PennState

CSE543
Introduction to Computer and Network Security
Module: Malware

Asst. Prof. Syed Rafiul Hussain

Malware

- Adversaries aim to get code running on your computer that performs tasks of their choosing
  - This code is often called malware
- Two main challenges for adversaries
  - How do they get trick you into getting their malware onto your computer?
  - How do they get their malware to run?
- Other practical concerns of malware distribution
  - Spread malware to as many systems as possible
  - Hide malware execution
  - Make malware difficult to remove

Viruses

- Is an attack that modifies programs on your host
- Approach
  1. Download a program …
  2. Run the program …
  3. Searches for binaries and other code (firmware, boot sector) that it can modify …
  4. Modifies these programs by adding code that the program will run

- What can an adversary do with this ability?

Viruses

- How does it work?
  - Modify the file executable format

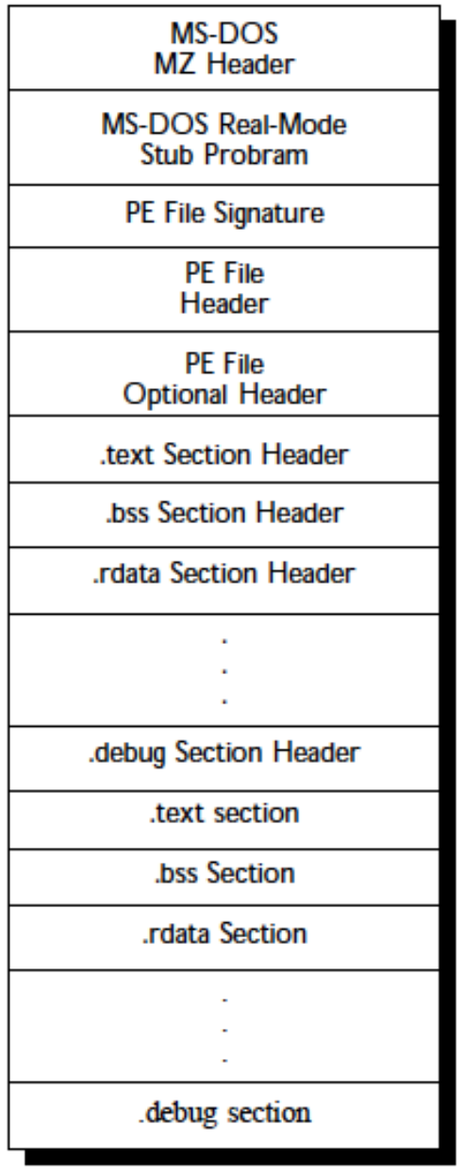


Figure 1. Overall structure of a Portable Executable file image.

Viruses

- How does it work?
  - Modify the file executable format
- What types of modifications?
  - Overwrite the "entry point"
  - Add code anywhere and change "address of entry point"
    - Add a new section header
    - Patch into a section
  - Add jump instruction to exploit
- All these were well known by 90s

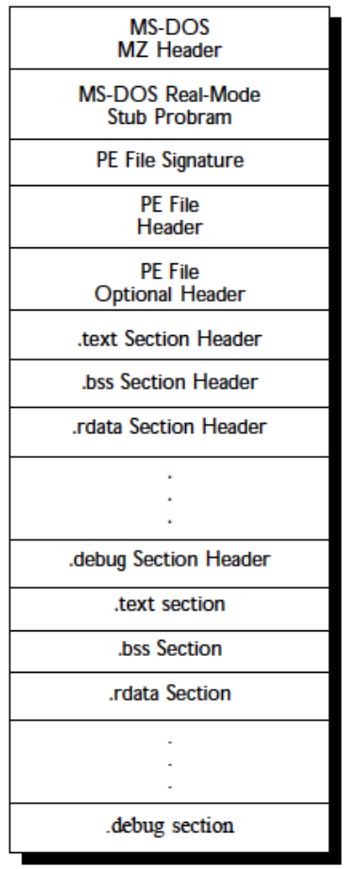


Figure 1. Overall structure of a Portable Executable file image.

Virus Infection

- Keeping with the virus analogy, getting a virus to run on a computer system is called infecting the system
  - Program that attaches itself to another (usually trusted, aka. benign program)

