PennState

CSE 543: Computer Security

Module: Malware

Prof. Syed Rafiul Hussain
Department of Computer Science and Engineering
The Pennsylvania State University

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

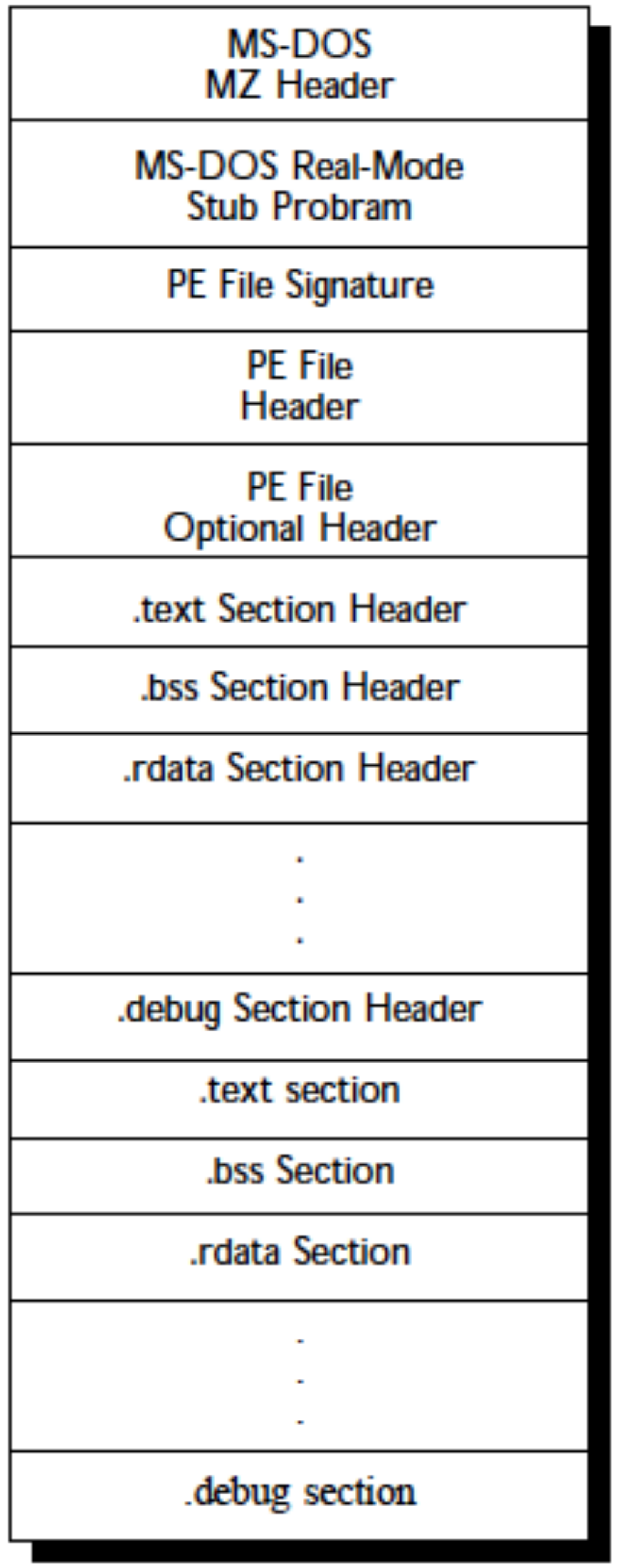
- Is an attack that modifies programs on your host
- Approach

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



MS-DOS
MZ Header

MS-DOS Real-Mode
Stub Probram

PE File Signature

PE File
Header

PE File
Optional Header

.text Section Header

.bss Section Header

.rdata Section Header

.
.
.

.debug Section Header

.text section

.bss Section

.rdata Section

.
.
.

