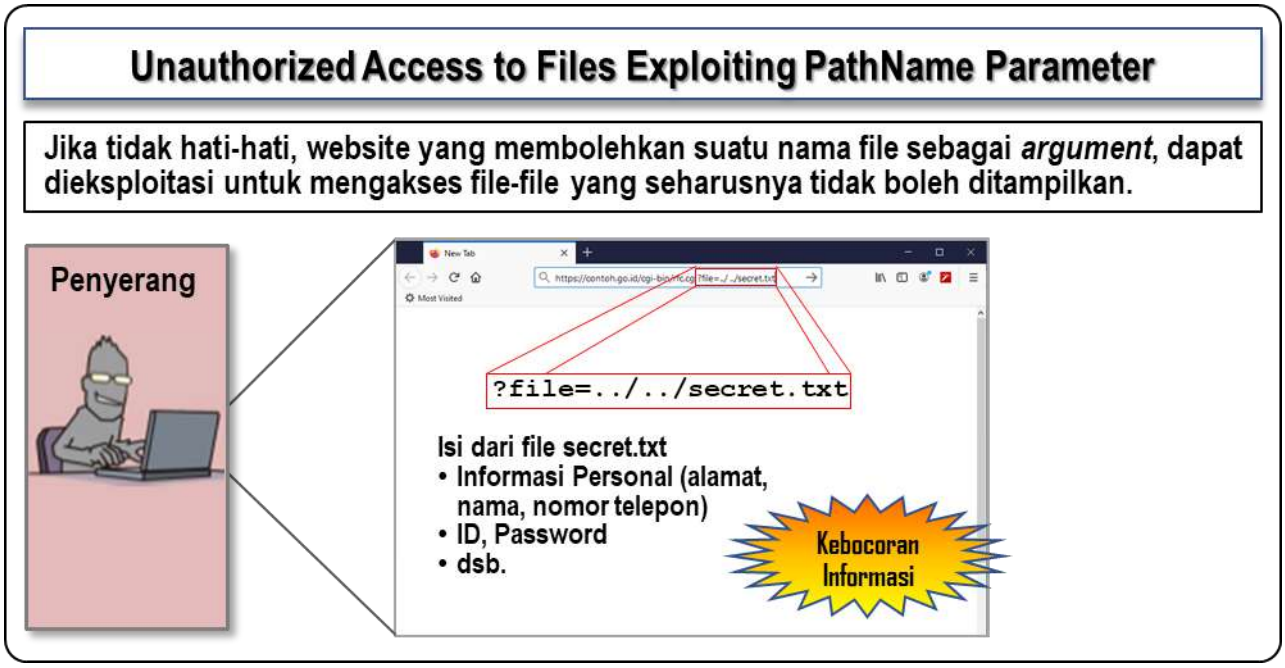
Beberapa aplikasi web memungkinkan untuk menentukan nama file yang disimpan di server web secara langsung menggunakan parameter eksternal. Jika aplikasi web semacam itu tidak diprogram dengan hati-hati, penyerang dapat menentukan file yang yang tidak diinginkan dan meminta aplikasi web menjalankan operasi yang tidak diinginkan. Masalah ini disebut " Kerentanan Direktori Traversal"dan metode penyerangan yang mengeksploitasi kerentanan ini disebut" Serangan Direktori Traversal".

IV.7.4.3.1 POTENSI ANCAMAN

Ancaman ini memungkinkan penyerang untuk:

- Melihat, memalsukan, dan menghapus file yang disimpan di server berupa:
  - Pengungkapan informasi sensitif
  - Pemalsuan dan penghapusan file konfigurasi, file data, dan kode sumber

Gambar 4.17 Visualisasi Proses Unauthorized Access to File



IV.7.4.3.2 SOLUSI

- Tidak menggunakan nama file yang disimpan di server web secara langsung menggunakan parameter eksternal

  Ketika aplikasi web dengan nama filenya ditentukan secara langsung menggunakan parameter eksternal, penyerang dapat memanipulasi parameter yang menentukan file secara yang tidak diinginkan dan melihat konten file yang seharusnya tidak diungkapkan. Misalnya, dalam kasus implementasi di mana nama file disimpan di web server ditentukan dalam parameter tersembunyi dan file itu digunakan dalam templat halaman web, seorang penyerang bisa output file yang tidak diinginkan sebagai halaman web dengan memanipulasi parameter.

  Dianjurkan untuk meninjau desain dan spesifikasi aplikasi, mempertimbangkan kembali apakah itu memang diperlukan untuk memungkinkan menentukan nama file yang disimpan di server web dalam parameter eksternal dan metode alternatif tersedia.

- Menggunakan direktori tetap untuk menangani nama file dan membatalkan nama direktori yang didalamnya nama file.

  Misalkan kita membuka file bernama "nama file" di direktori saat ini dan jika file itu terbuka fungsi diimplementasikan seperti open (nama file), penyerang dapat mengakses file yang tidak diinginkan dengan menentukan jalur absolut ke file. Untuk mencegah penggunaan jalur absolut, Kita bisa menggunakan direktori tetap, seperti “Dirname”, dan kode seperti open (dirname + nama file).

  Namun, hanya melakukan itu masih menyisakan kamar untuk serangan direktori traversal menggunakan "../". Untuk mencegahnya, kita dapat menggunakan API, seperti basename (), yang mengekstrak hanya nama file dan menghapus nama direktori dari jalur yang diberikan seperti berikut: open (dirname + basename (filename)

- Mengelola izin akses file dengan benar.

  Jika izin akses ke file di server web diterapkan dan dikelola dengan benar, server web mungkin dapat mencegah upaya serangan ketika aplikasi web mencoba untuk membuka file di direktori tersebut secara yang tidak diinginkan.

- Memeriksa nama file.

  Ketika nama file berisi string karakter yang digunakan untuk menentukan direktori arbitrer, seperti "/", "../" dan ".. \", batalkan prosesnya. Perhatikan bahwa jika Anda menggunakan pengodean dan dekode URL, Nilai-nilai yang disandikan URL seperti "% 2F", "..% 2F" dan "..% 5C" atau nilai ganda yang disandikan seperti "% 252F", "..% 252F" dan "..% 255C" dapat diartikan sebagai nilai input yang valid untuk nama file.

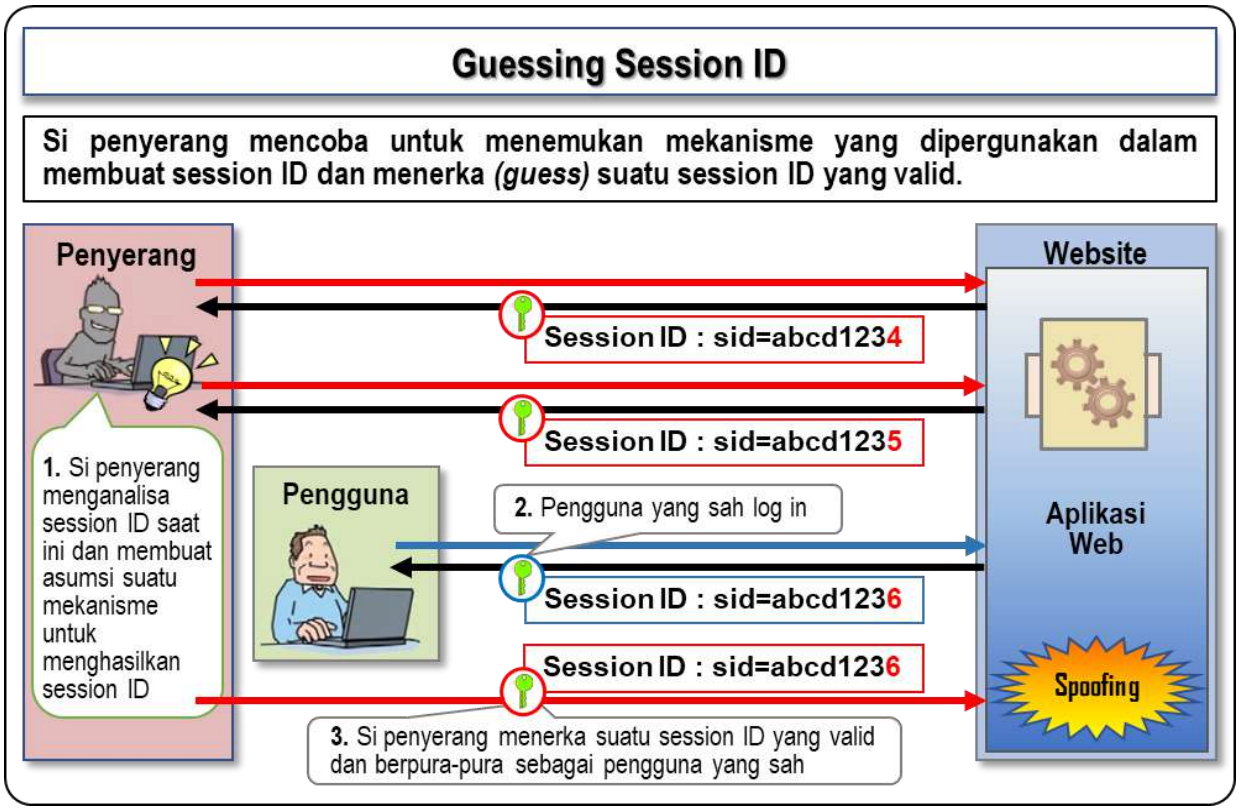
IV.7.4.4 MANAJEMEN SESI YANG TIDAK BENAR

Dalam pengelolaan sesi beberapa aplikasi web mengeluarkan ID sesi, yang merupakan informasi untuk mengidentifikasi pengguna.

Jika ID sesi tidak dikelola dengan benar, penyerang dapat mencuri ID sesi dari pengguna yang sah dan mendapatkan akses tidak sah ke layanan seolah - olah menjadi pengguna yang sah.

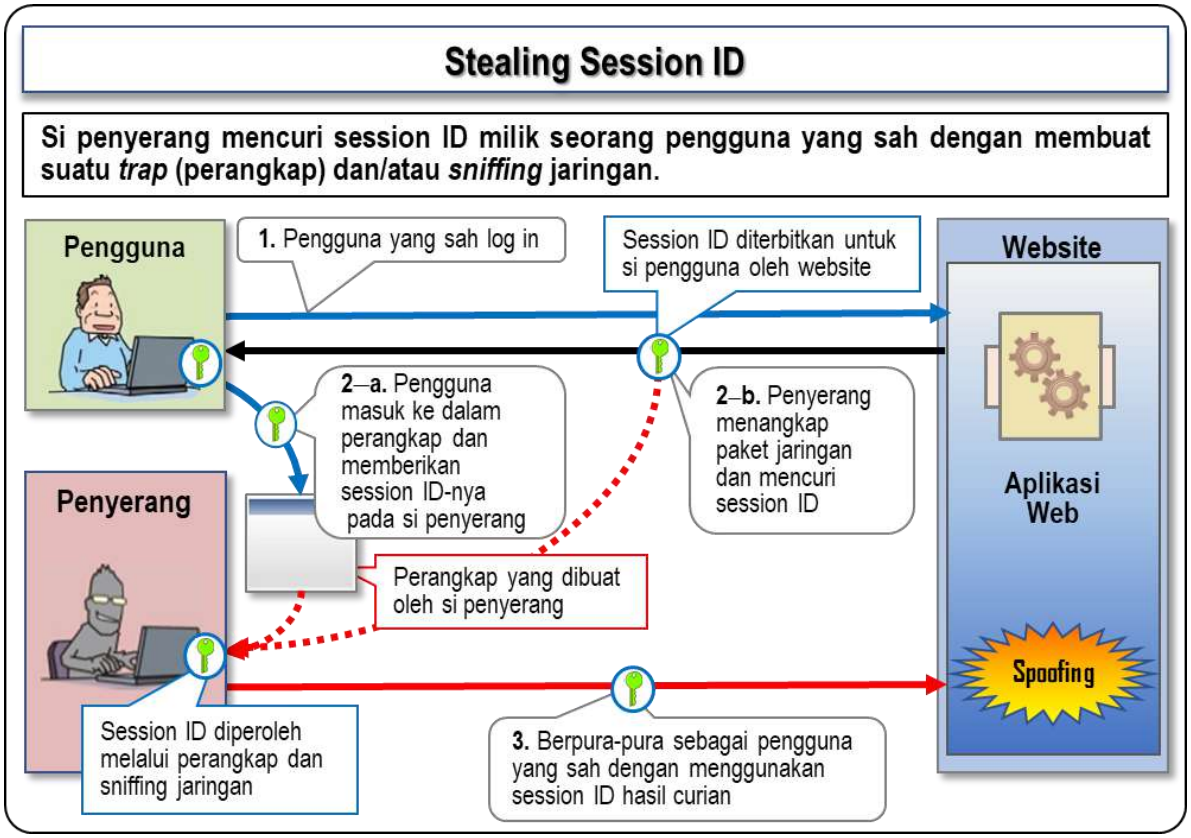
Metode penyerangan dengan mengeksploitasi kerentanan ini dalam manajemen sesi disebut "Session Hijacking".

