Government Computer Security Incident Response Team

BADAN PENGKAJIAN DAN PENERAPAN TEKNOLOGI

PANDUAN PENANGANAN INSIDEN

WEB DEFACEMENT

Diadopsi dari : SOP Incident Handling Web Defacement
Kementrian Komunikasi dan Informastika Republik Indonesia

BUKU PETUNJUK PELAKSANAAN

BPPT CSIRT
2014

DAFTAR ISI

BAGIAN 1 : PENDAHULUAN

Operasional Penanganan Web Defacement adalah adalah menghindari atatau menahan adanya serangan pada sebuah situs web yang mengubah tampilan visual dari situs atau halaman web. Ini biasanya disebabkan kerja Cracker, yang masuk ke server web dan mengganti situs host dengan salah satu hasil dari kerja mereka sendiri. Defacement umumnya dimaksudkan sebagai semacam grafiti elektronik, meskipun bisa juga menjadi sarana untuk menyebarkan pesan bermotif politik atau bentuk unjuk rasa.

1.1 TUJUAN

Penanganan yang terencana dan terorganisir sangatlah diperlukan dalam hal terjadinya insiden web defacement, supaya hal itu bisa dilakukan, maka perlu adanya suatu prosedur yang standar untuk melakukan penanganan terhadap insiden tersebut. Secara umum tujuan prosedur standar ini adalah untuk memberikan arahan secara best practices dalam penanganan insiden web defacement, sedangkan secara khusus adalah seperti yang dijabarkan sebagai berikut :

a. Memastikan adanya sumber daya yang memadai untuk menangani insiden yang terjadi
b. Menjamin pihak-pihak yang bertanggung jawab dalam penanganan insiden bekerja sesuai dengan tugas dan kewajiban masing-masing
c. Menjamin aktivitas dari penanganan insiden dapat terkoordinasi dengan baik
d. Melakukan pengumpulan informasi yang akurat
e. Sharing pengetahuan dan pengalaman di antara anggota tim penanganan insiden
f. Meminimalisir dampak dari insiden yang terjadi.
g. Mencegah adanya serangan lanjutan dan mencegah kerusakan agar tidak lebih meluas

1.2 RUANG LINGKUP

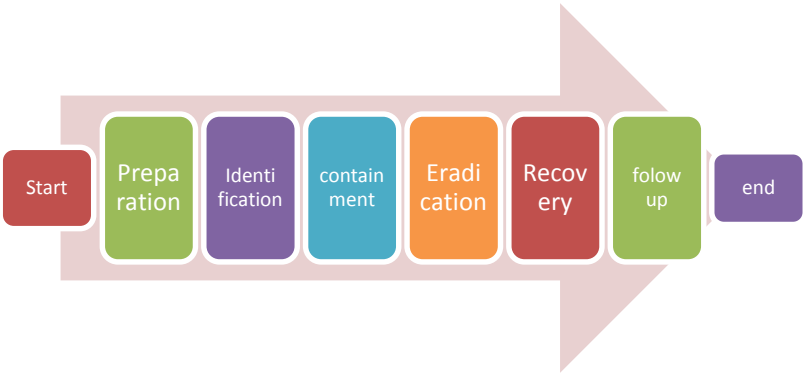
Prosedur standar penanganan insiden ini berisi langkah-langkah yang harus diambil apabila terjadi insiden web defacement, yang dimulai dari tahap persiapan sampai dengan tahap pembuatan laporan dari penanganan insiden. Web Defacement dapat terjadi pada semua halaman web, baik web utama www.bppt.go.id maupun web lain dalam subdomain bppt.go.id .

Prosedur standar penanganan insiden ini dapat dijadikan acuan bagi semua individu atau tim yang bertindak sebagai penanggung jawab/administrator dari suatu web server.

BAGIAN 2 :
PROSEDUR PENANGANAN WEB DEFACEMENT

Terdapat 2 cara bagi institusi/perorangan untuk meletakan suatu halaman web, yaitu meletakan pada server yang dikelola sendiri atau meletakan halaman webnya pada web hosting. Bagi yang meletakkan halaman webnya pada web hosting, maka apabila terjadi web deface harus melakukan koordinasi dengan pihak web hosting. Koordinasi ini ditujukan untuk memudahkan penanganan dari web yang telah terdeface. Setiap pengelolah web hosting seharunya memiliki prosedur untuk menangan insiden web defacement.

Penangan terhadap insiden WEB Defacement dapat dilakukan dalam beberapa tahap seperti pada gambar berikut,



Gambar 1 : Tahap-tahap penanganan insiden

2.1 Tahap Persiapan (Preparation)

a. Pembentukan tim penanganan insiden.
   Anggota tim penanganan insiden bisa berasal juga dari pihak luar institusi kalau memang sangat diperlukan.
b. Menentukan metode koordinasi dan komunikasi antara tim dan penanggung jawab dari web server. Kapan koordinasi harus dilakukan, dan melalui media apa komunikasi akan dilakukan harus ditentukan.
   Media untuk melakukan koordinasi dan komunikasi harus memenuhi beberapa syarat, antara lain
   - Harus selalu tersedia
   - Harus aman (informasi rahasia yang disalurkan melalui media tidak boleh bocor).
c. Menyiapkan dokumen yang dibutuhkan dalam proses penanganan insiden.
   Dokumen ini antara lain adalah :
   - Standar Operation Procedure
   - Form-form yang akan digunakan.
   - Gambaran diagram terbaru yang menggambarkan hubungan antar komponen-komponen aplikasi yang membangun web site (web server, aplikasi web, para user, diagram network).
   - Dokumentasi dari sistem operasi, aplikasi, protocoldan anti virus yang terdapat pada web server.
d. Menentukan tempat (ruangan) untuk menangani insiden.
e. Menyiapkan tool dan media yang dibutuhkan untuk menangani insiden.
f. Membangun website cadangan sementara,dimana organisasi dapat mempublikasikan konten website.
g. Menentukan prosedur untuk mengarahkan setiap pengunjung yang akan berinteraksi dengan website cadangan yang telah dibuat.
h. Memasang alat monitoring (misal Wireshark) untuk mendeteksi secara cepat terhadap setiap perilaku proses yang tidak lazim

terhadap aplikasi web.

