

Internet Security 2

(aka Advanced InetSec)

Malware 1

Christian Platzer	cplatzer@seclab.tuwien.ac.at
Paolo Milani Comparetti	pmilani@seclab.tuwien.ac.at
Clemens Kolbitsch	ck@seclab.tuwien.ac.at
Thorsten Holz	tho@seclab.tuwien.ac.at

News from the Lab

*Int. Secure Systems Lab
Technical University Vienna*

- Challenge 7 (reverse engineering & heap spraying) was solved by six people
 - congratulations since it was a hard challenge!
 - winner is GEORG “Ice Inferno” MERZDOVNIK
- Challenge 8 starts today (last challenge)
 - closely related to today's topic
 - implement a worm program

Malicious Code 1

Overview

*Int. Secure Systems Lab
Technical University Vienna*

- Introduction to malicious code
 - taxonomy, history, life cycle
- Virus
 - infection strategies, armored viruses, detection
- Worms
 - email- and exploit-based worms, spreading strategies
- Trojan horses and bots (in the next lecture)
 - keylogger, rootkits, botnets, and spyware
 - we show danger of malicious code on local host

Introduction

Int. Secure Systems Lab
Technical University Vienna

- Malicious Software (short: *Malware*)
 - software that fulfills malicious intent of author
 - term often incorrectly used equivalent with term *virus* (due to media coverage)
 - however, many different types exist!
 - classical viruses account for only a very small percentage of malware in the wild
- Virus - Definition

A virus is a program that reproduces its own code by attaching itself to other executable files in such a way that the virus code is executed when the infected executable file is executed

Taxonomy

Int. Secure Systems Lab
Technical University Vienna

Means of Distribution

Non-Spreading *Self-Spreading*

Computer Virus	Computer Worm
Trojan Horse Rootkit	Keylogger Spyware Dialers

Requires Host

Runs Independently

Dependency on Host

Taxonomy

*Int. Secure Systems Lab
Technical University Vienna*

- Virus
 - self-replicating, infects files (thus requires host)
- Worm
 - self-replicating, spreads over network
- Interaction-based worms (e.g., Bagle, Netsky, Sobig)
 - spread requires human interaction (typically via e-mail)
 - double-click and execute attachment
 - follow link to download executable
- Process-based worms (e.g., Code Red, Blaster, Slammer)
 - requires no human interaction
 - exploits vulnerability in network service

A Short History

Int. Secure Systems Lab
Technical University Vienna

- 1971 – *Creaper* is first self-replicating program on PDP-10
- 1981 – First outbreak of *Elk Cloner* on Apple II floppy disks
- 1983 – The first documented experimental virus
 - Fred Cohen's pioneering work; name *coined* by Len Adleman
- 1987 - File infectors
 - *Christmas worm* hit IBM Mainframes (500,000 replications / hour)
- 1988 - Internet worm (November 2, 1988)
 - Internet worm created by Robert Morris (CERT is created)
- 1995 – *Concept virus*, the first macro virus
- 1999 - *Melissa worm* (first large scale email virus)

A Short History

Int. Secure Systems Lab
Technical University Vienna

- 1999 – Distributed denial of service (DDoS) attacks
- 1999 – Kernel Rootkits become public
 - *Knark* (modification of system call table)
- 2000 – ILOVEYOU (large-scale e-mail worm)
- 2001 – Code Red (large-scale, exploit-based worm)
- 2003 – SQL Slammer worm (extremely fast propagation)
- 2004+ - Botnets (e.g., Storm Worm, Torpig, Koobface, Conficker, ...)

<http://www.viruslist.com/en/viruses/encyclopedia>

Reasons for Malware Prevalence

*Int. Secure Systems Lab
Technical University Vienna*

- Mixing of data and code
 - violates important design property of secure systems
 - unfortunately very frequent
- Homogeneous computing base
 - Windows is just a very tempting target with huge market share
- Unprecedented connectivity
 - easy to attack from safety of home
- Clueless user base
 - many targets available, social engineering successful
- Malicious code has become profitable
 - compromised computers can be sold (e.g., spam relay, DoS attack, steal sensitive information, ...)

