Virus Protection and Intrusion Detection

John Mitchell

Topics

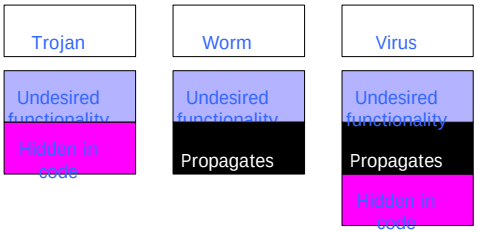
- Trojans, worms, and viruses
- Virus protection
  - Virus scanning methods
- Detecting system compromise
  - Tripwire
- Detecting system and network attacks
  - Scanning system call trace
  - Network intrusion detection

What is a Virus?

- Program embedded in file
- Spreads and does damage
  - Replicator
    - Portion of virus code that reproduces virus
  - Payload
    - Portion of virus code that does some other function
- Categories
  - Boot virus (boot sector of disk)
  - Virus in executable file
  - Macro virus (in file executed by application)

Virus scanner is large collection of many techniques

Three related ideas



Trojan Horse

!!! PKZIP Trojan Horse Version -
(Originally Posted May 1995) !!!

... a fake version of PKZIP is being distributed as PKZ300B.ZIP or PKZ300.ZIP. It is not an official version from PKWARE and it will attempt to erase your hard drive if run.

Not a virus since it doesn't replicate

Worm vs Virus

- A worm is a program
  - can run independently
  - consume the resources of its host
  - can propagate a complete working version of itself to other machines
- A virus is a piece of code
  - inserts itself into a host program
  - cannot run independently
  - requires that host program be run to activate it

Internet Worm

- Released November 1988
  - Program spread through Digital, Sun workstations
  - Exploited Unix security vulnerabilities
    - VAX computers and SUN-3 workstations running versions 4.2 and 4.3 Berkeley UNIX code
- Consequences
  - No immediate damage from program itself
  - Replication and threat of damage
    - Load on network, systems used in attack
    - Many systems shut down to prevent further attack

Consequences of attack

- Morris worm, 1988
  - Infected approximately 6,000 machines
    - 10% of computers connected to the Internet
  - cost ~ $10 million in downtime and cleanup
- Code Red worm, July 16 2001
  - Direct descendant of Morris' worm
  - Infected more than 500,000 servers
    - Programmed to go into infinite sleep mode July 28
  - Caused ~ $2.6 Billion in damages,

Statistics: Computer Economics Inc., Carlsbad, California

Love Bug worm: $8.75 billion ?

Internet Worm Description

- Two parts
  - Program to spread worm
    - look for other machines that could be infected
    - try to find ways of infiltrating these machines
  - Vector program (99 lines of C)
    - compiled and run on the infected machines
    - transferred main program to continue attack
- Security vulnerabilities
  - fingerd – Unix finger daemon
  - sendmail - mail distribution program
  - Trusted logins (.rhosts)
  - Weak passwords

Three ways the worm spread

- Sendmail
  - Exploit debug option in sendmail to allow shell access
- Fingerd
  - Exploit a buffer overflow in the fgets function
  - Apparently, this was the most successful attack
- Rsh
  - Exploit trusted hosts
  - Password cracking

sendmail

- Worm used debug feature
  - Opens TCP connection to machine's SMTP port
  - Invokes debug mode
  - Sends a RCPT TO that pipes data through shell
  - Shell script retrieves worm main program
    - places 40-line C program in temporary file called x$$,l1.c where $$ is current process ID
    - Compiles and executes this program
    - Opens socket to machine that sent script
    - Retrieves worm main program, compiles it and runs

fingerd

- Written in C and runs continuously
- Array bounds attack
  - Fingerd expects an input string
  - Worm writes long string to internal 512-byte buffer
- Attack string
  - Includes machine instructions
  - Overwrites return address
  - Invokes a remote shell
  - Executes privileged commands