.debug section

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

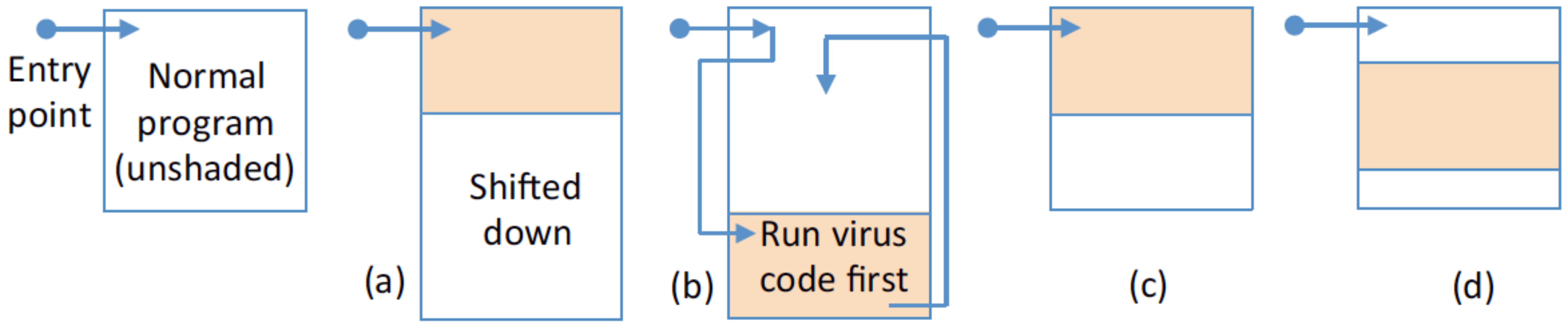


Figure 7.1: Virus strategies for code location. Virus code is shaded. (a) Shift and prepend. (b) Append. (c) Overwrite from top. (d) Overwrite at interior.

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

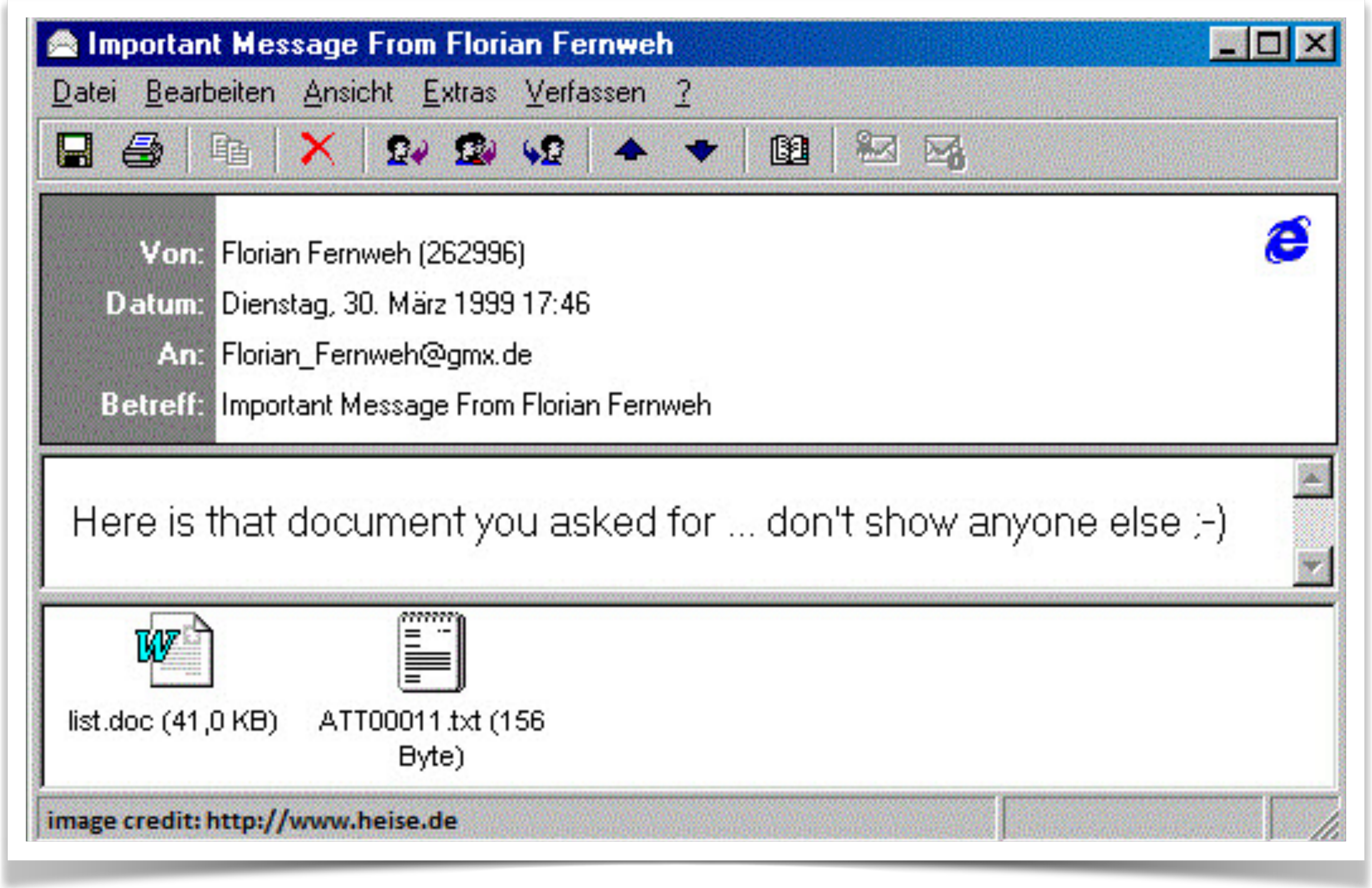
- Keeping with the virus analogy, getting a virus to run on a computer system is called infecting the system
  - Program that attaches itself to another (usually trusted program)
  - How can an adversary infect another's computer?
    - Tricking users into downloading their malware
      - Need to also 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.