Virus Lifecycle

*Int. Secure Systems Lab
Technical University Vienna*

- Lifecycle
 - reproduce, infect, run payload
- Reproduction phase
 - viruses balance infection versus detection possibility
 - variety of techniques may be used to hide viruses
- Infection phase
 - difficult to predict when infection will take place
 - many viruses stay resident in memory (TSR or process)
- Attack phase
 - e.g., deleting files, changing random data on disk
 - viruses often have bugs (poor coding) so damage can be done
 - Stoned virus expected 360K, floppy, corrupted sectors

Infection Strategies

*Int. Secure Systems Lab
Technical University Vienna*

- Boot viruses
 - master boot record (MBR) of hard disk (first sector on disk)
 - boot sector of partitions
 - e.g., Pakistani Brain virus, nowadays Mebroot/Torpig
 - rather old, but interest is growing again
 - diskless work stations, virtual machine virus (SubVirt)
- File infectors
 - simple overwrite virus (damages original program)
 - parasitic virus
 - append virus code and modify program entry point
 - cavity virus
 - inject code into unused regions of program code

Infection Strategies

*Int. Secure Systems Lab
Technical University Vienna*


- Entry Point Obfuscation
 - virus scanners quickly discovered to search around entry point
 - virus hijacks control later (after program is launched)
 - overwrite import table addresses
 - overwrite function call instructions
- Code Integration
 - merge virus code with program
 - requires disassembly of target
 - difficult task on x86 machines
 - W95/Zmist is a classic example for this technique

Macro Viruses

Int. Secure Systems Lab
Technical University Vienna

- Many modern applications support macro languages
 - Microsoft Word, Excel, Outlook, JavaScript in Acrobat Reader
 - macro language is powerful
 - embedded macros automatically executed on load
 - mail app. with Word as an editor
 - mail app. with Internet Explorer to render HTML

I made this program to all those people who want to write Word 2000 virii, but don't know what the hell to do.



The screenshot shows a configuration window for a macro virus. At the top, there are menu items: **•About•**, **Greets•**, **Contact•**, **Min•**, and **Exit•**. The main area contains several fields and options:

- Name of Author (Your name):** [Text input field]
- Name of Virus:** [Text input field]
- Special comments, shout outs:** [Text input field]
- Origin (The country you are in):** [Text input field]
- When would you like the infection to take place?**
 - On Open
 - On New
 - On Close
- When would you like the Message Box to be displayed?**
 - On Open
 - On New
 - On Close
- Where would you like the virus to be created?**
 - On Desktop
 - In Current Directory
- Click here to create your virus**
- Create•**

Companion Virus

*Int. Secure Systems Lab
Technical University Vienna*

- Companion virus
 - installs a COM file (the virus) for every EXE file found
 - idea is simple: DOS runs COM files before EXE
 - virus will stay memory resident and execute the original file
 - easy to find and eliminate

NTFS ADS Viruses

Int. Secure Systems Lab
Technical University Vienna

- NTFS contains a system called Alternate Data Streams (ADS)
 - sometimes used by malware
 - original intention of ADS is to store meta information with file
e.g., has it been downloaded from the Internet?

```
echo 'Hello World' > test.txt  
echo 'This is Hidden' > test.txt:hidden.txt  
notepad test.txt:hidden.txt
```

- Stream we have created is completely invisible
 - most commands do not work on ADSs (e.g., deleting).
 - Explorer and dir will not show the file
 - malware can make use of ADS to hide code, data, temporary files
 - tool called *streams.exe* from Sysinternals.com is useful for finding such streams

Fast and Slow Infectors

*Int. Secure Systems Lab
Technical University Vienna*

- A fast infector infects any file accessed
 - purpose of fast infection is to ride on the back of anti-virus software
 - infect files as they are being checked
 - can be defeated if the scanner is started from a floppy/USB stick
- A slow infector only infects files as they are created or modified
 - purpose of slow infection is to attempt to defeat integrity checking
 - piggyback on top of the process which legitimately changes a file
 - if integrity checker has a scanning component, virus can be caught

Virus Defense

*Int. Secure Systems Lab
Technical University Vienna*

- Antivirus Software
 - working horse is signature-based detection
 - database of byte-level or instruction-level signatures that match malware
 - wildcards can be used, regular expressions common
 - heuristics (check for signs of infection)
 - code execution starts in last section
 - incorrect header size in PE header
 - suspicious code section name
 - patched import address table
- Sandboxing
 - run untrusted applications in restricted environment
 - simplest variation, do not run as Administrator

Tunneling and Camouflage Viruses

*Int. Secure Systems Lab
Technical University Vienna*

- To minimize the probability of its being discovered, a virus could use a number of different techniques
- A tunneling virus attempts to bypass antivirus programs
 - idea is to follow the interrupt chain back down to basic operating system or BIOS interrupt handlers
 - install virus there
 - virus is “underneath” everything – including the checking program
- In the past, possible for a virus to spoof a scanner by camouflaging itself to look like something the scanner was programmed to ignore
 - false alarms of scanners make “ignore” rules necessary

