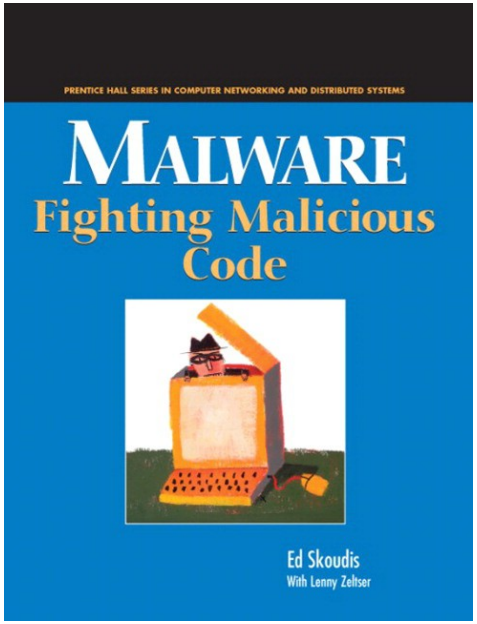
The golden age of hacking

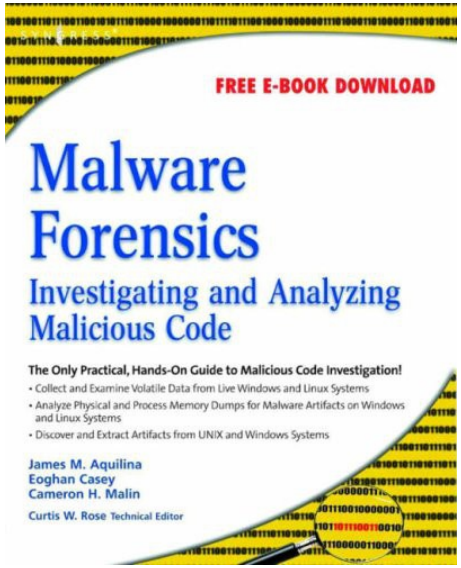
Malicious Code
Virus, malware, worms

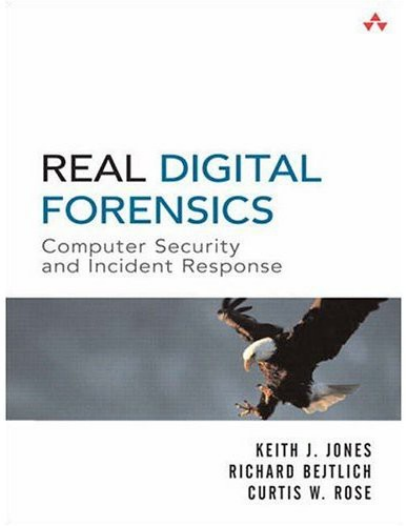
http://en.wikipedia.org/wiki/Portal:Computer_security

Books

- Malware Fighting …
  - Chapter 2. Viruses
  - Chapter 3. Worms
  - Chapter 11. Malware Analysis
    - Malware lab, static and dynamic analysis etc.
- Malware Forensics
  - http://www.malwareforensics.com
- Real Digital Forensics
  - Chapters 13, 14 and 15







