KAJIAN VULNERABILITY KEAMANAN DATA DARI EKSPLOITASI HASH LENGTH EXTENSION ATTACK

VULNERABILITY DATA SATISFACTION STUDY FROM EXPLOITATION HASH LENGTH EXTENSION ATTACK

Didi Juardi
Program Studi Teknik Informatika, FASILKOM – UNSIKA
Jl. H.S. Ronggowaluyo Teluk Jambe Karawang 41361
(0267) 641177, 641367, 642582
E-mail : info@unsika.ac.id, didi.juardi@staff.unsika.ac.id

ABSTRAK
Algoritma SHA-512 termasuk jenis fungsi hash yang merupakan pengembangan dari algoritma SHA-1.Fungsi hash memetakan pesan M dengan panjang berapapun menjadi nilai hash h dengan panjang tetap (tertentu, tergantung algoritmanya). Terdapat 3 jenis serangan terhadap fungsi hash, yaitu serangan yang memanfaatkan kelemahan struktural, kelemahan analitik dan eksploitasi keduanya. Metode untuk melakukan evaluasi terhadap keamanan dari sebuah sistem dan jaringan komputer dilakukan dengan cara melakukan sebuah simulasi serangan (attack) menggunakan Penetration Test (Pentest). Percobaan Hash Length Extension Attack dengan melakukan information gathering dengan tujuan untuk mencurangi sistem pada website target. Melalui hasil sniffing diketahui bahwa pencatatan nilai dilakukan terpusat di server website yang dijadikan target serangan dengan menggunakan HTTP GET Request.
Kata Kunci : Algoritma, Attack, Fungsi Hash, Server

ABSTRACT
The SHA-512 algorithm includes the type of hash function that is the development of the SHA-1 algorithm. The hash function maps M messages of any length to h hash values with fixed lengths (certain, depending on the algorithm). There are 3 types of attacks on the hash function, ie attacks that exploit structural weakness, analytic weakness and exploitation of both. The method to evaluate the security of a computer system and network is done by performing an attack simulation using Penetration Test (Pentest). Attempt Hash Length Extension Attack by doing information gathering in order to cheat the system on the target website. Through the sniffing results note that the recording of the value is done centrally in the website server which targeted the attack by using HTTP GET Request.
Keywords: *Algorithm, Attack, Hash Function, Server*

1. PENDAHULUAN

Masalah keamanan merupakan salah satu aspek penting dari sebuah sistem informasi. Sayang sekali masalah keamanan ini sering kali kurang mendapat perhatian dari para pemilik & pengelola sistem informasi. Seringkali masalah keamanan berada di urutan kedua, bahkan di urutan terakhir dalam daftar hal-hal yang dianggap penting.

Apabila menggangu performansi dari sistem, seringkali keamanan dikurangi/ditiadakan Informasi saat ini sudah menjadi sebuah komoditi yang sangat penting. Bahkan sudah menjadi sebuah "information-based society". Kemampuan untuk mengakses dan menyediakan informasi secara cepat dan akurat menjadi sangat esensial bagi sebuah organisasi, baik yang berupa organisasi komersial (perusahaan), perguruan tinggi, lembaga

pemerintahan, maupun individual (pribadi) dimungkinkan dengan berkembangannya bidang teknologi komputer & telekomunikasi.

Serangan Terhadap Keamanan Sistem Informasi (Security attack), dapat dilihat dari sudut peranan komputer/jaringan komputer yang fungsinya adalah sebagai penyedia informasi. Ada beberapa kemungkinan serangan (attack) salah satunya adalah Interception yaitu Pihak yang tidak berhak, berhasil mengakses aset/informasi. Serangan ini biasanya memanfaatkan celah keamanan yang lemah atau sering disebut sebagai vulnerabilities. Adapun salah satu contoh bentuk serangan ini adalah Injection Flaws yaitu Celah Injeksi, yang umumnya injeksi terhadap SQL (database) dari suatu aplikasi web. Hal ini mungkin terjadi apabila pengguna memasukkan data sebagai bagian dari perintah (query) yang menipu interpreter untuk menjalankan perintah tersebut atau merubah suatu data.

Pada data digital biasanya ada beberapa mekanisme pengujian integritas salah satu diantaranya adalah hash function. Hash function merupakan fungsi yang bersifat satu arah dimana jika kita masukkan data, ada beberapa hash function yang umum digunakan, antara lain MD5 dan SHA.

Dalam kajian penelitian ini akan membahas tentang hash length extension attack, bagaimana cara eksploitasinya dan bagaimana cara agar program yang kita buat tidak bisa dieksploitasi dengan teknik serangan ini.

Fungsi Hash

Hash adalah suatu teknik "klasik" dalam Ilmu Komputer yang banyak digunakan dalam praktek secara mendalam. Hash merupakan suatu metode yang secara langsung mengakses record-record dalam suatu tabel dengan melakukan transformasi aritmatik pada key yang menjadi alamat dalam tabel tersebut. Key merupakan suatu input dari pemakai di mana pada umumnya berupa nilai atau string karakter.