Gambar 4.18 Visualisasi Proses Guessing Session ID

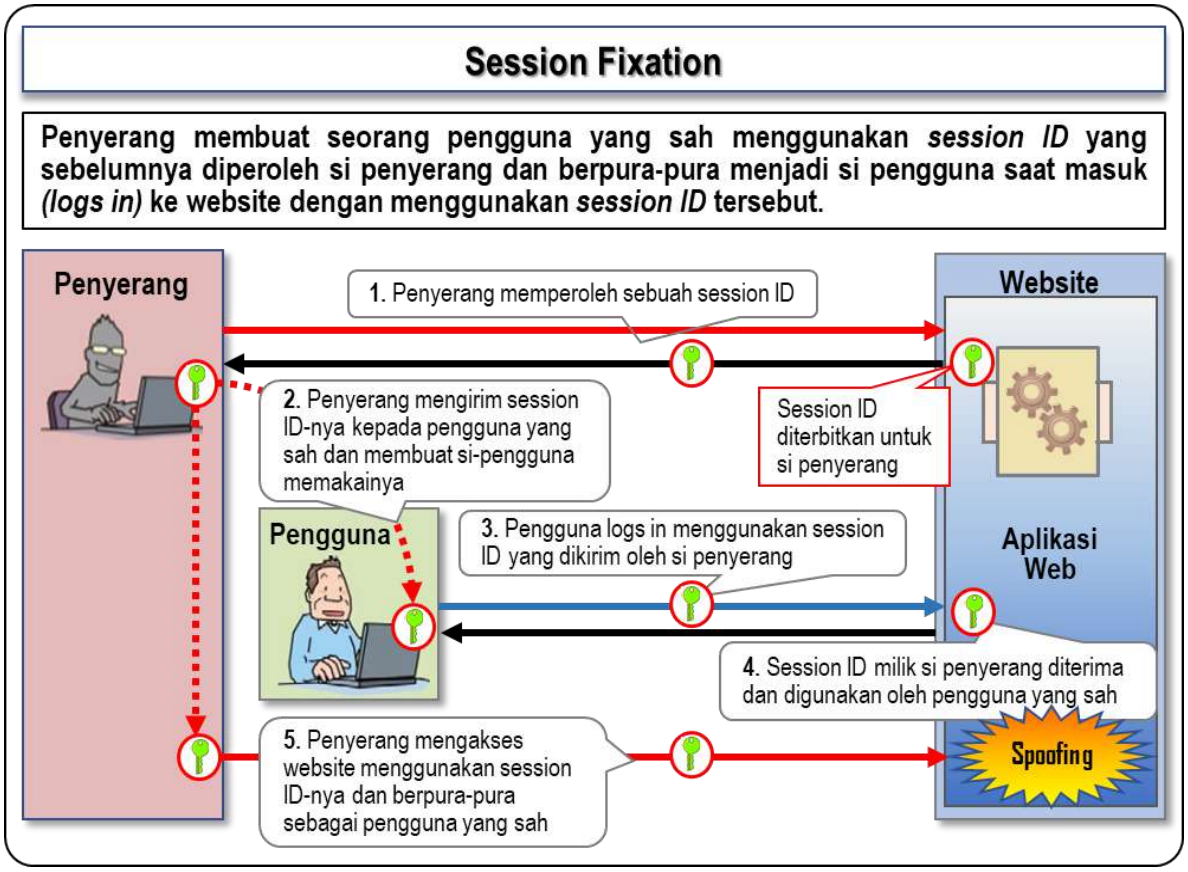


Selain menebak atau mencuri ID sesi, ada serangan lain yang mengeksploitasi sesi yang tidak patut yaitu "Sesi Fixation". Itu terjadi ketika penyerang menyiapkan ID sesi dan memiliki target pengguna menggunakan ID sesi dalam beberapa cara an pengguna target yang tidak menyadarinya login ke situs web. Jika berhasil, penyerang bisa seolah - olah menjadi pengguna yang ditargetkan menggunakan ID sesi yang telah ditetapkan oleh penyerang

Gambar 4.19 Visualisasi Proses Stealing Session ID



Gambar 4.20 Visualisasi Proses Session Fixation



IV.7.4.4.1 POTENSI ANCAMAN

Jika serangan yang mengeksploitasi manajemen sesi dimana penyerang seolah - olah menjadi pengguna yang sah dan melakukan operasi yang diizinkan untuk pengguna tersebut. Penyerang membuat pengguna yang sah menggunakan ID sesi yang telah ditentukan oleh penyerang

Hal ini dapat memungkinkan adanya:

- Akses layanan yang biasanya hanya tersedia untuk pengguna yang sah (misal: pengiriman uang tanpa izin, membeli barang yang tidak diinginkan, membatalkan keanggotaan bertentangan dengan keinginan pengguna)
- Menambah dan mengubah informasi yang biasanya diizinkan hanya untuk pengguna yang memilikinya masuk dengan benar (misal. perubahan pengaturan aplikasi yang tidak sah (kata sandi, fungsi administrator, dll.),
- Melihat informasi yang biasanya tersedia hanya untuk pengguna yang sah (misal: akses tanpa izin ke informasi pribadi, webmail, papan buletin khusus anggota)

IV.7.4.4.2 SOLUSI

- Membuat ID sesi yang sulit ditebak

  Menggunakan mekanisme yang aman, seperti generator nomor acak yang akan mengeluarkan ID rahasia dan terenkripsi.

- Tidak menggunakan parameter URL untuk menyimpan ID sesi.

  Jika ID sesi diatur dalam parameter URL, browser pengguna akan meneruskan URL ID yang didalamnya tertanam sesi ke situs web berikutnya. Jika penyerang mencegatnya, ia akan memungkinkan untuk membajak sesi. Simpan ID sesi dalam cookie atau parameter tersembunyi menggunakan metode POST untuk meneruskannya.

  Beberapa server aplikasi web dapat secara otomatis beralih untuk menggunakan parameter URL ketika browser pengguna diatur untuk menolak cookie. Jika demikian, ubah pengaturan server dan coba nonaktifkan fitur ini.

- Mengatur atribut aman cookie ketika menggunakan HTTPS.

  Cookie memiliki atribut aman yang memungkinkan cookie dikirim hanya melalui saluran HTTPS. Pastikan untuk mengatur atribut aman. Tetapi jika menggunakan cookie dalam komunikasi HTTP buat cookie baru, terpisah dari yang digunakan dalam komunikasi HTTPS.

- Memulai sesi baru setelah berhasil melakukan login.

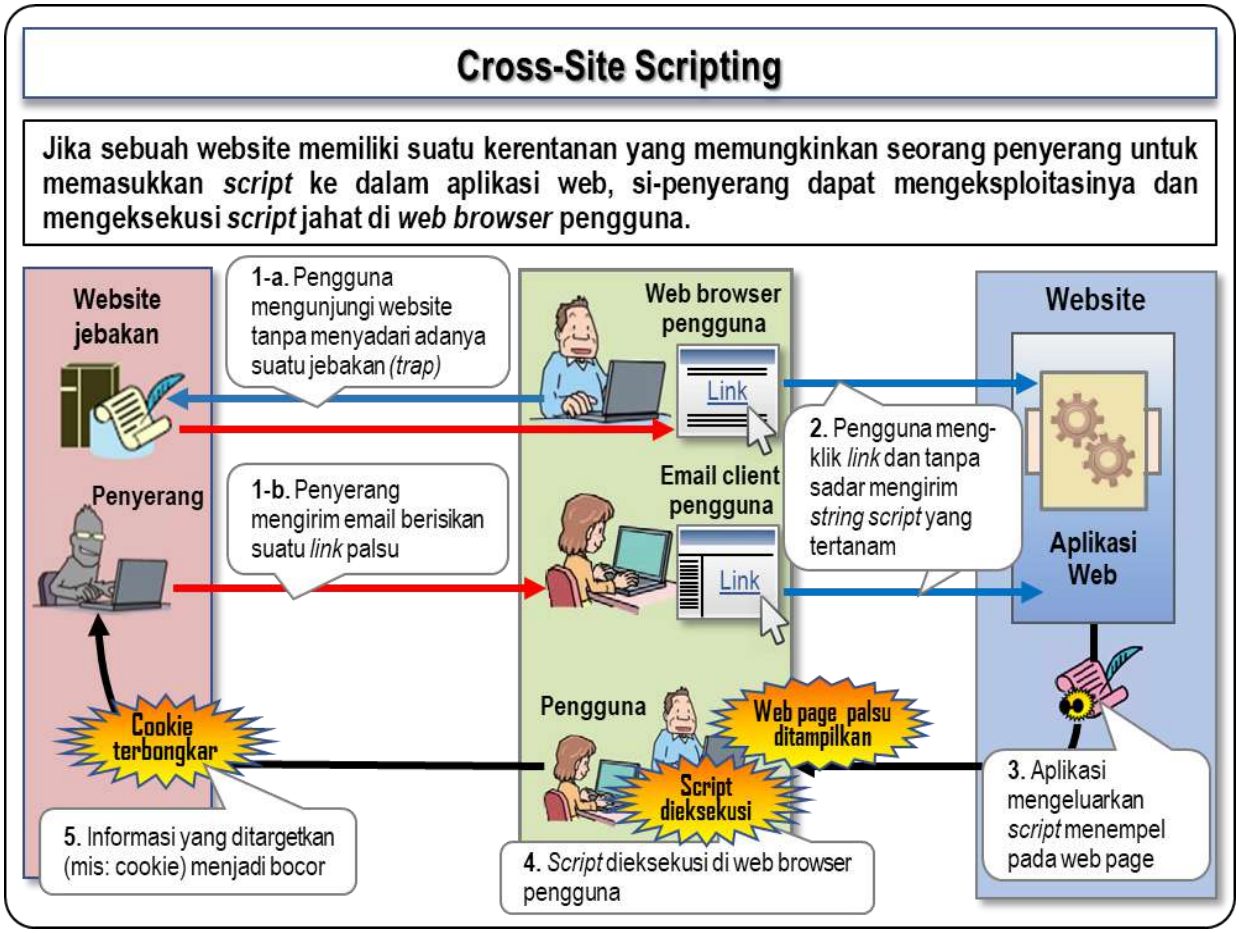
  Beberapa aplikasi web memulai sesi mengeluarkan ID sesi sebelum pengguna masuk, mungkin ketika pengguna pertama kali mengakses situs web, dan terus menggunakan sesi yang sama. Metode ini, bagaimanapun, rentan terhadap fiksasi sesi. Anda harus menghindarinya dan lebih baik memulai sesi baru setelah pengguna berhasil login in (kelola sesi dengan ID sesi baru). Menonaktifkan ID sesi lama saat menggantinya dengan ID sesi baru.

- Menetapkan tanggal kadaluwarsa cookie saat menyimpan ID sesi didalam cookie.

IV.7.4.5 CROSS SITE SCRIPTING

Beberapa aplikasi web menghasilkan halaman web berdasarkan input pengguna atau informasi header HTTP, seperti hasil pencarian, halaman konfirmasi pendaftaran pengguna, papan buletin dan laporan statistik web., penyerang dapat memasukkan konten jahat. Masalah ini disebut "Kerentanan Cross-Site Scripting " dan metode serangan yang mengeksploitasi kerentanan ini disebut "serangan Cross-Site Scripting". Tidak hanya membahayakan situs web itu sendiri tetapi juga akan mempengaruhi keamanan para pengunjung situs web.

Gambar 4.21 Visualisasi Proses Cross Site Scripting



IV.7.4.5.1 POTENSI ANCAMAN

Ancaman ini memungkinkan penyerang untuk:

- Menampilkan halaman web palsu di situs web yang sah (misal. kebingungan yang disebabkan oleh informasi yang salah, pengungkapan informasi sensitif melalui serangan phishing)
- Mencuri cookie yang disimpan oleh browser web
    - Pencurian ID sesi yang disimpan dalam cookie akan menyebabkan spoofing
    - Pengungkapan informasi pribadi dan data sensitif yang disimpan dalam cookie curian.

IV.7.4.5.2 SOLUSI

Tindakan untuk Aplikasi Web yang Tidak Mengizinkan Input Teks HTML

- Melakukan Escaping untuk semua yang akan dikeluarkan ke web halaman.

  Untuk mencegah skrip lintas situs, melakukan pelepasan untuk semua elemen halaman web, seperti konten dan nilai atribut HTML. Salah satu cara untuk mengimplementasikan melepaskan diri adalah dengan mengganti karakter khusus digunakan untuk mengontrol tata letak halaman web, seperti "<", ">" dan "&", dengan masing-masing entitas HTML "& lt;", “& gt;” dan “& amp;”. Jika aplikasi web perlu membuat tag HTML, pastikan untuk melampirkan semua nilai atribut dalam tanda kutip ganda, kemudian lakukan pelarian dengan mengganti tanda kutip ganda yang terkandung dalam nilai atribut dengan entitas HTML "& quot;".

