BAB III
Evaluasi Kemanan Sistem Informasi

Pendahuluan

Meski sebuah sistem informasi sudah dirancang memiliki perangkat pengamanan, dalam operasi masalah keamanan harus selalu dimonitor. Hal ini disebabkan oleh beberapa hal, antara lain:

- Ditemukannya lubang keamanan (*security hole*) yang baru. Perangkat lunak dan perangkat keras biasanya sangat kompleks sehingga tidak mungkin untuk diuji seratus persen..
- Kesalahan konfigurasi. Kadang-kadang karena lalai atau alpa, konfigurasi sebuah sistem kurang benar sehingga menimbulkan lubang keamanan.
- Penambahan perangkat baru (hardware dan/atau software) yang menyebabkan menurunnya tingkat security atau berubahnya metoda untuk mengoperasikan sistem.

Sumber lubang keamanan

- Lubang keamanan (*security hole*) dapat terjadi karena beberapa hal; salah disain (*design flaw*), salah implementasi, salah konfigurasi, dan salah penggunaan.
- Salah Disain

  -Lubang keamanan yang ditimbulkan oleh salah disain umumnya jarang terjadi. Akan tetapi apabila terjadi sangat sulit untuk diperbaiki. Akibat disain yang salah, maka biarpun dia diimplementasikan dengan baik, kelemahan dari sistem akan tetap ada.

  -Contoh lain lubang keamanan yang dapat dikategorikan kedalam kesalahan disain adalah disain urutan nomor (*sequence numbering*) dari paket TCP/IP. Kesalahan ini dapat dieksploitasi sehingga timbul masalah yang dikenal dengan nama "*IP spoofing*", yaitu sebuah host memalsukan diri seolah-olah menjadi host lain dengan membuat paket palsu setelah mengamati urutan paket dari host yang hendak diserang. Bahkan dengan mengamati cara mengurutkan nomor packet bisa dikenali sistem yang digunakan.

Sumber lubang keamanan

- Lubang keamanan (*security hole*) dapat terjadi karena beberapa hal; salah disain (*design flaw*), salah implementasi, salah konfigurasi, dan salah penggunaan.
- **Implementasi kurang baik**

  -Lubang keamanan yang disebabkan oleh kesalahan implementasi sering terjadi. Banyak program yang diimplementasikan secara terburu-buru sehingga kurang cermat dalam pengkodean. Akibatnya cek atau testing yang harus dilakukan menjadi tidak dilakukan. Sebagai contoh, seringkali batas ("*bound*") dari sebuah "*array*" tidak dicek sehingga terjadi yang disebut *out-of-bound array* atau *buffer overflow* yang dapat dieksploitasi (misalnya overwrite ke variable berikutnya).

  -Contoh lain sumber lubang keamanan yang disebabkan oleh kurang baiknya implementasi adalah kealpaan memfilter karakter-karakter yang aneh-aneh yang dimasukkan sebagai input dari sebuah program (misalnya input dari *CGI-script*2) sehingga sang program dapat mengakses berkas atau informasi yang semestinya tidak boleh diakses.

Sumber lubang keamanan

- Lubang keamanan (*security hole*) dapat terjadi karena beberapa hal; salah disain (*design flaw*), salah implementasi, salah konfigurasi, dan salah penggunaan.
- **Salah konfigurasi**
  - Meskipun program sudah diimplementasikan dengan baik, masih dapat terjadi lubang keamanan karena salah konfigurasi. Contoh masalah yang disebabkan oleh salah konfigurasi adalah berkas yang semestinya tidak dapat diubah oleh pemakai secara tidak sengaja menjadi "*writeable*".
  - Contoh lain adalah mekanisme sharing – dimana dengan membuka folder untuk publik kemungkinan konfigurasi hak akses didalamnya tidak dilakukan konfigurasi.
- **Salah menggunakan program atau sistem**
  - Salah penggunaan program dapat juga mengakibatkan terjadinya lubang keamanan. Kesalahan menggunakan program yang dijalankan dengan menggunakan account root (super user) dapat berakibat fatal.
  - misalnya, ingin melihat daftar berkas di sebuah direktori dengan memberikan perintah "dir *.*" ternyata salah memberikan perintah menjadi "del *.*" (yang juga menghapus seluruh file di direktori tersebut).

