

CIS/TCOM 551

Computer and Network Security

Spring 2005

Lecture 7

Outline for Today

- Detecting viruses and worms
- Intrusion detection in general
- Defenses against viruses/worms/general intruders
 - Tools for determining system vulnerability

- Research Paper: "Automated Worm Fingerprinting"
 - Singh, Estan, Varghese, and Savage
 - (may not cover until next time)

Virus & Worm Signatures

- Viruses & worms can't be completely invisible:
 - Code must be stored somewhere
 - They must do something (e.g. propagate) when they run
- Fragments of the virus/worm code itself
 - Strings “kindly check the attached LOVELETTER”
- Effects on the computing environment
 - Changes to the Windows registry
- Propagation Behavior
 - Copying/modifying system files.
- (More generally, any attack will have some observable effect...)

Detecting Viruses & Worms

- How do you even know there's a problem?
 - System administrators/users notice unusual behavior
 - Automatic intrusion/anomaly detection
 - Internet "telescopes"
 - Lure in an attack: honeypots / honeynets

- Techniques:
 - Integrity checks
 - Heuristic detection
 - Signature checking
 - System auditing

Virus Scanners

- Search the system for virus signatures
 - Main memory
 - All files in file system
 - Should also check boot sector
- Where to scan?
 - At each host (e.g. Norton Antivirus)
 - At the firewall
 - At the mail server
- When to scan?
 - On access (when a program is run)
 - On demand (at user's request, or scheduled)
 - When e-mail is received?
 - Before web content is displayed?

Virus/Worm Scanning

- Pros
 - Effectively detects *known* viruses/worms before they can cause harm
 - Few false alarms
- Cons
 - Can detect only viruses/worms with known signatures
 - Performance penalty (due to scanning)
 - Signature set must be kept up to date
 - Virus/worm writers can easily change signatures
- ==> Generate signatures automatically
 - Automated Worm Fingerprinting (more in a bit)

Software Integrity Checks

- Compute a hash or checksum of executable files
 - Assumes the software to be virus free!
 - Store the hash information for later verification
- Verify new hash vs. saved one during scan
 - Also used for ensuring that software is not corrupted/modified when shipped over the network.
- Pros:
 - Can detect corruption of executables too
 - Reliable
 - Doesn't require virus signatures
- Cons:
 - False positives (i.e. recompilation)
 - Can't use it on documents (they change too often)
 - Not supported by most vendors

Heuristic Detection

- Collection of ad hoc rules that identifies virus behavior or virus-like programs
 - Modification of system executables
 - Modification of “template documents” like normal.doc
 - Self-modifying and self-referential code
 - Atypical or abnormal behavior
- Pros
 - Perhaps able to detect unknown viruses/worms
 - Can build tools to look for these features
- Cons
 - Heuristics are expensive and hard to develop.
 - Too many false positives?

Detecting Attacks

- Attacks (against computer systems) usually consist of several stages:
 - Finding software vulnerabilities
 - Exploiting them
 - Hiding/cleaning up the exploit
- Attackers care about finding vulnerabilities:
 - What machines are available?
 - What OS / version / patch level are the machines running?
 - What additional software is running?
 - What is the network topology?
- Attackers care about not getting caught:
 - How detectible will the attack be?
 - How can the attacker cover her tracks?
- Programs can automate the process of finding/exploiting vulnerabilities.
 - Same tools that sys. admins. use to audit their systems...
 - A worm is just an automatic vulnerability finder/exploiter...

Attacker Reconnaissance

- Network Scanning
 - Existence of machines at IP addresses
 - Attempt to determine network topology
 - ping, tracert
- Port scanners
 - Try to detect what processes are running on which ports, which ports are open to connections.
 - Typical machine on the internet gets 10-20 port scans per day!
 - Can be used to find hit lists for flash worms
- Web services
 - Use a browser to search for CGI scripts, Javascript, etc.

Determining OS information

- Gives a lot of information that can help an attacker carry out exploits
 - Exact version of OS code can be correlated with vulnerability databases
- Sadly, often simple to obtain this information:
 - Just try telnet

```
playground~> telnet hpux.u-aizu.ac.jp
Trying 163.143.103.12 ...
Connected to hpux.u-aizu.ac.jp.
Escape character is '^]'.
HP-UX hpux B.10.01 A 9000/715 (ttyp2)

login:
```

Determining OS

- Or ftp:

```
$ ftp ftp.netscape.com 21
Connected to ftp.gftp.netscape.com.
220-36
220 ftpnscp.newaol.com FTP server (SunOS 5.8) ready.
Name (ftp.netscape.com:stevez):
331 Password required for stevez.
Password:
530 Login incorrect.
ftp: Login failed.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> system
215 UNIX Type: L8 Version: SUNOS
ftp>
```