Virus Infection

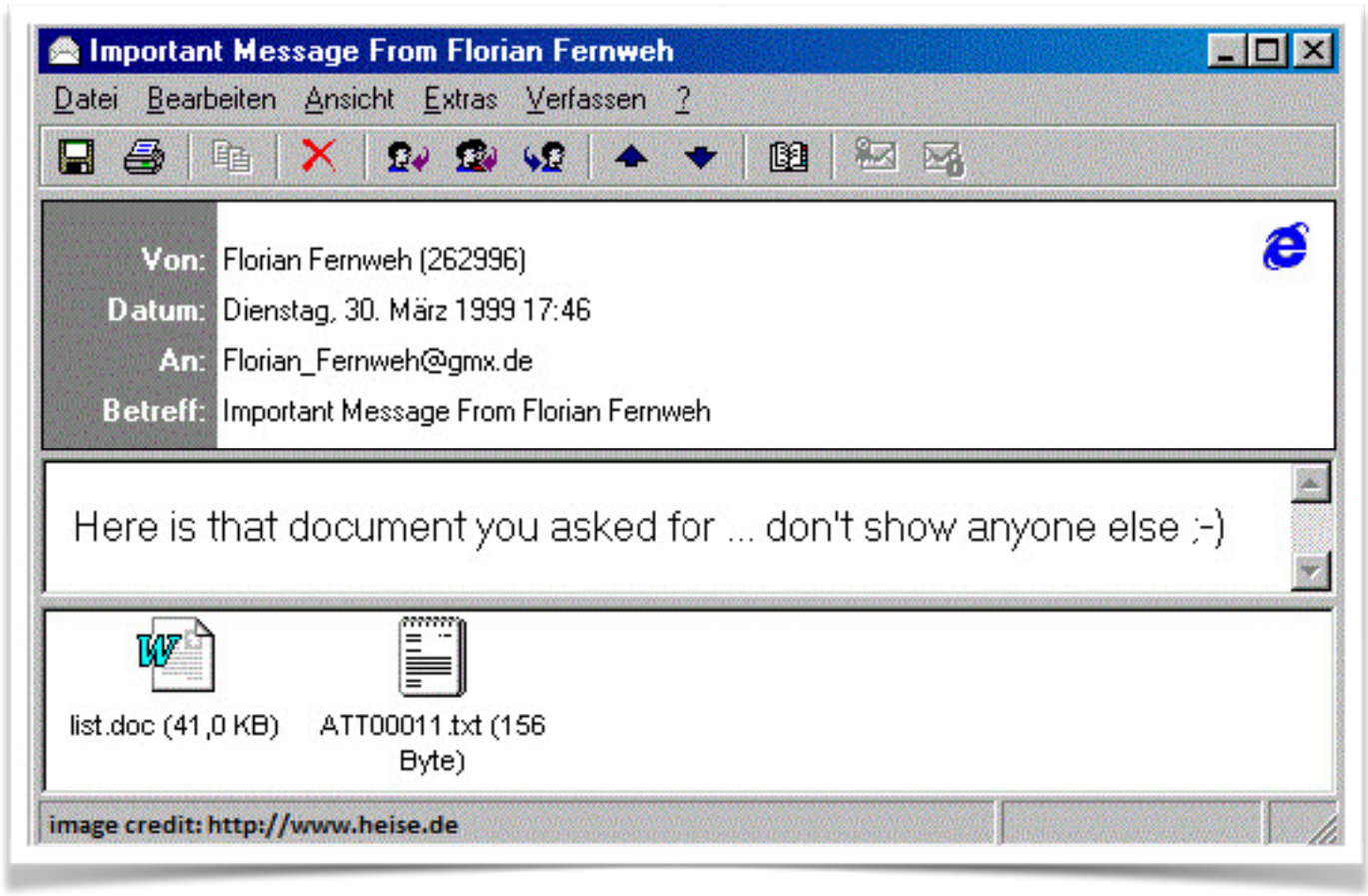
- Keeping with the virus analogy, getting a virus to run on a computer system is called infecting the system
  - Program that attaches itself to another (usually trusted program)
  - How can an adversary infect another's computer?
    - Tricking users into downloading their malware
      - Need to also trick the user into running the malware
    - Exploiting a vulnerable program to inject code
      - By exploiting a running process, the malware can run directly

An Easier Way

- Don't really need to modify existing executable to download and run code on a remote computer
  - Since the mid-90s systems have provided methods for you to get a remote system to run your code
  - First, email attachments, then client-side scripts
    - Enabled by phishing attacks (more later)
- In general, the idea is to get the user to run your code (in email or via web link)
  - Either run directly
  - Or exploit a vulnerability in the platform (e.g., browser)

Melissa Virus (1999)

- Came through email including an MS Word attachment
- Emailed itself to the first 50 people in the Outlook's contact list
- Infected ~20% of computers, $1.2B in damages.