Sparse Infectors and Armored Viruses

Int. Secure Systems Lab
Technical University Vienna

- Sparse infector
 - infect every n^{th} time a file is executed
 - infect files only with a certain name
- Armored virus
 - aims to make disassembly difficult
 - exploits fact that x86 code is hard to disassemble
 - Whale (early virus), made extensive use of such techniques
 - *Virtual machine packers*
 - manual disassembly is almost always possible, but takes more time and is not automated

Polymorphism and Metamorphism

*Int. Secure Systems Lab
Technical University Vienna*

- Polymorphic virus
 - change layout (shape) with each infection
 - payload is encrypted
 - using different key for each infection
 - makes static string analysis practically impossible
 - of course, encryption routine must be changed as well
 - otherwise, detection is trivial
- Metamorphic techniques
 - create different “versions” of code that look different but have the same semantics (i.e., do the same)

Chernobyl (CIH) Virus

*Int. Secure Systems Lab
Technical University Vienna*

```
5B 00 00 00 00    pop ebx
8D 4B 42          lea ecx, [ebx + 42h]
51              push ecx
50              push eax
50              push eax
0F 01 4C 24 FE    sidt [esp - 02h]
5B              pop ebx
83 C3 1C          add ebx, 1Ch
FA              cli
8B 2B           mov ebp, [ebx]
```

```
5B 00 00 00 00 8D 4B 42 51 50 50 0F 01 4C 24 FE 5B
83 C3 1C FA 8B 2B
```

Dead Code Insertion

*Int. Secure Systems Lab
Technical University Vienna*

```
5B 00 00 00 00    pop ebx
8D 4B 42          lea ecx, [ebx + 42h]
51              push ecx
50              push eax
90              nop
50              push eax
40              inc eax
0F 01 4C 24 FE    sidt [esp - 02h]
48              dec eax
5B              pop ebx
83 C3 1C         add ebx, 1Ch
FA              cli
8B 2B           mov ebp, [ebx]
```

```
5B 00 00 00 00 8D 4B 42 51 50 90 50 40 0F 01 4C 24
FE 48 5B 83 C3 1C FA 8B 2B
```

Instruction Reordering

*Int. Secure Systems Lab
Technical University Vienna*

5B 00 00 00 00	pop ebx
EB 09	jmp <S1>
S2:	
50	push eax
0F 01 4C 24 FE	sidt [esp - 02h]
5B	pop ebx
EB 07	jmp <S3>
S1:	
8D 4B 42	lea ecx, [ebx + 42h]
51	push ecx
50	push eax
EB F0	jmp <S2>
S3:	
83 C3 1C	add ebx, 1Ch
FA	cli
8B 2B	mov ebp, [ebx]

```
5B 00 00 00 00 00 EB 09 50 0F 01 4C 24 FE 5B EB 07 8D
4B 42 51 50 EB F0 83 C3 1C FA 8B 2B
```

Instruction Substitution

*Int. Secure Systems Lab
Technical University Vienna*

```
5B 00 00 00 00    pop ebx
8D 4B 42          lea ecx, [ebx + 42h]
51              push ecx
89 04 24          mov eax, [esp]
83 C4 04          add 04h, esp
50              push eax
0F 01 4C 24 FE    sidt [esp - 02h]
83 04 24 0C       add 1Ch, [esp]
5B              pop ebx
8B 2B           mov ebp, [ebx]
```

```
5B 00 00 00 00 8D 4B 42 51 89 04 24 83 C4 04 50 0F
01 4C 24 FE 83 04 24 0C 5B 8B 2B
```

Advanced Virus Defense

*Int. Secure Systems Lab
Technical University Vienna*

- Most malware techniques very effective against static analysis
- Thus, dynamic analysis techniques introduced
 - AV scanner equipped with emulation engine
 - executes actual instructions (no disassembly problems)
 - runs until polymorphic part unpacks actual malware
 - then, signature matching can be applied
 - emulation must be fast
 - ANUBIS (<http://anubis.iseclab.org>)
- Difficulties
 - malware can attempt to detect emulation engine
 - time execution, use exotic (unsupported) instructions, ...
 - insert useless instructions in the beginning of code to deceive scanner