i. Melakukan export file log web server ke server external, pastikan terjadi sinkronisasi waktu antar kedua server.
j. Menentukan kontent dari pihak ekternal (statik atau dinamik) dan membuat daftar dari konten-konten tersebut.
k. Jika website terletak pada web hosting, pastikan penyedia web hosting mempunyai aturan untuk mengaktifkan log terhadap semua kejadian pada web server.
l. Tidak mengubah peta dan konfigurasi jaringan yang ada.

2.2 Tahap Identifikasi dan Analisa

Tujuan dari proses analisa adalah:

a. Memahami sifat dan ruang lingkup kejadian
b. Mengumpulkan informasi yang cukup tentang insiden itu sehingga tim respon dapat memprioritaskan langkah selanjutnya dalam menangani insiden tersebut, yang bisaanya diikuti dengan penahanan sistem.
c. Menentukan apakah terdapat data rahasia yang terlibat dalam insiden tersebut.

Pada tahap ini dilakukan proses identifikasi untuk memastikan insiden yang telah terjadi. Konfirmasi bahwa insiden web defacement telah terjadi dapat dilakukan dengan prosedur dibawah ini,

a. Memonitor kembali halaman web untuk mengetahui halaman web yang telah diubah.

b. Mencari informasi atau pemberitahuan dari para pengguna halaman web tentang problem yang mereka temui saat melakukan browsing terhadap website

c. Menggunakan tool security checks (Zone-h)

Memastikan telah terjadi perubahan halaman web dan mendeteksi sumbernya:

a. Memeriksa file-file yang bersifat statis, apakah terjadi perubahan dan kapan perubahan itu terjadi.

b. Memeriksa komponen dan kandungan dari komponen mashups. Komponen mashup adalah komponen pembangun web site yang berasal dari berbagai situs web. Mashup adalah sebuah situs web atau aplikasi web yang mengombinasikan konten dari banyak sumber ke dalam sebuah portal yang terintegrasi.

c. Memeriksa semua link yang ada pada halaman web (src, meta, css, script).

d. Memeriksa semua log file,
Apache Server pada dasarnya memiliki 2 log:
1. Access Log
-Berisi semua request yang telah diproses oleh apache
-Lokasi file ini secara default berada pada
usr/local/apache/logs/access.log
2. Error log
-Berisi informasi diagnostik dan pesan kesalahan yang ditemukan saat memproses suatu request di apache server
-Lokasi file ini secara default berada di
/usr/local/apache/logs/error.log

e. Memeriksa apakah ada malicious file yang telah tertanam pada web server. Memeriksa adanya command line, string-string yang digunakan untuk menyerang, php file.

f. Memeriksa database untuk menemukan isi yang berbahaya.

Mengukur dampak dari terjadinya web defacement,

i. Terhadap kelangsungan proses bisnis, indikatornya adalah seberapa besar dari fungsi fungsi bisnis yang terdapat pada web mengalami gangguan.

ii. Terhadap sistem dan informasi, indikatornya adalah apakah ada data dan informasi yang berubah atau terhapus, baik pada web server maupun pada server lain yang terhubung langsung dengan web server.

iii. Kemampuan untuk recovery ke halaman web yang asli, indikatornya seberapa besar sumber daya dan waktu yang dibutuhkan untuk memulihkan web server.

2.3 Tahap Pengumpulan materi (Containtment)

Setelah dipastikan bahwa memang benar telah terjadi web deface, maka dilakukan proses berikutnya dengan tujuan:

1. Tidak terjadi kerusakan lebih dalam pada web server
2. Melindungi server-server lain yang terhubung dengan web server

Prosedur yang dilakukan pada tahap ini adalah:

a. Melakukan proses backup semua data yang terdapat pada web server untuk keperluan forensik dan pengumpulan bukti-bukti. Backup sebaiknya ditempatkan pada hard disk yang masih kosong.
b. Memastikan bahwa exploitasi kerentanan oleh penyerang terjadi apada web server yang akan diperiksa, bukan ditempat lain.
c. Melakukan pemeriksaan pada sistem dimana web server berjalan.
d. Melakukan pemeriksaan pada seluruh service yang berjalan pada server.
e. Melakukan pemeriksaan pada semua koneksi dengan sistem lain yang berhubungan dengan server yang mungkin telah menginfeksi server.
f. Jika sumber penyerangan berasal dari sistem lain pada jaringan, maka putuskan secara fisik koneksi tersebut dan lakukan investigasi terhadap sumber tersebut.
g. Temukan darimana, tempat, dan bagaimana cara penyerang memasuki sistem saat pertama kali.
h. Melakukan pemeriksaan untuk menemukan kerentanan dari komponen web yang bisa diakses untuk diedit (memiliki access

write).

i. Melakukan pemeriksaan apakah terdapat kode-kode berbahaya (trojan, backdoor) yang tersimpan dalam web server.
j. Memeriksa folder-folder yang bersifat publik.
k. Memeriksa kelemahan kode sql yang digunakan.
l. Melakukan inventarisasi terhadap semua kerentanan yang telah ditemukan.
m. Jika ternyata terjadi kesulitan (masalah sulit dipecahkan), maka harus membuat website baru yang sifatnya sementara dimana isinya sama dengan website yang telah rusak diserang.

