**KEBIJAKAN TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB**

**DI {{nama oraganisasi}}**

**{{logo organisasi}}**

|  |  |
| --- | --- |
| No Dokumen | : 0.1 (Draft) |
| Revisi | : 00 |
| Tanggal Terbit | : |

|  |  |  |
| --- | --- | --- |
| Disusun  oleh : | Diketetahui  oleh: | Disetujui / Disahkan  oleh : |
|  |  |  |
| **NN**  NN | **NN**  NN | **NN**  NN |

**Riwayat Dokumen**

|  |  |  |
| --- | --- | --- |
| **Versi** | **Tanggal** | **Perubahan** |
| 0.1 (Draft) | 28/02/2023 | Rilis draft awal dengan hanya menyalin sepenuhnya konten dari **PEDOMAN TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB - BSSN** |
| … | … | … |
|  |  |  |

**KEBIJAKAN TATA KELOLA KEAMANAN APLIKASI BERBASIS WEB**



**KATEGORISASI & MANAJEMEN RISIKO PENGAMANAN APLIKASI BERBASIS WEB**

**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.

**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*.
   * + 1. **JENIS RISIKO KEAMANAN APLIKASI BERBASIS WEB**
9. **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.

1. **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.

1. **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.

1. **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.

* + - 1. **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 |

* + - 1. **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/kerus akan data tetapi bukan data sensitive/rahasia | Ada kebocoran  /kehilangan/kerusak an data sensitif/rahasia dan berdampak signifikan secara internal | Ada kebocoran  /kehilangan/kerusakan data sensitif berdampak signifikan secara nasional. |

* + - 1. **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 | 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 |
| Prevalensi Kelemahan: 2 |
| Kelemahan Deteksi: 3 |

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | *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. |
| Prevalensi Kelemahan: 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| Kelemahan Deteksi: 2 |  | * 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. |
| Dampak Teknis: 3 |
|  | | | |
| Dapat Diekploitasi: 3 | *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 |
| Prevalensi Kelemahan: 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| Kelemahan Deteksi: 2 |  | 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. |
| Dampak Teknis: 3 |
|  | | | |
| Dapat Diekploitasi: 3 | *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 |
| Prevalensi Kelemahan: 3 |
| Kelemahan Deteksi: 3 |
| Dampak Teknis: 2 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 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 | *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, |
| Prevalensi Kelemahan: 2 |
| Kelemahan Deteksi: 3 |
| Dampak Teknis: 3 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | diteruskan ke kerangka kerja SOAP. | gateway keamanan API, atau malah Web Application Firewalls (WAFs) untuk mendeteksi, memantau, dan memblokir serangan XXE. |
|  | | | |
| Dapat Diekploitasi: 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. |
| Prevalensi Kelemahan: 3 |
| Kelemahan Deteksi: 3 |
| Dampak Teknis: 2 |
|  | | | |
| Dapat Diekploitasi: 3 | 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 |
| Prevalensi Kelemahan: 3 |
| Kelemahan Deteksi: 3 |
| Dampak Teknis: 2 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 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 | *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. |
| Prevalensi Kelemahan: 2 |
| Kelemahan Deteksi: 2 |
| Dampak Teknis: 3 |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Dapat Diekploitasi: 2 | 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. |
| Prevalensi Kelemahan: 2 |
| Kelemahan Deteksi: 2 |
| Dampak Teknis: 3 |
|  | | | |
| Dapat Diekploitasi: 2 | *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 |
| Prevalensi Kelemahan: 3 |

|  |  |  |  |
| --- | --- | --- | --- |
| Kelemahan Deteksi: 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. |
| Dampak Teknis: 2 |
|  | | | |
| Dapat Diekploitasi: 2 | *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. |
| Prevalensi Kelemahan: 3 |

|  |  |  |  |
| --- | --- | --- | --- |
| Kelemahan Deteksi: 2 |  | * 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 |
| Dampak Teknis: 3 |



