PERANGKAT LUNAK PEMINDAI CELAH KEAMANAN JARINGAN PADA BERBAGAI SISTEM OPERASI

Febriliyan Samopa, Imam Kuswardayan, Firman Fathoni
Jurusan Teknik Informatika,
Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Kampus ITS, Jl. Raya ITS, Sukolilo – Surabaya 60111, Telp. + 62 31 5939214, Fax. + 62 31 5913804
Email : iyan@its-sby.edu, imam@its-sby.edu

ABSTRAK

Keamanan jaringan merupakan tantangan yang besar dalam pengembangan layanan Internet yang dewasa ini tumbuh demikian cepat. Salah satu bentuk masalah keamanan yang paling menonjol adalah intrusi dari cracker yang memanfaatkan celah-celah keamanan yang ada sebagai kompensasi dari layanan yang diberikan oleh sebuah server. Cracker akan berusaha mengakses ports yang sedang digunakan oleh layanan suatu server dan berusaha mengekspoitasi kelemahan port tersebut untuk melakukan Denial of Service (DoS), defacing situs internet, maupun Information Retrieval terhadap data-data sensitif.

Tugas akhir ini berusaha menyajikan sebuah perangkat lunak yang dapat mencari celah-celah keamanan yang mungkin timbul pada sebuah komputer yang terhubung ke jaringan. Celah keamanan yang dapat dideteksi oleh perangkat lunak ini antara lain ARP Poisoning, Open Ports, NetBIOS Null Session, DCOM RPC dan IIS Unicode. Sebagai tambahan, perangkat lunak ini juga dapat memberikan estimasi mengenai sistem operasi apa yang dimiliki oleh komputer tersebut.

Dari hasil uji coba, perangkat lunak ini dapat menjalankan fungsi-fungsinya dengan baik. Namun demikian, terdapat sedikit kekurangan-kekurangan antara lain antisipasi terhadap firewall yang belum cukup baik dan penggunaan CPU time yang cukup tinggi.

Kata Kunci : *Jaringan, Keamanan, ARP Poisoning, IIS Unicode, DCOM RPC, Null Session, Fingerprinting, Port Scanning*

1. PENDAHULUAN

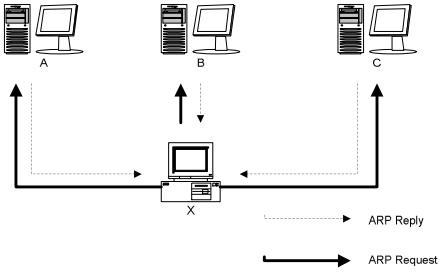
Dengan perkembangan Internet yang demikian cepatnya, semakin banyak orang yang menggunakan layanan berbasiskan internet dalam kehidupan mereka sehari-hari. Mereka bekerja, berbelanja, dan melakukan kegiatan perbankan secara *online*. Setiap komputer yang ada di muka bumi ini dapat saling terkoneksi dan melakukan pertukaran data-data penting secara *real-time* sehingga dapat menghemat waktu dengan cukup signifikan. Namun disamping kemudahan-kemudahan dan interkoneksitas tinggi yang ditawarkan internet, sebagian besar orang juga mencemaskan aspek keamanannya. Komputer yang terhubung ke internet berarti telah membuka dirinya untuk diakses oleh siapapun, bahkan oleh pihak-pihak yang tidak bertanggung jawab, yang biasa dikenal dengan istilah *cracker*. Keamanan jaringan merupakan tantangan yang besar dalam pengembangan layanan Internet.

Salah satu bentuk masalah keamanan yang paling menonjol adalah intrusi dari *cracker* yang memanfaatkan celah-celah keamanan yang ada sebagai kompensasi dari layanan yang diberikan oleh sebuah *server*. *Cracker* akan berusaha mengakses *ports* yang sedang digunakan oleh layanan suatu *server* dan berusaha mengekspoitasi kelemahan *port* tersebut untuk melakukan *Denial of Service* (DoS), *defacing* situs internet, maupun *Information Retrieval* terhadap data-data sensitif.