Beberapa tool yang bisa digunakan pada tahapan ini,
- Nmap, untuk mengetahui port yang terbuka
- Nessus, untuk mengetahui vulnerability

2.4 Tahap Penghapusan Konten (Eradication)

Proses ini bertujuan untuk,

m. Menyimpan bukti-bukti yang telah dimiliki
n. Melakukan analisa tambahan untuk menyelesaikan investigasi
o. Melepaskan komponen-komponen yang bisa menyebabkan gangguan pada sistem
p. Mengurangi vector serangan sehingga kejadian serupa tidak terjadi (misalnya, patch

Prosedur untuk melakukan proses ini dapat dilakukan dengan cara berikut,

a. Menghapus semua malicious content, termasuk content deface dan merubah kembali dengan konten asli.
b. Memulihkan kembali data dari backup file, namun perlu dipastikan bahwa pada data yang dipulihkan tidak terdapat vulnerability yang dapat menjadi celah masuk kembali oleh penyerang
c. Menghapus aplikasi-aplikasi yang mencurigakan yang masih berjalan, jika memungkinkan uninstall aplikasi yang sekiranya tidak dibutuhkan dan hanya menjalankan service yang perlu saja

d. Patching terhadap aplikasi yang rentan, melakukan upgrade terhadap web server, atau juga CMS / web aplikasi yang masih memiliki kerentanan.
e. Memeriksa apakah terdapat backdoor atau rootkit yang tertanam pada server, dan segera menghapus backdoor dan rootkit tersebut
f. Melakukan vulnerability assessment untuk melihat seberapa besar celah yang masih terdapat pada server dan aplikasi yang dijalankan.

2.5 Tahap Pemulihan (Recovery)

Pada tahapan ini bertujuan untuh memulihkan kembali halaman web kepada keadaan semula.

a. Jika untuk masuk menuju web server membutuhkan otentikasi pengguna, maka ubah semua id dan password pengguna. Hal ini dilakukan dalam rangka kehati-hatian apabila terdapat id dan password pengguna yang telah diambil oleh pihak lain (penyerang).
b. Jika terdapat backup komponen aplikasi dari web server, pasang kembali komponen- komponen aplikasi dari web server.
c. Jika peristiwa web defacement telah diketahui oleh publik, berikan penjelasan kepada publik mengenai insiden tersebut.
d. Memastikan bahwa semua langkah yang diambil sebelumnya sesuai dengan prosedur yang benar.
e. Memetakan semua kerentanan yang telah diketahui.
f. Menutup semua kerentanan yang terdapat pada web server dan melindungi/proteksi kembali web server sesuai dengan standard yang telah ditentukan. Standar keamanan secara periodik harus selalu ditingkatkan.
g. Melakukan penetration testing untuk mengetahui celah-celah keamanan yang mungkin masih ada pada web server.

2.6 Tahap Tindak Lanjut

Tujuan dari tahap ini adalah

I. Mengambil pelajaran dan membuat rekomendasi untuk mencegah insiden serupa terjadi lagi
II. Mengeluarkan laporan akhir
III. Menyediakan Bukti Arsip dan dokumentasi
IV. Menutup proses penanganan insiden

Prosedur yang dapat dilakukan adalah sebagai berikut:

a. Menjabarkan teknik serangan serta vulnerability yang terdapat pada webserver
b. Membuat semua dokumentasi dan laporan terkait kegiatan dan waktu yang dibutuhkan pada proses incident handling yang telah dilakukan
c. Menuliskan tools apa saja yang digunakan dalam membatu proses incident handling
d. Menuliskan bukti-bukti yang ditemukan, hal ini terkait dengan proses hukum kedepannya
e. Memberikan analisa dan penjelasan apa yang harus dilakukan sehingga incident serupa tidak terulang kembali.
f. Membuat evaluasi dan rekomendasi.

LAMPIRAN A - Informatif

A. Pengertian Web Defacement

Secara harfiah arti Web Defacement adalah pengubahan halaman web, pengubahan halaman web ini tidak diketahui oleh pemilih sah dari web. Para pemilik sah dari web, bisaanya akan mengetahui ada proses pengubahan apabila telah terjadi perubahan pada tampilan dari web mereka. Perubahan pada tampilan web bisa berupa berubahnya gambar atau teks maupun link terhadap halaman web lain.

Tujuan seseorang melakukan web defacement bisaanya adalah hanya ingin menunjukkan kemampuan dirinya. Pada umumnya situs web yang dirusak memanfaatkan kerentanan yang terdapat pada web server untuk mendapatkan shell root, yang kemudian menyuntikkan kode berbahaya ke halaman web target yang berada pada server.

Secara umum seorang hacker bisa melakukan penyerangan terhadap suatu system melalui beberapa tahap sebagai berikut:

1. Tahap pengintaian dan penjejakan

Pada tahap ini penyerang akan mengumpukan informasi sebanyak-banyaknya tentang target yang akan diserang. Informasi awal yang bisaanya dicari adalah mengenai sistem operasi dan web server yang digunakan pada target. Untuk mendapatkan informasi ini penyerang bisa menggunakan beberapa strategi, misalnya:

- Penyerang melakukan tindakan untuk meyakinkan orang-orang yang mempunyai akses terhadap suatu sistem, penyerang berpura-pura menjadi orang yang bisa dipercaya oleh pihak lain yang memiliki akses terhadapsuatu sistem.

- Memannfaatkan orang dalam, penyerang bekerja sama dengan seseorang dalam organisasi untuk melakukan terhadap sistem yang

dimiliki oleh organisasi tersebut.