Remote shell

- Unix trust information
  - /etc/host.equiv – system wide trusted hosts file
  - /.rhosts and ~/.rhosts – users' trusted hosts file
- Worm exploited trust information
  - Examining files that listed trusted machines
  - Assume reciprocal trust
    - If X trusts Y, then maybe Y trusts X
- Password cracking
    - Worm was running as daemon (not root) so needed to break into accounts to use .rhosts feature
    - Dictionary attack
    - Read /etc/passwd, used ~400 common password strings

The worm itself

- Program is called 'sh'
  - Clobbers argv array so a 'ps' will not show its name
  - Opens all its files, then unlinks (deletes) them so they can't be found
    - since files are open, worm can still access their contents
- Tries to infect as many other hosts as possible
  - When worm successfully connects, forks a child to continue the infection while the parent keeps trying new hosts

Some things the worm did not do

- ... did not delete a system's files,
- ... did not modify existing files,
- ... did not install trojan horses,
- ... did not record or transmit decrypted passwords,
- ... did not try to capture superuser privileges,
- ... did not propagate over UUCP, X.25, DECNET, or BITNET.

Detecting Internet Worm

- Files
  - Strange files appeared in infected systems
  - Strange log messages for certain programs
- System load
  - Infection generates a number of processes
  - Systems were reinfected => number of processes grew and systems became overloaded
    - Apparently not intended by worm's creator

Thousands of systems were shut down

Stopping the worm

- System admins busy for several days
  - Devised, distributed, installed modifications
- Perpetrator
  - Student at Cornell; discovered quickly and charged
  - Sentence: community service and $10,000 fine
    - Program did not cause deliberate damage
    - Tried (failed) to control # of processes on host machines
- Lessons?
  - Security vulnerabilities come from system flaws
  - Diversity is useful for resisting attack
  - "Experiments" can be dangerous

Sources for more information

- Eugene H. Spafford, The Internet Worm: Crisis and Aftermath, CACM 32(6) 678-687, June 1989
- IETF rfc1135
- ftp://coast.cs.purdue.edu/pub/doc/morris_worm
- Page, Bob, "A Report on the Internet Worm", http://www.ee.ryerson.ca:8080/~elf/hack/iworm.html

Other significant worms

- Code Red, July 2001
  - Affects Microsoft Index Server 2.0,
    - Windows 2000 Indexing service on Windows NT 4.0.
    - Windows 2000 that run IIS 4.0 and 5.0 Web servers
  - Exploits known buffer overflow in Idq.dll
- SQL Slammer, January 2003
  - Affects in Microsoft SQL 2000
  - Exploits known buffer overflow vulnerability
    - Server Resolution service vulnerability reported June 2002
    - Patched released in July 2002 Bulletin MS02-39

Code Red

- Sends its code as an HTTP request
- HTTP request exploits buffer overflow
- Malicious code is not stored in a file
  - Placed in memory and then run
- When executed,
  - Worm checks for the file C:\Notworm
    - If file exists, the worm thread goes into infinite sleep state
  - Creates new threads
    - If the date is before the 20th of the month, the next 99 threads attempt to exploit more computers by targeting random IP addresses

SQL Slammer

- Server Resolution vulnerability
  - Two buffer overflow vulnerabilities
    - packet to Resolution Service overwrites system memory
    - the heap in one case, the stack in the other
  - Attack code runs in security context of SQL Server
    - Security context chosen by administrator at installation
    - Default is a Domain User
    - Attacker does not have OS privileges
      - But can create threads and send HTTP requests
      - Damage caused by network overload

Before we talk about viruses ...

- Quiz question
  - What's the longest Starbucks coffee order?
  - Grandé decaf extra-hot blended no-foam caramel macchiato ...