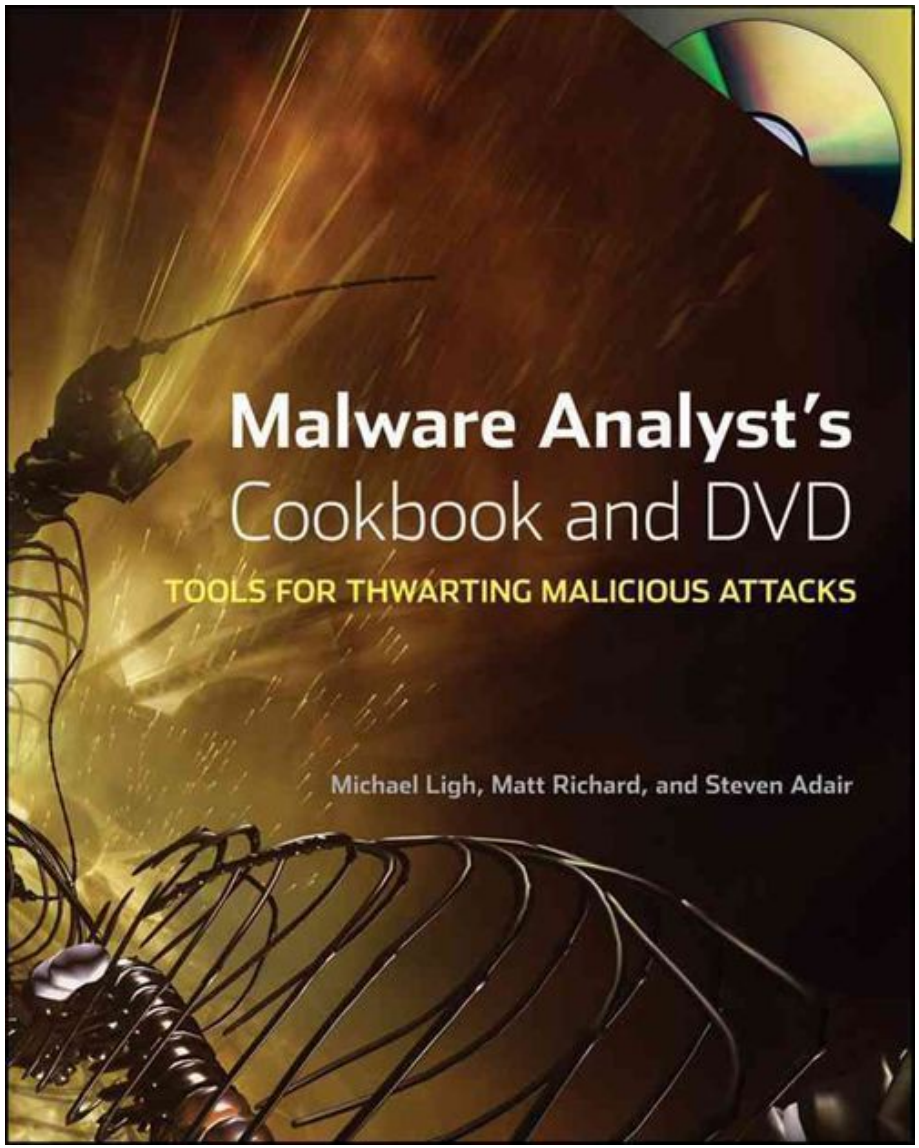
Books

- The best!
- DVD on digitalbrott share

\malware\malwarecookbook.com

- Password - infected
- ISBN:
  - 0470613033
  - 978-0470613030

http://www.malwarecookbook.com/

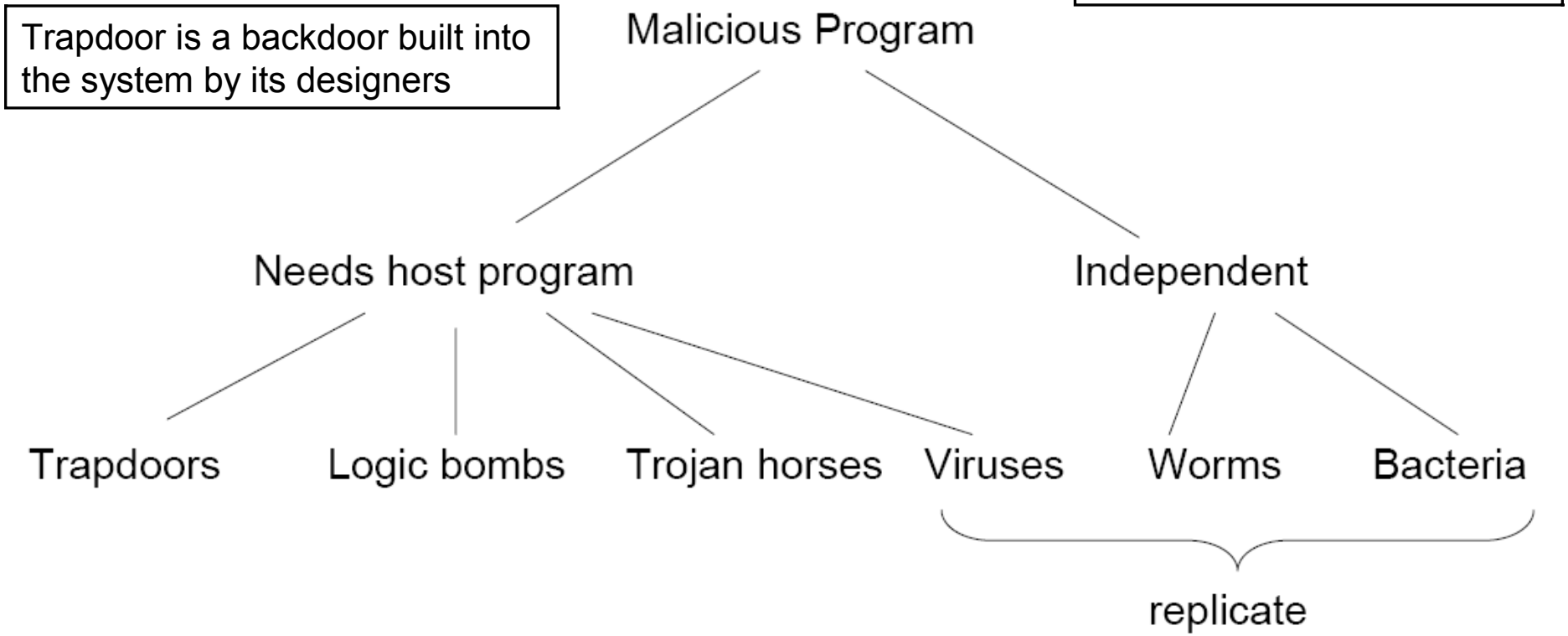


Malicious Code

- Malware terminology
  - Google is your friend…

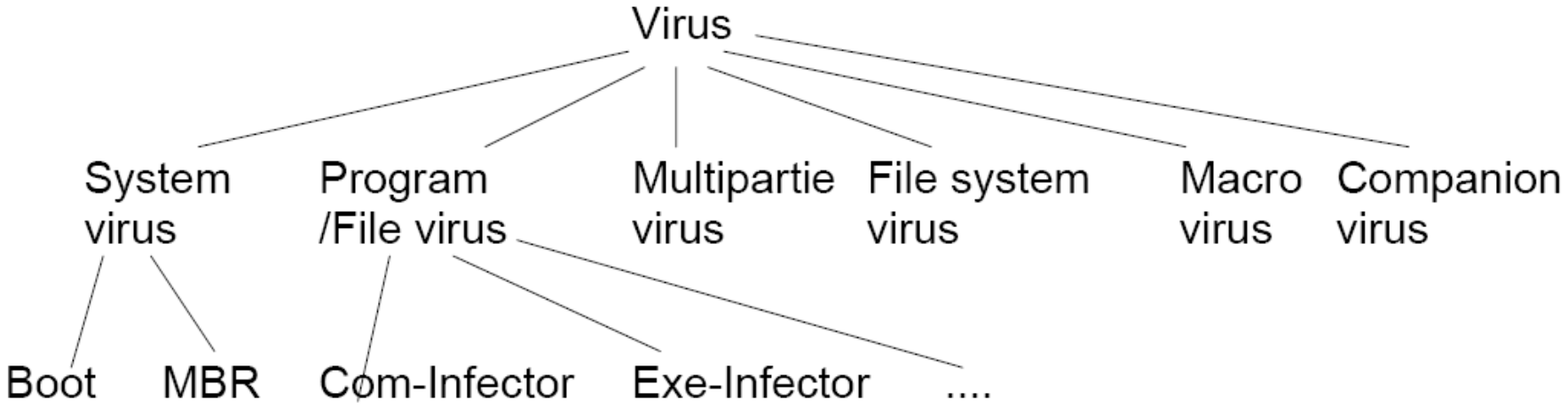
Bacteria are a type of malware that create many instances of themselves, or run many times simultaneously, in order to consume large amounts of system resources

Trapdoor is a backdoor built into the system by its designers



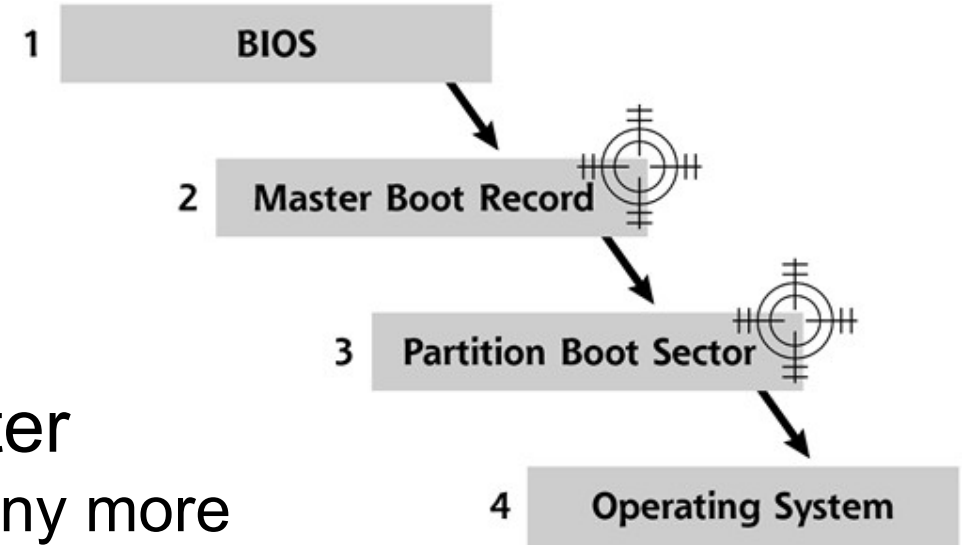
Virus

- Early history
  - 196? - 1980
- Non-autonomous segment of code or a macro that will copy itself into "host program" when it is activated
- Two major components
  - Infection routine
  - Payload task