  Dalam hal pencegahan kerentanan, data yang harus melalui proses pelolosan adalah input string karakter diteruskan ke aplikasi web oleh entitas eksternal, nilai-nilai yang diambil dari database atau file dan yang dihasilkan dari operasi aritmatika pada string karakter. Namun, kita bisa membuatnya tentu tidak akan melewatkan apapun dengan mengambil pendekatan yang lebih konsisten di mana semua elemen teks halaman web berada harus melalui proses pelolosan terlepas dari apakah perlu. Proses keluaran yang perlu menyertakan proses pelarian tidak terbatas pada itu untuk respons HTTP. Saat mengubah konten halaman web secara dinamis, misalnya menggunakan metode document.write dalam JavaScript atau properti innerHTML, proses yang sama diperlukan.

- Saat mengeluarkan URL dalam HTML, hanya diizinkan yang memulai dengan pola tertentu, seperti "http: //" dan "https: //".

  URL dapat dimulai dengan tidak hanya "http: //" atau "https: //" tetapi juga dengan "javascript:". Jika sebuah URL sumber daya atau gambar yang akan dimasukkan ke dalam halaman HTML secara dinamis dibuat berdasarkan input eksternal, penyerang dapat meluncurkan serangan skrip lintas situs dengan menanamkan skrip ke dalam URL.

  Misalnya, jika aplikasi web membuat halaman output HTML dengan mengatur URL yang ditentukan oleh pengguna seperti <a herf= *pengguna input* URL> , penyerang dapat menyisipkan skrip dengan string yang dimulai dengan "http: //" atau "https:: //" untuk nilai URL. Ambil pendekatan daftar putih di mana hanya string yang dimulai dengan http: // atau https: // yang diizinkan untuk nilai URL.

Tindakan untuk Aplikasi Web yang Mengizinkan Input Teks HTML

- Tidak membuat konten tag <script> ... </script> secara dinamis

  Jika nilai untuk tag <script> ... </script> dibuat secara dinamis berdasarkan input eksternal, skrip yang tidak diinginkan dapat dimasukkan di sana. Kita dapat memeriksa dan membatalkan skrip yang berisiko.

Disarankan untuk tidak membiarkan aplikasi menetapkan nilai tag <script> ... </script> secara dinamis karena akan sulit untuk menentukan skrip mana yang memang berbahaya.

- Tidak mengizinkan impor stylesheet dari situs web yang tidak diinginkan

  Skrip dapat ditulis ke dalam stylesheet menggunakan fungsi seperti ekspresi (). Itu artinya skrip berbahaya dapat dimasukkan ke halaman web jika desain situs web memungkinkan untuk mengimpor stylesheet dari situs web yang tidak diinginkan.

  Kita dapat memeriksa stylesheet yang diimpor dan membatalkan skrip berbahaya atau dengan tidak membiarkan aplikasi menggunakan stylesheet eksternal karena akan sulit untuk menghapusnya

- Memeriksa nilai input.

  Menjadikan aplikasi web memiliki fungsi untuk memeriksa nilai input dan meminta pengguna untuk memasukkan kembali ketika mereka tidak mengikuti aturan tertentu.

- Membuat pohon parse dari input teks HTML dan ekstrak saja elemen yang diperlukan yang tidak mengandung skrip.

  Parsing input teks HTML dan ekstrak hanya elemen yang diizinkan dalam daftar putih yang telah ditentukan. Langkah ini akan membutuhkan pemrograman yang kompleks dan beban pemrosesan akan tinggi.

Tindakan untuk Aplikasi Web Secara Umum

- Menetapkan parameter charset dari header jenis konten HTTP.
  Kita dapat mengatur kode karakter di Tipe-Konten dari header HTTP seperti: “Tipe-Konten: teks / html; charset = UTF-8 ”. Ketika parameter charset tidak ada dari bagian header Jenis-Konten, browser mengasumsikan kode karakter berdasarkan aturannya sendiri dan memproses string dengan karakter asumsi yang ditetapkan untuk menampilkannya di browser web. Sebagai contoh, beberapa browser diketahui menggunakan kode karakter tertentu ketika bagian pertama dari teks HTML berisi string karakter tertentu.

  Jika charset tidak ditentukan, penyerang dapat mengeksploitasi perilaku browser ini, dengan meminta browser untuk memilih set karakter tertentu dengan sengaja dengan memasukkan string karakter tertentu dan menanamkan karakter string yang akan muncul sebagai skrip ketika mereka diproses dengan set karakter itu.

  Misalnya, jika karakter string "+ ADw-script + AD4-alert (+ ACI-test + ACI -) + AdsAPA- / script + AD4-" dimasukkan ke dalam teks HTML, beberapa browser akan mengenalinya sebagai string yang dikodekan oleh UTF-7. Jika string ini diproses menggunakan UTF-7, akan menjadi "<script> alert ('test') </script>" dan skrip ini akan dieksekusi.

- Menghapus string skrip dalam input teks HTML.

  Identifikasi string skrip yang termasuk dalam input teks HTML dan membatalkan string tersebut dengan menggantinya dengan string yang tidak berbahaya. Misalnya kita mengganti "<script>" atau “Javascript:” dengan menambahkan karakter ke string tersebut seperti “<xscript>” atau “xjavascript:”. Alternatif lain dapat menghapus string skrip secara keseluruhan tetapi mungkin akan menghadirkan risiko baru dengan menghapus mereka yang pada gilirannya membuat string berbahaya.

  Tindakan pencegahan terhadap kerentanan cross-site scripting dengan benar, karakter yang ditunjukkan di atas, seperti "+ ADw-", tidak akan luput sejak karakter tersebut diproses oleh aplikasi web yang diatur dengan kode karakter lain, seperti UTF-8, EUC-JP atau SHIFT_JIS, yang tidak dikenali sebagai sesuatu yang harus diloloskan.

  Untuk mencegah masalah ini, dapat melakukan pelolosan lain untuk teks HTML dengan asumsi itu dikodekan oleh UTF-7, tetapi dengan asumsi hanya UFT-7 tidak cukup. Jadi, untuk mengatasi masalah ini, sangat efektif untuk menentukan parameter charset tanpa menghilangkan itu. Setel kode karakter yang akan digunakan aplikasi web dengan penanganan string karakter saat mengeluarkan halaman HTML dalam tipe-konten header HTTP yang menyertainya.

- Atur atribut HttpOnly dari cookie dan nonaktifkan metode TRACE untuk mencegah pengungkapan informasi cookie.

  "HttpOnly" adalah atribut yang dapat kita atur pada cookie dan akan menolak skrip dalam teks HTML akses ke cookie. Ini akan mencegah cookie dari dicuri bahkan jika situs web memiliki kerentanan cross – site scripting

  Mengatur atribut HttpOnly di header HTTP Set-Cookie saat membuat cookie seperti: “Set-Cookie: [snip]; **HttpOnly**

  Ada beberapa hal yang harus diketahui saat mengadopsi tindakan pencegahan ini

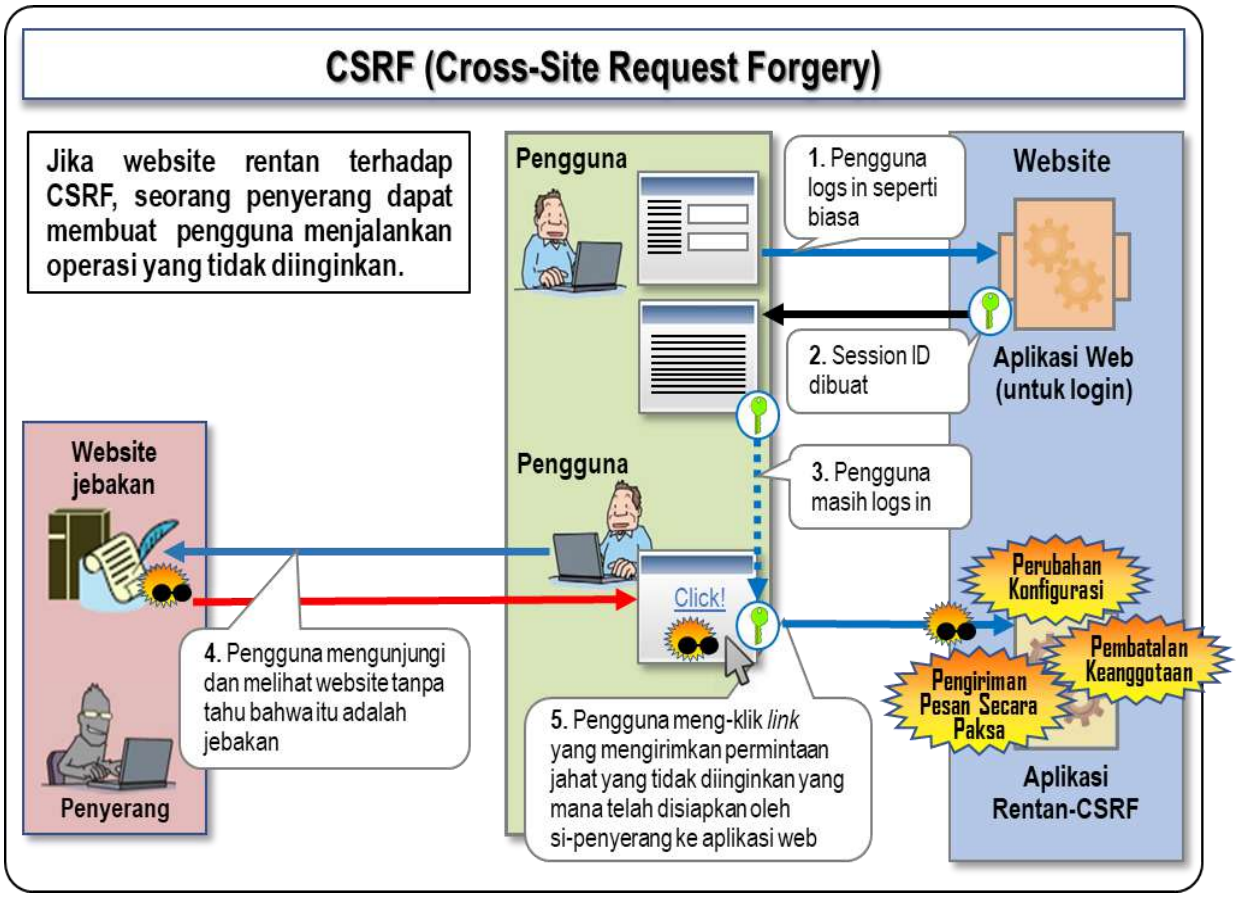
  Pertama, Kita perlu menonaktifkan metode TRACE di server web. Saat metode TRACE diaktifkan, jika situs web memiliki kerentanan skrip lintas situs, penyerang dapat memperoleh seluruh permintaan header HTTP yang dikirimkan browser menggunakan metode serangan yang disebut "Cross-Site Tracing". Permintaan HTTP header berisi informasi cookie

  Kedua, atribut HttpOnly tidak didukung oleh semua browser, jadi itu bukan solusi yang bisa bermanfaat dan melindungi semua pengunjung situs web. dengan demikian cookie akan 'dicuri' bahkan jika atribut HttpOnly adalah set.

IV.7.4.6 CROSS SITE REQUEST FORGERY

Beberapa situs web mengharuskan pengguna untuk masuk untuk menawarkan layanan mereka. Di sini, jika situs web tidak memiliki mekanisme untuk memverifikasi apakah permintaan yang dibuat oleh pengguna yang masuk adalah memang permintaan yang dimaksudkan oleh pengguna. Situs web dapat menerima permintaan jahat yang dibuat oleh pihak eksternal lainnya. Jika situs web memiliki ini kerentanan, penggunanya dapat menderita karena melakukan hal-hal yang tidak diinginkan di situs web melalui perangkap yang ditetapkan oleh penyerang. Hal ini disebut "kerentanan Pemalsuan Permintaan Lintas Situs" dan serangannya metode yang mengeksploitasi kerentanan ini disebut "serangan Pemalsuan Permintaan Situs-Lintas".