2. Pemindaian

Setelah penyerang mendapatkan informasi target dari serangannya, maka tahap selanjutnya adalah mencari celah untuk masuk ke dalam sistem.

3. Mendapatkan akses

Pada tahap ini terjadi proses untuk melakukan ekploitasi terhadap kesalahan implementasi dan kerentanan dari sistem komputer maupun jaringan. Bisaanya disebut juga dengan fase penetrasi.Penyerang mulai memanfaatkan kesalahan-kesalahan ataupun kelemahan-kelemahan yang terdapat pada suatu sistem.

4. Mempertahankan akses

Pada tahap ini terjadi proses penguasaan terhadap sistem, dan bisaanya disertai dengan proses untuk mempertahankan akses tersebut dengan cara memasang backdoor, rootkit, atau trojan. Web defacemant terjadi pada tahap ini. Sistem yang sudah dikuasai ini bisa juga dimanfaatkan untuk melakukan serangan terhadap sistem lain.

5. Membersihkan jejak

Pada tahap ini penyerang akan membersihkan jejak dengan cara memanipulasi atau menghapus bukti yang terdapat pada catatan/log sistem atau IDS yang merekam aktifitas yang telah dilakukan.

B. Metode untuk melakukan Web Defacement

Pada proses web defacement, hacker akan mencari kelemahan-kelemahan yang terdapat pada sistem web server. Karena tujuannya hanya mengubah halaman web, maka hacker akan berusaha untuk menemukan dan melakukan perubahan pada pada file index dan bahkan mengganti file index tersebut. Hacker bisaanya menggunakan tool-tool yang berfungsi untuk mengetahui kerentanan suatu website.

Tool-tool untuk memindai kerentanan ini bisa diunduh secara bebas dari internet.

Kerentanan suatu website bisa disebabkan beberapa hal, yaitu kerentanan dari program web server (IIS atau Apache), kerentanan dari sistem operasi yang digunakan, dan kerentanan dari kode program yang digunakan untuk membangun halaman web.Apabila kerentanan sudah diketahui, maka hacker akan mengexploitasi kerentanan tersebut untuk mendapatkan hak akses sebagai root/administrator dari sistem.

Dalam mengekploitasi kerentanan, para menyerang menggunakan kode-kode berbahaya atau exploit code. Exploit code adalah kode-kode program yang digunakan untuk melakukan exploitasi terhadap kerentanan suatu sistem. Kode-kode ini bisa dibuat dari bahasa pemrograman Java, C, Perl, dan lain-lain. Untuk membuat exploit code ini, seorang hacker harus menguasai bahasa pemrograman yang akan digunakan.

Exploitasi kerentanan dengan memanfaatkan kode-kode tersebut bisa dilakukan dengan berbagai cara seperti berikut:

1. SQL Injection

Sql injection merupakan jenis exploitasi keamanan halaman web, dimana penyerang akan menyisipkan kode-kode sql melalui formulir/form atau memanipulasi URL berdasarkan pada parameter sql.

Serangan injeksi SQL berupainjeksi perintah SQL berbahaya melalui form input data dari klien sesi aplikasi yang kemudian diteruskan menuju database untuk dieksekus, dan bertujuan untuk mempengaruhi pelaksanaan perintah SQL standar. Bentuk utama dari injeksi SQL terdiri dari penyisipan langsung kode ke variable masukkan pengguna yang digabungkan dengan perintah SQL dan dieksekusi. Sebuah serangan injeksi kode berbahaya ke dalam string yang ditujukan untuk tabel-tabel data dalam database atau metadata. Ketika string yang tersimpan selanjutnya berubah menjadi perintah dinamis SQL, kode berbahaya ini kemudian dieksekusi. Sebuah serangan SQL injection yang sukses dapat mengakses data sensitif dalam database. Jika

penyerang berhasil mengambil data-data root atau administrator, maka bisa dipastikan akan bisa mengambil alih sistem web server, dan bisa melakukan proses web defacement.

Contoh SQL Injection

Cara pencegahan SQL INJECTION

a. Membatasi panjang input box (jika memungkinkan), dengan cara membatasinya di kode program, jadi si cracker pemula akan bingung sejenak melihat input box nya yang tidak bisa diinject dengan perintah yang panjang.

b. Melakukan filter terhadap input yang dimasukkan oleh user, terutama penggunaan tanda kutip tunggal (Input Validation).

c. Mematikan atau menyembunyikan pesan-pesan error yang keluar dari SQL Server yang berjalan.

d. Mematikan fasilitas-fasilitas standar seperti Stored Procedures, Extended Stored Procedures jika memungkinkan.

e. Mengubah “Startup and run SQL Server” menggunakan low privilege user di SQL Server Security tab.

2. OS Command Injection

OS Command Injection adalah teknik menyerang website dengan menggunakan command/perintah yang terdapat pada sistem operasi, dengan memanfaatkan kerentanan dari sistem.

Injeksi perintah OS juga dikenal sebagai sanitisation yang tidak tepat pada elemen khusus yang digunakan dalam Command OS dan merupakan teknik yang digunakan melalui antarmuka web untuk mengeksekusi perintah OS pada server web.

Pengguna memasok semua atau bagian dari perintah OS yang tidak benar melalui antarmuka web. Jika antarmuka web yang inputnya tidak dibersihkan dengan benar akan rentan terhadap eksploitasi ini. Dengan

kemampuan untuk menjalankan perintah OS, pengguna dapat menyuntikkan perintah tak terduga dan berbahaya, meng-upload program berbahaya atau bahkan mendapatkan password secara langsung dari system operasi. Masalah ini diperparah jika proses yang baku gagal mengikuti prinsip least privileges, sebab penyerang bisa mengendalikan sistem secara penuh karena telah memiliki otorisasi/hak akses yang tinggi.