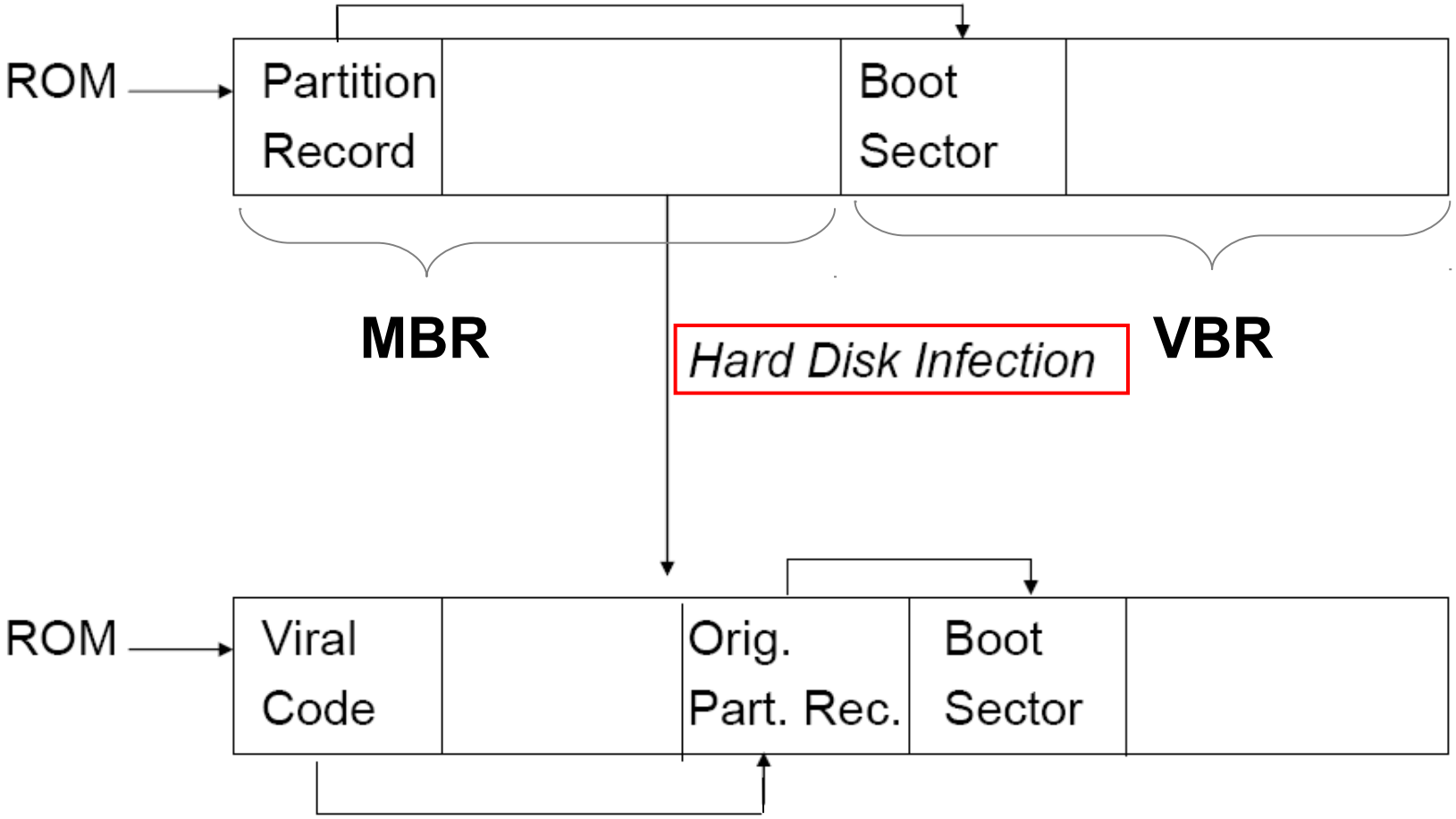
System Virus I

- Boot Sequence
  - Example: CIH aka Chernobyl or space filler
    - Erase MBR, reprogram BIOS, stored in PE files section-gap
    - http://en.wikipedia.org/wiki/CIH_virus
  - ROM BIOS routines
  - Master Boot Record (MBR) code
  - Boot Sector code
  - Historically
    - IO.sys and MSDOS.sys
    - Config.sys
    - Command.com
    - Autoexec.bat
- Windows NT and greater
  - Usually not a problem any more