Virus Naming

*Int. Secure Systems Lab
Technical University Vienna*

- Virus writers would like to name themselves
 - this is of course not possible 😊
 - Netsky --> Skynet discussion
- The first identifiers of a virus also get to name it
 - typically, this is a research in an antivirus company
 - however, people work in parallel...
- Each anti-virus company has its own notation
 - causes confusion and often multiple names
 - attempts to create a unique naming and identification notation

Number of Viruses

*Int. Secure Systems Lab
Technical University Vienna*

- More MS-DOS/Windows viruses than all other types
- Numbers are growing
 - 1991 - 600 to 1,000 viruses
 - 1996 - more than 10,000
 - 2000 - more than 50,000
 - 2005 - more than 125,000
- Only a small percentage is active in the wild
- Particular strong growth in recent years
 - problems with size of signature database
 - incremental updates are necessary
- No consensus on what a “new” virus is

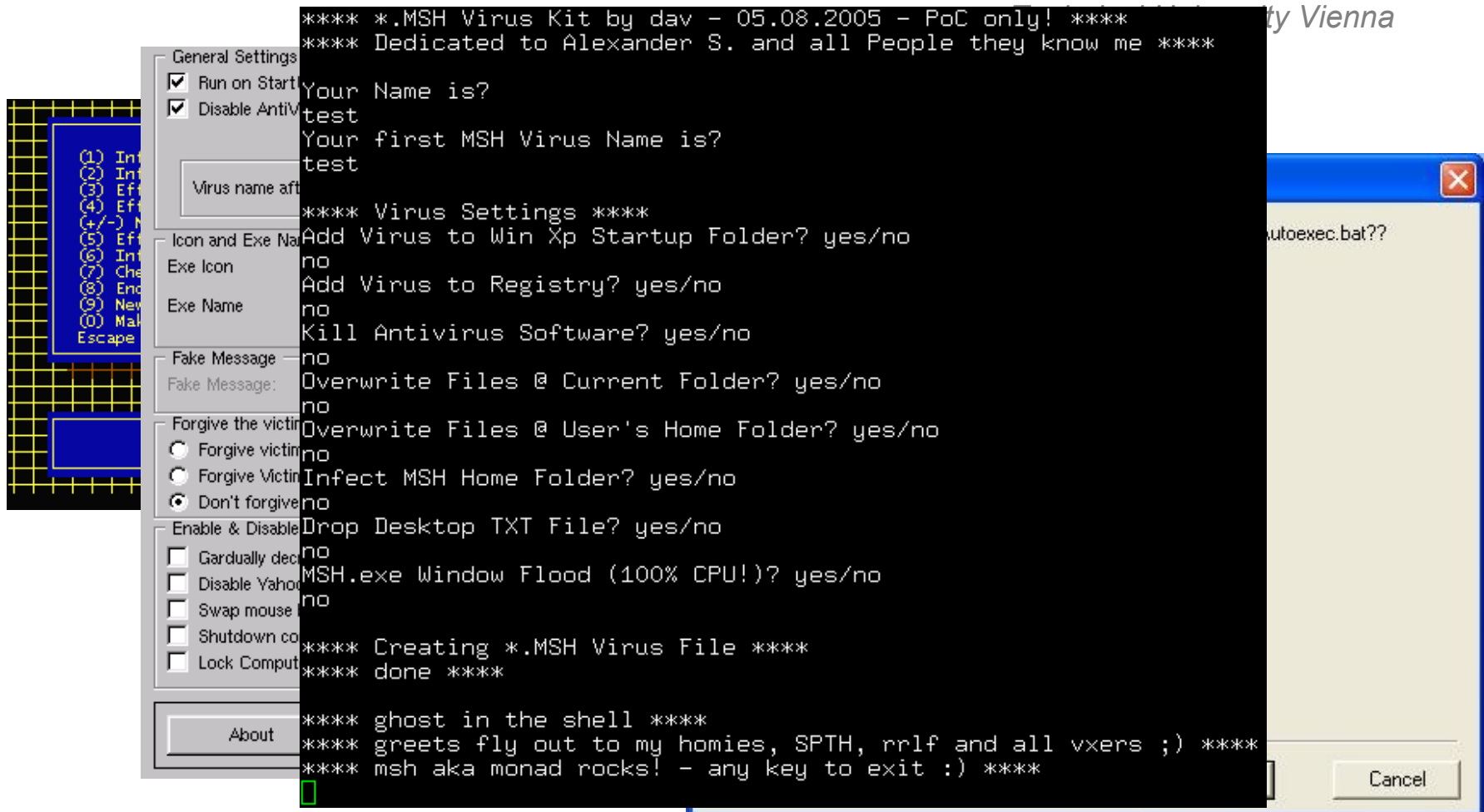
How Serious a Threat are Viruses?