2. ARP POISONING

ARP Poisoning adalah sebuah cara untuk memberikan pemetaan IP dan MAC Address yang keliru pada ARP cache sebuah host. ARP Poisoning ini akan berakibat pada pengalihan lalu lintas IP dari sebuah host ke host lainnya.

ARP Poisoning dapat dideteksi dengan cara menganalisa lalu lintas paket-paket data untuk memeriksa pasangan field MAC Address dengan IP Address dan membandingkannya dengan database yang telah ada. Jika pasangan MAC dan IP Address tersebut tidak terdapat dalam database, maka dapat dipastikan bahwa sebuah usaha percobaan ARP Poisoning telah dilakukan pada IP Address tersebut.

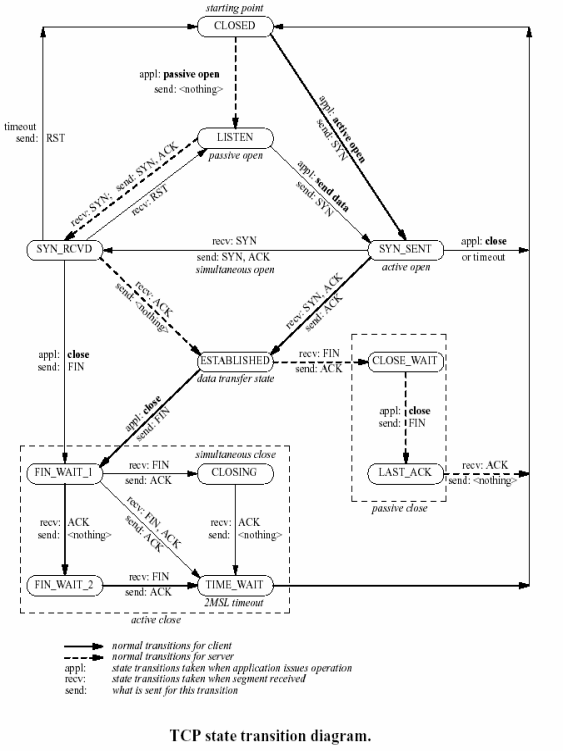


Gambar 1. ARP Detection

3. PORT SCANNING

Port Scanning adalah salah satu teknik yang paling populer untuk mengetahui dan memetakan layanan yang tersedia pada port tertentu dengan memanfaatkan karakteristik transisi state dalam koneksi TCP seperti yang dapat dilihat pada Gambar 2.

a. Open Scan
   Port terbuka:
   client → SYN
   server → SYN|ACK
   client → ACK
   Port tertutup:
   client → SYN
   server → RST|ACK
   client → RST



TCP state transition diagram.

Gambar 2. TCP state diagram

b. Syn Scan
   Port terbuka:
   client → SYN
   server → SYN|ACK
   client → RST
   Port tertutup:
   client → SYN
   server → RST|ACK

c. Syn|Ack Scan
   Port terbuka:
   client → SYN|ACK
   server → RST
   Port tertutup:
   client → SYN|ACK
   server → --

d. Fin Scan
   Port terbuka:
   client → FIN
   server → --
   Port tertutup:
   client → FIN
   server → RST

e. Ack Scan
   Port terbuka:
   client → ACK
   server → RST (TTL =< 64)
   Port tertutup:
   client → ACK
   server → RST|ACK WINDOW (non-zero)

f. Null Scan
   Port terbuka:
   client → NULL (no flags)
   server → --
   Port tertutup:
   client → NULL
   server → RST

g. XMAS Scan
   Port terbuka:
   client → XMAS (all flags)
   server → --
   Port tertutup:
   client → XMAS (all flags)
   server → RST

h. UDP Port Unreachable Scan
   Port terbuka:
   client → udp packet
   server → --
   Port tertutup:
   client → udp packet
   server → ICMP Port Unreachable

4. NETBIOS NULL SESSION ENUMERATION

NetBIOS Enumeration adalah teknik yang digunakan untuk menjelajahi file-sharing service NetBIOS yang ditawarkan oleh sistem target. Teknik ini mengimplementasikan pendekatan langkah demi langkah untuk mengumpulkan informasi dan berusaha untuk mendapatkan akses file tingkat sistem meskipun hanya sebagai client lokal biasa.