Virus Examples

- Jerusalem
  - One oldest and most common; many variants
  - Will infect both .EXE and .COM files
  - Every Friday 13th, deletes programs run that day
- Melissa
  - Word macro virus spread by email
  - Initially distributed in internet group alt.sex
  - Sent in a file called LIST.DOC
  - When opened, macro emails to 50 people listed in the address book of the user

Melissa Email

From: (name of infected user)
Subject: Important Message From (name of infected user)
To: (50 names from alias list)

Here is that document you asked for … don't show anyone else ;-)

- Recipients likely to open a document from someone they know

FunLove Virus

- Also called W32.FunLove.4099
- Modifies WinNT kernel
  - Works only if infected user is administrator
  - Modifies access control code so all users have access to all files

Viruses – What's Out There?

- Wild List http://www.wildlist.org/
  - Industry standard
  - Currently 64 participants
    - mostly from security companies
    - keep watch for active viruses
  - About 200 current sightings
    - Virus needs two independent sightings to stay on list
- Virus families
  - Many viruses reuse proven replicators

Who writes viruses?

- Limited scientific study
  - Sarah Gordon papers at http://www.research.ibm.com/antivirus/SciPapers.htm
- Identified four groups by survey
  - Early adolescent, College student, Adult/professional, Ex-writer of viruses
- Trends
  - "Those who have continued a normal ethical development have aged out of virus writing"
  - Some are older and more skilled than before
    - Viruses like Zhengxi and Concept point to an advanced knowledge of programming techniques

How hard is it to do?

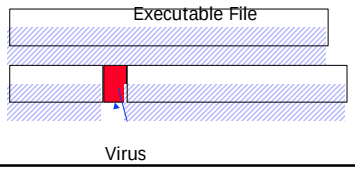
- Google search: virus construction toolkit
- First link:
  - Name: OVCT
  - Type: Virus Creation Kit
  - Info:
    Overwritting Virus Construction Toolkit is a virus source generator program designed for makeing overwritting virii.

spelling errors

- Links to ~40 other construction kits at http://www.ebcvg.com/creation_labs.php
  - I do not recommend downloading or running these!!

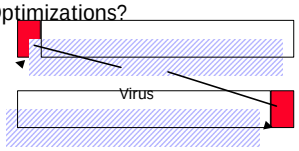
Simple File-Infecting Virus

- Propagate identical copy of itself
- Identified by "signature"
  - Characteristic bit pattern in virus code
  - Can detect family of viruses with similar replicator



Performance Issues

- Many files to scan, many signatures
- Optimizations?



  - Many viruses at beginning or end of a file
  - Almost all viruses are less than 4KB

More General Limitation

- Virus must be executed to be effective
  - Most viruses at an entry point or after non-branching code
- Antivirus programs check entry points
  1) Set E to program entry point
  2) scans instructions starting at location E
  3) Jump or call, set E to new location and go to 2

Reference: Nachenberg article

Virus Encryption

- Writer may encrypted main portion of virus
  - Decryption code
  - Encrypted Virus code
    - Does not need to be strong encryption
    - Just something to fool fast checker
- Encrypted code depends on key used
- Identify virus by decryption routine
  - Decryption routines are often unique
  - Most have at least 10-15 distinct bytes
  - Since small, increase probability of ident error

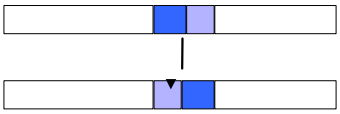
Virus Cleaning

- Virus detection
  - Determine whether there is a virus
- Virus identification
  - Determine the identity or family of virus
- Virus cleaning
  - Remove virus from file
  - Requires some knowledge of how virus works
    - How many bytes in replicator,
    - Identify beginning/end of payload,
    - ...