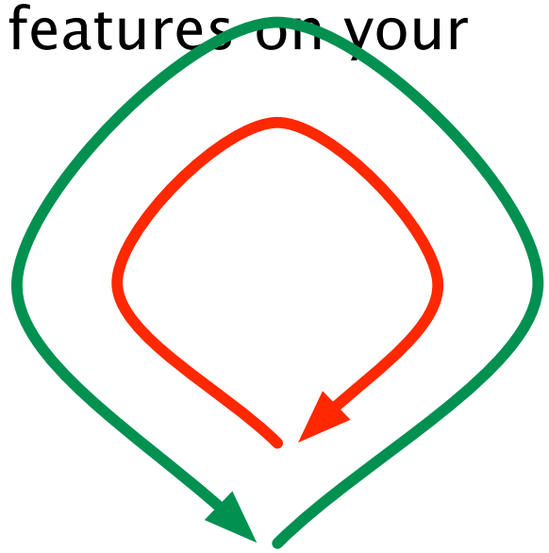


Worms

- A worm is a self-propagating program.
- As relevant to this discussion

1. Exploits some vulnerability on a target host (e.g., buffer overflow)…
2. (often) embeds itself into a host …
3. Searches for other vulnerable hosts without human interventions…
   - A worm takes advantage of file or information transport features on your system, which allows it to travel unaided.
4. Goto (1)

- Sometimes used to create botnets



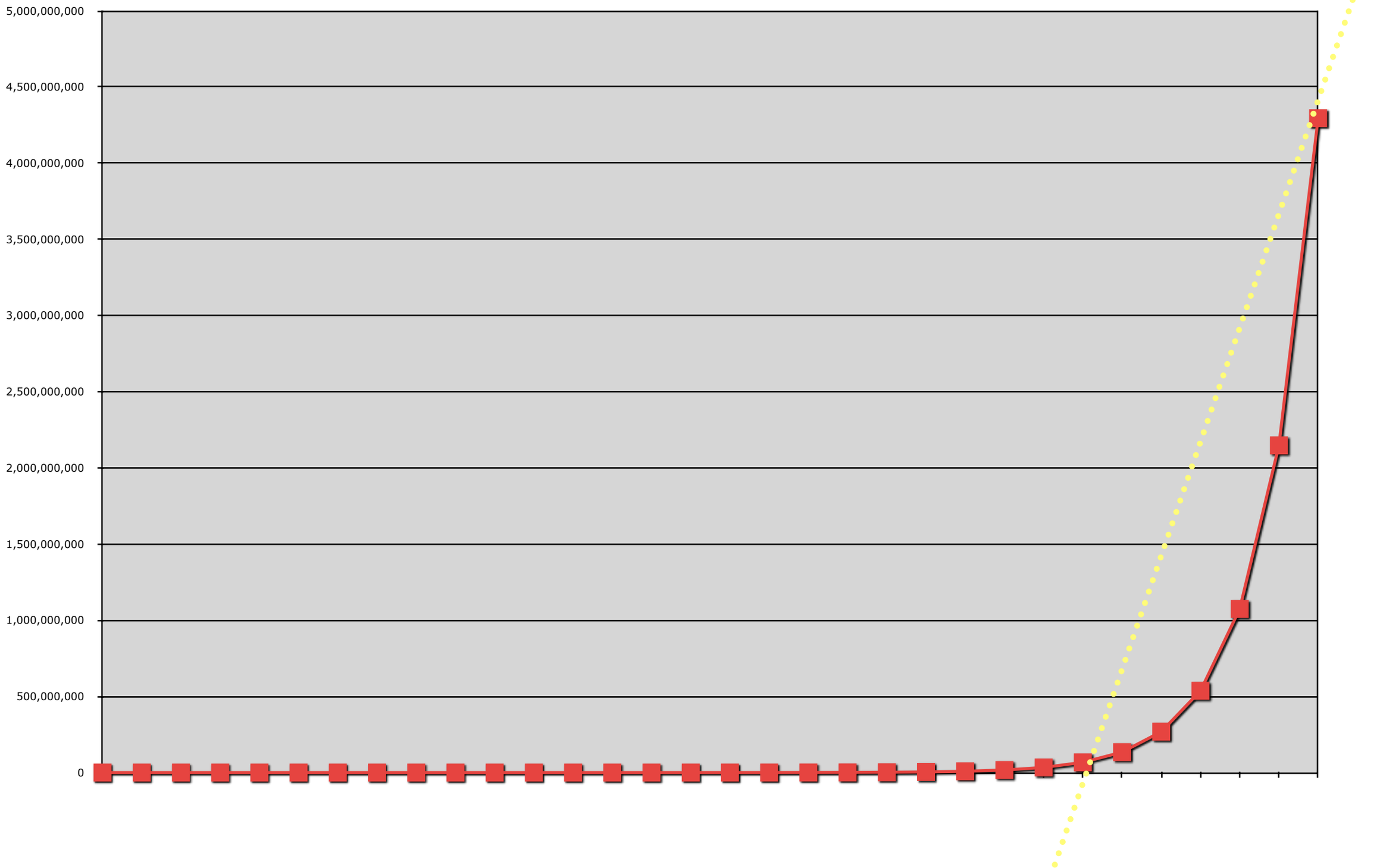
The Danger

- What makes worms so dangerous is that infection grows at an exponential rate
  - A simple model:
    - *s* (search) is the time it takes to find vulnerable host
    - *i* (infect) is the time is take to infect a host
  - Assume that t=0 is the *worm outbreak*, the number of hosts infected at t=j is

$$2^{(j/(s+i))}$$

  - For example, if (s+i = 1), what is it at time j=32?