Sumber lubang keamanan

- Lubang keamanan (*security hole*) dapat terjadi karena beberapa hal; salah disain (*design flaw*), salah implementasi, salah konfigurasi, dan salah penggunaan.
- **Salah konfigurasi**
  - Meskipun program sudah diimplementasikan dengan baik, masih dapat terjadi lubang keamanan karena salah konfigurasi. Contoh masalah yang disebabkan oleh salah konfigurasi adalah berkas yang semestinya tidak dapat diubah oleh pemakai secara tidak sengaja menjadi "*writeable*".
  - Contoh lain adalah mekanisme sharing – dimana dengan membuka folder untuk publik kemungkinan konfigurasi hak akses didalamnya tidak dilakukan konfigurasi.
- **Salah menggunakan program atau sistem**
  - Salah penggunaan program dapat juga mengakibatkan terjadinya lubang keamanan. Kesalahan menggunakan program yang dijalankan dengan menggunakan account root (super user) dapat berakibat fatal.
  - misalnya, ingin melihat daftar berkas di sebuah direktori dengan memberikan perintah "dir *.*" ternyata salah memberikan perintah menjadi "del *.*" (yang juga menghapus seluruh file di direktori tersebut).

Penguji keamanan sistem

- Dikarenakan banyaknya hal yang harus dimonitor, administrator dari sistem informasi membutuhkan "*automated tools*", perangkat pembantu otomatis, yang dapat membantu menguji atau meng-evaluasi keamanan sistem yang dikelola.
- **UNIX C***ops , Tripwire, Satan/Saint, SBScan*: localhost security scanner
- Untuk sistem yang berbasis Windows NT ada juga program semacam, misalnya program *Ballista* yang dapat diperoleh dari:<http://www.secnet.com>
- Selain program-program (tools) yang terpadu (*integrated*) seperti yang terdapat pada daftar di atas, ada banyak program yang dibuat oleh hackers untuk melakukan "coba-coba".
  - *crack*: program untuk menduga atau memecahkan password dengan menggunakan sebuah atau beberapa kamus (*dictionary*).
  - *land* dan *latierra*: program yang dapat membuat sistem Windows 95/NT menjadi macet (*hang*, *lock up*). Program ini mengirimkan sebuah paket yang sudah di"*spoofed*" sehingga seolah-olah paket tersebut berasal dari mesin yang sama dengan menggunakan port yang terbuka .
  - *ping-o-death*: sebuah program (*ping*) yang dapat meng-crash-kan Windows 95/NT dan beberapa versi Unix.
  - *winuke*: program untuk memacetkan sistem berbasis Windows

Probing Services

- Servis di Internet umumnya dilakukan dengan menggunakan protokol TCP atau UDP. Setiap servis dijalankan dengan menggunakan port yang berbeda, misalnya :
  - SMTP, untuk mengirim dan menerima e-mail, TCP, port 25
  - DNS, untuk domain, UDP dan TCP, port 53
  - HTTP, web server, TCP, port 80
  - POP3, untuk mengambil e-mail, TCP, port 110
- Pemilihan servis apa saja tergantung kepada kebutuhan dan tingkat keamanan yang diinginkan. Sayangnya seringkali sistem yang dibeli atau dirakit menjalankan beberapa servis utama sebagai “default”. Kadang- kadang beberapa servis harus dimatikan karena ada kemungkinan dapat dieksploitasi oleh cracker. Untuk itu ada beberapa program yang dapat digunakan untuk melakukan “*probe*” (meraba) servis apa saja yang tersedia. Program ini juga dapat digunakan oleh kriminal untuk melihat servis apa saja yang tersedia di sistem yang akan diserang dan berdasarkan data-data yang diperoleh dapat melancarkan serangan.

Probing Services

- Servis di Internet umumnya dilakukan dengan menggunakan protokol TCP atau UDP. Setiap servis dijalankan dengan menggunakan port yang berbeda, misalnya :
  - SMTP, untuk mengirim dan menerima e-mail, TCP, port 25
  - DNS, untuk domain, UDP dan TCP, port 53
  - HTTP, web server, TCP, port 80
  - POP3, untuk mengambil e-mail, TCP, port 110
- Pemilihan servis apa saja tergantung kepada kebutuhan dan tingkat keamanan yang diinginkan. Sayangnya seringkali sistem yang dibeli atau dirakit menjalankan beberapa servis utama sebagai “default”. Kadang- kadang beberapa servis harus dimatikan karena ada kemungkinan dapat dieksploitasi oleh cracker. Untuk itu ada beberapa program yang dapat digunakan untuk melakukan “*probe*” (meraba) servis apa saja yang tersedia. Program ini juga dapat digunakan oleh kriminal untuk melihat servis apa saja yang tersedia di sistem yang akan diserang dan berdasarkan data-data yang diperoleh dapat melancarkan serangan.