Identification errors make it harder to clean files

Polymorphic Viruses

- Change "shape" as they propagate
  - Specially designed mutation engines
    - can generate billions of mutation routines
    - mutation engine may be more complex than virus
  - Combine with encryption
    - change decryption routine by switching the order of instructions



Polymorphic Virus Detection

- Sandboxing
  - Run the file on a protected virtual computer
- Analyze virus body when decrypted
- Many performance problems
  - How long to run each program?
  - Solve the halting problem

Sophisticated viruses require sophisticated detection
Virus detection is an arms race

Intrusion detection

- Intrusion prevention
  - Network firewall
    - Restrict flow of packets; cover in another lecture
  - System security
    - Find buffer overflow vulnerabilities and remove them!
- Intrusion detection
  - Discover system modifications
    - Tripwire
  - Look for attack in progress
    - Network traffic patterns
    - System calls, other system events

Tripwire

- Steps in standard attack
  - Gain user access to system
  - Gain root access
  - Replace system binaries to set up backdoor
  - Use backdoor for future activities
- Tripwire detection point: system binaries
  - Compute hash of key system binaries
  - Compare current hash to hash stored earlier
  - Report problem if hash is different
  - Store reference hash codes on read-only medium

Is Tripwire too late?

- Typical attack on server
  - Gain access
  - Install backdoor
    - This can be in memory, not on disk!!
  - Use it
- Tripwire
  - Is a good idea
  - Wont catch attacks that don't change system files
  - Detects a compromise that has happened

Remember: Defense in depth

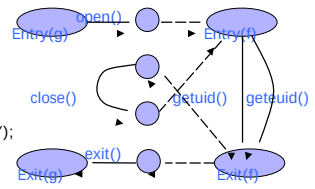
Detect modified binary in memory?

- Can use system-call monitoring techniques
- For example [Wagner, Dean IEEE S&P '01]
  - Build automaton of expected system calls
    - Can be done automatically from source code
  - Monitor system calls from each program
  - Catch violation

Results so far: lots better than not using code!

Example code and automaton

```
f(int x) {
  x ? getuid() : geteuid();
  x++
}
g() {
  fd = open("foo", O_RDONLY);
  f(0); close(fd); f(1);
  exit(0);
}
```



General intrusion detection

- Many intrusion detection systems
  - Close to 100 systems with current web pages
  - Network-based, host-based, or combination
- Two basic models
  - Misuse detection model
    - Maintain data on known attacks
    - Look for activity with corresponding signatures
  - Anomaly detection model
    - Try to figure out what is "normal"
    - Report anomalous behavior
- Continuing difficulty – too many false alarms

Misuse example - rootkit

- Rootkit sniffs network for passwords
  - Modifies netstat, ps, ls, du, ifconfig, login
    - Modified binaries hide new files used by rootkit
    - Modified login allows attacker to return for passwords
  - Fools simple Tripwire checksum
    - Modified binaries have same checksum
    - But better hash should detect rootkit
  - How else can we detect rootkit?
    - Rootkit does not alter the data structures normally used by netstat, ps, ls, du, ifconfig
    - Host-based intrusion detection can find rootkit files

Misuse example - port sweep

- Attacks can be OS specific
  - Bugs in specific implementations
  - Oversights in default configuration
- Attacker sweeps net to find vulnerabilities
  - Port sweep tries many ports on many IP addresses
  - If characteristic behavior detected, mount attack
    - SGI IRIX responds TCPMUX port (TCP port 1)
    - If machine responds, SGI IRIX vulnerabilities can be tested and used to break in
- Port sweep activity can be detected

Anomaly Detection

- Basic idea
  - Monitor network traffic, system calls
  - Compute statistical properties
  - Report errors if statistics outside established range
- Example – IDES (Denning, SRI)
  - For each user, store daily count of certain activities
    - E.g., Fraction of hours spent reading email
  - Maintain list of counts for several days
  - Report anomaly if count is outside weighted norm

[Hofmeyr, Somayaji, Forrest]

Anomaly – sys call sequences

- Build traces during normal run of program
  - Example program behavior (sys calls)

    open read write open mmap write fchmod close
  - Sample traces stored in file (4-call sequences)

    open read write open
    read write open mmap
    write open mmap write
    open mmap write fchmod
    mmap write fchmod close
  - Report anomaly if following sequence observed