MD5

MD5 adalah salah satu dari serangkaian algortima(merupakan salah satu fungsi Hash) message digest yang didesain oleh Profesor Ronald Rivest dari MIT (Rivest, 1994). Saat kerja analitik menunjukkan bahwa pendahulu MD5, yaitu MD4 mulai tidak aman, MD5 kemudian didesain pada tahun 1991 sebagai pengganti dari MD4 (kelemahan MD4 ditemukan oleh Hans Dobbertin).

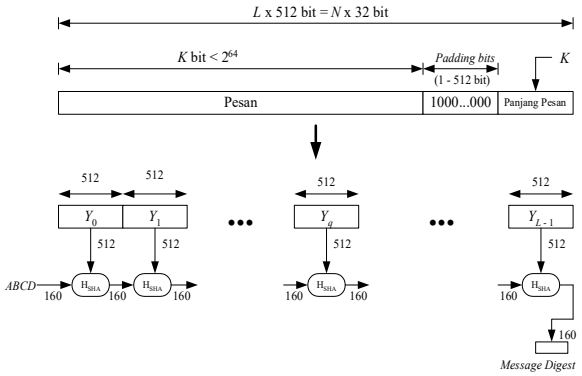
MD5 memproses variasi panjang pesan kedalam keluaran 128-bit dengan panjang yang tetap. Pesan masukan dipecah menjadi dua gumpalan blok 512-bit; Pesan ditata sehingga panjang pesan dapat dibagi 512. Penataan bekerja sebagai berikut: bit tunggal pertama, 1, diletakkan pada akhir pedan. Proses ini diikuti dengan serangkaian nol (0) yang diperlukan agar panjang pesan lebih dari 64-bit dan kurang dari kelipatan 512. Bit-bit sisa diisi dengan 64-bit integer untuk menunjukkan panjang pesan yang asli. Sebuah pesan selalu ditata setidaknya dengan 1-bit tunggal, seperti jika panjang pesan adalah kelipatan 512 dikurangi 64-bit untuk informasi panjang (panjang mod(512) = 448), sebuah blok baru dari 512-bit ditambahkan dengan 1-bit diikuti dengan 447 bit-bit nol (0) diikuti dengan panjang 64-bit.

***Secure Hash Algorithm* (SHA)**

SHA adalah fungsi hash satu-arah yang dibuat oleh NIST dan digunakan bersama DSS (Digital Signature Standard). Oleh NSA, SHA dinyatakan sebagai standard fungsi hash satu-arah. SHA didasarkan pada MD4 yang dibuat oleh Ronald L. Rivest dari MIT. SHA disebut aman (secure) karena ia dirancang sedemikian sehingga secara komputasi tidak mungkin menemukan pesan yang berkoresponden dengan message digest yang diberikan.

Algoritma SHA menerima masukan berupa pesan dengan ukuran maksimum 264 bit (2.147.483.648 gigabyte) dan menghasilkan message digest yang panjangnya 160 bit, lebih panjang dari message digest yang dihasilkan oleh MD5. Gambaran pembuatan message

digest dengan algoritma SHA diperlihatkan pada Gambar berikut:

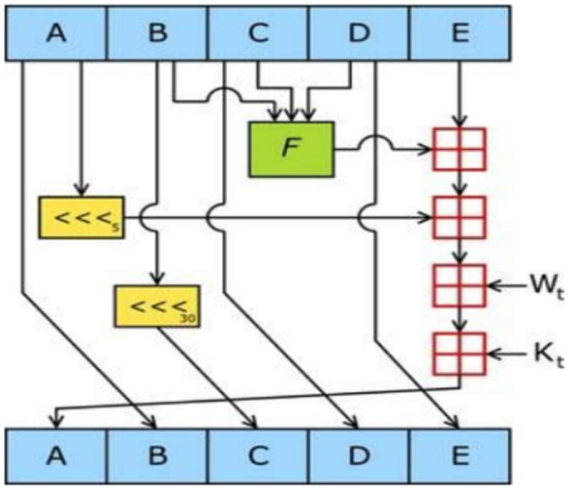


Gambar 1. Pembuatan message digest dengan algoritma SHA

Algoritma Hash-512

Algoritma SHA-512 adalah algoritma yang menggunakan fungsi hash satu arah yang diciptakan oleh Ron Rivest. Algoritma merupakan pengembangan dari algoritma-algoritma sebelumnya yaitu algoritma SHA-0, SHA-1, SHA-256 dan algoritma SHA-384.

Cara kerja kriptografi algoritma SHA-512 adalah menerima input berupa pesan dengan ukuran sembarang dan menghasilkan message diggest yang memiliki panjang 512 bit. Berikut illustrasi gambar dari pembuatan message diggest pada kriptografi algoritma SHA-512



Gambar 2. Message Digest Dengan Algoritma SHA-512

3. METODOLOGI PENELITIAN

Metode penelitian ini menggunakan metode Fungsi Hash dengan studi kasus yang bertujuan untuk memberikan gambaran yang lebih mendalam dan lengkap dari subyek yang akan diteliti, yaitu dengan menggunakan Metode Pentest yaitu Evaluasi dilakukan dengan cara melakukan sebuah simulasi serangan (attack). Hasil dari pentest ini sangat penting sebagai feedback bagi pengelola sistem untuk memperbaiki tingkat keamanan dari sistem komputernya. Dan Metode Algoritma Hash adalah yang menghasilkan sedikit hash collision. Sudah banyak algoritma fungsi hash yang diciptakan, namun fungsi hash yang umum digunakan saat ini adalah MD5 dan SHA (Secure Hash Algorithm).