Contoh OS Command Injection

BPencegahan

Dengan memanfaatkan API

https://www.owasp.org/index.php/Command_Injection

3. Directory Transversal

Pada metode ini seorang penyerang situs akan melakukan penelusuran terhadap direktori yang terdapat pada komputer server. Penelusuran dimaksudkan untuk mencari beberapa file yang terdapat pada suatu direktori, misalnya untuk mencari lokasi file /etc/passwd pada sistem operasi Linux. Apabila lokasi file untuk password diketahui, maka penyerang akan berusaha untuk membuka file tersebut, apabila berhasil maka bisa dipastikan password untuk root/admin akan diketahui. Dengan begitu maka penyerang akan dengan mudah mengambil alih sistem.

Salah satu dari teknik ini dikenal dengan nama “unicode Directory transversal unicode”. Cara ini menggunakan protokol TCP pada port 80 HTTP untuk melewatkan code/script pada URL. Dengan menulis script URL dibawah pada web browser, dan ditujukan pada alamat website yang memiliki kerentaan, maka akan didapatkan gambaran listing direktori dari drive c:\

http://10.10.1.1/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir+c:\

Cara pencegahan

a. Menonaktifkan cmd.exe atau menjadikan cmd.exe hanya boleh diakses langsung dari local host

b. Tidak menempatkan directory server meyatu dengan partisi untuk sistem operasi, sehingga struktur direktori dari web server tidak diketahui.

4. Cross-Site Scripting

Cross-site scripting (XSS) adalah kerentanan yang bisaanya ditemukan di aplikasi Web interaktif yang memungkinkan injeksi kode berbahaya oleh pengguna web ke dalam halaman webyang juga sedang dilihat oleh pengguna lain. Hal ini bisaanya terjadi pada halaman Web yang tidak melakukan pemeriksaan terhadap data input oleh pengguna. Exploitasi pada kerentanan ini dapat digunakan oleh penyerang untuk berkompromi dengan komputer pengguna lain atau untuk menerima data dari sesi Web pengguna lain (misalnya, user ID dan password atau cookie sesi). Dengan demikian, apabila penyerang bisa berkompromi dengan komputer pengguna lain yang memiliki hak akses administrator terhadap suatu server, maka penyerang bisa menguasai server tersebut. Dengan menguasai server, berarti penyerang mempunyai kemampuan untuk melakukan

web defacement.

Konten berbahaya yang dikirim ke browser web dapat berupa beberapa bentuk termasuk JavaScript, VBScript, ActiveX, HTML, Flash atau jenis lain dari kode yang dapat dieksekusi oleh browser. Serangan XSS umumnya dapat dikategorikan menjadi tiga jenis: Tersimpan (stored), Tercermin (reflected) dan berbasis Document Object Mode ("DOM Based").

Stored XSS (Persistent)

Serangan XSS Tersimpan berarti bahwa kode berbahaya yang disuntikkan disimpan secara permanen pada server target sebagai papan buletin, log pengunjung, atau kolom komentar. Ketika berinteraksi dengan server target, pengguna tidak sengaja mengambil dan

mengeksekusi kode berbahaya dari server.

Reflected XSS (Non-Persistent)

Serangan XSS tercermin adalah kode yang disuntikkan dikirim ke server web yang rentan dan kemudian mengarahkan serangan cross-site kembali ke browser pengguna. Jenis serangan ini bertujuan untuk mengelabui pengguna dengan mengklik link berbahaya atau mengirimkan formulir yang telah dibuat secara khusus. Browser pengguna kemudian mengeksekusi kode berbahaya, dengan asumsi itu berasal dari server terpercaya.

DOM (Document Object Model) Based XSS

Berbeda dengan dua sebelumnya, XSS berbasis DOM tidak memerlukan web server untuk menerima XSS jahat. Sebaliknya, dalam XSS berbasis DOM, serangan tertanam dalam objek DOM di browser korban yang digunakan oleh script asli pada sisi klien, sehingga kode pada sisi klien berjalan secara "tak terduga". Artinya, halaman itu sendiri (HTTP response) tidak berubah, tetapi kode yang terkandung pada halaman sisi klien melakukan eksekusi yang berbeda karena modifikasi berbahaya yang telah terjadi di lingkungan DOM lokal. Serangan ini bisaanya dilakukan dengan mengirimkan URL jahat ke pengguna.

Secara singkat dapat dijelaskan proses penyerangan XSS sebagai berikut :

a. Penyerang melakukan investigasi pada situs-situs dimana diperlukan hak akses tertentu, dan mencuri hak akses suatu user melalui cookies atau ID sessions.

b. Penyerang menemukan titik rawan pada halaman suatu situs.

c. Penyerang menciptakan link khusus ke suatu situs dan menempelkannya dalam suatu email HTML yang dikirimkan ke korban potensial.

d. Link khusus tersebut juga berisi kode yang didesain khusus untuk

mengirimkan salinan cookie korban ke penyerang

e. Tanpa sepengtahuan korban, penyerang mendapatkan informasi (cookie) milik pengunjung situs target.

f. Dengan memanfaatkan informasi tersebut, penyerang mengeksploitasi situs target.

Solusi untuk serangan XSS adalah memvalidasi semua input pengguna dan menghapus data yang tidak terduga atau berpotensi berisiko. Solusi lain adalah dengan menggunakan HTML-quoted version 45 dari setiap input pengguna yang disajikan kembali ke pengguna lain

5. Cross-Site Request Forgery

Cross-site request forgery, dikenal juga dengan one click attack atau session riding disingkat dengan CSRF atau XSRF, merupakan bentuk eksploitasi website. Cross-site request forgery menipu Web site melalui request dari user yang dipercaya.