**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.

**UPAYA PREVENTIF TERHADAP KEAMANAN KONTEN APLIKASI BERBASIS WEB**

* + - 1. **INFORMASI/DATA YANG TIDAK DAPAT DIPUBLIKASI**

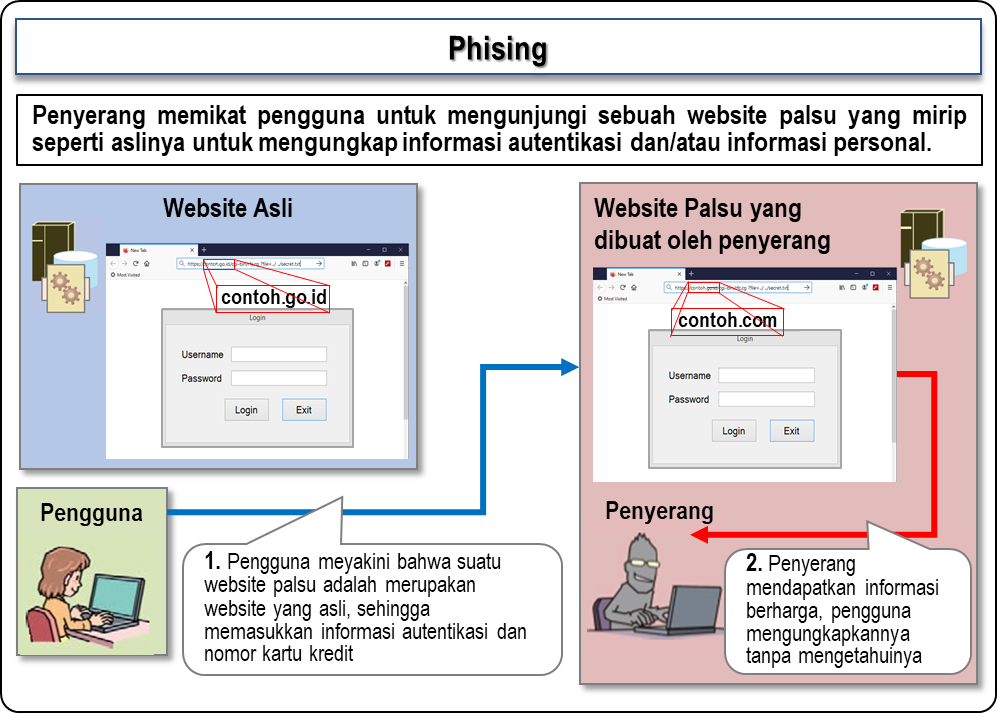
Setiap instansi harus memiliki daftar informasi/data yang tidak dapat dipublikasikan melalui Aplikasi Berbasis Web diantaranya:

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](mailto: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.
    * + 1. **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.
    * + 1. **MENGURANGI SERANGAN TIDAK LANGSUNG TERHADAP KONTEN**
11. Serangan Phising. Untuk meminimalisasi dampak dari serangan phising, maka setiap instansi harus:
    1. 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.
    2. Memvalidasi komunikasi resmi dengan mempersonalkan email (membuat email menurut selera) dan menyediakan informasi identifikasi yang unik yang seharusnya hanya diketahui oleh instansi dan pengguna.
    3. Menggunakan signature pada e-mail sekalipun signature tidak dapat divalidasi secara otomatis oleh aplikasi e-mail pengguna.
    4. Menjalankan validasi konten dalam lingkup aplikasi Aplikasi Berbasis Web. Kerentanan dalam aplikasi Aplikasi Berbasis Web mungkin digunakan dalam suatu serangan phishing.
    5. Membuat konten Aplikasi Berbasis Web tambahan sebagai identifikasi atas situs web yang sah.
    6. Menggunakan otentikasi berbasis token atau otentikasi dua arah (mutual authentication) pada situs web untuk mencegah para pelaku phising dari penggunaan kembali informasi.
    7. Situs Web yang menyimpan *Personally Identifiable Information*/PII harus mempertimbangkan dengan cermat untuk menerapkan anti-phishing yang lebih kuat.
    8. Situs Web dengan informasi publik yang tidak sensitif tidak perlu mengimplementasikan anti-phishing canggih yang menghabiskan banyak biaya.