Gambar 4.22 Visualisasi Proses Site Request Forgery



IV.7.4.6.1 POTENSI ANCAMAN

Ancaman ini dapat memungkinkan penyerang untuk:

- Mengkkses layanan yang biasa tersedia untuk pengguna yang memiliki akses masuk dengan bena (misal. mentransfer uang, membeli barang, atau membatalkan keanggotaan yang tidak diinginkan oleh pengguna)
- Menambahkan dan mengubah informasi yang biasanya diizinkan hanya untuk pengguna yang memilikinya (misal. pengaturan aplikasi (kata sandi, fungsi administrator, dll.), menulis entri yang tidak benar)

IV.7.4.6.2 SOLUSI

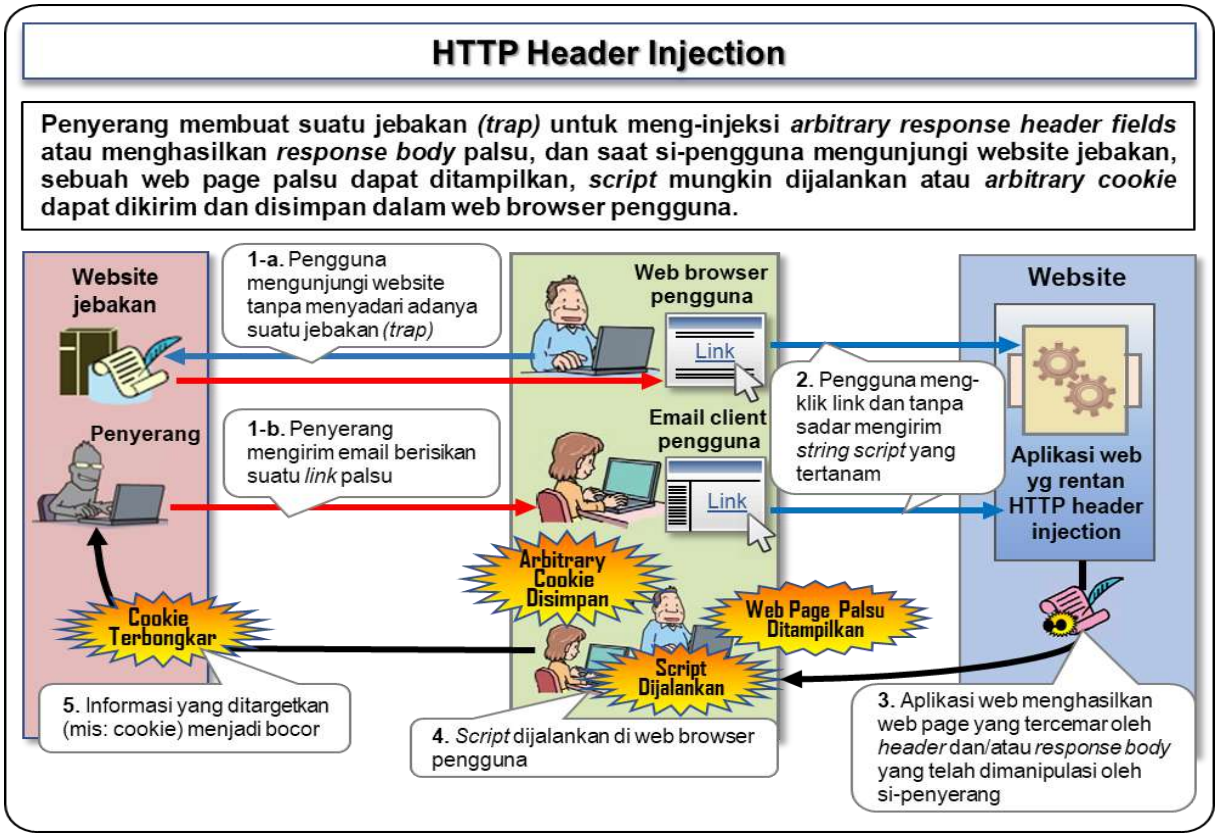
- Akses halaman web, di mana operasi tertentu akan dieksekusi, melalui metode POST rahasia yang memiliki halaman web sebelumnya masukkan ke dalam file tersembunyi, dan menjalankan operasi yang diminta hanya ketika rahasianya benar.
- Meminta kata sandi tepat sebelum menjalankan operasi yang diminta dan dilanjutkan ketika kata sandi itu benar.
- Memeriksa pengarah (referrer) URL yang diharapkan dan dilanjutkan hanya untuk URL yang benar.
- Beri tahu alamat email yang ditentukan sebelumnya secara otomatis ketika operasi penting telah dilakukan.
- Email dikirim ketika pasca-insiden dan karenanya tidak dapat mencegah serangan CSRF, tetapi bisa memunculkan warna merah yang ditandai bahwa ada sesuatu yang salah ketika serangan itu benar-benar terjadi. Hati-hati untuk tidak memasukkan informasi sensitif terkait privasi di badan email.

IV.7.4.7 HTTP HEADER INJECTION

Beberapa aplikasi web secara dinamis menetapkan nilai bagian header respons HTTP berdasarkan nilai yang disahkan oleh parameter eksternal. Misalnya, pengalihan HTTP diterapkan dengan mengatur pengalihan ke URL yang ditentukan dalam parameter ke bagian header lokasi, atau aplikasi web dapat menetapkan nama dimasukkan dalam papan buletin ke header Set-Cookie yang diajukan.

Jika proses membangun respons header HTTP dalam aplikasi web tersebut memiliki kerentanan, penyerang dapat menambahkan bagian header, memanipulasi badan respons dan minta aplikasi web menghasilkan beberapa respons. Masalah ini disebut “Kerentanan Injeksi Header HTTP ”dan metode serangan yang mengeksploitasi kerentanan ini disebut “Serangan Injeksi Header HTTP “. Secara khusus, serangan yang mengarahkan aplikasi web untuk menghasilkan beberapa respons biasa disebut "HTTP Response Splitting Attack".

Gambar 4.23 Visualisasi Proses HTTP Header Injection



IV.7.4.7.1 POTENSI ANCAMAN

Kerentanan ini dapat memungkinkan penyerang untuk:

- Menghadirkan ancaman yang sama dengan yang ditimbulkan oleh kerentanan site cross scripting

  Jika arbitrary response body dimasukkan, akn mengakibatkan browser pengguna menampilkan informasi yang salah atau dipaksa untuk mengeksekusi skrip yang tidak diinginkan tersebut.

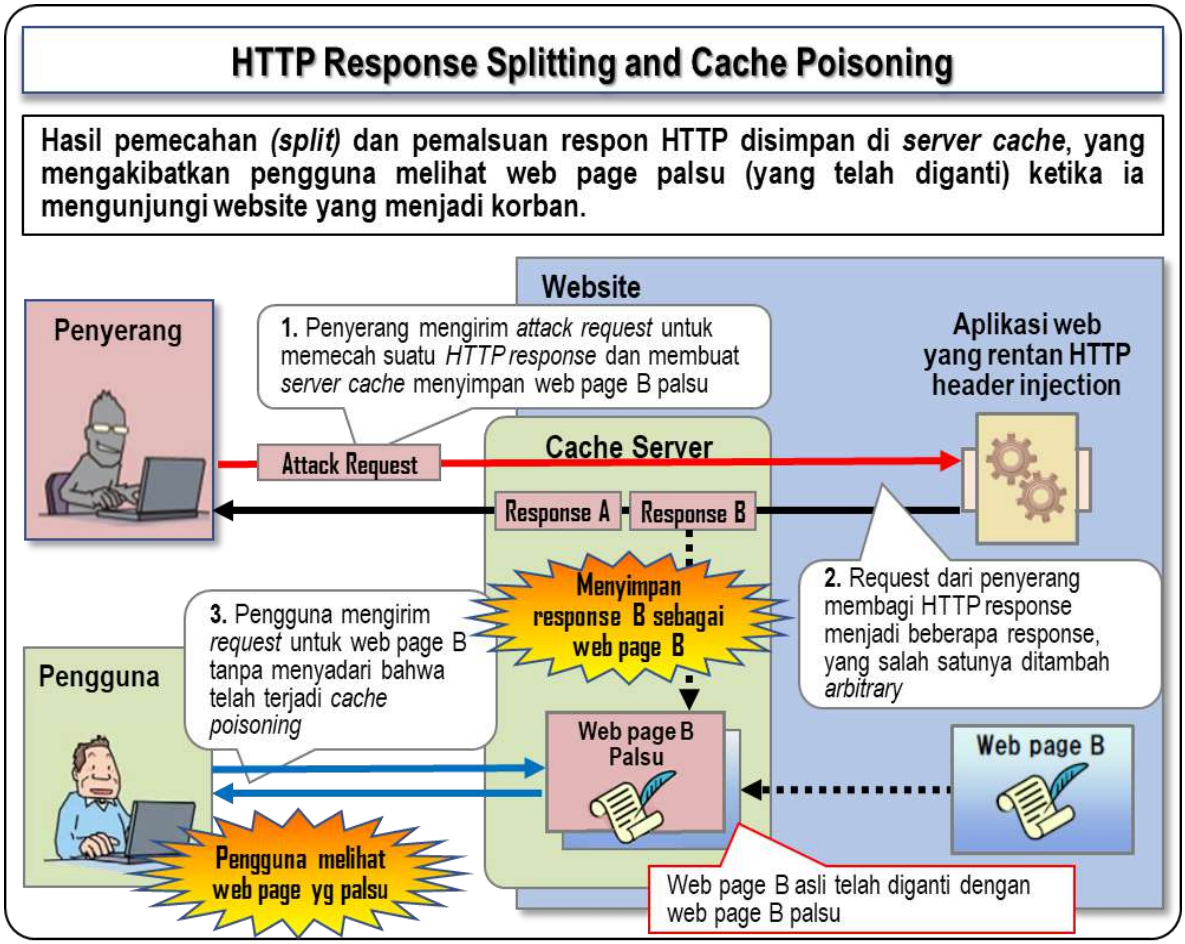
- Membuat cookie yang tidak diinginkan

  Ketika header HTTP Set-Cookie dimasukkan, cookie yang tidak diinginkan dibuat dan disimpan di browser pengguna.

- Poison web cache

  Pemisahan respons HTTP memaksa server web untuk menghasilkan beberapa respons HTTP dan bisa menyebabkan poison cache, yang mengakibatkan pemalsuan halaman web, dan memiliki server proxy cache sebagai tanggapan HTTP yang tidak dinginkan dan mengganti halaman web cache yang asli. Pengguna yang mengunjungi situs web korban akhirnya lebih pada melihat halaman web palsu yang diganti. Ancaman poison web cache akan memengaruhi lebih banyak pengguna dan cenderung bertahan lama.

Gambar 4.24 Visualisasi Proses HTTP Response Splitting and Cache Poisoning



IV.7.4.7.2 SOLUSI

- Tidak mencetak header HTTP secara langsung dan melakukannya melalui HTTP API header yang disediakan oleh lingkungan eksekusi atau bahasa pemrograman.

  Di beberapa lingkungan eksekusi, aplikasi web dapat langsung mencetak HTTP header respons yang menentukan bagian Content-Type. Dalam kasus ini, jika aplikasi mencetak nilai input dilewatkan oleh parameter eksternal secara langsung, karakter umpan baris dapat diatur sepanjang. Karakter umpan baris digunakan untuk memisahkan header HTTP sehingga memungkinkan penyisipan umpan baris yang dapat menjadi penyebab injeksi / respons header / badan yang tidak diinginkan Karena struktur header cukup kompleks seperti yang dilihat pada opsi non-breaking space dan sulit untuk menangani semuanya secara manual, maka sebaiknya menggunakan API header HTTP yang ditawarkan oleh bahasa pemrograman atau lingkungan eksekusi
- Jika HTTP header API yang menawarkan netralisasi umpan baris tidak tersedia maka dapat diimplementasikan secara manual.

  Dapat menambahkan ruang tanpa putus setelah umpan baris yang tidak terduga, dengan menghapus string itu mengikuti umpan garis yang tidak terduga

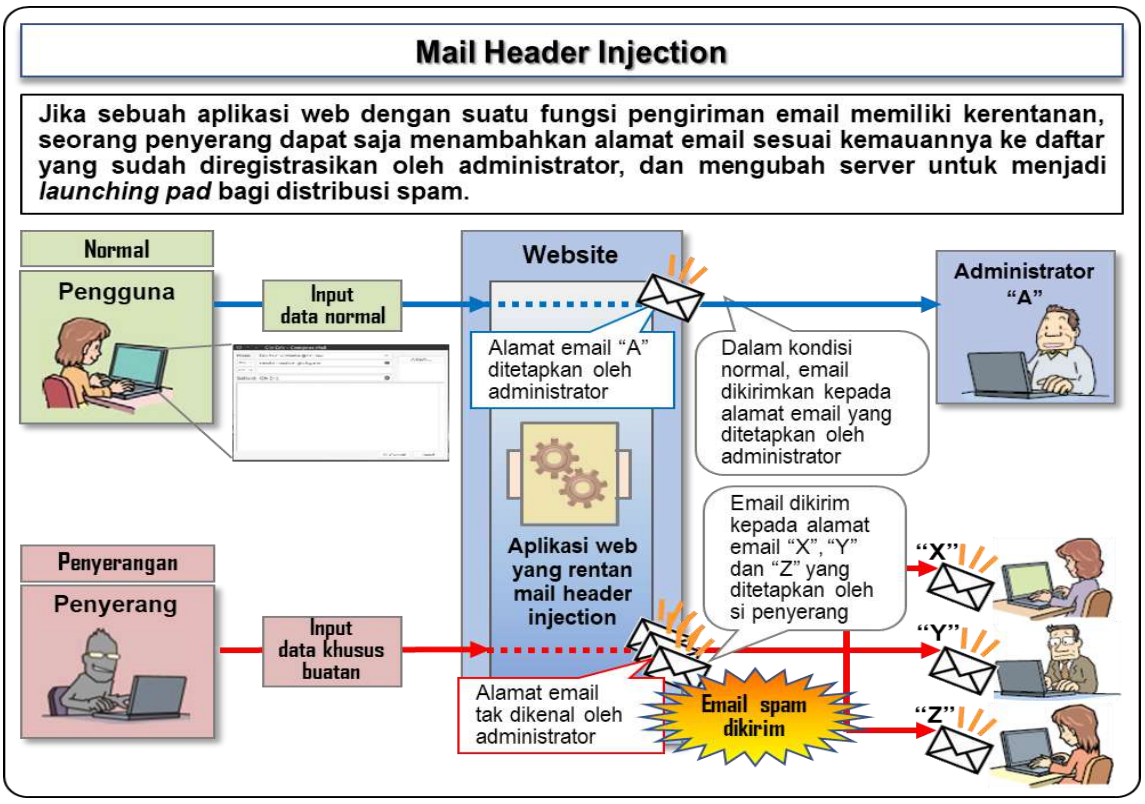
- Menghapus semua karakter umpan baris yang muncul di input teks eksternal

  Menghapus semua karakter umpan baris yang muncul dalam teks input yang dilewatkan oleh parameter eksternal. Bahkan mungkin ingin menghapus semua karakter kontrol seolah – olah hanya karakter umpan baris. Perhatikan bahwa jika web aplikasi perlu menerima string karakter yang mungkin berisi umpan baris, seperti input data di <textarea> ... </textarea> tag, secara sistematis menghapus setiap umpan baris dari semua input data yang dapat menghalangi operasi aplikasi web.