*Int. Secure Systems Lab
Technical University Vienna*

- Important to keep virus threat in perspective
 - antivirus industry is also a business
 - only a small fraction found in the wild
 - according to some, the chance that a cup of coffee will do more damage to your computer is higher
 - probably not true, however ...
 - ... there are other threats besides classic file infecting viruses that are more problematic nowadays (worms, Trojan horses, bots)
 - one should also consider the existence of toolkits to generate viruses
 - makes entrance easy for script kiddies

Virus Generation Toolkits

Int. Secure Systems Lab
University of Vienna



The image shows a screenshot of a virus generation toolkit interface. On the left, there is a vertical menu with options: (1) Int, (2) Int, (3) Ef, (4) Ef, (4/-) N, (5) Ef, (6) Int, (7) Che, (8) Enc, (9) New, (0) Mal, and Escape. The main interface has several sections: 'General Settings' with checkboxes for 'Run on Start' and 'Disable AntiV'; 'Virus name aft'; 'Icon and Exe Na'; 'Exe Icon'; 'Exe Name'; 'Fake Message' with a 'Fake Message:' field; 'Forgive the victim' with radio buttons for 'Forgive victim', 'Forgive Victim', and 'Don't forgive'; 'Enable & Disable' with checkboxes for 'Gardually dec', 'Disable Yahoo', 'Swap mouse', 'Shutdown co', and 'Lock Comput'; and an 'About' button. A large black terminal window is overlaid on the interface, displaying the following text:

```
**** *MSH Virus Kit by dav - 05.08.2005 - PoC only! ****
**** Dedicated to Alexander S. and all People they know me ****

Your Name is?
test
Your first MSH Virus Name is?
test

**** Virus Settings ****
Add Virus to Win Xp Startup Folder? yes/no
no
Add Virus to Registry? yes/no
no
Kill Antivirus Software? yes/no
no
Overwrite Files @ Current Folder? yes/no
no
Overwrite Files @ User's Home Folder? yes/no
no
Infect MSH Home Folder? yes/no
no
Drop Desktop TXT File? yes/no
no
MSH.exe Window Flood (100% CPU!)? yes/no
no

**** Creating *.MSH Virus File ****
**** done ****

**** ghost in the shell ****
**** greets fly out to my homies, SPTH, rrlf and all vxers ;) ****
**** msh aka monad rocks! - any key to exit :) ****
```

On the right side, there is a small dialog box titled 'autoexec.bat??' with a 'Cancel' button.

Computer Worms

Int. Secure Systems Lab
Technical University Vienna

A self-replicating program able to propagate itself across networks, typically having a detrimental effect.

(Oxford English Dictionary)

- Worms either
 - exploit vulnerabilities that affect large number of hosts
 - send copies of worm body via email
- Difference to classic virus is *autonomous* spread over network
- Speed of spreading is constantly increasing
- Make use of techniques known by virus writers for long time

Worm Components

*Int. Secure Systems Lab
Technical University Vienna*

- Target locator
 - how to choose new victims
- Infection propagator
 - how to obtain control of victim
 - how to transfer worm body to target system
- Life cycle manager
 - control different activities depending on certain circumstances
 - often time depending
- Payload
 - nowadays, often a Trojan horse (we talk about that in next lecture)

Target Locator

*Int. Secure Systems Lab
Technical University Vienna*

- Email harvesting
 - consult address books (W32/Melissa)
 - files might contain email addresses
 - inbox of email client (W32/Mydoom)
 - Internet Explorer cache and personal directories (W32/Sircam)
 - even Google searches are possible
- Network share enumeration
 - Windows discovers local computers, which can be attacked
 - some worms attack everything, including network printers
prints random garbage (W32/Bugbear)

Target Locator

*Int. Secure Systems Lab
Technical University Vienna*

- Scanning
 - randomly generate IP addresses and send probes
 - interestingly, many random number generators flawed
 - static seed
 - not complete coverage of address space
 - scanning that favors local addresses (topological scanning)
 - some worms use hit-list with known targets (shorten initial phase)
- Service discovery and OS fingerprinting performed as well