**Gambar 2.1 Visualisasi Mekanisme Phising**



1. Serangan Pharming. Untuk meminimalisasi dampak dari serangan pharming, maka setiap instansi harus:
   1. 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.
   2. Mengimplementasikan Proteksi Server-Side DNS Terhadap Pharming―Ada beberapa tool yang tersedia untuk mengurangi ancaman terhadap perangkat lunak DNS, seperti DNS Security Extensions
   3. 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.
   4. 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](http://www.pajak.go.id/)
   5. 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.
   6. 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.
      * 1. **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.

**UPAYA PREVENTIF TERHADAP KEAMANAN WEB SERVER**

* + - 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.
   * + 1. **KEAMANAN APLIKASI APLIKASI BERBASIS WEB**
6. 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.
7. 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.
8. Identifikasi kebutuhan keamanan dapat dikategorikan menjadi *functional security requirement, non-functional security requirement, derived security requirement, user stories, dan abuse case*
9. Perancangan keamanan sistem perlu dimulai dengan kajian risiko keamanan sistem.
10. Menentukan dan mendokumentasikan peran dan tanggungjawab keamanan informasi dalam SDLC.
11. Menentukan individu-individu yang memiliki peran dan tanggungjawab dalam keamanan informasi dan mengintegrasikan proses manajemen risiko Instansi ke dalam aktivitas SDLC.
12. Lingkungan Pengembangan (Development Environment) harus menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak dengan fasilitas sebagai berikut:
    * [Editor,](https://id.wikipedia.org/wiki/Editor) yaitu fasilitas untuk menuliskan kode sumber dari perangkat lunak.
    * [Compiler,](https://id.wikipedia.org/wiki/Compiler) yaitu fasilitas untuk mengecek sintaks dari kode sumber kemudian mengubah dalam bentuk binari yang sesuai dengan bahasa mesin.
    * [Linker,](https://id.wikipedia.org/w/index.php?title=Linker&action=edit&redlink=1) yaitu fasilitas untuk menyatukan data binari yang beberapa kode sumber yang dihasilkan [compiler](https://id.wikipedia.org/wiki/Compiler) sehingga data-data binari tersebut menjadi satu kesatuan dan menjadi suatu program komputer yang siap dieksekusi.
    * [Debuger,](https://id.wikipedia.org/w/index.php?title=Debuger&action=edit&redlink=1) yaitu fasilitas untuk mengetes jalannya program, untuk mencari [*bug*/](https://id.wikipedia.org/wiki/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.
13. 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

1. 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
2. Lingkungan Pengujian:
   * Lingkungan untuk uji integrasi (integration testing) dan jaminan kualitas (quality assurance).
   * Mengelola rilis versi aplikasi untuk menjaga stabilitas
3. Standarisasi penggunaan tools pemrograman yang dapat mengendalikan ancaman injeksi SQL baik bagi vendor ataupun internal berupa penggunaan API / Interpreter yang aman.
4. Memasukan kegiatan verifikasi keamanan aplikasi untuk ancaman injeksi SQL dalam setiap kegiatan ujicoba aplikasi yang sedang dikembangkan dan akan dioperasionalkan
5. Skenario ujicoba yang mengevaluasi hasil konstruksi aplikasi berdasarkan pedoman keamanan aplikasi harus dibuat dan dilaksanakan oleh fungsi audit.
6. 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
7. 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
8. Penggunaan token unik terenkripsi untuk setiap URL permintaan persetujuan yang dikirimkan melalui suatu email yang kode parameternya tidak mudah dipahami user.
9. 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.
10. Standar konfigurasi server web aplikasi yang telah memperhatikan faktor keamanan dari ancaman serangan.
11. Arsitektur aplikasi yang kuat yang menyediakan pemisahan dan keamanan yang tegas antar komponen.
12. Menjalankan scan dan melakukan audit secara periodik untuk membantu mendeteksi kesalahan konfigurasi atau patch yang hilang di masa mendatang.
13. 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.
14. 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.
15. 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.
16. 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.
17. 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.
18. 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.