IV.7.4.8 MAIL HEADER INJECTION

Beberapa aplikasi web menyediakan fungsi yang mengirim email ke alamat email tertentu tentang, untuk misalnya, barang dagangan yang dibeli pengguna atau balasan survei. Secara umum, alamat email ini adalah ditentukan sebelumnya dan hanya administrator web yang dapat mengubah. Penyerang mungkin dapat mengatur dan mengubahnya ke alamat email yang tidak diinginkan. Kerentanan ini disebut “Kerentanan Mail Header Injection ”dan metode serangan yang mengeksploitasi kerentanan ini disebut“ Serangan Mail Header Injection".

Gambar 4.25 Visualisasi Proses Mail Header Injection



IV.7.4.8.1 POTENSI ANCAMAN

Kerentanan ini dapat memungkinkan penyerang untuk:

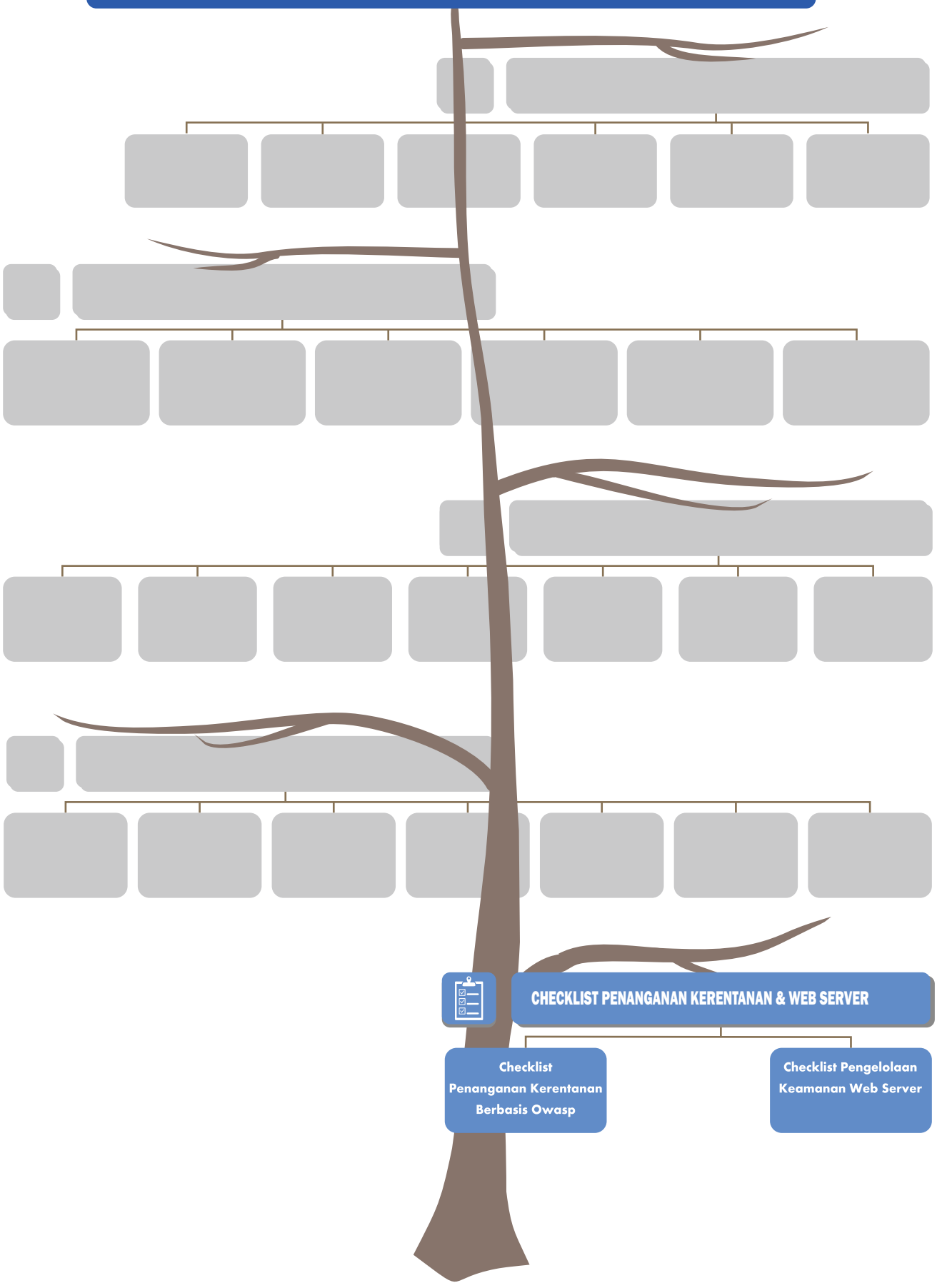
- Relai email pihak ketiga (digunakan sebagai landasan untuk distribusi spam)

IV.7.4.8.2 SOLUSI

- Menggunakan nilai tetap untuk elemen header dan menghasilkan semua input eksternal ke badan email.
  Dalam kasus dimana nilai elemen header email, seperti "To", "Cc", "Bcc" dan "Subject", diatur berdasarkan input eksternal, atau proses output data ke fungsi pengiriman email yang rentan, jika input eksternal langsung digunakan sebagai nilai output, karakter umpan baris termasuk dalam eksternal input akan dimasukkan sebagai jeda baris yang tidak perlu. Jika ini dibolehkan, penyerang akan mengeksploitasi memasukkan header email yang tidak diinginkan, mengubah badan email atau mengirim email ke alamat tidak diinginkan. Disarankan tidak menggunakan parameter eksternal untuk mengatur nilai elemen header email.
- Jika nilai tetap tidak dapat digunakan untuk header, dapat menggunakan API pengiriman email yang ditawarkan oleh lingkungan eksekusi aplikasi web atau bahasa pemrograman.

  Contoh di mana Anda tidak bisa menggunakan nilai tetap untuk elemen header adalah case yang Anda inginkan untuk mengubah topik pembicaraan.
  Jika perlu menggunakan input eksternal sebagai nilai header, disarankan untuk menggunakan API pengiriman email ditawarkan oleh lingkungan eksekusi aplikasi web atau bahasa pemrograman yang digunakan. Namun, beberapa API tidak dapat menangani karakter umpan baris dengan tepat atau dapat menyisipkan beberapa header. Dalam kasus tersebut, dapat menerapkan tambalan keamanan atau mengimplementasikan modifikasi yang diperlukan untuk tidak membiarkan garis pemisah sendiri. Misalnya, untuk mencegah jeda baris, dapat memasukkan spasi atau tab horizontal setelah umpan garis karakter agar program memproses garis sebagai satu baris berkelanjutan. Menghapus karakter setelah baris karakter atau menghentikan pembuatan halaman web jika jeda baris terdeteksi.
- Tidak mencantumkan alamat email dalam HTML.
  Tidak mencantukan alamat email penerima secara langsung di parameter tersembunyi. Implementasi seperti ini akan diteruskan aplikasi web sehingga dapat dieksploitasi oleh serangan relai email pihak ketiga dengan mengubah parameter nilai.

PEDOMAN TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB



V. CHECKLIST PENANGANAN KERENTANAN & WEB SERVER

V.1 CHECKLIST PENANGANAN KERENTANAN BERBASIS OWASP

Tabel 5.1 Checklist Penanganan Kerentanan Berbasis OWASP

No.	Jenis Kerentanan	Checkbox	Aktivitas
1.	SQL Injection	□ Done □ Not Done □ N/A	Membuat semua pernyataan SQL menggunakan placeholder.
		□ Done □ Not Done □ N/A	Saat membuat pernyataan SQL melalui concatenation maka gunakan aplikasi khusus yang ditawarkan oleh mesin database untuk melakukan escaping dan memperbaiki literal pernyataan SQL dengan benar.
		□ Done □ Not Done □ N/A	Tidak menulis pernyataan SQL langsung di parameter yang akan dilewatkan ke aplikasi web.
		□ Done □ Not Done □ N/A	Membatasi informasi untuk ditampilkan dalam pesan kesalahan pada browser web.
		□ Done □ Not Done □ N/A	Memberikan hak minimum untuk akun basis data.
2.	Injeksi perintah – perintah OS	□ Done □ Not Done □ N/A	Menghindari penggunaan fungsi yang dapat memanggil perintah shell.
		□ Done □ Not Done □ N/A	Saat menggunakan fungsi yang dapat memanggil perintah shell, periksa semua variabel yang membentuk parameter shell dan pastikan untuk mengeksekusi hanya mereka yang diberikan untuk dieksekusi.
3.	Parameter Nama Jalur Tidak Dicari / Direktori Traversal yang Tidak Benar	□ Done □ Not Done □ N/A	Tidak mencantumkan nama file yang disimpan di server web secara langsung menggunakan parameter eksternal.
			Menggunakan direktori tetap untuk menangani nama file dan membatalkan nama direktori dalam nama file.
		□ Done □ Not Done □ N/A	Mengelola izin akses file dengan benar.
		□ Done □ Not Done □ N/A	Mengecek nama file
4.	Manajemen Sesi yang tidak benar	□ Done □ Not Done □ N/A	Membuat ID sesi yang sulit ditebak.
		□ Done □ Not Done □ N/A	Tidak menggunakan parameter URL untuk menyimpan ID sesi.

			□ Done □ Not Done □ N/A	Mensetting atribut aman pada cookie ketika menggunakan HTTPS.
			□ Done □ Not Done □ N/A	Mulai sesi baru setelah login berhasil.
			□ Done □ Not Done □ N/A	Mengeluarkan rahasia setelah login dan autentikasi pengguna setiap kali pengguna bergerak di situs web.
			□ Done □ Not Done □ N/A	Menggunakan ID sesi acak
			□ Done □ Not Done □ N/A	Menetapkan tanggal kedaluwarsa cookie dengan hati-hati saat menyimpan id sesi dalam cookie.
5.	Cross-SiteScripting	Tindakan agar Aplikasi Web tidak mengizinkan input teks HTML	□ Done □ Not Done □ N/A	Melakukan Escaping untuk semua yang akan dikeluarkan ke halaman web.
			□ Done □ Not Done □ N/A	Saat mengeluarkan URL dalam HTML, diizinkan hanya yang memulai dengan pola tertentu, seperti "http: //" dan "https: //".
			□ Done □ Not Done □ N/A	Tidak membuat konten tag <script> ... </script> secara dinamis.
			□ Done □ Not Done □ N/A	Tidak mengizinkan mengimpor stylesheet dari situs web yang tidak jelas.
			□ Done □ Not Done □ N/A	Mengecek nilai input
		Tindakan untuk Aplikasi Web yang mengizinkan input teks HTML	□ Done □ Not Done □ N/A	Membuat pohon parse dari input teks HTML dan ekstrak hanya elemen yang tidak mengandung skrip.
			□ Done □ Not Done □ N/A	Hapus string skrip dalam input teks HTML.
		Tindakan umum untuk semua aplikasi web	□ Done □ Not Done □ N/A	Setting parameter rangkaian karakter dari header jenis konten HTTP.
			□ Done □ Not Done □ N/A	Setting atribut HttpOnly cookie dan nonaktifkan metode TRACE untuk mencegah pengungkapan informasi cookie.
6.	CSRF(Cross-Site Request Forgery)		□ Done □ Not Done □ N/A	Akses halaman web, di mana operasi tertentu akan dijalankan, melalui metode POST rahasia memiliki halaman web sebelumnya menyisipkannya dalam file tersembunyi yang disembunyikan, dan menjalankan operasi yang diminta hanya jika kebijakan ini benar.
			□ Done □ Not Done □ N/A	Meminta hak password sebelum menjalankan operasi yang diminta dan melanjutkannya hanya ketika password tersebut benar.
			□ Done □ Not Done □ N/A	Memeriksa pengarah apakah URL yang diharapkan jika ya berlanjut hanya ketika URL benar
			□ Done □ Not Done □ N/A	Memberi tahu alamat email yang ditentukan sebelumnya secara otomatis ketika operasi penting telah dilakukan.