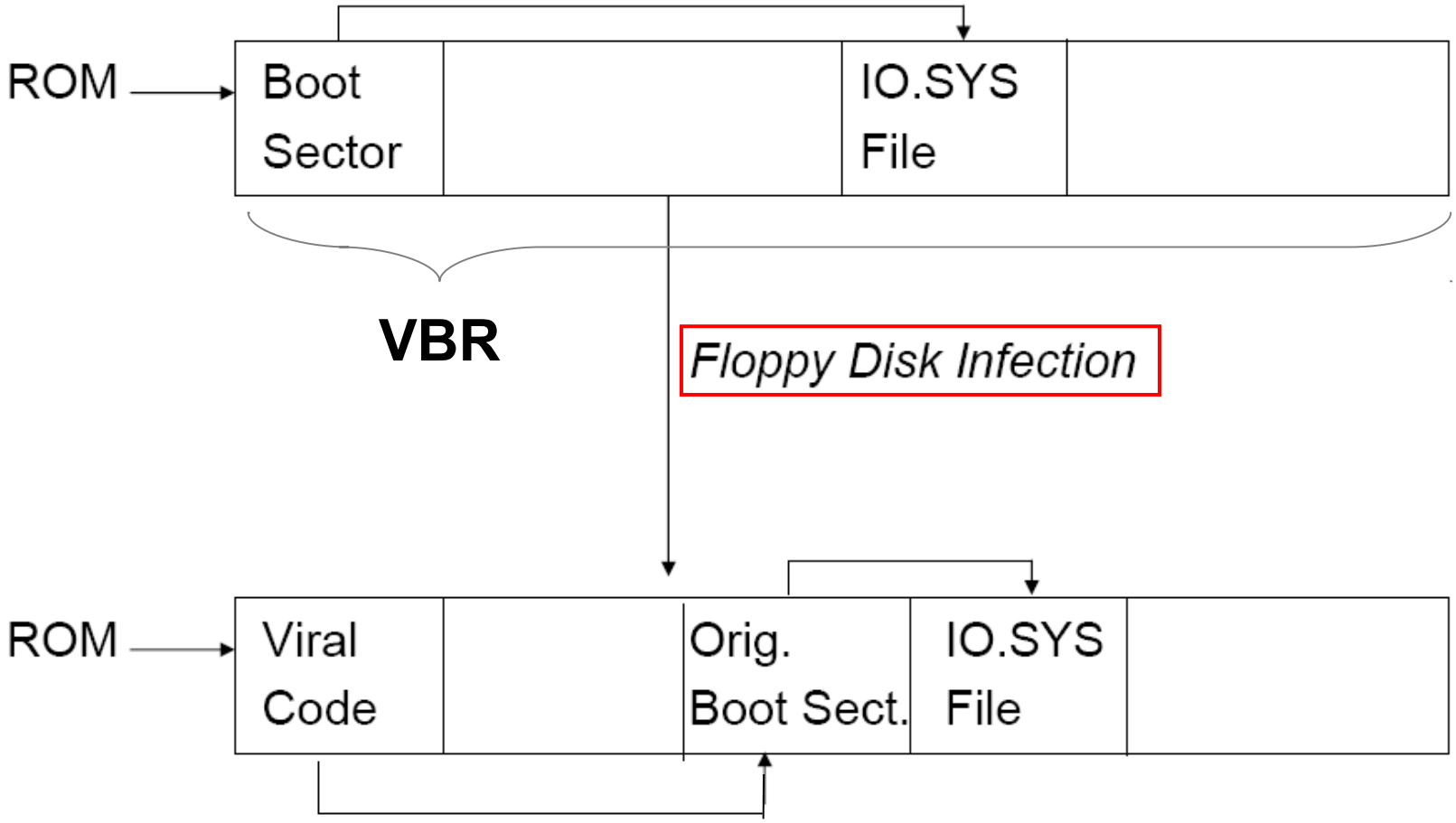
System Virus II

- MBR Infection

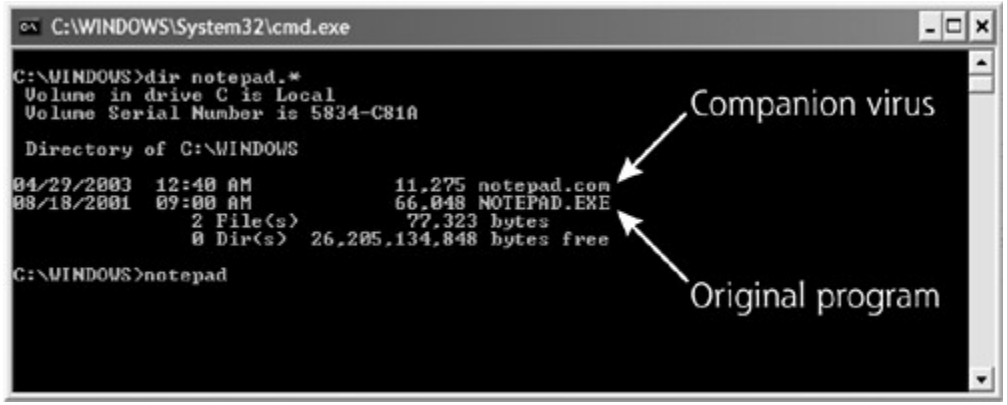


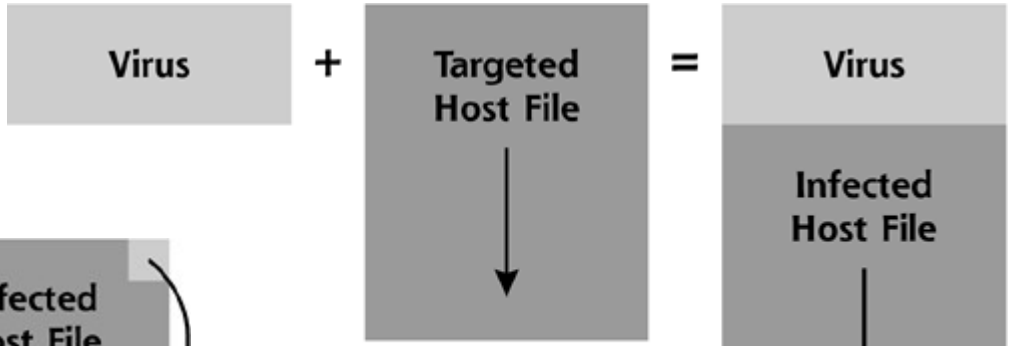
System Virus III

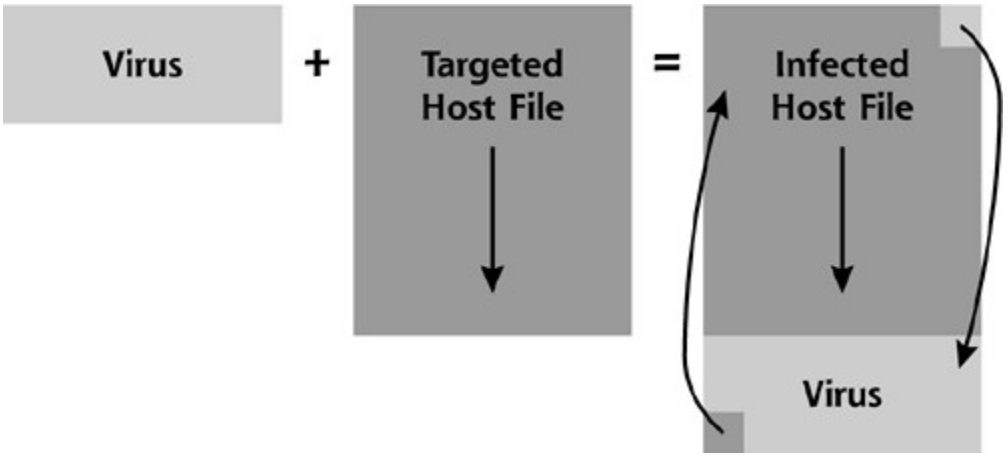
- Boot Sector Infection



Companion, prepending and appending virus







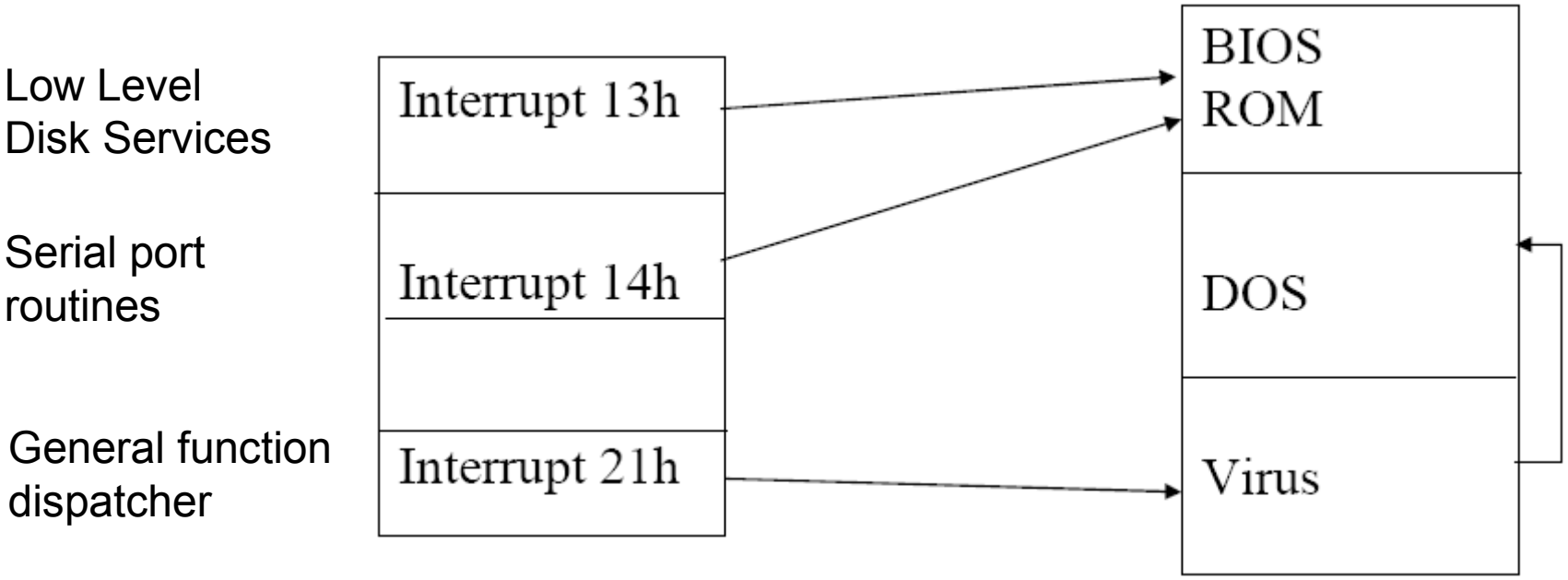
Transient virus

- Runs when its host runs - select a target to be infected and then transfer control to host
  - Example of a simple transient virus in pseudo code

```
class Viri {
    signature = 0x4D5A009000FA…;
    infect-executable() {
        loop:
        file = get-random-executable-file;
        if (first-line-of-file = signature) then goto loop // target already infected
        else append or prepend Viri to file;
    }
    do-damage() {
        whatever damage is to be done
    }
    trigger-pulled() {
        return true if some condition holds
    }
    main() {
        infect-executable();
        if (trigger-pulled()) then do-damage();
    }
}
```