**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.

**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



**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.



**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



**PENGENDALIAN INSIDEN KEAMANAN APLIKASI BERBASIS WEB**

**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:   1. Web Deface   2. Account Compromise * Cyber Harashment:   1. Cyber Bully   2. Cyber Stalking * Fraud:   1. Phishing/Pharming   2. 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:   1. Port Scanning   2. Brute Force * Malicous code:   1. Botnet CNC   2. Malware   3. 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:   1. Junk mail   2. Spam Relay   3. 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 |

**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:
   1. Helpdesk
   2. Personel pemantauan deteksi intrusi
   3. Administrator sistem
   4. Administrator firewall
   5. Mitra/Pihak ketiga
   6. Manajer
   7. Petugas keamanan
   8. Sumber lainnya

Mereka yang di Unit Pengelola TI mungkin memiliki prosedur kontak yang berbeda dari yang di luar unit pengelola TI.

1. Jika orang yang menemukan insiden tersebut adalah anggota Unit Pengelola TI atau departemen yang terdampak-insiden, mereka akan melanjutkan ke langkah 5.
2. 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.
3. 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:
   1. Nama pelapor
   2. Waktu panggilan
   3. Informasi kontak tentang pelapor
   4. Jenis kejadian
   5. Peralatan atau orang yang terlibat
   6. Lokasi peralatan atau orang yang terlibat
   7. Bagaimana insiden itu terdeteksi
   8. Ketika peristiwa pertama kali diperhatikan, gagasan apakah yang mendukung bahwa ada terjadi sebuah insiden
4. 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:
   1. Apakah peralatan mempengaruhi bisnis yang kritis?
   2. Seberapa parah dampak potensial?
   3. Nama sistem yang terdampak, sistem operasi, alamat IP, dan lokasi
   4. Alamat IP dan informasi apa pun tentang asal serangan
5. Anggota tim respons yang dihubungi akan meninjau atau membahas situasi melalui telepon dan menentukan strategi respons.
   1. Apakah insiden itu nyata atau diduga?
   2. Apakah insiden itu masih berlangsung?
   3. Data atau apa saja yang terancam dan seberapa kritisnya?
   4. Apa dampaknya terhadap bisnis jika serangan berhasil? Minimal, serius, atau kritis?
   5. Sistem atau bagian apa yang terdampak, di mana mereka ditempatkan secara fisik dan di struktur jaringan?
   6. Apakah insiden di dalam jaringan-terpercaya (trusted network)?
   7. Apakah responsnya mendesak?
   8. Bisakah insiden itu dengan cepat terkendali?
   9. Apakah respons akan memperingatkan penyerang dan apakah kita peduli bila penyerang tahu?
   10. Jenis insiden apa ini? Contoh: virus, worm, intrusi, penyalahgunaan, kerusakan
6. Tiket insiden dikategorikan ke dalam level tertinggi yang berlaku dari salah satu kategori berikut:
   1. Kategori tinggi - Ancaman terhadap keselamatan publik, reputasi instansi & data sensitif
   2. Kategori medium - Ancaman terhadap sistem
   3. Kategori rendah - Gangguan layanan
7. Anggota tim akan menetapkan dan mengikuti salah satu dari prosedur berikut berdasarkan respons mereka pada penilaian insiden, misalnya:
   1. Prosedur respons worm
   2. Prosedur respons virus
   3. Prosedur kegagalan sistem
   4. Prosedur respons intrusi aktif
   5. Prosedur respons intrusi tidak aktif
   6. Prosedur penyalahgunaan sistem
   7. Prosedur respons pencurian properti
   8. Prosedur penolakan situs web terhadap layanan
   9. Prosedur penolakan layanan database atau file
   10. Prosedur respons spyware
   11. 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.