Determining OS

- Exploit different implementations of protocols
 - Different OS's have different behavior in some cases
- Consider TCP protocol, there are many flags and options, and some unspecified behavior
 - Reply to bogus FIN request for TCP port (should not reply, but some OS's do)
 - Handling of invalid flags in TCP packets (some OS's keep the invalid flags set in reply)
 - Initial values for RWS, pattern in random sequence numbers, etc.
 - Can narrow down the possible OS based on the combination of implementation features
- Tools can automate this process

Auditing: Remote auditing tools

- Several utilities available to “attack” or gather information about services/daemons on a system.
 - SATAN (early 1990’s):
[Security Administrator Tool for Analyzing Networks](#)
 - SAINT - Based on SATAN utility
 - SARA - Also based on SATAN
 - Nessus - Open source vulnerability scanner
 - <http://www.nessus.org>
 - Nmap
- Commercial:
 - ISS scanner
 - Cybercop

Nmap screen shot

File View Help

Target(s): Scan Exit

Scan Discover Timing Files Options

Scan Type
SYN Stealth Scan
Relay Host:

Scanned Ports
Most Important [fast]
Range:

Scan Extensions
 RPC Scan Identd Info OS Detection Version Probe

```
Starting nmap 3.49 ( http://www.insecure.org/nmap/ ) at 2003-12-19 14:28 PST
Interesting ports on www.insecure.org (205.217.153.53):
(The 1212 ports scanned but not shown below are in state: filtered)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 3.1p1 (protocol 1.99)
25/tcp    open  smtp     gmail smtpd
53/tcp    open  domain   ISC Bind 9.2.1
80/tcp    open  http     Apache httpd 2.0.39 ((Unix) mod_perl/1.99_07-dev Perl/v5.6.1)
113/tcp   closed auth
Device type: general purpose
Running: Linux 2.4.X|2.5.X
OS details: Linux Kernel 2.4.0 - 2.5.20
Uptime 212.119 days (since Wed May 21 12:38:26 2003)

Nmap run completed -- 1 IP address (1 host up) scanned in 33.792 seconds
```

Command

Kinds of Auditing done

- Nessus web pages:
 - Backdoors
 - CGI abuses
 - Denial of Service
 - Finger abuses
 - Firewalls
 - FTP
 - Gain a shell remotely
 - Gain root remotely
 - Netware
 - NIS
 - Port scanners
 - Remote file access
 - RPC
 - Settings
 - SMTP problems
 - SNMP
 - Useless services
 - Windows
 - Windows : User management
- Doing this kind of auditing by hand is complex and error prone
- These tools aren't fool proof or complete.

Defenses

- Disable ping service, unneeded ports
 - Not always possible (you do want some connectivity!)
- Firewalls and NATs
 - Filter out inappropriate packets
 - Scanners use broadcast, multicast packets, clever port/flag combinations to thwart firewall filters
- Keep audit logs of requests
 - Generates a lot of data, hard to sift through
 - Clever port scan packages use a “drip scan” approach, sprinkling their scan packets sparsely across several hours or days
- Intrusion detection

Intrusion Detection

- The process of monitoring computers and networks to analyze and detect signs of security threats.
- Layered security is good
 - More layers means more chances to thwart attacks and more places where the attack might be discovered
- Basic detection alone is insufficient
 - Localization (i.e. What machines are infected.)
 - Identification (i.e. What is the nature of the attack.)
 - Assessment (i.e. What is the extent of the damage, what action to take.)
- Evaluation of intrusion detection scheme based on:
 - False positives: alerts to situations that aren't attacks
 - False negatives: attacks that are missed

Intrusion Detection Systems

- Network-based IDSs:
 - Use raw packet data to look for possible attacks
 - Run network adapter in promiscuous mode
- Host-based IDSs:
 - Log OS specific events (e.g. system calls)
- Looking for:
 - Known attack patterns (e.g. versus SMTP or port scans)
 - Frequency statistics (to detect DoS)
 - Anomalous behavior (based on some profile of “usual” behavior)

- Example: ZoneAlarm software can detect port scans.