Probing Services

- Untuk beberapa servis yang berbasis TCP/IP, proses probe dapat dilakukan dengan menggunakan program telnet. Misalnya untuk melihat apakah ada servis e-mail dengan menggunakan SMTP digunakan telnet ke port 25.
- Untuk servis lain, seperti POP atau POP3 dapat dilakukan dengan cara yang sama dengan menggunakan nomor “port” yang sesuai dengan servis yang diamati.

```
unix% telnet target.host.com 25
Trying 127.0.0.1...
Connected to target.host.com.
Escape character is '^]'.
220 dma-baru ESMTP Sendmail 8.9.0/8.8.5; Mon, 22 Jun 1998 10:18:54 +0700
```

```
unix% telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK QPOP (version 2.2) at dma-baru.paume.itb.ac.id starting.
+<20651.898485542@dma-baru.paume.itb.ac.id>
quit
+OK Pop server at dma-baru.paume.itb.ac.id signing off.
Connection closed by foreign host.
```

Probing Services

- Proses probing tersebut dapat dilakukan secara otomatis, sehinga menguji semua port yang ada, dengan menggunakan beberapa program paket seperti didaftarkan di bawah ini.
  - Paket probe untuk sistem UNIX : *nmap , strobe , tcpprobe*
  - Probe untuk sistem Window 95/98/NT : *NetLab, Cyberkit, Ogre*

- **Mendeteksi Probling**
- Apabila anda seorang sistem administrator, anda dapat memasang program yang memonitor adanya probing ke sistem yang anda kelola. Probing biasanya meninggalkan jejak di berkas log di sistem anda.
- Selain itu, ada juga program untuk memonitor probe seperti paket program *courtney, portsentry* dan *tcplogd*.

```
root# tail /var/log/syslog
May 16 15:40:42 epson tcplogd: "Syn probe"
notebook[192.168.1.4]:[8422]->epson[192.168.1.2]:[635]
May 16 15:40:42 epson tcplogd: "Syn probe"
notebook[192.168.1.4]:[8423]->epson[192.168.1.2]:ssl-ldap
May 16 15:40:42 epson tcplogd: "Syn probe"
notebook[192.168.1.4]:[8426]->epson[192.168.1.2]:[637]
May 16 15:40:42 epson tcplogd: "Syn probe"
notebook[192.168.1.4]:[8429]->epson[192.168.1.2]:[638]
May 16 15:40:43 epson tcplogd: "Syn probe"
notebook[192.168.1.4]:[8430]->epson[192.168.1.2]:[639]
```

OS fingerprinting

- Mengetahui *operating system* (OS) dari target yang akan diserang merupakan salah satu pekerjaan yang dilakukan oleh seorang cracker. Setelah mengetahui OS yang dituju, dia dapat melihat database kelemahan sistem yang dituju. *Fingerprinting* merupakan istilah yang umum digunakan untuk menganalisa OS sistem yang dituju.
- *Fingerprinting* dapat dilakukan dengan berbagai cara. Cara yang paling konvensional adalah melakukan telnet ke server yang dituju. Jika server tersebut kebetulan menyediakan servis telnet, seringkali ada banner yang menunjukkan nama OS beserta versinya.
- Cara fingerprinting yang lebih canggih adalah dengan menganalisa respon sistem terhadap permintaan (request) tertentu. Misalnya dengan menganalisa nomor urut packet TCP/IP yang dikeluarkan oleh server tersebut dapat dipersempit ruang jenis dari OS yang digunakan.
- Ada beberapa tools untuk melakukan deteksi OS ini antara lain:
  - *Nmap*
  - *queso*

Penggunaan Program Penyerang

- Salah satu cara untuk mengetahui kelemahan sistem informasi anda adalah dengan menyerang diri sendiri dengan paket-paket program penyerang (*attack*) yang dapat diperoleh di Internet. Dengan menggunakan program ini anda dapat mengetahui apakah sistem anda rentan dan dapat dieksploitasi oleh orang.

jangan menggunakan program-program tersebut untuk menyerang sistem lain

- Selain program penyerang yang sifatnya agresif melumpuhkan sistem yang dituju, ada juga program penyerang yang sifatnya melakukan pencurian atau penyadapan data. Untuk penyadapan data, biasanya dikenal dengan istilah "***sniffer***". Meskipun data tidak dicuri secara fisik (dalam artian menjadi hilang), sniffer ini sangat berbahaya karena dia dapat digunakan untuk menyadap password dan informasi yang sensitif. Ini merupakan serangan terhadap aspek privacy.
- Contoh program penyadap (*sniffer*) antara lain: *pcapture* (Unix), *sniffit* (Unix), *tcpdump* (Unix), *WebXRay* (Windows), Chain & abel (Windows)