1. Anggota tim akan menggunakan teknik forensik, yaitu:
   1. Mengidentifikasi
   2. mempreservasi
   3. Mengkoleksi
   4. Mengeksaminasi
   5. Menganalisis
   6. Mempresentasi
   7. Memutuskan

Hanya personel yang berwenang yang boleh melakukan wawancara atau memeriksa bukti, dan personel yang berwenang dapat berbeda-beda tergantung situasi dan organisasi.

1. Anggota tim akan merekomendasikan perubahan untuk mencegah terjadinya infeksi pada sistem lainnya.
2. Setelah persetujuan manajemen, perubahan akan diterapkan.
3. Anggota tim akan mengembalikan sistem yang terdampak ke kondisi kerja-normal. Mereka dapat melakukan salah satu atau lebih hal berikut ini:
   1. Menginstal ulang sistem yang terkena dampak dari awal dan memulihkan data dari backup/cadangan jika perlu. Simpan bukti sebelum melakukan ini.
   2. Membuat pengguna mengubah kata sandi jika kata sandi mungkin telah diretas.
   3. Memastikan sistem telah diperkeras (harden) dengan mematikan atau menghapus instalasi layanan yang tidak digunakan.
   4. Memastikan sistem telah ditampal (patch) sepenuhnya.
   5. Memastikan perlindungan virus dan deteksi intrusi waktu-nyata berjalan.
   6. Memastikan sistem mencatat peristiwa yang benar dan ke tingkat yang tepat.
4. Dokumentasi, berikut ini harus didokumentasikan:
   1. Bagaimana insiden ditemukan
   2. Kategori insiden
   3. Bagaimana insiden itu terjadi, baik melalui email, firewall, dll
   4. Dari mana insiden itu berasal, seperti alamat IP dan informasi terkait lainnya tentang insiden
   5. Bagaimana rencana responsnya
   6. Tindakan apa yang dilakukan sebagai respons?
   7. Efektif atau tidaknya dari respons
5. Preservasi/Perlindungan Bukti, tim akan membuat:
   1. Bagaimana insiden ditemukan
   2. Membuat salinan log, email, dan komunikasi lainnya
   3. Menyimpan daftar saksi
   4. Menyimpan bukti selama diperlukan
6. Memberi tahu pihak eksternal yang tepat bila diperlukan, misalnya:
   1. Memberi tahu polisi
   2. Memberi tahu agensi lain yang sesuai jika penuntutan terhadap penyusup dimungkinkan
7. Membuat penilaian kerusakan dan biaya, yaitu:
   1. Menilai kerusakan pada organisasi
   2. Memperkirakan biaya kerusakan
   3. Menghitung biaya upaya pencegahan
8. Meninjau respons dan memperbarui kebijakan untuk merencanakan dan mengambil langkah pencegahan agar gangguan tidak terjadi lagi, yaitu antara lain:
   1. Mempertimbangkan apakah kebijakan tambahan dapat mencegah intrusi
   2. 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
   3. Apakah respons insiden sesuai? Bagaimana itu bisa diperbaiki?
   4. Apakah setiap pihak yang tepat mendapat informasi tepat waktu?
   5. Apakah prosedur penanganan insiden terperinci dan apakah mereka mencakup seluruh situasi? Bagaimana mereka dapat ditingkatkan?
   6. Sudahkah perubahan dibuat untuk mencegah infeksi ulang? Apakah semua sistem telah ditampal, sistem dikunci, kata sandi diubah, pembaruan anti-virus, kebijakan email ditetapkan, dll?
   7. Apakah telah dilakukan perubahan untuk mencegah infeksi baru dan serupa?
   8. Haruskah ada kebijakan keamanan diperbarui?
   9. Pelajaran apa yang bisa dipetik dari pengalaman ini?



**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:

|  |
| --- |
| **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 |
| **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 |

* + - 1. **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

|  |
| --- |
| **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 |



|  |
| --- |
| * + - 1. **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       2. **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* |
| **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

|  |
| --- |
| **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* |