Sebuah status query UDP dikirimkan ke target, yang umumnya akan dibalas dengan data berupa nama komputer NetBIOS dari target tersebut. Data ini yang akan digunakan untuk melakukan sebuah sesi. Balasan tersebut juga mangandung sejumlah informasi lain seperti workgroup dan nama account yang tengah aktif pada sistem. Agar dapat menerima data balasan ini, hak akses root atau administrator harus diperoleh karena data balasan tersebut akan dikirim ke port UDP 137, meskipun query awal dikirimkan dari port yang berbeda.

Setelah itu koneksi TCP dilakukan ke port NetBIOS 139, dan meminta sebuah sesi dengan menggunakan nama komputer yang telah kita terima sebelumnya. Berbagai cara untuk menebak nama

komputer dapat dilakukan, jika proses query gagal untuk mendapatkan informasi yang diinginkan. Jika percobaan untuk melakukan sebuah sesi ini gagal, maka dapat dikatakan bahwa host mampu bertahan terhadap serangan NetBIOS meskipun port 139 berhasil dicapai.

Jika sesi NetBIOS dapat dilakukan pada port TCP 139, maka dengan ini target dinyatakan rawan (vulnerable). Informasi diperoleh melalui kelemahan-kelemahan sistem yang ada pada tahap ini, karena pada tiap titik selanjutnya query dapat saja diblokir oleh konfigurasi keamanan sistem target. Mayoritas server berbasis Microsoft dan Unix SAMBA akan menghasilkan nama komputer dan daftar share, namun tidak mengijinkan file-sharing yang sesungguhnya tanpa username/password yang sah.

5. DCOM RPC

Terdapat bug pada bagian RPC yang menangani pertukaran data melalui TCP/IP. Bug yang terjadi karena penanganan yang salah atas data yang invalid ini terdapat pada RPC antarmuka Distributed Component Object Model (DCOM), yang melakukan proses listening pada port RPC. Antarmuka ini menangani permintaan aktivasi obyek DCOM yang dikirim oleh komputer klien kepada server, sebagai contoh yaitu pada permintaan path Universal Naming Convention (UNC). Penyerang yang dapat mengeksploitasi kelemahan ini akan memiliki akses penuh untuk menjalankan kode-kode pada sistem yang terkena, termasuk melakukan proses instalasi program, manipulasi data, bahkan membuat sebuah account baru yang memiliki hak akses penuh.

Interface yang dimaksud diatas adalah seperti dibawah ini:

```
HRESULT CoGetInstanceFromFile(
  COSERVERINFO * pServerInfo,
  CLSID * pclsid,
  IUnknown * punkOuter,
  DWORD dwClsCtx,
  DWORD grfMode,
  OLECHAR * szName,
  ULONG cmq,
  MULTI_QI * rgmqResults
);
```

Parameter keenam, yaitu szName, menurut MSDN adalah nama file yang akan diinisialisasi dengan menggunakan IPersistFile::Load. Nilai parameter ini tidak boleh berisi NULL. Parameter ini yang dapat menyebabkan kondisi buffer overflow.