Metode yang Digunakan Dalam Fungsi Hash

Metode Pembagian

Pada metode pembagian ini, kita memetakan suatu kunci k ke dalam salah satu slot dari m buah slot dengan mengambil sisa dari k dibagi oleh m. Maka, fungsi Hash ini adalah:

h(k) = k mod m

Dimana :

h(k) = fungsi Hash
k = kunci yang akan dihitung/di-hash,
m = jumlah keseluruhan slot.

Metode Perkalian

Metoda perkalian untuk memperoleh fungsi Hash dilakukan melalui dua langkah. Langkah pertama, kita mengalikan kunci dengan suatu konstanta didalam range 0 < A < 1 dan mengambil nilai fraksional dari kA. Kemudian, kita mengalikan nilai ini dengan m dan dapat diperoleh hasilnya. Singkatnya, fungsi Hash ini adalah ;

h(k) = ⌊m (k A mod 1)⌋

Dimana :

A » (Ö5 - 1)/2 = 0,6180339887..
(Golden Number),

"k A mod 1" artinya adalah nilai fraksional dari kA, atau kA – ⌊kA⌋ .

Algoritma MD5

Algoritma yang diberikan disini diambil dari RFC 1321, yang disusun oleh Ron Rivest. Dimisalkan kita memiliki pesan sepanjang "b"-bit, dan akan dicari message digestnya. Untuk menghitung message digest dari sebuah pesan, pada MD5 dilakukan lima langkah sebagai berikut :

- Penambahan Panjang Bit Pesan diperpanjang sampai sebesar 448 bit, dengan modulo 512. artinya jika panjang pesan telah melebihi 448 bit ini, maka perpanjangan pesan akan dilakukan sampai sebesar 512 + 448 bit, dan begitu seterusnya. Penambahan panjang pesan ini dilakukan dengan cara sebagai berikut, sebuah bit "1" ditambahkan ke dalam pesan. Kemudian bit "0" ditambahkan sampai panjang pesan menjadi 448 bit. Tujuan dari penambahan pesan ini adalah membuat panjang pesan menjadi (kelipatan) 512 bit, dikurangi 64 bit. Kekurangan 64 bit ini akan diatasi pada tahap kedua.
- Penambahan Panjang Pesan Total Representasi sebesar 64 bit dari "b" (panjang pesan awal) ditambahkan ke dalam pesan. Jika representasi "b" ini ternyata lebih besar dari 64 bit, maka yang akan diambil hanyalah 64 bit awal (low-order) saja. Panjang pesan total sampai pada tahap ke-dua ini sebesar (kelipatan dari) 512 bit. Tujuan dari penambahan ukuran pesan sampai sebesar kelipatan dari 512 bit ini adalah agar pesan memiliki panjang tepat kelipatan dari 16 word (satu word memiliki ukuran 32 bit). Pengolahan pesan pada tahap keempat nanti akan dilakukan untuk setiap blok sebesar 16 word.
- Inisialisasi Buffer MD Pada tahap ini digunakan 4 buah register sebagai buffer untuk perhitungan pesan (A, B, C, dan D). Setiap buffer ini memiliki ukuran 32 bit. Ke empat register ini diinisialisasi dengan nilai-nilai berikut (LSB di sebelah kiri):

Tabel 1. Tabel Inisialisasi Buffer MD

A	0	1	2	3	4	5	6	7
B	8	9	A	B	C	D	E	F
C	F	E	D	C	B	A	9	8
D	7	6	5	4	3	2	1	0

- Pemrosesan Pesan dalam blok 16-word Pada awal tahap ini didefinisikan 4 buah fungsi dengan input 3 buah word 32-bit, dan menghasilkan output satu buah word 32-bit. Fungsi-fungsi ini dideskripsikan sebagai berikut :

F(X,Y,Z) = XY or (~X)Z
G(X,Y,Z) = XZ or Y(~Z)
H(X,Y,Z) = X xor Y xor Z
I(X,Y,Z) = Y xor (X or (~Z))

Pada proses enkripsi pesan dengan MD5 ini juga diperlukan sebuah array T[i] yang berisi 64 elemen. Seluruh elemen ini didapat dengan perhitungan tertenu menggunakan fungsi sinus. Seluruh nilai dari tiap elemen array ini diberikan dalam RFC 1321. Pengolahan pesan dilakukan untuk setiap potongan blok dari pesan dengan ukuran tepat 16 word. Dengan demikian pada langkah awal harus dilakukan looping untuk setiap potongan pesan ini. Langkah-langkah yang harus dilakukan dijelaskan dalam tahapan berikut :

- Loop untuk setiap 16 word
- Definisikan suatu array x[i] yang berisi setiap word dari pesan (total ukuran array adalah 16 elemen).
- Definisikan variabel AA, BB, CC, dan DD, yang berisi nilai A, B, C, dan D pada tahap tiga.
- Tahap ini dibagi menjadi 4 bagian, biasa disebut tahap FF, GG, HH, dan II.