7.	Http Header Injection	□ Done □ Not Done □ N/A	Tidak mencetak header HTTP secara langsung tetapi melalui API HTTPheader yang disediakan oleh lingkungan eksekusi atau bahasa pemrograman.
		□ Done □ Not Done □ N/A	Jika HTTP header API yang menawarkan netralisasi umpan tidak tersedia maka terapkan secara manual.
		□ Done □ Not Done □ N/A	Hapus semua karakter umpan baris yang muncul di input teks eksternal.
8.	Mail Header Injection	□ Done □ Not Done □ N/A	Gunakan nilai tetap untuk elemen header dan output semua external input ke badan email.
		□ Done □ Not Done □ N/A	Jika hal diatas tidak diterapkan, nilai tetap tidak dapat digunakan untuk header, gunakan API pengiriman email yang ditawarkan oleh lingkungan atau bahasa eksekusi aplikasi web.
		□ Done □ Not Done □ N/A	Tidak mencantumkan alamat email dalam HTML.
		□ Done □ Not Done □ N/A	Hapus semua karakter umpan baris yang muncul di input teks eksternal.
9		□ Done □ Not Done □ N/A	Ketika sebuah situs web memerlukan kontrol akses, terapkan mekanisme autentikasi yang mengharuskan pengguna memasukkan semacam informasi rahasia, seperti kata sandi.
		□ Done □ Not Done □ N/A	Terapkan otorisasi serta otentikasi untuk memastikan bahwa pengguna login tidak dapat berpura-pura menjadi pengguna lain dan mengakses data mereka.

V.2 CHECKLIST PENGELOLAAN KEAMANAN WEB SERVER

Tabel 5.2 Checklist Pengelolaan Keamanan Web Server

No	Checkbox	Aktivitas
1.	**Perencanaan konfigurasi dan deployment Web Server**	
	□ Done □ Not Done □ N/A	Mengidentifikasi fungsi- fungsi dari Web server
	□ Done □ Not Done □ N/A	Mengidentifikasi kategori informasi yang akan disimpan, diproses, dan dikirim melalui Web server
	□ Done □ Not Done □ N/A	Mengidentifikasi kebutuhan keamanan informasi
	□ Done □ Not Done □ N/A	Mengidentifikasi bagaimana informasi dipublikasikan ke Web server
	□ Done □ Not Done □ N/A	Mengidentifikasi kebutuhan keamanan *host* lain yang terlibat (misalnya backend database atau Web server)
	□ Done □ Not Done □ N/A	Mengidentifikasi *host* yang ditunjuk untuk menjalankan Web server
	□ Done □ Not Done □ N/A	Mengidentifikasi layanan jaringan yang akan diberikan atau didukung oleh Web server
	□ Done □ Not Done □ N/A	Mengidentifikasi kebutuhan keamanan dari suatu layanan tambahan yang diberikan atau didukung oleh Web server
	□ Done □ Not Done □ N/A	Mengidentifikasi bagaimana Web server akan dikelola
	□ Done □ Not Done □ N/A	Mengidentifikasi para pengguna dan kategori para pengguna dari Web server dan tentukan priviledge dari setiap kategori pengguna
	□ Done □ Not Done □ N/A	Identifikasi metode otentikasi pengguna pada Web server dan bagaimana data otentikasi akan diproteksi
	□ Done □ Not Done □ N/A	Mengidentifikasi bagaimana akses ke sumber informasi akan diberlakukan
	□ Done □ Not Done □ N/A	Mengidentifikasi mekanisme keamanan fisik yang tepat

	□ Done □ Not Done □ N/A	Mengidentifikasi mekanisme ketersediaan yang tepat
	□ Done □ Not Done □ N/A	Pemilihan OS yang tepat untuk Web server
	□ Done □ Not Done □ N/A	Mengidentifikasi penyingkapan kerentanan yang paling minimal
	□ Done □ Not Done □ N/A	Kemampuan untuk membatasi aktifitas tingkat administratif atau root hanya untuk para pengguna yang sah
	□ Done □ Not Done □ N/A	Kemampuan untuk mengontrol akses terhadap data pada server
	□ Done □ Not Done □ N/A	Kemampuan untuk menonaktifkan layanan jaringan yang tidak diperlukan dalam perangkat lunak OS atau server
	□ Done □ Not Done □ N/A	Kemampuan untuk mengontrol akses ke berbagai bentuk program yang dapat dieksekusi, seperti CGI script dan server plug-in
	□ Done □ Not Done □ N/A	Kemampuan untuk merekam setiap aktifitas server untuk mendeteksi penyerangan dan usaha penyerangan
	□ Done □ Not Done □ N/A	Ketersediaan kapabilitas firewall host-based
	□ Done □ Not Done □ N/A	Ketersediaan staf yang berpengalaman untuk menginstal, mengkonfigurasi, dan memelihara OS
	□ Done □ Not Done □ N/A	Pemilihan Platform yang tepat untuk Web Server • General purpose OS • Trusted OS • Web server appliance • Pre-hardened OS and Web server • Virtualized Platform
2.	**Patch dan upgrade OS**	
	□ Done □ Not Done □ N/A	Melakukan dan mendokumentasikan proses *patching*
	□ Done □ Not Done □ N/A	Menjaga server agar tidak terhubung ke jaringan atau terisolasi yang membatasi koneksi sampai seluruh *patch* telah dipasang
	□ Done □ Not Done □ N/A	Mengidentifikasi dan menginstalasi seluruh *patch* dan *upgrade* yang dibutuhkan ke OS
	□ Done □ Not Done □ N/A	Mengidentifikasi dan menginstalasi seluruh *patch* dan *upgrade* yang dibutuhkan pada aplikasi dan layanan yang terkait dengan OS

	□ Done □ Not Done □ N/A	Mengidentifikasi dan meminimalkan kerentanan yang tidak di-*patch*
3.	**Menonaktifkan layanan dan aplikasi yang tidak diperlukan**	
	□ Done □ Not Done □ N/A	Non-aktifkan atau hilangkan layanan dan aplikasi yang tidak diperlukan
4.	**Konfigurasikan otentikasi pengguna OS**	
	□ Done □ Not Done □ N/A	Hilangkan atau nonaktifkan akun default dan grup yang tidak dibutuhkan
	□ Done □ Not Done □ N/A	Non-aktifkan akun yang tidak interaktif
	□ Done □ Not Done □ N/A	Menciptakan grup-grup pengguna untuk komputer- komputer tertentu
	□ Done □ Not Done □ N/A	Menciptakan akun-akun pengguna untuk komputer- komputer tertentu
	□ Done □ Not Done □ N/A	Memeriksa kebijakan password organisasi dan atur akun password secara tepat (misal panjang, kompleksitas dan sebagainya)
	□ Done □ Not Done □ N/A	Mencegah penebakan password (misalnya memberikan jeda waktu antara percobaan login password, tolak login setelah sejumlah percobaan gagal yang telah ditentukan)
	□ Done □ Not Done □ N/A	Menginstalasi dan mengkonfigurasi mekanisme keamanan lain untuk memperkuat otentikasi
	□ Done □ Not Done □ N/A	Konfigurasi kontrol sumber daya secara tepat
	□ Done □ Not Done □ N/A	Menolak akses untuk membaca file-file dan direktori- direktori yang tidak diperlukan
	□ Done □ Not Done □ N/A	Menolak akses untuk menulis pada file-file dan direktori-direktori yang tidak diperlukan
	□ Done □ Not Done □ N/A	Membatasi priviledge dalam hal eksekusi dari tool sistem hanya pada administrator sistem
	□ Done □ Not Done □ N/A	Instalasi dan Konfigurasi control keamanan tambahan
	□ Done □ Not Done □ N/A	Memilih, menginstalasi, dan mengkonfigurasikan perangkat lunak tambahan untuk menyediakan kontrol-kontrol yang dibutuhkan yang tidak termasuk dalam OS

5.	**Uji keamanan dari OS**	
	□ Done □ Not Done □ N/A	Mengidentifikasikan sistem identik yang terpisah
	□ Done □ Not Done □ N/A	Uji OS dilakukan setelah instalasi awal untuk menentukan kerentanan
	□ Done □ Not Done □ N/A	Uji OS secara berkala (missal triwulan) untuk menentukan kerentanan baru
6.	**Instalasi Web server secara aman**	
	□ Done □ Not Done □ N/A	Menginstalasi perangkat lunak Web server pada suatu host yang telah ditetapkan atau guest OS yang divirtualkan dan ditetapkan
	□ Done □ Not Done □ N/A	Mengaplikasikan suatu patches atau upgrade untuk memperbaiki kerentanan yang diketahui
	□ Done □ Not Done □ N/A	Membuat suatu phisical disc atau logical portion(terpisah dari OS dan aplikasi Web server) untuk konten Web
	□ Done □ Not Done □ N/A	Menghilangkan atau nonaktifkan seluruh layanan yang dipasang oleh aplikasi Web server namun tidak dibutuhkan (misal gopher, FTP, administrasi remote)
	□ Done □ Not Done □ N/A	Menghilangkan atau nonaktifkan seluruh akun login default yang tidak dibutuhkan yang dibuat pada saat instalasi Web server
	□ Done □ Not Done □ N/A	Menghilangkan seluruh dokumentasi manufaktur dari server
	□ Done □ Not Done □ N/A	Menghilangkan file contoh atau file uji apapun dari server, termasuk script dan kode yang dapat dijalankan
	□ Done □ Not Done □ N/A	Mengaplikasikan template keamanan yang sesuai atau hardening script ke server
	□ Done □ Not Done □ N/A	Mengkonfigurasikan kembali banner layanan HTTP (dan layanan lain yang dibutuhkan) untuk tidak melaporkan tipe dan versi dari Web server dan OS
7.	**Mengkonfigurasi OS dan Access Control Web server**	
	□ Done □ Not Done □ N/A	Mengkonfigurasikan proses Web server untuk dapat dijalankan oleh pengguna dengan priviledge yang dibatasi dengan ketat
	□ Done □ Not Done □ N/A	Mengkonfigurasikan Web server sedemikian hingga file konten Web dapat dibaca namun tidak dapat ditulis oleh proses layanan

	□ Done □ Not Done □ N/A	Mengkonfigurasikan Web server sedemikian hingga proses layanan tidak dapat menulis ke direktori dimana konten Web disimpan
	□ Done □ Not Done □ N/A	Mengkonfigurasikan Web server sedemikian rupa hingga hanya proses yang berhak untuk administrasi Web server yang dapat menulis file konten Web
	□ Done □ Not Done □ N/A	Mengkonfigurasikan OS host sedemikian hingga Web server dapat menulis file log namun tidak dapat membacanya
	□ Done □ Not Done □ N/A	Mengkonfigurasikan host OS sehingga file temporer yang dibuat oleh aplikasi Web server terbatas untuk subdirektori yang khusus dan dilindungi dengan baik.
	□ Done □ Not Done □ N/A	Mengkonfigurasikan host OS sehingga akses untuk file temporer manapun yang dibuat oleh aplikasi Web server terbatas untuk proses layanan yang telah membuat file.
	□ Done □ Not Done □ N/A	Menginstalasikan konten Web pada suatu hard drive atau logical partition yang berbeda dari OS dan aplikasi Web server
	□ Done □ Not Done □ N/A	Jika upload ke Web server diperbolehkan, konfigurasikan sedemikian hingga ada suatu batasan tentang jumlah ruang hard drive yang ditetapkan untuk tujuan ini; upload sebaiknya ditempatkan dalam partisi yang terpisah
	□ Done □ Not Done □ N/A	Pastikan bahwa file log tersimpan dalam suatu lokasi yang diukur secara tepat; file log sebaiknya ditempatkan pada suatu partisi terpisah
	□ Done □ Not Done □ N/A	Konfigurasi jumlah maksimum proses Web server dan/atau koneksi jaringan yang sebaiknya diperbolehkan oleh Web server
	□ Done □ Not Done □ N/A	Pastikan bahwa guest OS virtual apapun mengikuti checklist ini
	□ Done □ Not Done □ N/A	Pastikan bahwa para pengguna dan administrator dapat mengubah password
	□ Done □ Not Done □ N/A	Non-aktifkan para pengguna setelah tidak-aktif untuk periode tertentu
	□ Done □ Not Done □ N/A	Pastikan para pengguna dan administrator yang memiliki ID yang unik
8.	**Mengkonfigurasikan suatu direktori konten web yang aman**	
	□ Done □ Not Done □ N/A	Tetapkan suatu hard drive atau logical partition tunggal untuk konten Web dan bangun subdirektori terkait khusus untuk file konten Web server, termasuk grafik namun tidak termasuk script dan program lain
	□ Done □ Not Done □ N/A	Tentukan suatu direktori tunggal khusus untuk seluruh script atau program program eksternal yang dijalankan sebagai bagian dari konten Web server (missal CGI, ASP)