Sebagai contoh:

```
hr = CoGetInstanceFromFile(pServerInfo, NULL,0,
CLSCTX_REMOTE_SERVER,STGM_READWRITE,L"C:\\12345611111111111111111111111111.doc",1,&qi);
```

Jika nama file terlalu panjang, sistem akan menghasilkan buffer overflow lokal, karena fungsi GetPathForServer RPCSS hanya mengalokasikan nama file hanya sebesar 20h. Namun demikian, karena API ini memeriksa file lokal terlebih dahulu, dan kita pun tidak dapat membuat sebuah file yang panjangnya 20h, maka API ini tidak dapat digunakan untuk mengeksploitasi. Namun kita dapat menggunakan fungsi LPC dengan membuat paket sendiri, yang nantinya akan menghasilkan sebuah remote stack overflow.

Setelah komputer klien mengirimkan parameter ke server, server akan menerjemahkannya ke bentuk "\\servername\c$\12345611111111111111111111.doc". Kemudian server akan mengambil servername terlebih dahulu. Namun Windows tidak memeriksa panjang parameter, hanya mengalokasikan stack sebesar 20h, dimana 20h adalah panjang maksimum sebuah NetBIOS name. Hal inilah yang memicu terjadinya buffer overflow.

6. IIS UNICODE

Bug dalam IIS Unicode extensions adalah proses decoding karakter Unicode yang dilakukan setelah proses path-checking. Secara default, IIS memeriksa URL yang diminta dan mencari substring "..\.." didalamnya. Jika ditemukan, berarti seseorang berupaya untuk mengakses direktori diatas direktori virtual web-access. Biasanya, IIS tidak memperbolehkan akses semacam ini, namun dengan mengganti karakter '/' atau '\' dengan nilai Unicode-nya, proses path-checking IIS dapat terkecoh dan akses dapat dilakukan.

Sebagai contoh:

http://www.aurora.com/../../../../../winnt/repair/sam._

Script diatas sangat mudah untuk dimengerti: web server diminta untuk mencari file dalam direktori "../../../../../winnt/repair/sam._". Substring "../" memerintahkan web server untuk naik satu direktori, sehingga lima kali "../" secara berurutan akan membuat web server untuk mencari file "winnt/repair/sam._" dalam document root. Banyaknya jumlah "../" tidak menjadi masalah selama direktori root dapat dicapai.

7. ACTIVE FINGERPRINTING

Active OS Fingerprinting adalah sebuah metode untuk mengenali sistem operasi pada sebuah host dengan cara "memancing" host target untuk mengirimkan paket-paketnya sesuai dengan keinginan kita. Serangkaian paket tertentu dikirimkan ke host target untuk mendapatkan respon yang tertentu pula. Dari karakteristik masing-masing respon inilah, jenis sistem operasi dapat diprediksikan.

Setiap sistem operasi memberikan respon yang berbeda. Namun tidak setiap sistem operasi yang sama akan memberikan respon yang sama pula. Oleh karena itu fingerprinting memuat serangkaian pengujian untuk mencocokkan respon dengan

database yang dimiliki untuk mendapatkan probabilitas keakuratan terbesar.

Pengujian dilakukan dengan mengirimkan sembilan (9) paket pengujian yang berbeda, seperti yang tercantum pada tabel 1.

Tabel 1. Pengujian Fingerprinting

TEST	PENJELASAN
TSeq	Serangkaian paket SYN dikirim ke host target untuk mengetahui pola TCP *sequence number*-nya
T1	Sebuah paket SYN dengan opsi (WNMTE) dikirimkan ke sebuah port TCP yang terbuka
T2	Sebuah paket NULL dengan opsi (WNMTE) dikirimkan ke sebuah port TCP yang terbuka
T3	Sebuah paket SYN, FIN, PSH, URG dengan opsi (WNMTE) dikirimkan ke sebuah port TCP yang terbuka
T4	Sebuah paket ACK dengan opsi (WNMTE) dikirimkan ke sebuah port TCP yang terbuka
T5	Sebuah paket SYN dengan opsi (WNMTE) dikirimkan ke sebuah port TCP yang tertutup
T6	Sebuah paket ACK dengan opsi (WNMTE) dikirimkan ke sebuah port TCP yang tertutup
T7	Sebuah paket SYN, FIN, PSH, URG dengan opsi (WNMTE) dikirimkan ke sebuah port TCP yang tertutup
PU	Sebuah paket dikirimkan ke port UDP yang tertutup