Output yang dihasilkan dari tahap empat ialah A, B, C, dan D. Hasil dari fungsi MD5 didapat dengan rumusan berikut :

Output =
LSB (A) + LSB (B) + LSB (C) + LSB (D)

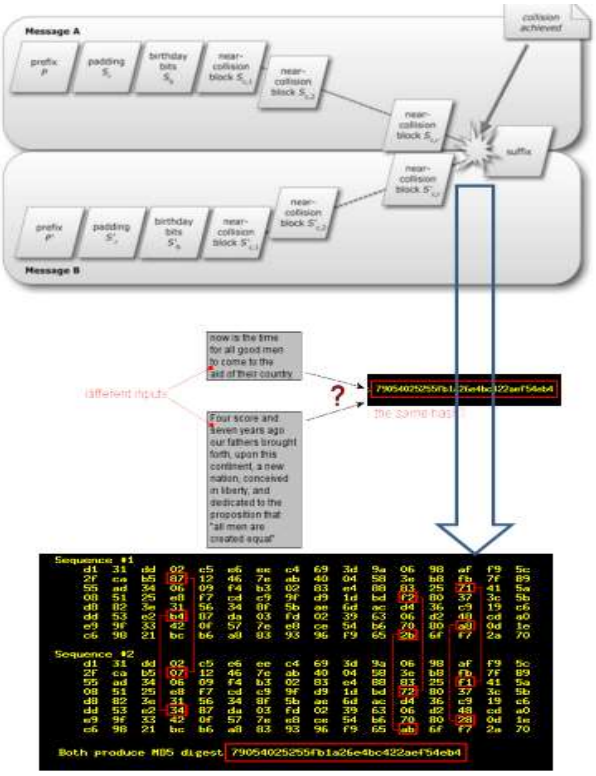
Pengamanan Data dengan Algoritma SHA-512

Algoritma SHA-512 termasuk jenis fungsi hash yang merupakan pengembangan dari algoritma SHA-1. Fungsi hash memetakan pesan M dengan panjang berapapun menjadi nilai hash h dengan panjang tetap (tertentu, tergantung algoritmanya). Untuk algoritma SHA-512 panjang nilai hash yang dihasilkan adalah 512 bit. Fungsi hash yang menghasilkan keluaran dengan ukuran yang kecil mudah diserang oleh birthday attack. Serangan ini dilakukan dengan cara mendapatkan dua pesan secara acak yang memiliki nilai hash h sama. SHA-512 sebagai fungsi hash mempunyai sifat-sifat sebagai berikut:

- **h** mudah dihitung bila diberikan M. Sifat ini merupakan keharusan, karena jika h sukar dihitung, maka fungsi hash tersebut tidak dapat digunakan.
- M tidak dapat dihitung jika hanya diketahui h. Sifat ini disebut juga one-way function, atau mudah untuk menghitung h dan sukar untuk dikembalikan ke M semula. Sifat ini sangat penting dalam teknik kriptografi, karena jika tanpa sifat tersebut maka penyerang dapat menemukan nilai M dengan mengetahui nilai hash-nya h
- Tidak mungkin dicari M dan M' sedemikian sehingga H(M)=H(M'). Sifat ini disebut juga collision free. Sifat ini mencegah kemungkinan pemalsuan

Collision Vulnerability

Salah satu masalah yang mungkin terjadi dari fungsi hash adalah collision. Maksudnya adalah ada 2 atau lebih teks yang menghasilkan nilai hash yang sama. Anda sendiri telah melihat dengan MD5 bahwa masukan sepanjang berapapun, akan menghasilkan nilai hash sepanjang 128 bit. Itu artinya kemungkinan inputnya sangat banyak jumlahnya, tak terhingga, namun kemungkinan nilai hashnya hanya sejumlah 2^128. Sebagai ilustrasi, bayangkan apa yang terjadi bila dalam suatu negara jumlah wanitanya sangat banyak, hingga 5 kali lipat jumlah pria. Maka kemungkinan akan ada 2 atau lebih wanita yang memiliki suami yang sama. Inilah yang disebut collision. Ada 2 atau lebih input teks yang memiliki nilai hash yang sama. Berikut gambar MD5 Collision, dimana terdapat dua atau lebih input teks yang memiliki nilai hash yang sama(perbedaan pesan tapi nilai hash-nya sama)



Gambar 3. MD5 Collison

Serangan Terhadap Fungsi Hash

Jenis-Jenis Serangan

Kriptografi fungsi hash untuk autentikasi dan integritas data, satu persatu telah berhasil dipecahkan. Terdapat 3 jenis serangan terhadap

fungsi hash, yaitu serangan yang memanfaatkan:

- kelemahan struktural
- kelemahan analitik
- eksploitasi keduanya

Kriptografi fungsi hash $H: \{0,1\} \rightarrow \{0,1\}^n$ memetakan set input infinit ke sekumpulan terbatas n-bit *hash value*. Jelasnya, sebuah fungsi hash *H* sebaiknya berkelakuan ideal (seperti sebuah *oracle random*). Hal ini tidak akan berguna untuk sebuah definisi formal. Sebagai gantinya adalah tujuan keamanan yang lebih sederhana untuk $H: \{0,1\} \rightarrow \{0,1\}^n$.