The result


5,000,000,000
4,500,000,000
4,000,000,000
3,500,000,000
3,000,000,000
2,500,000,000
2,000,000,000
1,500,000,000
1,000,000,000
500,000,000
0

The Morris Worm (1988)

PennState

- Robert Morris, a 23 doctoral student from Cornell
  - Wrote a small (99 line) program
  - Launched on November 3rd, 1988
  - Simply disabled the Internet
- How it did it
  - Exploited a buffer overflow in the "finger" daemon
  - Used local /etc/hosts.equiv, .rhosts, .forward to identify hosts that can be accessed without passwords
  - Reads /etc/password to perform password cracking
  - Scanned local interfaces for network information
  - Covered its tracks (set is own process name to sh, prevented accurate cores, re-forked itself)
- Morris claimed the worm was intended to gauge the size of the internet but accidentally replicated itself.

Code Red

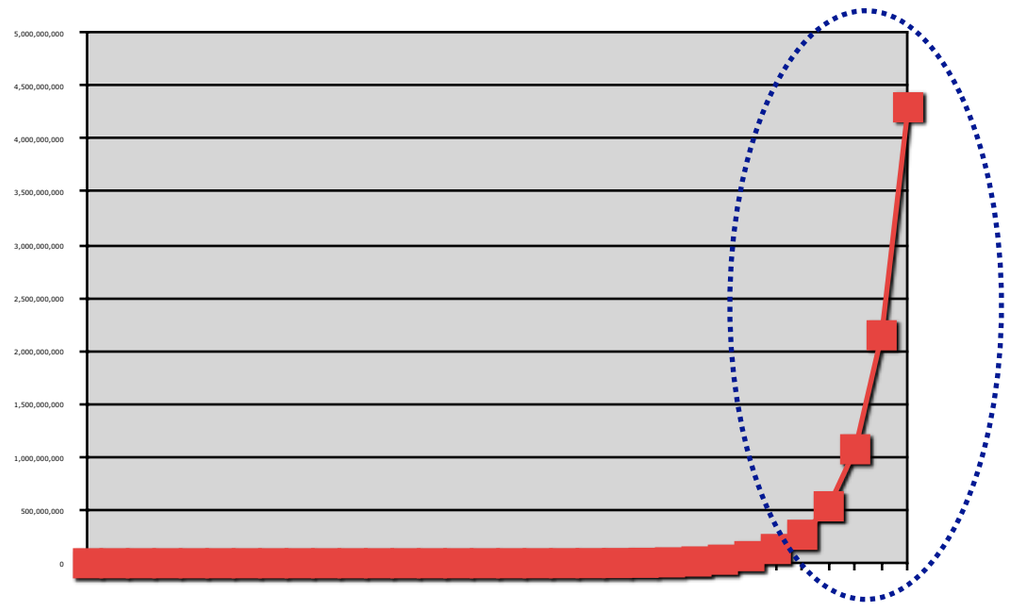
- Exploited a Microsoft IIS web-server vulnerability
  - A vanilla buffer overflow (allows adversary to run code)
  - Scans for vulnerabilities over random IP addresses
  - Sometimes would deface the served website
- July 16th, 2001 - outbreak
  - CRv1- contained bad randomness (fixed IPs searched)
  - CRv2 - fixed the randomness,
    - added DDOS of www.whitehouse.gov
    - Turned itself off and on (spread 1st-19th of month, attack 20-27th, dormant 28-31st)
  - August 4 - Code Red II
    - Different code base, same exploit
    - Added local scanning (biased randomness to local IPs)
    - Killed itself in October of 2001

Worms and infection

- The effectiveness of a worm is determined by how good it is at identifying vulnerable machines
  - Morris used local information at the host
  - Code Red used what?
- Multi-vector worms use lots of ways to infect
  - E.g., network, email, drive by downloads, etc.
  - Others' backdoors… - another worm, Nimda did this
- Lots of scanning strategies
  - Signpost scanning (using local information, e.g., Morris)
  - Random IP - good, but waste a lot of time scanning "dark" or unreachable addresses (e.g., Code Red)
  - Permutation scanning - instance is given part of IP space
- What is the fastest way to infect as many machines as possible?