Serangan bekerja melalui link atau script pada halaman site yang diakses user. Link tersebut dapat berupa gambar/image yang terhubung ke website tertentu. Jika website menyimpan informasi otentikasi dalam sebuah cookie yang belum expire, maka dengan melakukan klik ke link tersebut akan menyebabkan website diakses menggunakan cookie user yang melakukan klik. Dengan kata lain, penyerang menipu browser user untuk mengirimkan HTTP requests ke website target.

Pencegahan :

a. Otentikasi pada website beralih dari metoda persistent authentication (menggunakan otentikasi dengan cookie atau HTTP) ke metoda transient authentication (menggunakan hidden field oleh setiap form).

b. Menyertakan token user-specific rahasia yang ditambahkan ke setiap form.

c. Meskipun cross-site request forgery pada dasarnya adalah masalah dengan aplikasi web, user dapat membantu melindungi accountnya

dengan log off site sebelum mengunjungi web yang lain atau membersihkan cookie browsernya pada akhir session browser.

d. Membatasi waktu setiap session

6. LDAP Injection

Lightweight Directory Access Protocol (LDAP) adalah protokol standar terbuka untuk proses query dan memanipulasi layanan direktori. LDAP protokol berjalan diatas protokol transport internet, seperti TCP. Aplikasi Web dapat memanfaatkan input yang disediakan pengguna untuk membuat pernyataan LDAP yang meminta layanan pada web dinamis.

LDAP injection adalah teknik serangan pada aplikasi web yang memanfaatkan data yang disediakan pengguna dalam pernyataan LDAP tanpa terlebih dahulu memeriksa karakter berpotensi berbahaya dari permintaan yang dilakukan.

Ketika sebuah aplikasi web gagal membersihkan masukan yang disediakan pengguna dengan benar, adalah mungkin bagi penyerang untuk mengubah konstruksi pernyataan LDAP. Setelah penyerang dapat memodifikasi pernyataan LDAP, proses akan berjalan dengan hak akses yang sama sebagai komponen yang dapat mengeksekusi perintah. (misalnya server Database, aplikasi server Web, server Web, dll). Hal ini dapat menyebabkan masalah keamanan serius dalam hal pemberian hak ijin untuk melakukan query, memodifikasi atau menghapus apapun yang terdapat dalam struktur LDAP.

Contoh LDAP Injection

C. Mencegah terjadinya Web Defacement

Untuk mencegah terjadinya Web Defacement, secara umum dapat dilakukan dengan cara-cara sebagai berikut :
1. Memperketat keamanan web server
Web server adalah server untuk menempatkan web site, agar web site bisa terhindar dari web defacement, maka pada web server juga harus

diterapkan pengamanan yang ketat.
Beberapa pengamanan itu harus dilakukan pada
a. Level sistem operasi
 Melakukan update sistem
Untuk sistem operasi windows : http://update.microsoft.com
 Memperbaiki sistem manajemen proses keamanan pada sistem operasi
Untuk windows:
http://www.microsoft.com/technet/security/tools/mbsa2/default.mspx
 Mematikan semua service yang tidak dipakai
 Menutup port yang tidak perlu dipublish ke internet
 Mencegah sistem supaya tidak bisa menjalankan command prompt (cmd.exe, ftp.exe, dll)
 Memonitor Log sistem, Kebanyakan penyerang (defacer) telah memasang backdoor ketika telah berhasil melakukan deface website, hal ini dimungkinkan agar dapat melakukan deface ulang terhadap website. Wajib untuk memeriksa perubahan folder, file, database dan source terakhir dari website.
 Membuat Back up sistem, file configuration.
 Membuat access control terhadap file-file tertentu (configuration file, log and system audit file, password hash file)
 Tidak membuka akses superuser/root server dari luar, artinya akses superuser/root hanya diberikan dari localhost.
b. Level web server
 Melakukan patching dan update security
Untuk IIS :http://www.iis.net/
Untuk
Apache :http://httpd.apache.org/docs/misc/security_tips.html
 Menempatkan file-file untuk konten web secara terpisah dari sistem operasi maupun aplikasi, letakkan pada disk atau partisi tersendiri
 Mematikan semua layanan yang tidak digunakan (contoh ftp, tftp, dll)
 Menghapus semua file dokumentasi dari vendor
 Mengubah semua akun login default, yang dibuat pada waktu instalasi

 Menghapus semua file contoh dari vendor, termasuk script dan file executable

 Mengubah file permission untuk file-file tertentu, seperti: htaccess, Index, file admin, file includes, file config, file content

 Konfigurasi ulang “HTTP service banner” supaya tidak memberitahukan type dan versi dari sistem operasi dan web server yang digunakan

c. Level sistem aplikasi

Apabila menggunakan CMS

 Mengubah semua konfigurasi default pada CMS

 Update dan Upgrade

 Gunakanlah tambahan plugin/component yang tepat, sehingga dapat meminimalisasi terjadinya kegiatan defacing dari third party. Pastikan hasil review & ranking plugin bereputasi baik dan sudah di verified oleh penyedia CMS yang bersangkutan.

Code scaning

Code scaning adalah melakukan pemeriksaan dan validasi terhadap kode-kode yang diinputkan ke dalam halaman web, aplikasi pada web, maupun database pendukung web dengan menggunakan aturan-aturan yang telah ditetapkan.

Keamanan kode program

Aplikasi pada halamam web bisa dibuat dari beberapa bahasa pemrograman, diantaranya php, asp, javascript. Teknik dan aturan yang salah dalam mengimplementasikan kode-kode program tersebut dapat mengakibatkan kerentanan pada aplikasi web. Dibawah ini terdapat sumber-sumber yang dapat dijadikan sebagai acuan agar tidak terjadi kesalahan dalam mengimplementasikan kode program pada aplikasi halaman web.