Sementara *collision* (input $X \neq Y$ dengan $H(X) = H(Y)$) perlu ada, sebagaimana terdapat lebih banyak input daripada output, sebuah fungsi hash sebaiknya memiliki ketahanan terhadap *collision*; diberikan *H*, sebaiknya infeasible terhadap musuh untuk menemukan setiap *collision*.

Serangan Penggunaan Fungsi Hash

Berikut serangan atau attack terhadap database dengan penggunaan fungsi Hash

sha → biasanya ada pada fungsi mysql
jalankan phpmyadmin dan lakukan perintah SQL : select sha ("pesan anda") tekan tombol GO. Dan lihat hasilnya!
Contoh : select sha ("thank") hasilnya adalah :

c4121f6a6c697280868b940534f5b0cbd0277ca1

panjang 40digit
password → juga ada dalam fungsi mysql
misal : select password ("thank") hasilnya ;
04d7ce640ee7b52f → panjang 16 digit

md5, crypt dan crc32 → pada php

Hasilnya adalah :
hasil dari
md5=e274648aed611371cf5c30a30bbe1d65
jumlah digit=32
hasil dari crypt=thFCUcmih9JWE
jumlah digit =13
hasil dari crc32=332224512

3. HASIL DAN PEMBAHASAN

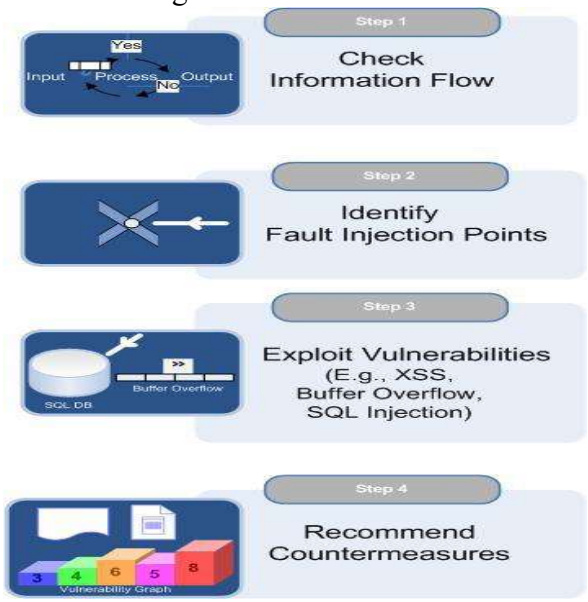
Security Attack Models

Jenis serangan terhadap keamanan sebagai berikut:

- ***Interruption***
- ***Interception***
- ***Modification***
- ***Fabrication***

Penetration Test System

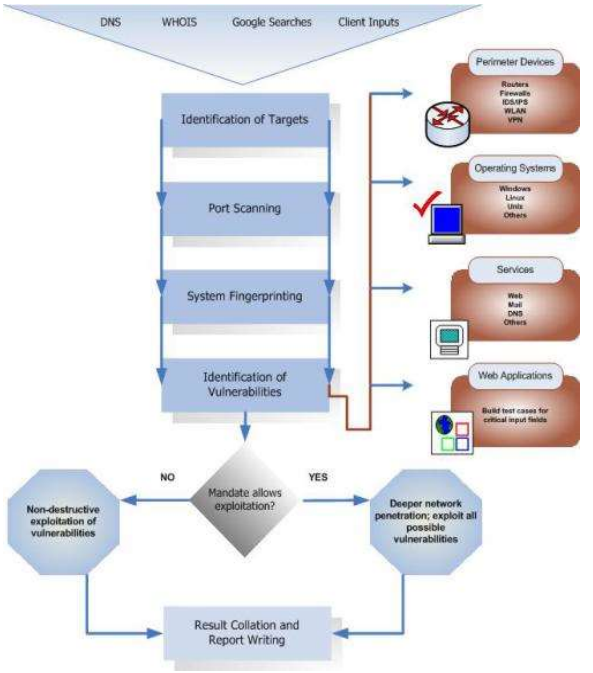
Penetration Test (Pentest) adalah sebuah metode untuk melakukan evaluasi terhadap keamanan dari sebuah sistem dan jaringan komputer. Evaluasi dilakukan dengan cara melakukan sebuah simulasi serangan (attack). Hasil dari pentest ini sangat penting sebagai feedback bagi pengelola sistem untuk memperbaiki tingkat keamanan dari sistem komputernya. Laporan hasil Pentest akan memberikan masukan terhadap kondisi vulnerabilitas sistem sehingga memudahkan dalam melakukan evaluasi dari sistem keamanan komputer yang sedang berjalan. Aktivitas ini kadang disebut juga dengan istilah ethical hacking.



(Sumber: http://www.niiconsulting.com/services/security_assessment/pentest.html)

Gambar 4. Aktivitas Penetration Test

Secara umum, terdapat 4(empat) langkah dasar untuk melakukan aktivitas Pentest sebagaimana pada Gambar diatas. Langkah pertama adalah mengumpulkan sejumlah informasi penting dari sistem, langkah kedua melakukan analisis untuk menentukan jenis serangan yang akan dilakukan, langkah ketiga adalah melakukan aktivitas serangan untuk mengeksploitasi vulnerabilitas sistem dan langkah keempat adalah melakukan laporan serta rekomendasi perbaikan sistem.
Gambaran lebih detail dari aktivitas Pentest adalah sebagaimana ilustrasi pada Gambar dibawah ini.