Other scanning strategies

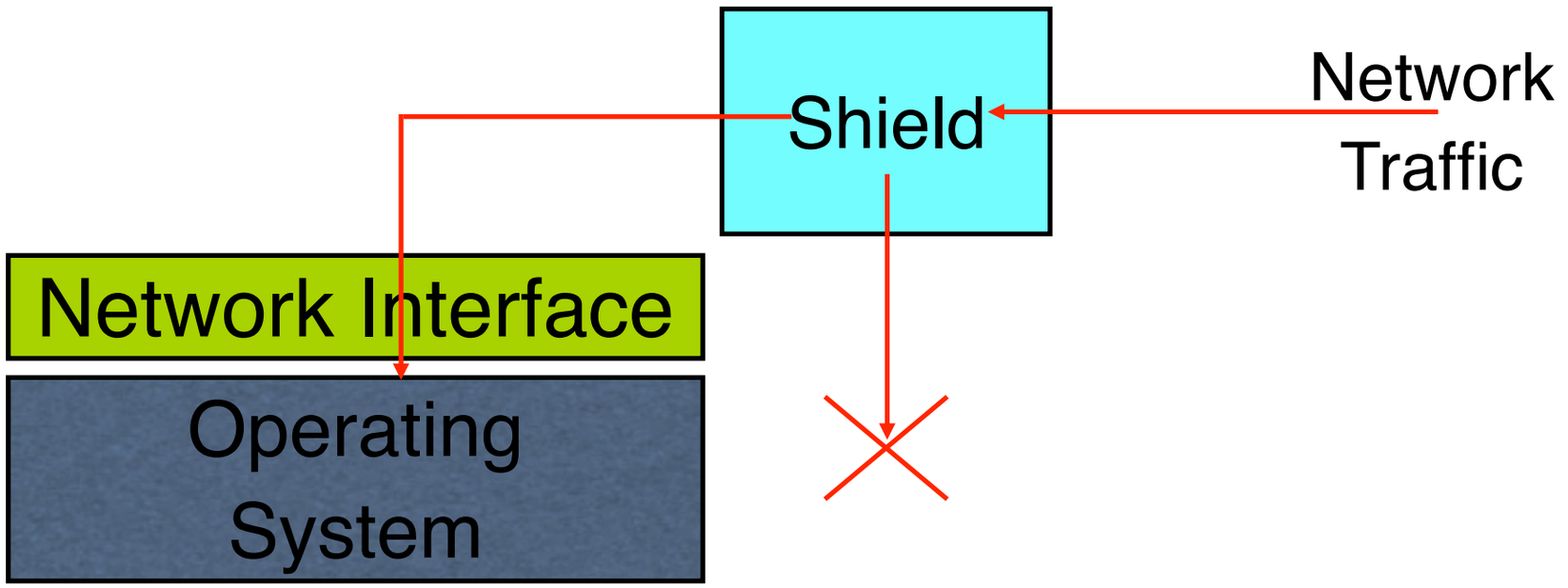
- The doomsday worm: a flash worm
  - Create a hit list of all vulnerable hosts
    - Staniford et al. argue this is feasible
    - Would contain a 48MB list
  - Do the infect and split approach
  - Use a zero-day vulnerability



- Result: saturate the Internet in less than *30 seconds*!

Worms: Defense Strategies

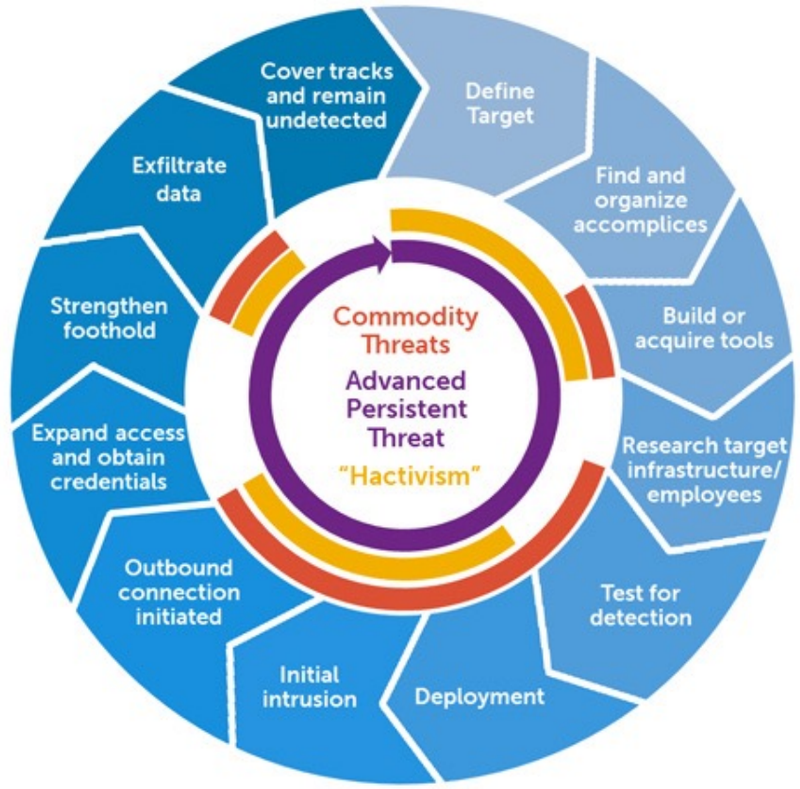
- (Network) Packet Filtering: look for unnecessary or unusual communication patterns, then drop them on the floor
  - This is the dominant method, sophisticated
- (Network) Heterogeneity: use more than one vendor for your networks