Dari masing-masing pengujian, ada beberapa metriks yang akan dianalisa, seperti dijelaskan dalam tabel 2:

Tabel 2. Metriks Fingerprinting

Metriks	Nilai yang valid	Penjelasan
Response	Y = ada respon N = tidak ada respon	Untuk mengetahui apakah host target merespon paket yang dikirimkan.
Don't Fragment	Y = bit DF diset N = bit DF tidak diset	Untuk mengetahui apakah paket respon dari host target berisi bit DF.
Window Size	Nilai integer 2-byte yang direpresentasikan dalam bilangan heksadesimal	Untuk mengetahui besar tiap paket respon
ACK Sequence	O = ack zero S = ack sequence number S++ = ack sequence + 1	Tipe penomoran acknowledge sequence pada setiap TCP session
Flags	S = flag SYN A = flag ACK R = flag RST F = flag FIN U = flag URG P = flag PSH	Untuk mengetahui flag apa saja yang berada dalam paket respon
Options	M = MSS E = Echoed MSS W = Window Scale T = Timestamp N = No Option	Data opsi yang dikirim dalam paket respon dari host target; dapat terdiri dari beberapa opsi dalam urutan yang tidak menentu

8. UJI COBA

a. Hasil Uji Coba Kebenaran
   - Modul OS Fingerprinting

Tabel 3. Hasil pengujian kebenaran OS Fingerprinting

Host	Pengujian (%)					Rata-rata (%)
	I	II	III	IV	V	
10.126.10.245	75	75	75	75	75	75
10.126.11.109	84.62	84.62	84.62	84.62	84.62	84.62
10.126.11.212	84.62	84.62	84.62	84.62	84.62	84.62
10.126.11.246	88.46	88.46	88.46	88.46	88.46	88.46

   - Modul Port Scanning

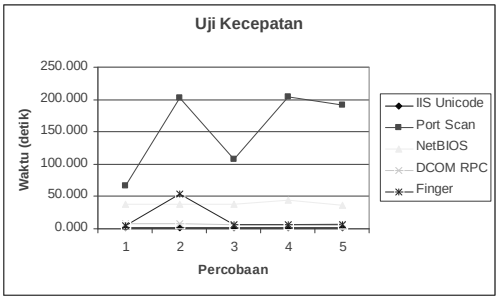
Tabel 4. Hasil pengujian kebenaran Port Scan

Host	Pengujian (%)					Rata-rata (%)
	I	II	III	IV	V	
10.126.10.245	100	100	100	100	100	100
10.126.11.109	100	100	100	100	100	100
10.126.11.212	100	100	100	100	100	100
10.126.11.246	100	100	100	100	100	100

b. Hasil Uji Coba Kecepatan

Tabel 5. Hasil pengujian kecepatan modul

Modul	Jumlah Host	Rata-rata (detik)	Rata-rata/host (detik)
IIS Unicode	39	1.994	0.051118
Port Scan	12	154.481	12.87343
NetBIOS	26	38.806	1.492546
DCOM RPC	39	7.130	0.182815
Fingerprint	26	15.412	0.592785