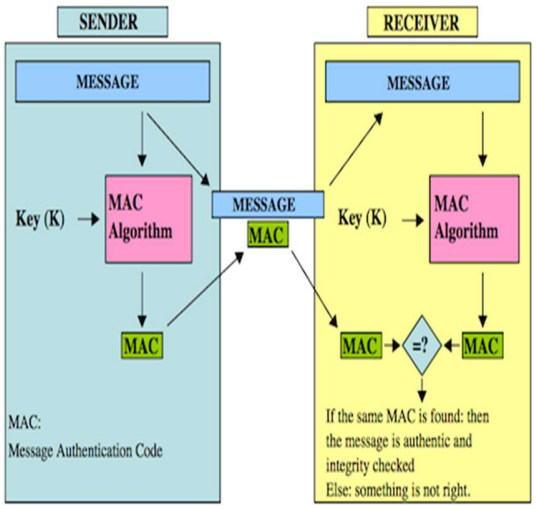


(Sumber:http://www.niiconsulting.com/services/security_assessment/pentest.html)

Gambar 5. Detail Aktivitas Penetration Test System

Fungsi Hash untuk MAC

Dengan fungsi hash seperti ini, pihak ketiga yang tidak mengetahui secret key tidak bisa membuat hash yang valid dari suatu data. Sebagai contoh, bila seseorang ingin mengirimkan data X dia harus menyertakan pula MD5 (secretkey+data X) sebagai MAC, bila dia mengetahui secretkey maka dia bisa menghitung nilai MAC dengan mudah. Namun bila secretkey tidak diketahui bagaimana cara menghitung MD5 (secretkey+data X), mungkinkah menghitung MD5 (secretkey+data X) tanpa mengetahui secretkey ?



Gambar 6. Fungsi Hash untuk MAC

Autotentikasi dengan Message Authentication Code (MAC)

MAC adalah suatu data yang digunakan sebagai otentikasi data dan menjamin keasliannya. Dalam gambar di bawah ini (sumber: wikipedia) menunjukkan salah satu use-case dari MAC, diilustrasikan bahwa X akan mengirim pesan ke Y:

- X dan Y sebelumnya harus sudah sepakat dengan suatu kunci rahasia
- X menghitung MAC dari pesan dengan kunci rahasia
- X mengirim MAC dan pesan ke Y
- Y menghitung MAC dari pesan yang diterima dengan kunci rahasia

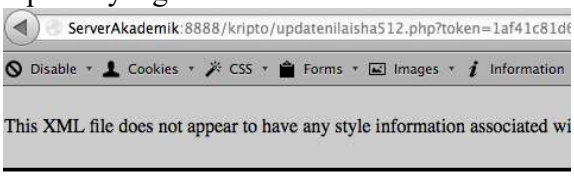
Eksploitasi Serangan Hash Length Extension

Percobaan Hash Length Extension Attack dengan melakukan information gathering dengan tujuan untuk mencurangi sistem akademik pada website target tersebut. Melalui hasil sniffing diketahui bahwa pencatatan nilai dilakukan terpusat di server website yang

dijadikan target serangan dengan menggunakan HTTP GET request sebagai berkut:

http://ServerAkademik:8888/kripto/updatenilaisha512.php?token=1af41c81d665f0e8542cafbe333255d47b65c0e650d1c3fd919947d237b81e86f1aa4cd31fbe4254abc9b959e10f23b92bb0f932ac5c0414014b507f048acdc9&nilai=MTMwMDAwMDAyM3xDUzMyMT1DO0NTNDQyPUI7

Dengan mengetikkan URL pada browser, dan response yang muncul adalah:




```
- <xml>
    <status>OK</status>
    <nim>1300000023</nim>
  - <matakuliah>
      <kode>CS321</kode>
      <nilai>C</nilai>
    </matakuliah>
  - <matakuliah>
      <kode>CS442</kode>
      <nilai>B</nilai>
    </matakuliah>
  </xml>
```

Gambar 7. Respon URL Web Browser

Proses Padding

Secara sederhana hash length extension attack bisa diuraikan sebagai berikut:

Bila diketahui data dan nilai hash dari (secret+data), maka untuk menghitung hash dari (secret+data+data tambahan) walaupun tidak mengetahuinya secretnya.

Jika diketahui SHA-512(secret +'abcd') adalah:

b51ca01e1054cd0cfa09316e53a1272ed43cf6286a18380b7758546026edf2c6af9f11251768b7510728e5c35324f0715b0d7717228865cf621a96ed3cef05a1

Untuk menghitung SHA-512(secret+'abcd'+'efghijklmnopqrstuvwxyz') walaupun tidak mengetahui secret. Untuk memahami bagaimana hash length extension ini terjadi kita harus melihat bagaimana hash SHA-512 dihitung.

Padding pada SHA-512

Bila data yang akan di-hash tidak tepat berukuran kelipatan 1024 bit, maka dibutuhkan pre-processing berupa menambahkan bit-bit padding sebagai pengganjal agar ukurannya menjadi tepat kelipatan 1024 bit.