TSR (Terminate and Stay Resident) Virus

- Locates itself in memory and remains active
  - Is activated even after host ends
- Interrupt vector with TSR virus - virus alters standard interrupts
  - Virus code is invoked when other applications make service requests
  - http://en.wikipedia.org/wiki/Interrupt_vector
  - http://en.wikipedia.org/wiki/BIOS_interrupt_call
- Windows NT and greater does not rely on BIOS for low level access



Shamoon (worm) - 2012-08

http://en.wikipedia.org/wiki/Shamoon

- Shamoon, also known as Disttrack, is a modular computer virus that attacks computers running the Microsoft Windows "NT" line of operating systems.
- The virus is being used for cyber espionage in the energy sector and has been noted as unique for having differing behaviour from other malware cyber espionage attacks.
- Shamoon is capable of spreading to other computers on the network, through exploitation of shared hard drives. Once a system is infected, the virus continues to compile a list of files from specific locations on the system, erase and then send information about these files back to the attacker. Finally, the virus will overwrite the master boot record of the system to prevent it from booting.
- The virus has hit companies within the oil and energy sectors. A group named "Cutting Sword of Justice" claimed responsibility for an attack on 30,000 Saudi Aramco workstations, causing the company to spend a week restoring their services.

https://kc.mcafee.com/corporate/index?page=content&id=PD23936&cat=CORP_MCAFEE_LABS&actp=LIST&showDraft=false

Methods to avoid detection

- Keep MAC(E) metadata intact
- Keep file size intact
  - Fill/overwrite empty space in binary
- Kill AV agent
- Avoid bait files
  - Sparse infection
- Stealth
  - ADS files
  - Intercept AV agent and OS (hooking etc.)
  - Polymorphic
    - Self-mutating/modification
    - Encryption, for example XOR-ing the code
    - Combinations of encryption and self-modification
  - Metamorphic
    - Rewrite themselves completely or change function each time they are invoked to infect new executables
- http://en.wikipedia.org/wiki/Computer_virus

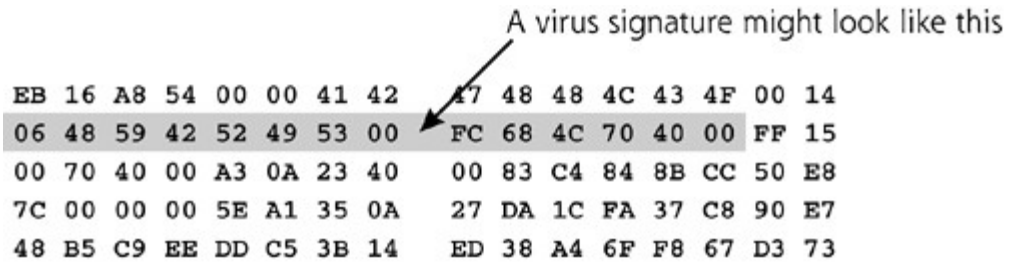
1. put in nop
2. put in garbage code

```
; Polymorfing 1
...
mov   ecx, 0x100
nop
xor   byte[eax], ebx
nop
inc   eax
...
```

```
; Polymorfing 2
...
mov   ecx, 0x100
mov   eax, eax
push  edx
xor   byte[eax], ebx
pop   edx
inc   eax
...
```

Malware flying in under the radar

- Foiling the AV agent signature check
- There are two basic ways of editing a program
  - Either change the original source code and/or just recompile it
  - Using binary editors to change various bytes in the compiled binary
- Tools needed
  - Hex editor
  - Debugger as OllyDbg etc.



- Find the addresses where the signature check is
  - With hex editor fill half program with zeros - scan with AV - repeat
  - After a while you know where the signature(s) is/are
- Analyze the signature part with debugger and make adjustments in assembly code if possible with NOP etc.
  - Check if malware works as it should and AV don't detect it
  - http://packetstormsecurity.org/papers/virus/ Taking_Back_Netcat.pdf

Macro virus

- Set of macro commands, specific to an application, which automatically execute in an unsolicited manner and spread to that application's documents
- Properties and risks
  - Platform independent
  - Spread easily by electronic mail attachments
- Common technique for spreading macro viruses
  - Auto-macro is attached to a word document that was sent by email
  - When documents is opened, macro executes and copies itself to global macro file
- Example: Melissa (1999)
  - Shut down internet when it emailed 50 entries in address book
  - http://en.wikipedia.org/wiki/Melissa_(computer_virus)

Virus defense I

- Signature Scanning
  - Recognizes a virus unique “signature”
- Limitations
  - Totally depended on maintaining up-to-date signature files
  - False positives
  - Detects only “known” viruses
- Heuristic Scanning
  - Uses heuristic rules to search for probable virus infections, similar to how a SPAM filter works
    - Attempts to access the boot sector
    - Attempts to locate all documents in a current directory
    - Attempts to write to an EXE file
    - Attempts to delete hard drive contents
- Limitations
  - Higher rate of false positive and false negatives

Virus defense II

- Activity Monitors
  - Monitor program resident in memory (fingerprint)
  - Raise warning, take special actions in event of suspicious activity
    - For example registry entries it creates
- Limitations
  - False positives
  - Software solution is vulnerable to virus alterations
  - Insecure if virus activates earlier in the boot sequence than monitor code
- Integrity checkers
  - Generate check codes for monitored files
  - Check codes are periodically recalculated and compared against the saved versions
- Limitations
  - Requires maintenance of a virus-free checksum database
  - Initial calculation must be performed on a known unaltered version of a file
  - May have trouble to detect stealth viruses, “slow infectors”
  - Cannot detect viruses by type or name

Virus defense III

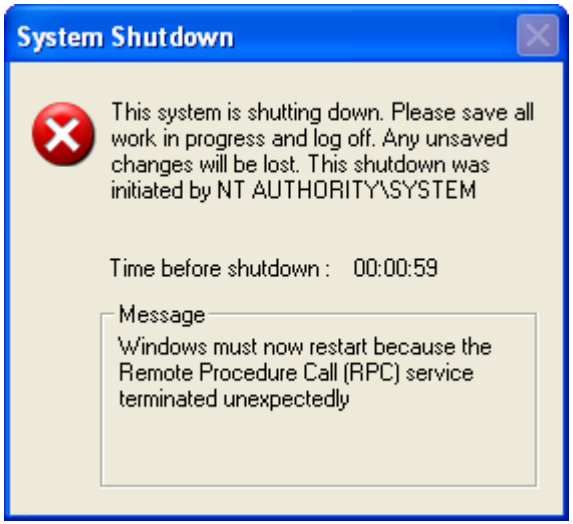
- Configuration Hardening
  - Obey the principle of least privilege
  - Minimizing the number of active system components
  - And so on…
- User Education
  - Do not attempt to disable defensive mechanisms
  - Be cautious of attachments that are unusual
  - Do not download and install programs from external sources
  - Do not connect your own systems to the organization's network
  - Learn to recognize signs of a virus infection
  - Do not forward virus warnings to your friends and colleagues as soon as you receive them (Hoax)