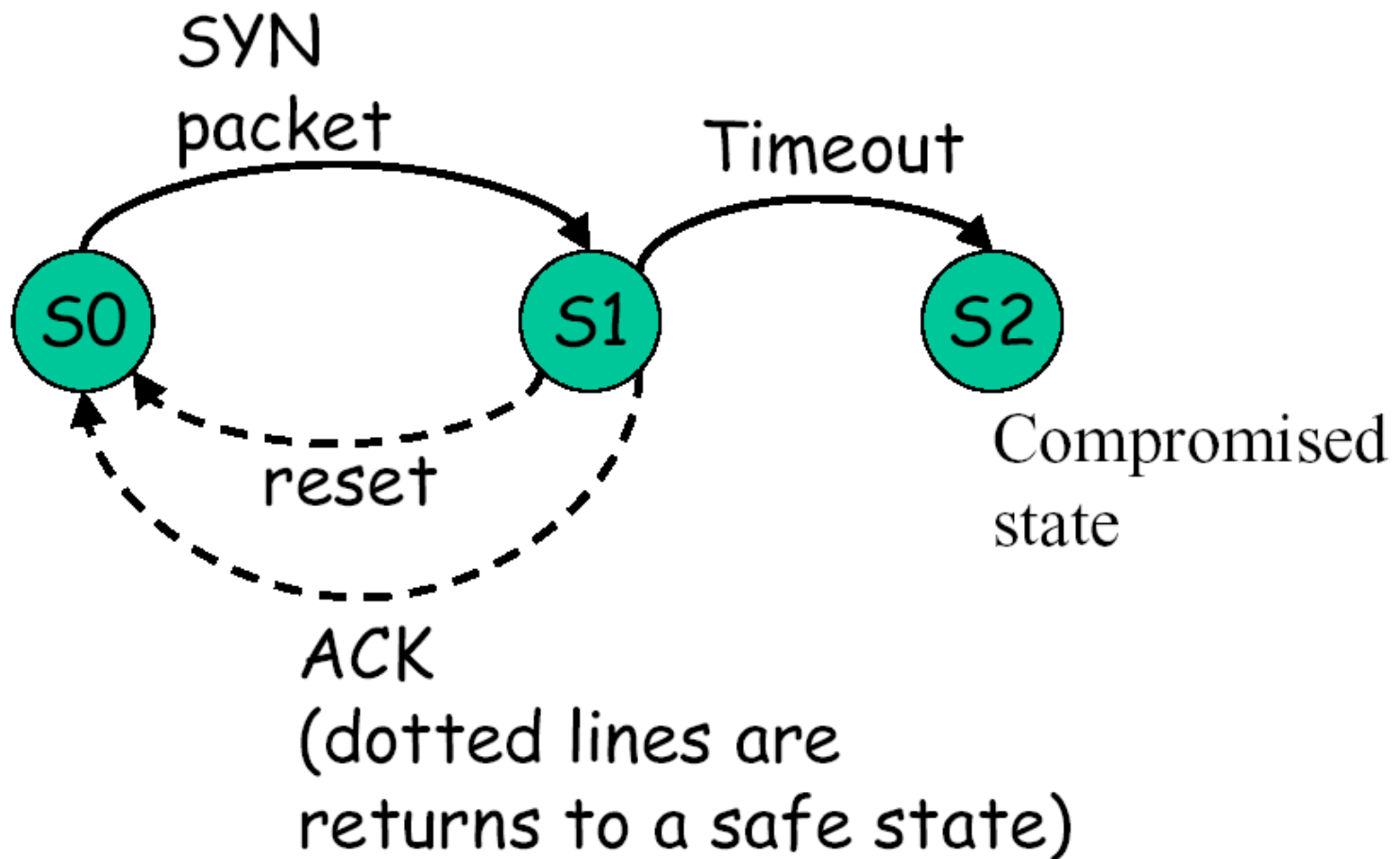
Intrusion Detection

- Known attack signatures
 - As for worms and viruses
- Anomaly detection:
 - Build a profile of “typical” or “normal” behavior
 - What programs are usually running
 - Where do people usually log in from
 - What is the normal timing behavior of your keystrokes
 - How do you arrange your computer desktop...

State Transition Analysis Techniques

- STAT: developed by Eckmann, Vigna, and Kemmerer.
- Defines attack scenarios as sequence of transitions in the security state
 - “state” captures information relevant to security
 - e.g. file ownership, userids, protocol stack states, etc.
- Transitions are the necessary actions that lead to compromise

STAT state diagram example



Snort



- Snort is a lightweight intrusion detection system:
 - Real-time traffic analysis
 - Packet logging (of IP networks)
- Rules based logging to perform content pattern matching to detect a variety of attacks and probes:
 - such as buffer overflows, stealth port scans, CGI attacks, SMB probes, etc.
- Example Rule:

```
alert tcp any any -> 192.168.1.0/24 143 (content:"|E8C0 FFFF  
FF|/bin/sh"; msg:"New IMAP Buffer Overflow detected!");
```