	□ Done □ Not Done □ N/A	Non aktifkan eksekusi script yang tidak secara khusus dibawah kontrol akun administratif. Tindakan ini dicapai dengan pembuatan dan pengontrolan akses ke suatu direktori terpisah yang dimaksudkan untuk memuat script yang berhak Non aktifkan penggunaan hard atau symbolic link (misal, shortcut untuk Windows)
	□ Done □ Not Done □ N/A	Identifikasikan folder dan file mana dalam dokumen Web server yang harus dibatasi dan mana yang sebaiknya dapat diakses (dan oleh siapa)
	□ Done □ Not Done □ N/A	Cek kebijakan password dari organisasi dan atur akun password secara tepat misal, panjang, kompleksitas)
	□ Done □ Not Done □ N/A	Menggunakan file robots.txt, jika sesuai
	□ Done □ Not Done □ N/A	Konfigurasikan perlindungan anti spambot, jika ada (misal, CAPTCHA, nofollow, atau keyword filtering)
	□ Done □ Not Done □ N/A	Pastikan tidak ada tipe-tipe informasi berikut ini yang terdapat pada atau melalui suatu Web server
	□ Done □ Not Done □ N/A	Rekaman (Record) yang diklasifikasikan
	□ Done □ Not Done □ N/A	Aturan dan prosedur personil lingkup internal
	□ Done □ Not Done □ N/A	Informasi sensitif atau berklasifikasi
	□ Done □ Not Done □ N/A	Informasi pribadi tentang personil suatu organisasi
	□ Done □ Not Done □ N/A	Nomor telepon, alamat email, atau daftar umum staf kecuali jika diperlukan untuk memenuhi persyaratan yang bersifat organisasi
	□ Done □ Not Done □ N/A	Jadwal pimpinan organisasi atau lokasi tepat mereka (apakah ada atau tidak di lokasi kantor)
	□ Done □ Not Done □ N/A	Informasi komposisi, persiapan atau penggunaan dari materi berbahaya.
	□ Done □ Not Done □ N/A	Informasi sensitif yang berkaitan dengan keamanan negara
	□ Done □ Not Done □ N/A	Catatan investigasi

	□ Done □ Not Done □ N/A	Catatan keuangan (diluar yang telah tersedia secara publik)
	□ Done □ Not Done □ N/A	Catatan medis
	□ Done □ Not Done □ N/A	Prosedur keamanan fisik dan informasi organisasi
	□ Done □ Not Done □ N/A	Informasi tentang jaringan organisasi dan infrastruktur sistem informasi
	□ Done □ Not Done □ N/A	Informasi yang mengkhususkan atau mengimplikasikan kerentanan keamanan fisik
	□ Done □ Not Done □ N/A	Rencana, peta, diagram, foto udara, dan rencana arsitektural organisasi gedung, properti, atau instalasi gedung
	□ Done □ Not Done □ N/A	Materi hak cipta tanpa ijin tertulis dari pemilik
	□ Done □ Not Done □ N/A	Kebijakan privasi atau keamanan yang menunjukkan tipe adanya tindakan keamanan hingga tingkat yang mungkin berguna bagi penyerang
9.	**Menetapkan suatu kebijakan formal dan proses terdokumentasi dalam lingkup organisasi (organizational-wide documented) mengenai konten web yang ditampilkan**	
	□ Done □ Not Done □ N/A	Informasi teridentifikasi yang sebaiknya dipublikasikan pada Web
	□ Done □ Not Done □ N/A	Sasaran audiens teridentifikasi
	□ Done □ Not Done □ N/A	Identifikasi efek negatif yang mungkin muncul akibat publikasi informasi
	□ Done □ Not Done □ N/A	Penanggung jawab yang jelas untuk pembuatan, publikasi, dan pemeliharaan informasi khusus.
	□ Done □ Not Done □ N/A	Menyediakan pedoman dalam hal gaya dan bentuk yang sesuai untuk publikasi Web
	□ Done □ Not Done □ N/A	Menyediakan tinjauan ulang yang sesuai terhadap informasi dalam hal sensifitas dan distribusi/kontrol peluncuran (termasuk sensifitas informasi dalam suatu kumpulan)
	□ Done □ Not Done □ N/A	Menentukan kontrol akses dan keamanan yang sesuai

	□ Done □ Not Done □ N/A	Menyediakan pedoman tentang informasi yang dimuat dalam source code dari konten Web
10.	**Mengelola privasi pengguna Web**	
	□ Done □ Not Done □ N/A	Kelola suatu kebijakan privasi yang dipublikasikan
	□ Done □ Not Done □ N/A	Lakukan larangan pengumpulan data identifikasi secara pribadi tanpa ijin eksplisit dari pengguna dan hanya kumpulkan data yang diperlukan
	□ Done □ Not Done □ N/A	Lakukan larangan penggunaan cookies yang “menetap ”
	□ Done □ Not Done □ N/A	Gunakan session cookie hanya jika diidentifikasikan secara jelas dalam kebijakan privasi yang dipublikasikan
	□ Done □ Not Done □ N/A	Kurangi serangan tidak langsung pada konten
	□ Done □ Not Done □ N/A	Pastikan para pengguna situs waspada terhadap bahaya serangan phishing dan pharming dan bagaimana menghindarinya
	□ Done □ Not Done □ N/A	Validasi komunikasi resmi dengan membuat emails yang khas (personilized email) dan menyediakan informasi identifikasi yang unik (tetapi tidak rahasia) yang sebaiknya hanya organisasi dan pengguna yang tahu
	□ Done □ Not Done □ N/A	Gunakanan signature pada email jika sesuai
	□ Done □ Not Done □ N/A	Jalankan validasi konten dalam aplikasi Web untuk menghindari serangan phishing yang lebih rumit (misal serangan berbasis scripting antar-situs)
	□ Done □ Not Done □ N/A	Buatlah konten Web (personalize) untuk mengidentifikasi situs Web yang sah
	□ Done □ Not Done □ N/A	Gunakan otentikasi berbasis token atau otentikasi mutual jika dapat diaplikasikan
	□ Done □ Not Done □ N/A	Sarankan penggunaan Web browser atau browser toolbars dengan perlindungan terhadap phishing/pharming
	□ Done □ Not Done □ N/A	Gunakan versi terkini dari software DNS dengan patch keamanan versi terakhir
	□ Done □ Not Done □ N/A	Instalasi mekanisme perlindungan DNS server-side

	□ Done □ Not Done □ N/A	Monitor domain organisasi dan domain yang serupa
	□ Done □ Not Done □ N/A	Sederhanakan struktur nama domain organisasi
	□ Done □ Not Done □ N/A	Gunakan koneksi yang aman untuk login
	□ Done □ Not Done □ N/A	Jika perlu, ikutsertakan suatu vendor untuk menyediakan tindakan anti-phishing/ anti-pharming yang lebih kuat
	□ Done □ Not Done □ N/A	Pertimbangkan resiko dan keuntungan active contentclient-side
	□ Done □ Not Done □ N/A	Jangan ambil tindakan tanpa pernyataan ijin dari pengguna
	□ Done □ Not Done □ N/A	Jika mungkin, gunakan hanya active content yang diadopsi secara luas seperti JavaScript, PDF, dan Flash
	□ Done □ Not Done □ N/A	Jika mungkin, sediakan beberapa alternatif (misal, HTML yang disediakan bersama PDF)
11.	**Kelola keamanan active content server-side**	
	□ Done □ Not Done □ N/A	Hanya kode yang sederhana, mudah untuk dimengerti yang sebaiknya digunakan
	□ Done □ Not Done □ N/A	Batasi atau tidak ada pembacaan atau penulisan kepada file sistem yang sebaiknya diijinkan
	□ Done □ Not Done □ N/A	Batasi atau tidak ada interaksi dengan program- program lain (misal, sendmail) yang semestinya diijinkan
	□ Done □ Not Done □ N/A	Sebaiknya tidak ada persyaratan untuk menjalankan dengan suid privileges pada UNIX atau Linux
	□ Done □ Not Done □ N/A	Nama path yang jelas sebaiknya digunakan (yaitu tidak bergantung pada variablel path)
	□ Done □ Not Done □ N/A	Tidak ada direktori yang memiliki ijin untuk menulis dan eksekusi
	□ Done □ Not Done □ N/A	Seluruh file yang dapat dijalankan ditempatkan dalam suatu folder yang telah ditetapkan
	□ Done □ Not Done □ N/A	SSL di non-aktifkan atau fungsi eksekusi di nonaktifkan

	□ Done □ Not Done □ N/A	Seluruh input pengguna di validasi
	□ Done □ Not Done □ N/A	Kode pembangkit konten Web harus di scan atau di audit
	□ Done □ Not Done □ N/A	Halaman yang dibuat secara dinamis tidak menghasilkan meta characters yang berbahaya
	□ Done □ Not Done □ N/A	Pengkodean himpunan karakter harus diatur dengan jelas dalam setiap halaman
	□ Done □ Not Done □ N/A	Data pengguna harus di scan untuk menjamin hanya mengandung input yang diharapkan, (misal a-z, A_Z, 0-9); care should be taken dengan karakter khusus atau HTML tags
	□ Done □ Not Done □ N/A	Cookies harus diperiksa dalam hal karakter khusus apapun
	□ Done □ Not Done □ N/A	Mekanisme enkripsi harus digunakan untuk mengenkripsi password yang dimasukkan melalui bentuk script
	□ Done □ Not Done □ N/A	Untuk aplikasi Web yang dibatasi oleh nama pengguna dan password, tidak ada halaman Web dalam aplikasi yang semestinya dapat di akses tanpa mengeksekusi proses login yang sesuai
	□ Done □ Not Done □ N/A	Seluruh script contoh dihilangkan
	□ Done □ Not Done □ N/A	Tidak ada script pihak ketiga atau kode yang dapat di eksekusi yang digunakan tanpa memverifikasi source code
12.	**Melindungi terhadap serangan brute force**	
	□ Done □ Not Done □ N/A	Gunakan otentikasi yang kuat jika memungkinkan
	□ Done □ Not Done □ N/A	Gunakan delay setelah usaha login yang gagal
	□ Done □ Not Done □ N/A	Lock-out suatu akun setelah satu set usaha login yang gagal
	□ Done □ Not Done □ N/A	Berlakukan suatu kebijakan password
	□ Done □ Not Done □ N/A	Blacklist IP address atau domain yang diketahui untuk usaha serangan brute force
	□ Done □ Not Done □ N/A	Gunakan software pengawasan log untuk mendeteksi serangan brute force

	□ Done □ Not Done □ N/A	Konfigurasikan teknologi otentikasi dan enkripsi Web
	□ Done □ Not Done □ N/A	Untuk sumber daya Web yang memerlukan proteksi minimal dan terdiri dari peserta yang sedikit dan jelas, konfigurasikan otentikasi yang berdasarkan alamat
	□ Done □ Not Done □ N/A	Untuk sumber daya Web yang memerlukan perlindungan tambahan akan tetapi terdiri dari peserta yang sedikit dan jelas, konfigurasikan otentikasi berdasarkan alamat sebagai garis pertahanan kedua
	□ Done □ Not Done □ N/A	Untuk sumber daya Web yang memerlukan perlindungan minimal tetapi tidak terdiri atas peserta yang didefinisikan dengan jelas, konfigurasikan otentikasi dasar atau digest (lebih baik)
	□ Done □ Not Done □ N/A	Untuk sumber daya Web yang memerlukan perlindungan dari malicious bots, konfigurasikan otentikasi dasar atau digest (lebih baik) atau implementasikan teknik-teknik mitigasi
	□ Done □ Not Done □ N/A	Untuk sumber daya Web yang memerlukan perlindungan maksimal, konfigurasikan SSL/TLS dengan cipher suite yang kuat
	□ Done □ Not Done □ N/A	Konfigurasikan SSL/TLS
13.	**Memastikan implementasi SSL/TLS telah dilakukan patch sepenuhnya**	
	□ Done □ Not Done □ N/A	Gunakan suatu issued certificate pihak ketiga untuk otentikasi server (kecuali seluruh sistem yang menggunakan server di atur secara organisasi dapat digunakan self-signed certificate.
	□ Done □ Not Done □ N/A	Untuk konfigurasi yang membutuhkan otentikasi klien tingkat menengah, konfigurasikan server untuk membutuhkan nama pengguna dan password melalui SSL/TLS
	□ Done □ Not Done □ N/A	Untuk konfigurasi yang membutuhkan otentikasi klien tingkat tinggi, konfigurasikan server untuk membutuhkan sertifikat klien melalui SSL/TLS
	□ Done □ Not Done □ N/A	Pastikan cipher suit yang lemah dinonaktifkan sesuai dengan ketentuan yang berlaku.
	□ Done □ Not Done □ N/A	Konfigurasikan pengecek keutuhan file untuk mengawasi sertifikat Web server
	□ Done □ Not Done □ N/A	Jika hanya SSL/TLS yang digunakan dalam Web server, pastikan akses melalui port TCP daripada 443 di nonaktifkan
	□ Done □ Not Done □ N/A	Jika sebagian besar lalu lintas pada Web server akan melalui SSL/TLS yang terenkripsi, pastikan bahwa mekanisme logging dan deteksi yang sesuai digunakan dalam Web server (karena pengawasan jaringan tidak efektif menghadapi sesi SSL/TLS)
14.	**Melakukan penilaian terhadap konfigurasi firewall**	
	□ Done □ Not Done □ N/A	Web server dilindungi oleh suatu firewall