- (Host) Patch Your Systems (auto): most, if not all, large worm outbreaks have exploited known vulnerabilities (with patches)
- Network and Host Intrusion Detection Systems (more later)

Modern Malware

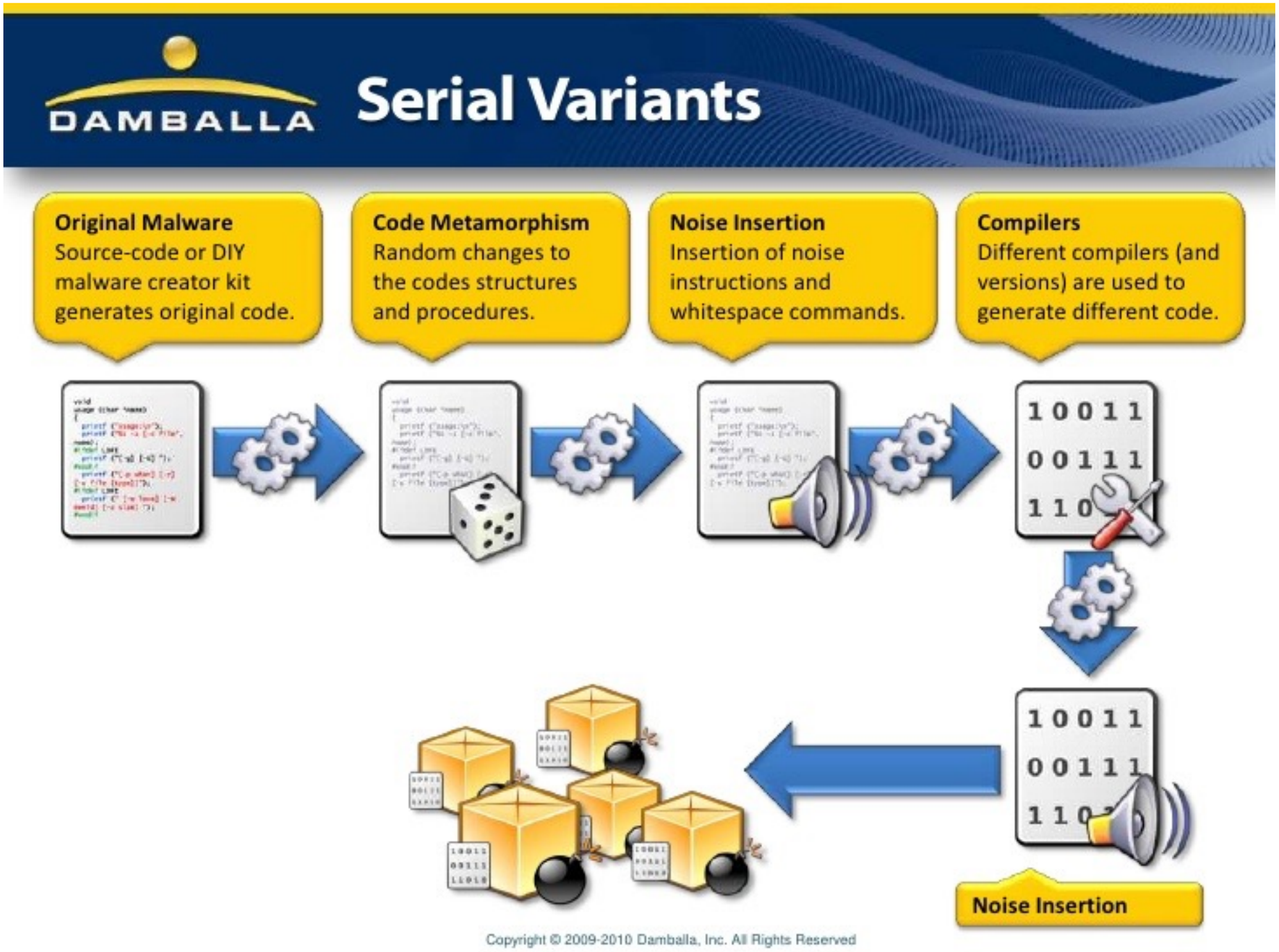
- Now malware has a whole other level of sophistication
- Now we speak of …
  - Advanced Persistent Malware
    - Target specific organizations for a singular objective
    - Attempt to gain a foothold in the environment (common tactics include phishing emails)
    - Escalate privileges – use exploits and password cracking to acquire administrator privileges
    - Use the compromised systems as access into the target network
    - Collect information on surrounding infrastructure
    - Move laterally and deploy additional tools that hel fulfill the attack objective
    - Cover tracks to maintain access for future initiatives