Computer Worm

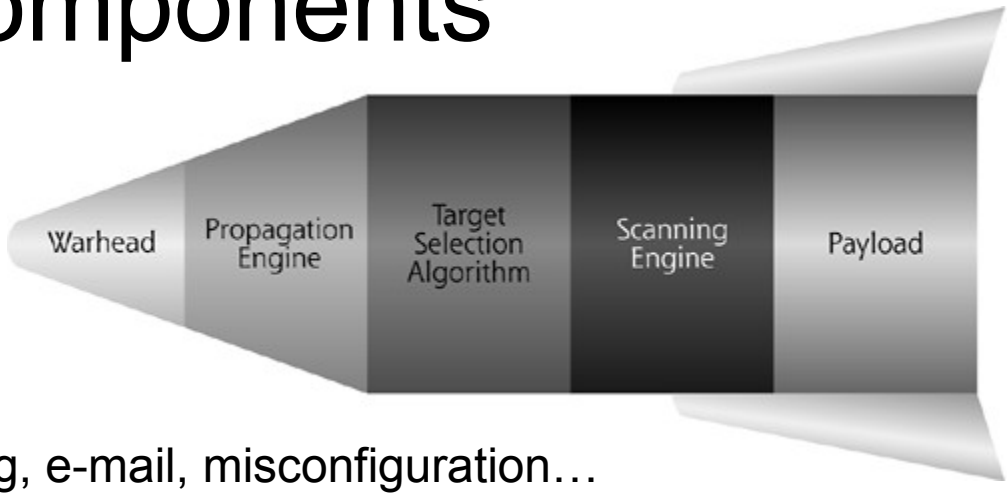
http://en.wikipedia.org/wiki/Computer_worm
http://en.wikipedia.org/wiki/Timeline_of_notable_computer_viruses_and_worms

- Program that can run independently and travel from machine to machine across network connections
- Famous viruses and worms
  - http://www.pbs.org/newshour/science/computer_worms/famous.html
- Examples
  - 1988: Morris Worm aka Internet worm
    - Exploited known Unix flaws:
      - Password guessing / dictionary attacks
      - Bug in finger program (buffer overflow)
      - Trapdoor in sendmail / debug
  - 2001: Code Red worm
    - IIS Web server
    - http://en.wikipedia.org/wiki/Code_Red_worm
  - 2001: Nimda
    - IIS Web server, Windows, Outlook, file sharing…
  - 2003: SQL Slammer worm
    - MS SQL
  - 2003: Blaster worm aka Lovesan or Lovsan
    - Buffer overflow in Windows XP RPC
    - http://en.wikipedia.org/wiki/Blaster_worm

XP SP2 released -04



Worm components



- Warhead
  - Exploit some vulnerability
    - Buffer overflow, file sharing, e-mail, misconfiguration…
- Propagation Engine
  - Transfer itself with protocols as FTP, TFTP, HTTP, SMB etc.
- Target Selection Algorithm
  - Random IP scan or some intelligent technique finding vulnerable hosts, files with host/client info
- Scanning Engine
  - Find a new target for the warhead
- Payload
  - Backdoor, DDoS agent, password cracking, …, anything!

Nimda worm I

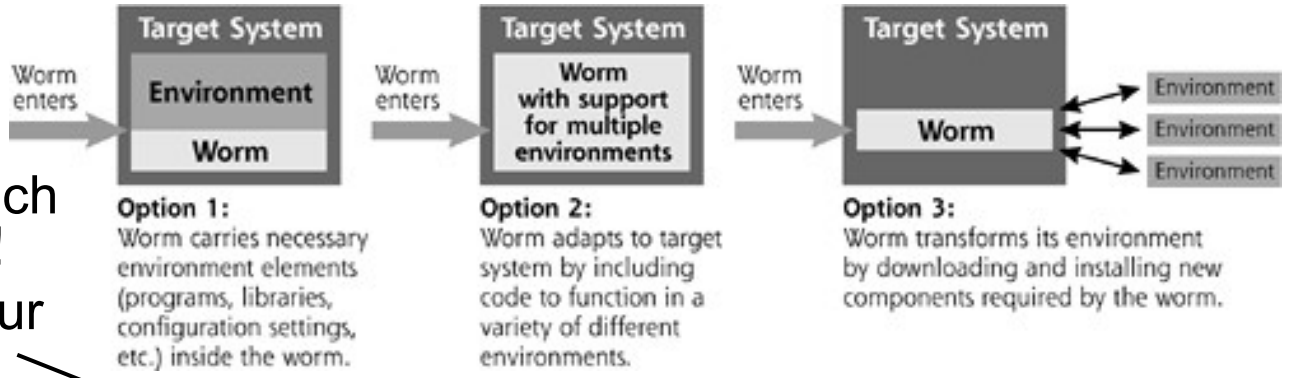
- Warhead and exploits
  - Flaws in Microsoft's IIS Web Server
    - Directory traversal flaws let an attacker run arbitrary code
  - Browsers that surfed to an infected web server gave the Nimda client new carriers
  - Outlook E-Mail clients
    - If a user read or even previewed an e-mail message infected with the Nimda code, the worm would install itself on the machine
  - Windows File Sharing
    - Nimda looked for Web content on the local system and any accessible network file shares trying to infect them and exe files with virus tech
  - Backdoors from previous worms
    - If found Nimda took advantage of it and eradicating the earlier worm
- Propagation engine was bundled tightly with its warhead
  - The worm propagated from Web sites using HTTP, from e-mail clients using various Outlook e-mail protocols, and from Windows file shares using the SMB protocol
  - Additionally, when scanning for Web servers with directory traversal vulnerabilities, the worm copied itself using TFTP

Nimda worm II

- Nimda's target selection algorithm operated in two modes
  - First, it focused on e-mail addresses
    - If Microsoft's Outlook e-mail program was installed, the worm searched the user's contact lists to harvest e-mail addresses
    - It also scanned the hard drive for any e-mail addresses referred to inside of HTM and HTML files. Nimda would then e-mail a copy of itself to various acquaintances of the user, spreading its code further. To disguise itself from users and evade e-mail filters, the worm morphed the subject line and length of the e-mail message.
  - Second, the Nimda target selection algorithm would generate a list of target IP addresses to scan for directory traversal vulnerabilities and the presence of the Code Red II and Sadmind IIS backdoors.
- Nimda's payload cracked the system wide open for further attacks and possibly even backdoor access
  - The worm enabled file sharing on infected systems by allowing unfettered access of the C:\ primary hard drive partition by activating the Guest account, and then adding the Guest account to the Administrators group on the victim machine
- Nimda is probably one the most determined worms we've witnessed to date
  - http://en.wikipedia.org/wiki/Nimda