Padding dilakukan dalam dua langkah:

- Menambahkan bit 1 di akhir data dan diikuti dengan bit 0 sejumlah yang diperlukan agar jumlahnya menjadi 128 bit kurang dari kelipatan 1024 bit.
- Sisa 128 bit yang akan melengkapi blok menjadi 1024 bit adalah panjang dari data (sebelum ditambahkan padding)

Komputasi SHA-512

SHA-512 menghitung nilai hash dengan cara memproses blok-blok berukuran 1024 bit. Gambar di bawah ini menunjukkan proses penghitungan SHA-512 data berupa deretan huruf A sebanyak 300 karakter. Data tersebut dipotong-potong dan ditambahkan padding sehingga menjadi 3 blok masing-masing berukuran 1024 bit.

Penghitungan hash suatu blok membutuhkan dua masukan, blok data 1024 bit dan hash dari blok sebelumnya. Kemudian hash dari suatu blok akan menjadi input untuk menghitung hash blok selanjutnya, dan proses ini terus berlanjut sampai semua blok telah dihitung hashnya. Hash blok terakhir adalah nilai hash final dari data. Khusus untuk memroses blok pertama, hash yang dipakai sebagai input adalah intial hash value yang didefinisikan dalam FIPS 180-3 sebagai:

H0 = 0x6a09e667f3bcc908
H1 = 0xbb67ae8584caa73b
H2 = 0x3c6ef372fe94f82b
H3 = 0xa54ff53a5f1d36f1
H4 = 0x510e527fade682d1

H5 = 0x9b05688c2b3e6c1f
H6 = 0x1f83d9abfb41bd6b
H7 = 0x5be0cd19137e2179

Gabungan dari 8 variabel di atas membentuk initial hash value :

6a09e667f3bcc908bb67ae8584caa73b3c6ef37
2fe94f82ba54ff53a5f1d36f1510e527fade682d
19b05688c2b3e6c1f1f83d9abfb41bd6b5be0cd
19137e2179
yang diperlukan untuk menghitung hash suatu blok adalah hash (bukan isi) blok sebelumnya

Serangan Server dengan SHA-512

Untuk melakukan serangan terhadap website akademik yang akan dijadikan target dengan mengetahui Hash Length Extension dari server target dengan panjang kunci rahasia tersebut adalah 14 karakter yang akan di-inject ke website target.

Karena setiap nilai dipisahkan dengan titik-koma, maka perubahan atau menambahkan data ‘;CS114=A;CS521=A;CS221=A;CS125=A;CS444=A;’ di akhir data aslinya supaya nilainya berubah menjadi A, dalam kasus ini sebanyak 5(lima) mata kuliah yang dilakukan perubahan yang dinginkan.

Berdasarkan serangan tersebut walaupun tidak diketahui byte key-nya (*unknown 14 byte key*), akan tetap bisa menghitung token yang valid karena mengetahui SHA-512

(‘unknown14bytekey’+’1300000023|CS321=C;CS442=B;’).

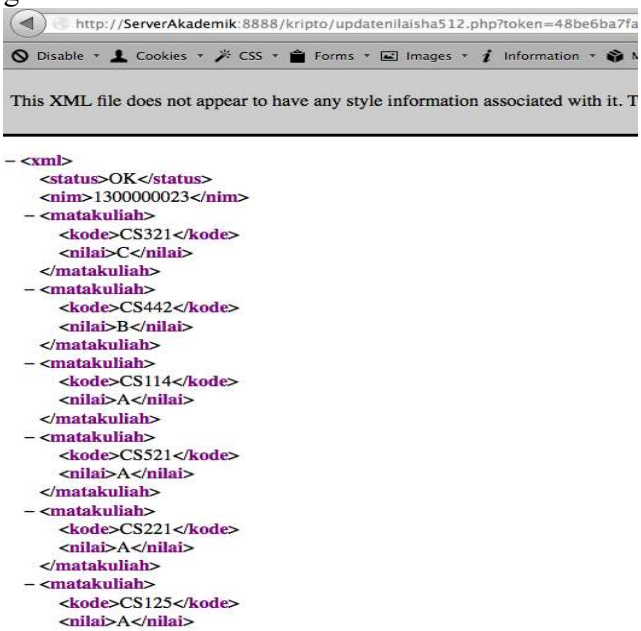
Kalau sudah tahu SHA-512(A+B), mencari SHA-512(A+B+C) itu mudah walaupun tidak tahu Nilai A dan B.

Dengan hash length extension attack tersebut, tinggal melanjutkan penghitungan hash-nya dengan blok data baru untuk mendapatkan nilai hash yang baru. Dia memakai tools sha512-extender untuk menghitung MAC yang valid.

Hasil Eksploitasi Hash Lentgh Extension Attack

Hasil dari ekspolitasi Hash Length Extension Attack, menandakan bahwa website target memiliki karentanan dari serangan-serangan para hacker salah satunya melalui fungsi hash.

Akhirnya, setelah melakukan Hash Length Extension Attack terhadap website target tersebut, terjadi perubahan sesuai yang diinginkan oleh penyerang, seperti terlihat pada gambar berikut:



Gambar 8. Perubahan akibat Hash Length Extension Attack

Pencegahan dari Eksploitasi Kerentanan Website

Agar website tidak mudah diexploitasi dengan solusinya sebagai berikut:

- Hindari membuat sendiri MAC dengan bentuk-bentuk HASH(kunci+data). Gunakan HMAC (Hash-based MAC) yang memang sudah dirancang untuk membuat MAC yang aman.
- Pengembang perangkat lunak, Otoritas Sertifikasi, pemilik website, dan pengguna harus menghindari menggunakan algoritma MD5 dalam kapasitas apa pun.