    ~~open read read~~ open mmap write fchmod close

    Compute # of mismatches to get mismatch rate

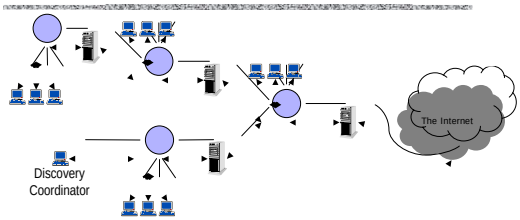
Difficulties in intrusion detection

- Lack of training data
  - Lots of "normal" network, system call data
  - Little data containing realistic attacks, anomalies
- Data drift
  - Statistical methods detect changes in behavior
  - Attacker can attack gradually and incrementally
- Main characteristics not well understood
  - By many measures, attack may be within bounds of "normal" range of activities
- False identifications are very costly
  - Sys Admin spend many hours examining evidence

Response to intrusion?

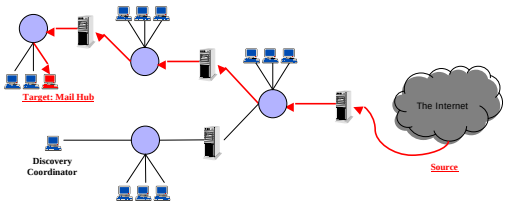
- Ideally,
  - Identify attack (possible if misuse, hard if anomaly)
  - Limit damage, stop attack, block further attacks
  - Restore system, identify and prosecute attacker
- Cliff Stoll
  - Detected attacker at Lawrence Berkeley
  - Created large file with nuclear weapon keywords
  - Traced international phone call during download

Example (UCD Computer Security Lab)



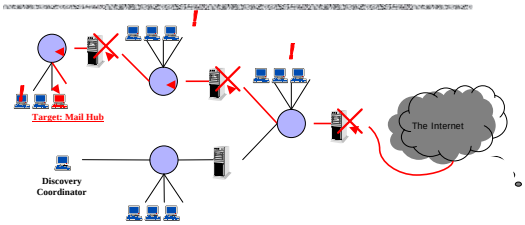
- Hosts grouped into neighborhoods
- Nbhd boundary protected by filtering router/firewall
- Centralized Discovery Coordinator directs global activity

SYN-flood Attack from Internet



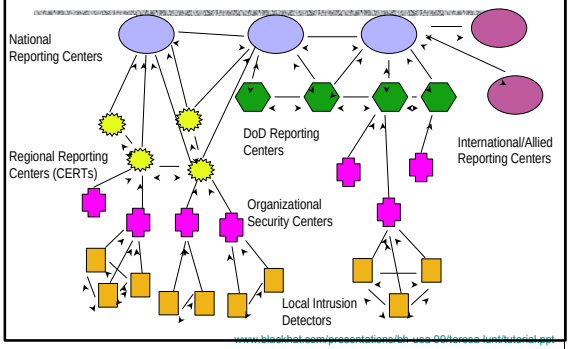
- Attack:SYN-flood to port 25 of central e-mail hub
- IP-header is forged: random, fake source addresses
- Result: E-mail effectively blocked by the attacker

Response from ID System



- IDS detects attack, reports to the DC
- DC correlates the sightings, selects response
- Result: Attack is prevented at the cost of blocking e-mail communications from arbitrary hosts

Strategic Intrusion Assessment [Lunt]



www.blackhat.com/presentations/bh-usa-99/teresa-lunt/tutorial.ppt

Strategic Intrusion Assessment [Lunt]

- Test over two-week period
  - AFIWC's intrusion detectors at 100 AFBs alarmed on 2 million sessions
  - After manual review, reduced to 12,000 suspicious events
  - After further manual review, these were reduced to four actual incidents
- Conclusion
  - Most alarms are false positives
  - Most true positives are trivial incidents
  - Of the significant incidents, most are isolated attacks to be dealt with locally

SNORT

- http://www.snort.org/