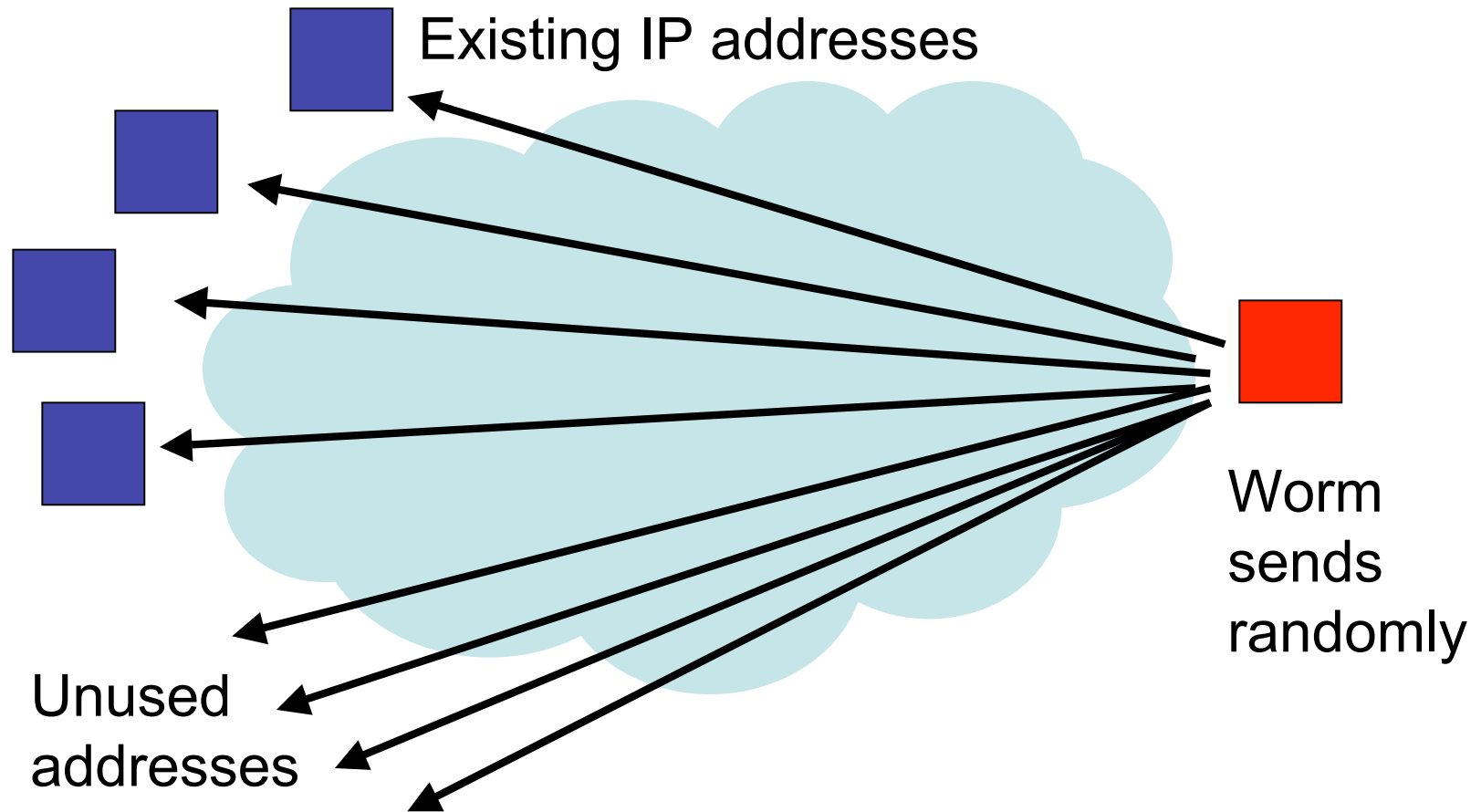
 - Generates an alert on all inbound traffic for port 143 with contents containing the specified attack signature.
- The Snort web site:
 - <http://www.snort.org/docs/>
- Question: How do you come up with the filter rules?

Detection & Prevention Recap

- Many strategies for intrusion detection
 - So far, the techniques we've seen are local to a machine or local network.
- What about large scale behavior?
- Virus/worm scanners work well if known signatures are available
 - Constructing signatures can be hard
 - Reaction time must be very quick (see last lecture)

Internet Telescopes

- Can be used to detect large-scale, wide-spread attacks on the internet.



Internet Telescopes

- Can be used to detect large-scale, wide-spread attacks on the internet.