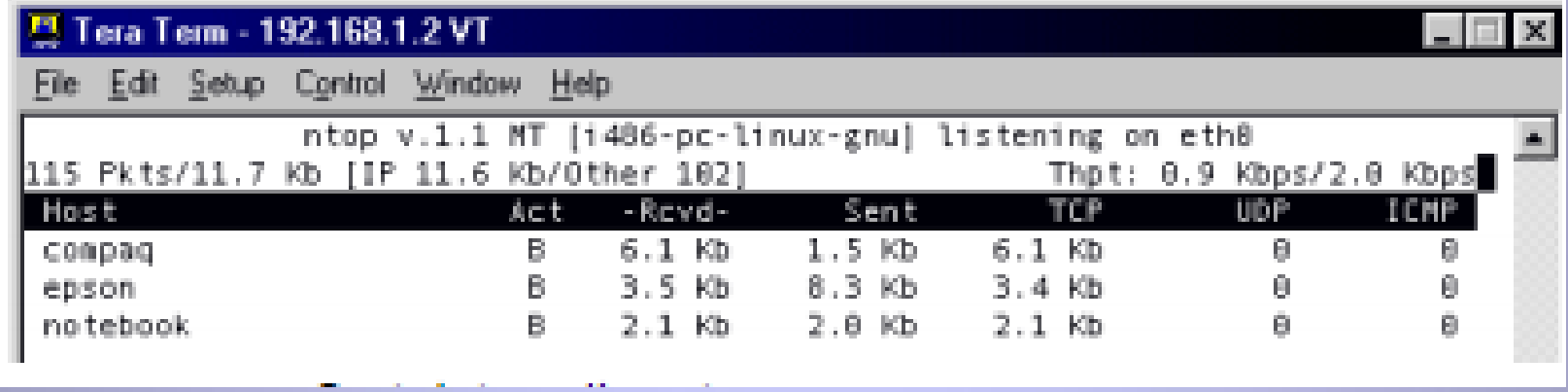
Penggunaan sistem pemantau jaringan

- Sistem pemantau jaringan (*network monitoring*) dapat digunakan untuk mengetahui adanya lubang keamaman. Misalnya apabila anda memiliki sebuah server yang semetinya hanya dapat diakses oleh orang dari dalam, akan tetapi dari pemantau jaringan dapat terlihat bahwa ada yang mencoba mengakses melalui tempat lain. Selain itu dengan pemantau jaringan dapat juga dilihat usaha-usaha untuk melumpuhkan sistem dengan melalui *denial of service attack* (DoS) dengan mengirimkan packet yang jumlahnya berlebihan.
- Network monitoring biasanya dilakukan dengan menggunakan protokol SNMP *(Simple Network Management Protocol*)
- Contoh-contoh program network monitoring / management antara lain: *Etherboy* (Windows), *Etherman* (Unix) , HP *Openview* (Windows), *Packetboy* (Windows), *Packetman* (Unix), SNMP Collector (Windows), *Webboy* (Windows)
- Contoh program pemanatu jaringan yang tidak menggunakan SNMP antara lain:
  - *iplog, icmplog, updlog,* yang merupakan bagian dari paket *iplog* untuk memantau paket IP, ICMP, UDP.
  - *iptraf*, sudah termasuk dalam paket Linux Debian *netdiag*
  - *netwatch*, sudah termasuk dalam paket Linux Debian *netdiag*
  - *ntop*, memantau jaringan seperti program *top* yang memantau proses di sistem Unix (lihat contoh gambar tampilannya)
  - *trafshow*, menunjukkan traffic antar hosts dalam bentuk text-mode

Penggunaan sistem pemantau jaringan

- Contoh peragaan *trafshow* di sebuah komputer yang bernama epson, dimana ditunjukkan sesi *ssh* (dari komputer compaq) dan *ftp* (dari komputer notebook).

```
epson (traffic) 0 days 00 hrs 00 min 46 sec
tcp   epson.insan.co.id    ssh     compaq 558 3096          832
tcp   epson.insan.co.id    ftp     notebook 1054 422        381
9K total, 0K bad, 0K nonip - 9K tcp, 0K udp, 0K icmp, 0K unkn
```



Merci bien
ありがとう
Matur Nuwun
Hatur Nuhun
Obrigado
Dank
Thanks
Matur se Kelangkong
Syukron
Kheili Mamnun
ευχαριστιες
Danke
Grazias
谢谢
Terima Kasih


Irawan_afrianto@yahoo.com


Irawan.afrianto


@Irawan_afrianto


+628170223513