 Php

Setting pada file php.ini seperti dibawah ini bisa memproteksi server untuk tidak melakukan eksekusi pada shell

disable_functions = php_uname, getmyuid, getmypid, passthru, leak, listen, diskfreespace, tmpfile, link, ignore_user_abord, shell_exec, dl, set_time_limit, exec, system, highlight_file, source, show_source, fpaththru, virtual, posix_ctermid, posix_getgid, posix_getcwd, posix_getgrgid, posix_getegid, posix_getgrnam, posix_geteuid,

posix_getgroups, posix_getlogin, posix_getpgid, posix_getpgrp, posix_getpid, posix, _getppid, posix_getpwnam, posix_getpwuid, posix_getrlimit, posix_getsid, nposix_mkfifo, posix_setegid, posix_getuid, posix_isatty, posix_kill, posix_seteuid, posix_setgid, posix_setpgid, posix_times, posix_ttyname, posix_uname, posix_setsid, posix_setuid, proc_open, proc_close, proc_get_status,proc_nice, proc_terminate, phpinfo, system, passthru, shell_exec, scapeshellarg, escapeshell cmd , proc_close, proc_open, ini_alter, dl,popen, popen,pcntl_exec,socket_accept, socket_bind, socket_clear_error, socket_close, socket_connect

safe_mode = On

register_globals = Off

display_errors = Off

allow_url_fopen = Off

allow_url_include = Off

enable open_basedir(set it to webroot path)

Hal-hal lain yang perlu diperhatikan bisa dilihat dari sumber sebagai berikut: http://www.phpide.com/php-tutorials/secure-php-programming/

 ASP,

http://msdn.microsoft.com/en-us/library/ff649100.aspx

http://www.asp.net/web-api/overview/security

d. Level sistem database

 Melakukan patching dan service pack terakhir

 Memonitor log secara rutin

 Tidak memberikan akses superuser/root database pada aplikasi yang membutuhkan database

 Hanya memberikan akses database dari alamat IP tertentu

 Mengubah port default untuk koneksi antara aplikasi dan database, misalkan default port mysql 3306 dirubah ke port lain.

 Prosees pertukaran data harus menggunakan tool tertentu, misalnya:

- Key exchange: Diffie-Hellman
- Authentication: RSA
- Encryption: AES (128)
- Message digest algorithm: SHA1

2. Melindungi dari "Network Sniffing"

Network Sniffing adalah kegiatan untuk melakukan mata-mata terhadap

jaringan komputer. Tujuan kegiatan ini adalah untuk mengambil data-data penting yang sedang dilewatkan pada jaringan. Untuk mencegah terjadinya efek negatif yang dihasilkan oleh kegiatan Network Sniffing (bocornya data administrator/root beserta password), bisa melakukan langkah-langkah sebagai berikut:

a. Memasang Network IPS/IDS

Intrusion Prevention System (IPS) adalah sebuah aplikasi yang bekerja untuk monitoring lalu-lintas jaringan, mendeteksi aktivitas yang mencurigakan, dan melakukan pencegahan dini terhadap intrusi atau kejadian yang dapat membuat jaringan menjadi berjalan tidak seperti sebagaimana mestinya. Bisa jadi karena adanya serangan dari luar, dan sebagainya. Produk IPS sendiri dapat berupa perangkat keras (hardware) atau perangkat lunak (software).

Network-based Intrusion Prevention System (NIPS) tidak melakukan pantauan secara khusus di satu host saja. Tetapi melakukan pantauan dan proteksi dalam satu jaringan secara global. NIPS menggabungkan fitur IPS dengan firewall dan kadang disebut sebagai In-Line IDS atau Gateway Intrusion Detection System (GIDS).Sistem kerja IPS yang populer yaitu pendeteksian berbasis signature, pendeteksian berbasis anomali, dan monitoring berkas-berkas pada sistem operasi host.

NIDS adalah jenis IDS yang menganalisa lalulintas paket dalam jaringan. Oleh NIDS, paket-paket data yang dikirimkan melalui jaringan akan diperiksa apakah berbahaya untuk keseluruhan jaringan. Apabila ada paket data yang berbahaya atau mencurigakan, NIDS akan membuat log mengenai paket tersebut, yang disertai informasi-informasi tambahan.

Berdasarkan paket yang mencurigakan tadi, NIDS akan memeriksa database miliknya mengenai ciri-ciri paket yang merupakan serangan terhadap jaringan. Kemudian NIDS akan memberikan label mengenai tingkat bahaya dari setiap paket yang dicurigai. Apabila tingkat bahaya cukup tinggi, NIDS bisa mengirimkan email peringatan kepada administrator agar dilakukan analisa lebih lanjut.

NIDS mendapatkan input dari sensor-sensor yang berada di lokasi-lokasi yang strategis dalam sebuah jaringan. Beberapa lokasi strategis yang bisa dipakai untuk menempatkan sensor antara lain switch, router, firewall, atau berada di sebuah host.

b. Memasang Web Application Firewall

Web application Firewall (WAF) adalah suatu alat pada layer aplikasi, yang bisa berupa plugin pada server atau filter yang berupa seperangkat aturan untuk penanganan HTTP. Umumnya, aturan ini digunakan untuk menangkal serangan seperti Cross-site Scripting (XSS) dan SQL Injection. Dengan membuat aturan untuk sebuah aplikasi, banyak serangan terhadap aplikasi dapat diidentifikasi dan diblokir.