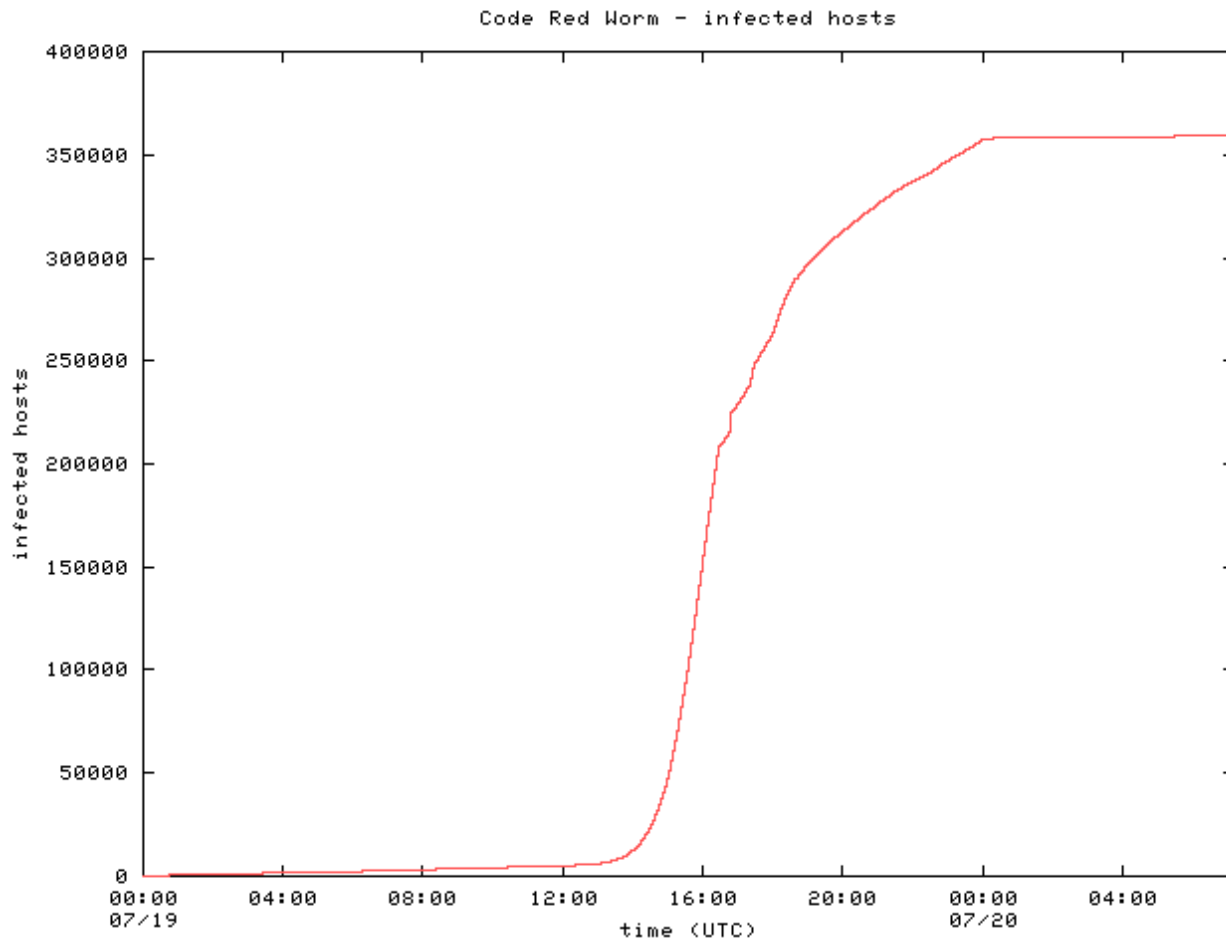
Exploit-Based Worms

*Int. Secure Systems Lab
Technical University Vienna*

- Require no human interaction
 - typically exploit well-known network services
 - can spread much faster
- Propagation speed limited either
 - by network latency
worm thread has to establish TCP connection (Code Red)
 - by bandwidth
worm can send (UDP) packets as fast as possible (Slammer)
- Spread can be modeled using classic disease model
 - worm starts slow (only few machines infected)
 - enters phase of exponential growth
 - final phase where only few uncompromised machines left

Exploit-Based Worms

*Int. Secure Systems Lab
Technical University Vienna*



Conclusion

*Int. Secure Systems Lab
Technical University Vienna*

- We started looking at malware today
 - Virus
 - Worms
- Next time we continue looking at malware and especially botnets
- Challenge 8 deals with worms
 - starts today
- Next lecture on Jan. 11th, 2010
 - Enjoy the vacation, merry Christmas!