Advanced

- More like a software engineering approach
  - Growing demand for "reliable" malware
  - Want malware to feed into existing criminal enterprise
  - Online - criminals use online banking too
- Malware ecosystem
  - *Measuring Pay-per-Install: The Commoditization of Malware Distribution*, USENIX 2011
  - Tool kits
  - Sharing of exploit materials
  - Combine multiple attack methodologies
- Not hard to find DIY kits for malware

Malware Lifecycle



Persistent

- Malware writers are focused on specific task
  - Criminals willing to wait for gratification
  - Cyberwarfare
- Low-and-slow
  - Can exfiltrate secrets at a slow rate, especially if you don't need them right away
- Plus can often evade or disable defenses

Threat

- Coordinated effort to complete objective
  - Not just for kicks anymore
- Well-funded
  - There is money to be made
    - … At least that is the perception

Example: Sirefef

- Windows malware - Trojan to install rootkit
  - See http://antivirus.about.com/od/virusdescriptions/a/What-Is-Sirefef-Malware.htm
- Attack: "Sirefef gives attackers full access to your system"
  - Runs as a Trojan software update (GoogleUpdate)
  - Runs on each boot by setting a Windows registry entry
  - Some versions replace device drivers
- Downloads code to run a P2P communication
  - Steal software keys and crack password for software piracy
  - Downloads other files to propagate the attack to other computers