Advanced worm components

- Methods a worm uses to adapt to an unsuitable environment
  - Crashing hosts limits spread!
  - Spreading to much may kill network!
  - Don't step on your own toes!
- Coming superworms
  - Multiplatform Worms
  - Multiexploit Worms
  - Zero-Day Exploit Worms
  - Fast-Spreading Worms
  - Polymorphic Worms
  - Metamorphic Worms
  - Truly Nasty Worms

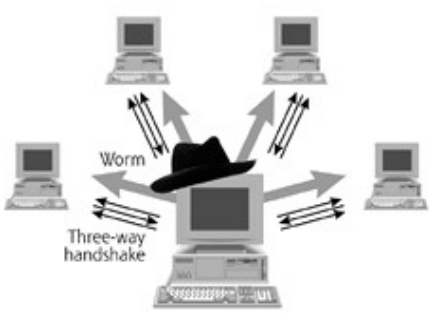


In a way as the Spanish flu H1N1 virus
Almost to deadly (25% peak) to be effective

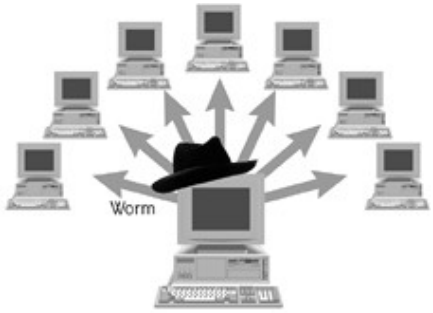
Stuxnet: 2.5 - 15 man-years of development!

Bigger Isn't Always Better!

- 2003: SQL Slammer worm (MS SQL)
  - Buffer overflow, 376 bytes, fitted in one packet
  - UDP, fire and forget
  - Warhol worm
    - Infect 99% of vulnerable systems in less than 15 min!
- Nuclear plant control system in Ohio USA
  - The cooling system control were shut down for 5 hours
- South Korea had to shut down Internet services
  - Routers collapsed
  - Routers still alive sent routing table updates
  - Router restart casued another flood of updates
  - The only one getting thru was slammer
  - http://en.wikipedia.org/wiki/SQL_slammer_worm

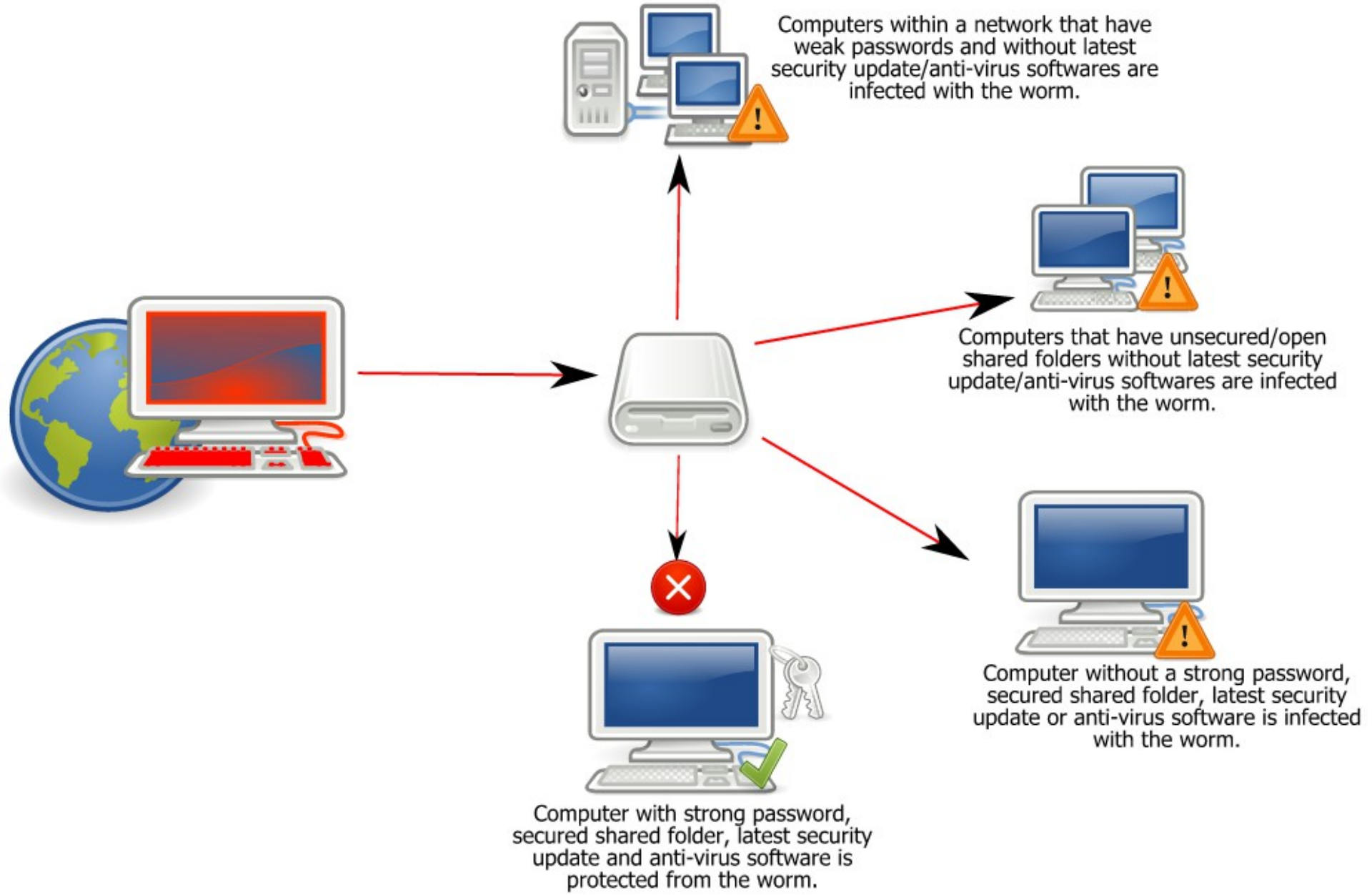


Worm spread via TCP



Worm spread via UDP

Worm:Win32 Conficker

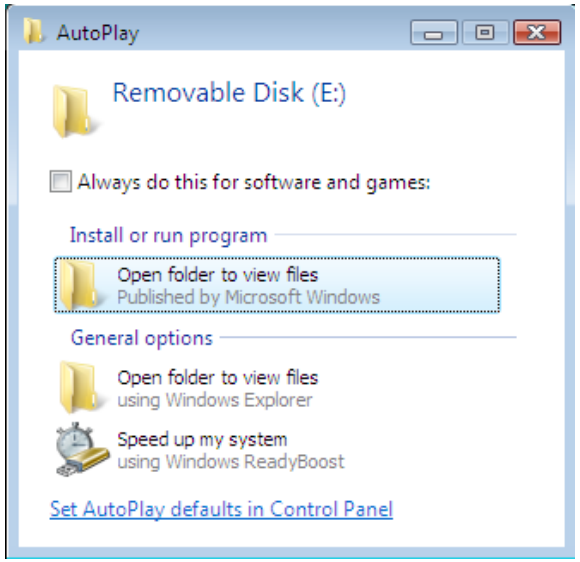


Conficker, also known as Downup, Downadup and Kido

- Conficker is supplied as a DLL
- MS Patch MS08-067 (2008-10-23)
  - The vulnerability is a rpc call to the API NetpwPathCanonicalize (), which is found in Netapi32.dll over a smb connection on port 445
  - A variant of buffer overflow - "aaa\bbb\..\ccc" becomes "aaa\ccc"
- Shell code injects Conficker via the Windows Server Service and then installs itself as an auto-started services via svchost.exe to survive reboots
  - Other possible installations is through USB drives or shared folders on networks whereby the dll started via rundll32.exe
- The worm then creates a set of domain names through a time-based random number function and then try to connect to those domains
  - In 2009-03-15 security experts discovered that some nodes downloaded the encrypted binaries from these generated domain names