	□ Done □ Not Done □ N/A	Web server, jika berhadapan dengan ancaman yang lebih tinggi atau jika lebih rentan, dilindungi oleh suatu firewall layer aplikasi
	□ Done □ Not Done □ N/A	Firewall mengendalikan semua lalu lintas antara Internet dan Web server
	□ Done □ Not Done □ N/A	Firewall memblokir semua inbound traffic ke Web server kecuali TCP ports 80 (HTTP) dan/atau 443 (HTTP), jika dibutuhkan
	□ Done □ Not Done □ N/A	Firewall memblokir (dalam hubungannya dengan IDPS) alamat IP atau subnet yang dilaporkan IDPS sedang menyerang jaringan organisasi
	□ Done □ Not Done □ N/A	Firewall memberitahukan jaringan atau administrator Web server mengenai kegiatan yang mencurigakan melalui suatu cara yang sesuai
	□ Done □ Not Done □ N/A	Firewall menyediakan penyaring konten (firewall layer aplikasi)
	□ Done □ Not Done □ N/A	Firewall dikonfigurasi untuk melindungi dari serangan DoS
	□ Done □ Not Done □ N/A	Firewall mendeteksi permintaan URL yang salah format atau serangan terhadap permintaan URL yang dikenal
	□ Done □ Not Done □ N/A	Firewall me-log kejadian kritis
	□ Done □ Not Done □ N/A	Firewall dan OS firewall dipatch hingga tingkat aman yang paling mutakhir atau yang paling tinggi
15.	**Melakukan evaluasi terhadap deteksi intrusi dan sistem pencegahan**	
	□ Done □ Not Done □ N/A	IDPS berbasis host digunakan untuk Web server yang beroperasi terutama menggunakan SSL/TLS
	□ Done □ Not Done □ N/A	IDPS dikonfigurasikan untuk memonitor lalu intas jaringan dari dan ke Web server setelah firewall
	□ Done □ Not Done □ N/A	IDPS dibentuk untuk memonitor perubahan atas file kritis pada Web server (IDPS berbasis host atau pengecek integritas file)
	□ Done □ Not Done □ N/A	IDPS memblokir (dalam hubungannya dengan firewall) alamat IP atau subnet yang menyerang jaringan organisasi
	□ Done □ Not Done □ N/A	IDPS memberitahu para administrator IDPS atau administrator Web server mengenai serangan melalui cara yang sesuai
	□ Done □ Not Done □ N/A	IDPS dikonfigurasikan untuk memaksimalkan deteksi dengan suatu tingkat penerimaan dari kesalahan yang positif

	□ Done □ Not Done □ N/A	IDPS dikonfigurasikan untuk me-log kejadian
	□ Done □ Not Done □ N/A	IDPS di-update dengan tanda serangan baru secara berkala (misal berdasarkan harian)
	□ Done □ Not Done □ N/A	IDPS berbasis host dikonfigurasikan untuk memonitor sumber daya sistem yang ada pada Web server host
16.	**Melakukan penilaian terhadap switch jaringan**	
	□ Done □ Not Done □ N/A	Switch jaringan digunakan untuk memproteksi jaringan dari penyadap jaringan
	□ Done □ Not Done □ N/A	Switch jaringan dikonfigurasikan pada model keamanan yang tinggi untuk mengalahkan ARP spoofing dan serangan-serangan ARP poisoning
	□ Done □ Not Done □ N/A	Switch jaringan dikonfigurasikan untuk mengirimkan semua lalu lintas pada segmen jaringan ke IDPS berbasis jaringan
	□ Done □ Not Done □ N/A	Lakukan evaluasi terhadap load balancer
	□ Done □ Not Done □ N/A	Load balancer digunakan untuk meningkatkan ketersediaan Web server
	□ Done □ Not Done □ N/A	Load balancer diperbesar dengan Web cache, jika dapat diaplikasikan
	□ Done □ Not Done □ N/A	Lakukan evaluasi terhadap reverse proxy
	□ Done □ Not Done □ N/A	Reverse proxy digunakan sebagai suatu gerbang pengamanan untuk meningkatkan ketersediaan Web server
	□ Done □ Not Done □ N/A	Reverse proxy diperbesar dengan penambahan
	□ Done □ Not Done □ N/A	Kecepatan enkripsi, otentikasi pengguna, dan kemampuan penyaring konten, jika dapat diaplikasikan
17.	**Melakukan Logging**	
	□ Done □ Not Done □ N/A	Gunakan format log kombinasi untuk menyimpan Log Transfer atau mengkonfigurasi secara manual informasi yang dijabarkan oleh format log kombinasi menjadi format standar untuk Log Transfer
	□ Done □ Not Done □ N/A	Aktifkan Log Pengacu atau Log Agent jika format log kombinasi tidak tersedia

	□ Done □ Not Done □ N/A	Buatlah nama-nama file log yang berbeda untuk situs Web virtual yang berbeda yang mungkin di implementasikan sebagai bagian dari suatu Web server fisik tunggal
	□ Done □ Not Done □ N/A	Gunakan identitas pengguna remote sebagaimana yang ditentukan dalam RFC 1413
	□ Done □ Not Done □ N/A	Simpanlah log pada suatu host yang terpisah (syslog)
	□ Done □ Not Done □ N/A	Pastikan terdapat kapasitas yang cukup untuk log
	□ Done □ Not Done □ N/A	Arsipkan log berdasarkan pada persyaratan organisasi
	□ Done □ Not Done □ N/A	Periksa kembali log harian
	□ Done □ Not Done □ N/A	Periksa kembali log mingguan (untuk trend long- term yang lebih banyak)
	□ Done □ Not Done □ N/A	Gunakan tool analisis file log yang otomatis
18.	**Melakukan pembuatan backup Web server**	
	□ Done □ Not Done □ N/A	Buatlah suatu kebijakan backup Web server
	□ Done □ Not Done □ N/A	Lakukan back up Web server secara differential atau incremental berbasis harian hingga mingguan
	□ Done □ Not Done □ N/A	Buatlah back up Web server secara penuh berbasis mingguan hingga bulanan
	□ Done □ Not Done □ N/A	Buatlah back up arsip secara periodik
	□ Done □ Not Done □ N/A	Pelihara suatu copy otoritatif situs Web
19.	**Melakukan pemulihan terhadap suatu kebobolan/insiden**	
	□ Done □ Not Done □ N/A	Laporkan insiden kepada pihak tanggap insiden keamanan komputer dari organisasi
	□ Done □ Not Done □ N/A	Isolasi sistem yang bobol atau ambil langkah-langkah lain untuk memagari serangan sehingga informasi tambahan dapat dikumpulkan
	□ Done □ Not Done □ N/A	Periksa host-host yang serupa untuk menentukan jika penyerang juga membobol sistem-sistem lain

	□ Done □ Not Done □ N/A	Konsultasikan, sebagaimana mestinya, dengan manajemen, konsultan hukum, dan kantor penegak hukum secara cepat.
20.	**Analisis intrusi yang terjadi**	
	□ Done □ Not Done □ N/A	Mengembalikan sistem seperti semula
	□ Done □ Not Done □ N/A	Menguji sistem untuk memastikan keamanan
	□ Done □ Not Done □ N/A	Menghubungkan kembali sistem ke jaringan
	□ Done □ Not Done □ N/A	Monitor sistem dan jaringan terhadap tanda-tanda bahwa penyerang berusaha mengakses kembali sistem atau jaringan kembali.
21.	**Scanning Kerentanan & Penetration Test**	
	□ Done □ Not Done □ N/A	Melakukan secara berkala scan kerentanan pada Web server, konten yang dihasilkan secara dinamis, dan jaringan pendukung
	□ Done □ Not Done □ N/A	Lakukan update scanner kerentanan sebelum pengujian
	□ Done □ Not Done □ N/A	Perbaiki kekurangan yang teridentifiikasi oleh scanner kerentanan
	□ Done □ Not Done □ N/A	Lakukan Penetration Testing pada Web server dan infrastruktur jaringan pendukung
	□ Done □ Not Done □ N/A	Perbaiki kekurangan apapun yang teridentifikasi oleh uji penetrasi
	□ Done □ Not Done □ N/A	Gunakan suatu mekanisme otentikasi yang kuat (misal, pasangan kunci publik/privat, otentikasi dua-faktor)
	□ Done □ Not Done □ N/A	Batasi host yang dapat digunakan untuk mengelola secara remote atau mengupdate konten pada Web server dengan IP address dan ke jaringan internal
22.	**Menggunakan protokol yang aman (misal, SSH, HTTPS)**	
	□ Done □ Not Done □ N/A	Terapkan konsep least privilege pada administrasi yang dilakukan secara remote dan update konten (misalkan meminimalkan hak akses untuk kedua hal ini)
	□ Done □ Not Done □ N/A	Rubah akun atau password default dari perangkat atau aplikasi administrasi yang dilakukan secara remote
	□ Done □ Not Done □ N/A	Tidak membolehkan administrasi dilakukan secara remoter melalui internet kecuali dengan menggunakan Secure Shell atau VPN
	□ Done □ Not Done □ N/A	Tidak melakukan sharing file Web server pada jaringan internal

VI. DAFTAR REFERENSI

1. OWASP (The Open Web Application Security Project)

   Web : http://www.owasp.org

   a. OWASP Top 10 – 2017 The Ten Most Critical Web Application Security Risks, 2017

   b. OWASP Testing Guide 4.0

   c. OWASP Code Review Guide 2.0

   d. OWASP Application Security Verification Standard 3.0.1, 2016

   e. OWASP Hardening IIS

   f. OWASP Secure Coding Practices Quick Reference Guide, 2010

2. ISO 27001:2015 – Information Security Management System, 2015

3. NIST (National Institute of Standards & Technology)

   a. Managing Information Security Risk, Special Publication 800-39, 2011

   b. Risk Management Framework For Information Systems And Organizations : A System Life Cycle Approach for Security and Privacy, NIST Special Publication 800-37, Revision 2, 2018

   c. Computer Security Incident Handling Guide, Special Publication 800-61 Revision 2, 2012

   d. Guide To Intrusion Detection And Prevention Systems (IDPS), Special Publication 800-94, 2007

   e. Guide to Computer Security Log Management, Special Publication 800-92, 2006

   f. Guide to Enterprise Patch Management Technologies, Special Publication 800-40 Revision 3, 2013

4. Peta Okupasi Nasional Dalam Kerangka Kualifikasi Nasional Indonesia Pada Area Fungsi Keamanan Siber (Mapping The Future of Cybersecurity Workforce in Indonesia), Badan Siber & Sandi Negara, 2019.

5. How to Secure Your Website 5th Editon - Approaches to Improve Web Application and Website Security, IT Security Center (Isec) Information-Technology Promotion Agency, Japan, 2011.

   Web : http://www.ipa.go.jp/security/english/third.html#websecurity

6. IIS-8-Server-Hardening-Handbook, The Centre for Internet Security (CIS), 2020.

   Web : https://www.goa.gov.in/wp-content/uploads/2020/01/IIS-8-Server-Hardening-Handbook.pdf

7. Apache Web Server Security and Hardening Tips, Tecmint, 2016

   Web : https://www.tecmint.com/apache-security-tips/

8. Top 25 Nginx Web Server Best Security Practices, Nixcraft, 2017

Web : https://www.cyberciti.biz/tips/linux-unix-bsd-nginx-webserver-security.html

9. Panduan Operasional Penanganan Insiden Siber, Pusat Operasi Siber, Kementerian Pertahanan, 2013

10. Panduan Keamanan Web Server, Direktorat Keamanan Informasi, Direktorat Jenderal Aplikasi Informatika, Kementerian Komunikasi & Informatika, 2011

11. Suwarno Pribadi, Karya Akhir : Perancangan Pedoman Keamanan Pengembangan Aplikasi, Fakultas Ilmu Komputer Program Studi Magister Teknologi Informasi Jakarta, 2013.

12. Permenhan RI No. 82 Tahun 2014 tentang Pedoaman Pertahanan Siber, 2014