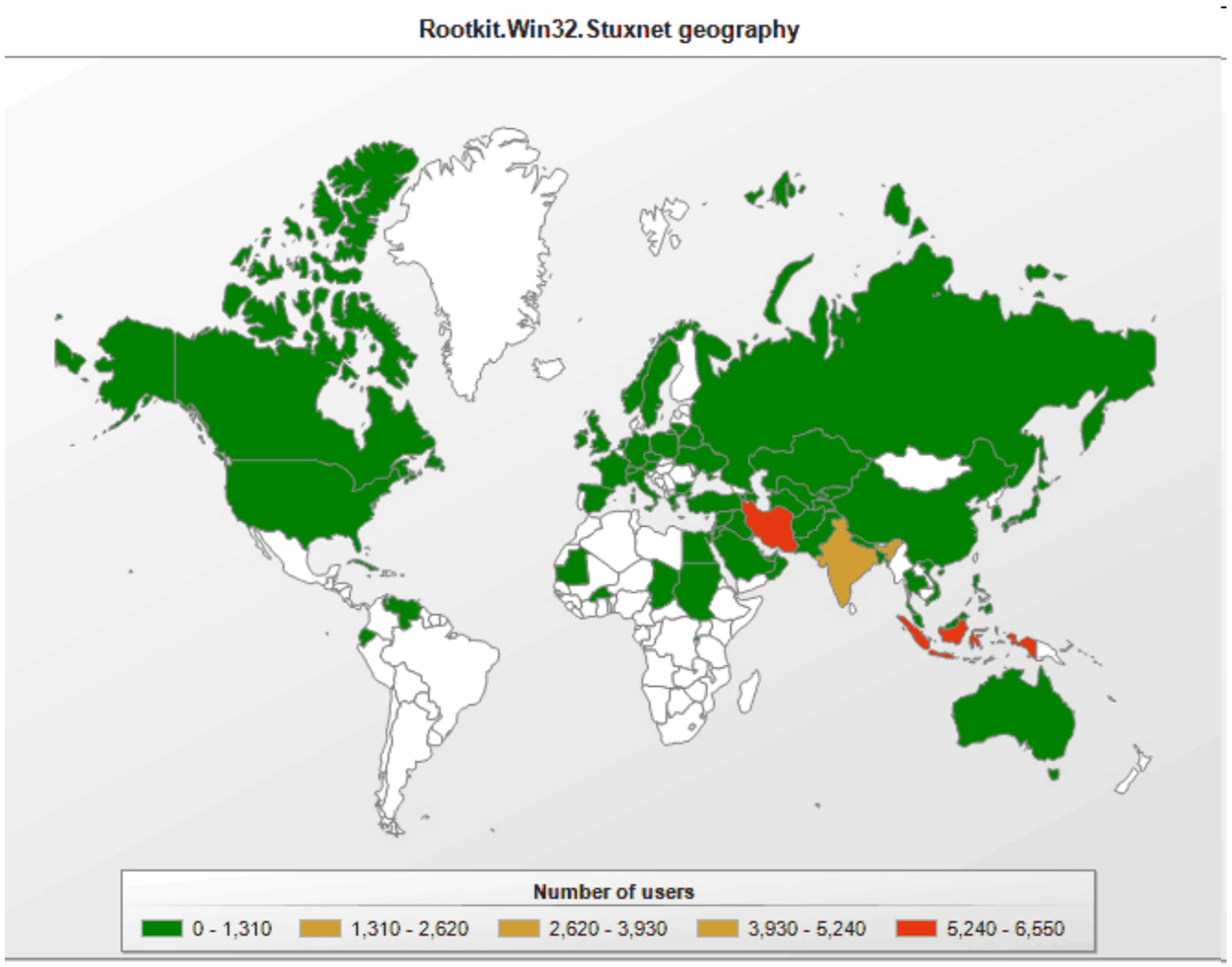
Example: Sirefef

- Windows malware - Trojan to install rootkit
  - See http://antivirus.about.com/od/virusdescriptions/a/What-Is-Sirefef-Malware.htm
- Stealth: "while using stealth techniques in order to hide its presence"
  - "altering the internal processes of an operating system so that your antivirus and anti-spyware can't detect it."
    - Disable: Windows firewall, Windows defender
    - Changes: Browser settings
    - Join bot
- Microsoft: *"This list is incomplete"*

Example: Stuxnet

- Symantec's slides

Real world example: Stuxnet Worm



- https://securelist.com/myrtus-and-guava-episode-3/29616/

Example: Stuxnet

- Symantec's slides

Stuxnet: Overview

- June 2010: A worm targeting Siemens WinCC industrial control system.
- Targets high speed variable-frequency programmable logic motor controllers from just two vendors: Vacon (Finland) and Fararo Paya (Iran)
- Only when the controllers are running at 807Hz to 1210Hz. Makes the frequency of those controllers vary from 1410Hz to 2Hz to 1064Hz.
- http://en.wikipedia.org/wiki/Stuxnet

2

Example: Stuxnet

- Symantec's slides

Timeline

- 2009 June: Earliest Stuxnet seen
  - Does not have signed drivers
- 2010 Jan: Stuxnet driver signed
  - With a valid certificate belonging to Realtek Semiconductors
- 2010 June: Virusblokada reports W32.Stuxnet
  - Verisign revokes Realtek certificate
- 2010 July: Anti-virus vendor Eset identifies new Stuxnet driver
  - With a valid certificate belonging to JMicron Technology Corp
- 2010 July: Siemens report they are investig[illegible]re SCADA systems
  - Verisign revokes JMicron certificate

Example: Stuxnet

- Symantec's slides

Possible Attack Scenario (Conjecture)

- Reconnaissance
  - Each PLC is configured in a unique manner
  - Targeted ICS's schematics needed
  - Design docs stolen by an insider?
  - Retrieved by an early version of Stuxnet
  - Stuxnet developed with the goal of sabotaging a specific set of ICS.
- Development
  - Mirrored development Environment needed
    - ICS Hardware
    - PLC modules
    - PLC development software
  - Estimation
    - 6+ man-years by an experienced and well funded development team

Example: Stuxnet

- Symantec's slides

Attack Scenario (2)

- The malicious binaries need to be signed to avoid suspicion
  - Two digital certificates were compromised.
  - High probability that the digital certificates/keys were stolen from the companies premises.
  - Realtek and JMicron are in close proximity.
- Initial Infection
  - Stuxnet needed to be introduced to the targeted environment
    - Insider
    - Third party, such as a contractor
  - Delivery method
    - USB drive
    - Windows Maintenance Laptop
    - Targeted email attack

Example: Stuxnet

- Symantec's slides

Attack Scenario (3)

- Infection Spread
  - Look for Windows computer that program the PLC's
    - The Field PG are typically not networked
    - Spread the Infection on computers on the local LAN
      - Zero-day vulnerabilities
      - Two-year old vulnerability
      - Spread to all available USB drives
  - When a USB drive is connected to the Field PG, the Infection jumps to the Field PG
    - The "airgap" is thus breached

Example: Stuxnet

- Symantec's slides

Attack Scenario (4)

- Target Infection
  - Look for Specific PLC
    - Running Step 7 Operating System
  - Change PLC code
    - Sabotage system
    - Hide modifications
  - Command and Control may not be possible
    - Due to the "airgap"
    - Functionality already embedded

Take Away

- Malware is now very functional and effective
  - Tools for building and hiding malware from detection
  - Malware can be difficult to notice much less detect and remove
- Malware leverages multiple vulnerabilities to escalate privileges and disable defenses
  - Getting code running on the host enables control of host
  - And there are lots of ways to download code to hosts

- *What are the nature of the vulnerabilities? Next time*