Conficker, cont.

http://en.wikipedia.org/wiki/Conficker



- Conficker uses RSA-signatures to validate the downloads and repels them if the signature does not match
- Conficker is clogging the hole (MS08-067) to counter attacks from other malware programs
  - By hooking the vulnerable function NetpwPatchCanonicalize () Conficker then accepts RPC
- Conficker A/B/C/D is a masterpiece of elaborate coding that was deliberately over complicated and has so far not been possible to interpret in detail
- Over 10 million computers are estimated to have been infected (worst attack so far?)
- Knocked out French Air Force and much of the British armed forces (battleships and submarines, etc..) And thousands of Region Skåne computers
- Bi effects on patched computers - DoS on user accounts (login reset)
- How Conficker works (very good article)
  - http://sakerhet.idg.se/2.1070/1.221789
- Microsoft's price on the creator's head: $250000

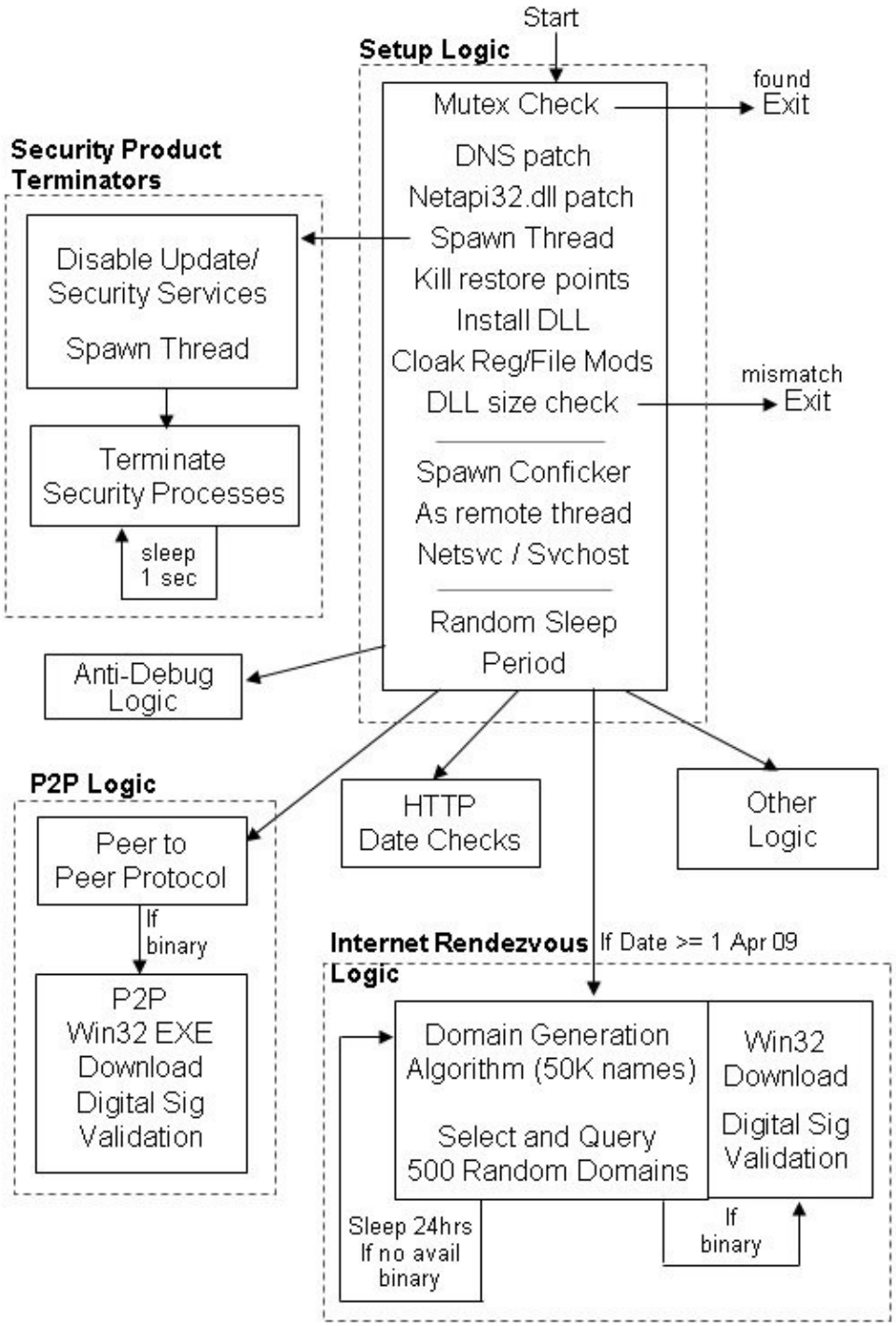
Conficker, cont.

- Conficker.C program logic is given to the right
- Source:

http://ethicalhackernet.blogspot.com/2009/04/silence-of-storm-worm-welcome-rolling.html

Is still one of the 5 largest bot-net

http://www.idg.se/2.1085/1.342233/har-ar-de-fem-storsta-botnaten



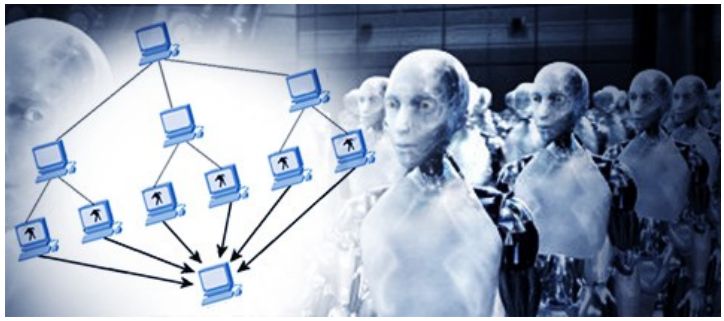
Stuxnet - 2010-06

http://en.wikipedia.org/wiki/Stuxnet

- Targeted industry worm mainly against Iran and other countries in the region towards east
- Stuxnet use at least three 0-day exploits and two stolen valid certificates from Taiwan (JMicron and Realtek)
- Spreading via flash drives, through network shares, through a RPC vulnerability and through a recently patched MS10-061 Print Spooler vulnerability
- Targeting specific types of industrial control equipment as Siemens Programmable Logic Controllers (the first discovered rootkit for PLCs)
- It cleans up very well after itself when moving and must have been developed with enormous resources according to specialists
- Stuxnet analysis

  [server]\malware\worms

http://www.eset.com/resources/white-papers/Stuxnet_Under_the_Microscope.pdf



Worm defense

- Ethical worms aka white worms
  - Distributing patches
- Deploy vendor patches asap
  - Secunia Personal Software Inspector (PSI)
- Hardening configuration settings on all publicly accessible systems
- Block outgoing initialized connections from servers
- Establish incident response capabilities
- Everything from virus defense