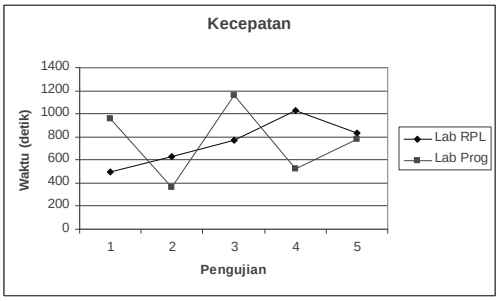


Gambar 3. Grafik pengujian kecepatan modul

c. Hasil Uji Coba Kehandalan

Tabel 6. Kecepatan hasil pengujian kehandalan

Host	Jumlah Host	Rata-rata (detik)	Rata-rata/host (detik)
Lab RPL	27	753	27.88889
Lab Prog	39	757	19.41026



Gambar 4. Grafik kecepatan pengujian kehandalan

Tabel 6. Tingkat kesalahan hasil pengujian kehandalan

Host	Jumlah Host	Pengujian (%)					Rata-rata (%)
		I	II	III	IV	V	
Lab RPL	27	7.4	7.4	7.4	18.5	11.1	10.3
Lab Prog	39	2.5	2.5	2.5	0	0	2.2

d. Analisa Hasil Uji Coba

Dari hasil uji coba diatas dapat diketahui bahwa semakin besar jumlah host tidak berarti semakin menurunkan kecepatan pemindaian. Faktor yang paling mempengaruhi kecepatan pemindaian adalah topologi dan kondisi lalu lintas jaringan. Semakin jauh jarak antara domain perangkat lunak dengan domain komputer target, maka kecepatan dan tingkat akurasi akan semakin menurun. Kemudian diperoleh juga data bahwa kinerja perangkat lunak menjadi lebih baik jika host yang akan dipindai tidak terlalu besar. Jumlah yang disarankan adalah satu hingga sepuluh target.

Komputer yang memiliki firewall juga mempengaruhi kecepatan dan reliabilitas pemindaian, karena firewall cenderung untuk tidak merespon terhadap percobaan pemindaian sehingga perangkat lunak terus berusaha menunggu respon sampai batas waktu tertentu. Hal ini yang menyebabkan akurasi data dan kecepatan pemindaian menurun.

9. KESIMPULAN DAN SARAN

Berdasarkan pada perancangan dan pembuatan sistem terhadap permasalahan yang diangkat, maka dapat diambil kesimpulan sebagai berikut:

1). Perangkat lunak dapat melakukan fungsinya dengan baik.
2). Kecepatan pemindaian relatif cepat jika memperoleh respon dengan baik dari komputer target, namun akan menurun drastis jika perangkat lunak tidak memperoleh respon yang umumnya disebabkan oleh kondisi lalu lintas jaringan yang padat, maupun komputer yang dilindungi oleh firewall.
3). Perangkat lunak memiliki kinerja yang lebih baik jika digunakan untuk memindai tidak lebih dari sepuluh komputer.
4). CPU load tinggi menyebabkan menurunnya efektifitas perangkat lunak.

Berikut ini adalah saran untuk kemungkinan pengembangan lebih lanjut dari hasil perancangan dan pembuatan aplikasi sistem dalam tugas akhir ini:

1). Mengoptimasi kode program untuk meminimalkan penggunaan CPU time
2). Menggunakan metode yang lebih baik pada pencocokan OS fingerprint
3). Mengoptimasi kinerja multi-threading.
4). Menyediakan fasilitas laporan hasil pemindaian ke format yang dapat dicetak seperti HTML, DOC, ataupun JPG.

10. DAFTAR PUSTAKA

1. San Bergmans, “My TCP/IP Projects”, http://www.xs4all.nl_sbp\projects\tcpip\tcpip.htm. 2003.

2. Ryan Spangler, “Analysis of Remote Active Operating System Fingerprinting Tools”. osdetection.pdf. 2003.

3. Gary R. Wright & W. Richard. Stevens, “TCP/IP Illustrated, Volume 2: The Implementation”. Addison-Wesley. 1995.
4. Fyodor Yarochkin, “Remote OS detection via TCP/IP Stack FingerPrinting”. http://www.insecure.org/nmap/nmap-fingerprinting-article.txt. 2002.