- Jangan menyimpan SHA1 (password) dalam database.
- SHA3 (Keccak) tidak vulnerable terhadap hash length extension, jadi kalau tetap menggunakan bentuk HASH (kunci+data) gunakan SHA3 (kunci+data).
- Hindari DNS (Domain Name System) spoofing. Kerentanan DNS spoofing ditemukan oleh Dan Kaminsky tahun 2008 memungkinkan penyerang untuk mengarahkan permintaan untuk setiap situs web ke web server yang dikendalikan oleh mereka.

4. SIMPULAN

- Serangan Terhadap Keamanan Sistem Informasi (Security attack), dapat dilihat dari sudut peranan komputer/jaringan komputer yang fungsinya adalah sebagai penyedia informasi. Ada beberapa kemungkinan serangan (attack) salah satunya adalah *Interception* yaitu Pihak yang tidak berhak, berhasil mengakses aset/informasi. Serangan ini biasanya memanfaatkan celah keamanan yang lemah atau sering disebut sebagai vulnerabilities. Untuk melakukan evaluasi terhadap keamanan dari sebuah sistem dan jaringan komputer dilakukan dengan cara melakukan sebuah simulasi serangan (attack) yang disebut Pentest(Penetration Test). Hasil dari pentest ini sangat penting sebagai feedback bagi pengelola sistem untuk memperbaiki tingkat keamanan dari sistem komputernya. Laporan hasil Pentest akan memberikan masukan terhadap kondisi vulnerabilitas sistem sehingga memudahkan dalam melakukan evaluasi dari sistem keamanan komputer yang sedang berjalan.
- Fungsi kriptografis hash seperti MD5, SHA1, SHA2 bisa dipakai untuk membuat MAC(Message Authentication Code) dengan cara menghitung hash dari gabungan secret key dan data yang akan dilindungi oleh MAC. Percobaan Hash Length Extension Attack dengan melakukan information gathering dengan tujuan untuk mencurangi sistem akademik pada website target. Melalui hasil sniffing diketahui bahwa pencatatan nilai dilakukan terpusat di server website yang dijadikan target serangan dengan menggunakan HTTP GET request. Hindari membuat sendiri MAC dengan bentuk-bentuk HASH (kunci+data). Gunakan HMAC (Hash-based MAC) yang memang sudah dirancang untuk membuat MAC yang aman.

DAFTAR PUSTAKA

Adriansyah, A, Arifandi,W, dan Wicaksono N, 2005 ,Keamanan Web Service, Teknik Informatika, Institut Teknologi Bandung, Bandung.

Aiden Bruen, David Wehlau, Mario Forcinito, 2001, Hash Functions Based on Sylvester Matrices, Patents O_ce Kilkenny, September 20th 2001.

Bart Van Rompay, 2004, Analysis and Design of Cryptographic Hash Functions, Mac Algorithms and Block Ciphers; Ph.D. Thesis, june 2004.

Bart PRENEEL, 2003, Analysis and Design of Cryptographic Hash Functions, Doctoral dissertation, February 2003.

Burrows, James, 2005, Securer Hash Standard, USA: US National Institute and Technology.

Coron, J.S. 2002. ***Security Proof for Partial-Domain Hash Signature Schemes***. Gemplus Card International

Dai, W., 2012, Speed Benchmarks for Various Ciphers and Hash Functions, http://www.weidai.com/, diakses terakhir tanggal 11 Juli 2012.

Deriz,Spyware,Ancaman Baru Bagi Komputer Dmitry K, Christian R and Alexandra Savelieva, 2011. "Bicliques for Preimages: Attacks on Skein-512 and the SHA-2 family". *IACR Cryptology ePrint Archive*. 2011:286.

Duo Lei, Feng Guozhu, Li C, Feng K, Longjiang Qu , The Design Principle of Hash Function with Merkle-Damgård onstruction.(http://eprint.iacr.org/2006/135.pdf)

Douglas R.Stinson, Cryptography Theory and Practice,CRC Press,1995.

Farrel – Vinay, Peter, 2008, Manage Software Testing, Auerbach Publications.

Federal Information Processing Standards Publication 180-2. 2002. ***Secure Hash Standard.*** National Institute of Standards and Technology.

Federal Information Processing Standards Publication 186-2. 2000. ***Digital Signature Standard (DSS).*** National Institute of Standards and Technology.

Fernanfo, Gilbert F, 2007, Penggunaan Fungsi Hash dalam Kriptografi, Makalah IF2153 Matematika Diskrit, Teknik Informatika Institut Teknologi Bandung.

Gustafson, David, 2002, Theory and Problems of Software Engineering, Mc Graw Hill.

Harry Lim, Kesalahan Utama dalam Keamanan, http://students.ukdw.ac.id/~22033120/5security.html

IEEE Computer Society, 1990, Standard Glossary of Software Engineering Terminology, IEEE Press.