Contoh beberapa WAF :

Fortinet, http://www.fortinet.com/solutions/firewall.html

Barracuda, https://www.barracuda.com/products

Stinger, https://www.owasp.org/index.php/OWASP_Stinger_Version_2

3. Memasang tool/sistem anti web defacement

Salah satu jenis sistem anti web defacement bekerja dengan cara membandingkan hash code dari halaman web dalam interval waktu tertentu. Dari suatu halaman web yang asli akan dihasilkan sebuah hash code, setiap membuka halaman web, akan menghasilkan hash code yang baru, hash code inilah yang dibandingkan dengan hash code yang asli untuk melihat apakah ada perubahan yang terjadi dari halaman web.

Beberapa toolyang bisa digunakan antara lain:

- Dotdefender, http://www.applicure.com/Products/dotdefender
- Nagios, http://www.nagios.com/solutions/website-defacement-detection
- Webguard, http://www.urtechnologies.net/products-services/webguard.html

4. Memperketat akses terhadap web

Peraturan akses terhadap web server, aplikasi-aplikasi pada web, dan database yang mendukung web harus diterapkan dengan ketat. Proses akses terhadap web bias dilakukan dengan beberapa tahap, diantaranya otentikasi dan otorisasi.

Otentikasi

Beberapa hal yang bisa digunakan untuk keamanan proses otentikasi

- Otentikasi melalui saluran yang aman
- Melindungi password pengguna dengan menerapkan standar
  - Ketentuan jumlah karakter
  - Ketentuan jenis karakter yang digunakan

    - Huruf kecil dan besar
    - Angka
    - Karakter khusus
  - Ketentuan waktu, perubahan password diperlukan setiap kurun waktu tertentu dan password sebelumnya tidak dapat digunakan kembali.
- Pembatasan jumlah percobaan login yang gagal, dan mengunci penggunanya
- Menggunakan fungsi hashing kriptografi
- Menggunakan proses re-authentication untuk opersi yang sensitif (missal perubahan password)

Manajemen sesi (session management)

Beberapa hal yang bisa digunakan untuk mengamankan suatu sesi

- Setelah proses otentikasi berhasil, sebuah token baru harus dihasilkan dan diberikan kepada pengguna. Token ini harus dikirim melalui saluran(channel) yang aman, dan disimpan dalam cookies.
- Aplikasi harus mendediakan fasilitas logout
- Token sesi harus berakhir setelah jangka waktu aktif yang wajar
- Sesi harus diterminasi, jika terjadi perilaku aneh atau anomali dari pengguna, dan melakukan pelaporan kepada administrator.
- Ketika terjadi pembatalan sesi baik melalui prosese logout yang wajar maupun terminasi, token harus dibuang dari server.

Access Control

Access Control berhubungan dengan pengendalian akses terhadap resource dari suatu sistem, dalam hal ini adalah server untuk menempatkan situs web. Cara paling umum dalam hal pengendalian akses adalah pemberian otorisasi kepada user untuk mengakses bagian/file tertentu. Beberapa langkah yang bisa digunakan untuk keaamanan acess control

- Memahami isi dan fungsi dari seluruh sumbr daya yang terdapat pada system
- Memetakan dengan benar akses pada sumber daya sesuai dengan profil dan role dari pengguna
- Secara default mematikan semua akses, kecuali yang telah

diterapkan secara eksplisit

- Memberikan proteksi (hanya read access) terhadap sumber yang bersifat statis, seperti dokumen, spreedsheets, dan gambar.

5. *Melakukan Security Audit/Penetration Testing secara berkala*

Penetration Testing adalah sebuah proses untuk melakukan pengujian terhadap celah keamanan yang terdapat pada suatu sistem komputer atau jaringan. Penetration testing yang khusus ditujukan untuk melakukan pengujian terhadap celah keamanan pada halaman web bisaanya disebut dengan Penetration Testing Web application. Pada proses penetration testing, sebuah web server akan dijadikan target serangan dengan melakukan beberapa simulasi serangan.

Pengujian terhadap sustu sistem dapat dilakukan dengan 2 pendekatan, yaitu

- Black Box Testingadalah penetration testing dilakukan tanpa mengetahui informasi-informasi yang berkaitan dengan sistem/jaringan seperti, sistem operasi yang digunakan sebagai target, topologi jaringan, port yang terbuka, dan service apa saja yang sedang berjalan.
- White Box Testing adalah penetration testing dilakukan dengan terlebih dahulu mengetahui informasi-informasi mengenai sistem/jaringan. Tetapi hal tersebut tidak serta-merta memberikan kemudahan dalam melakukan penetrasi, hal tersebut tergantung dari pelaku penetarsi testing yang melakukan pengujian dalam menilai sejauh mana kelemahan-kelemahan yang terdapat di dalam sistem/jaringan.

Terdapat duametode untuk “Web Application Penetration Testing“, yaitu:

- Passive Penetration Testing: Pada mode ini pentes tester mencoba untuk mengetahui logika dari aplikasi web. Tool yang ada digunakan untuk mengetahui beberapa informasi seperti kontrol yang ada didalam web application, login dan konfigurasinya, sehingga kita bisa memetakan target sistem.
- Active Penetration Testing: Yaitu melakukan kegiatan aktif

dalam pengujian terhadap keamanan sistem dengan melakukan manipulasi input, pengambilan akses, dan melakukan pengujian terhadap vulnerability-vulnerability yang sudah ada.

Khusus untuk menghindari web defacement, penetration testing dilakukan pada pengujian sebagai berikut:

- Configuration Management Testing
- Authentication Testing
- Session Management Testing
- Authorization Testing
- Data Validation Testing
- Web Service Testing

Banyak tools penetration testing yang bisa digunakan, yaitu:

Arachni, http://www.arachni-scanner.com

OWASP Zed Attack Proxy Project, https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

websploit, http://sourceforge.net/projects/websploit/

Acunetic, http://www.acunetix.com/