Important Message From Florian Fernweh
Datei Bearbeiten Ansicht Extras Verfassen ?

Von: Florian Fernweh (262996)
Datum: Dienstag, 30. März 1999 17:46
An: Florian_Fernweh@gmx.de
Betreff: Important Message From Florian Fernweh

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

list.doc (41,0 KB)
ATT00011.txt (156 Byte)

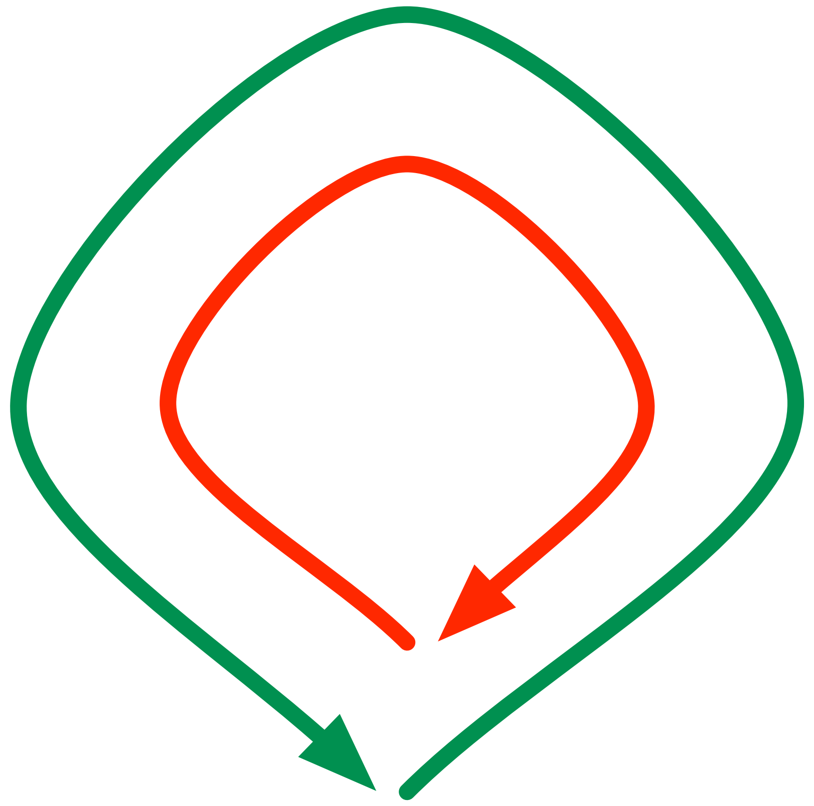
image credit: http://www.heise.de

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

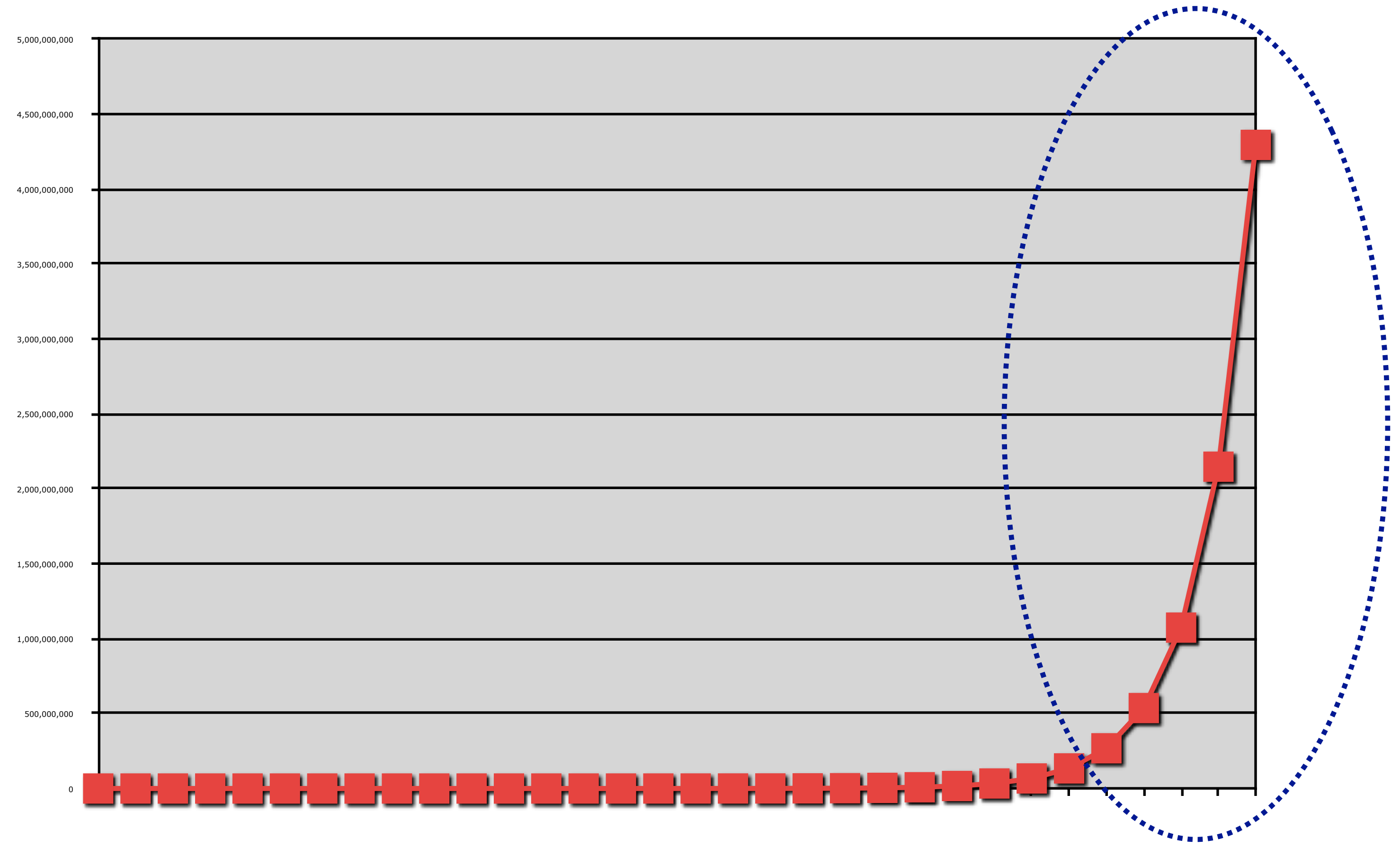


The Morris Worm (1988)

- Robert Morris, a 23 year old 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.

Other scanning strategies

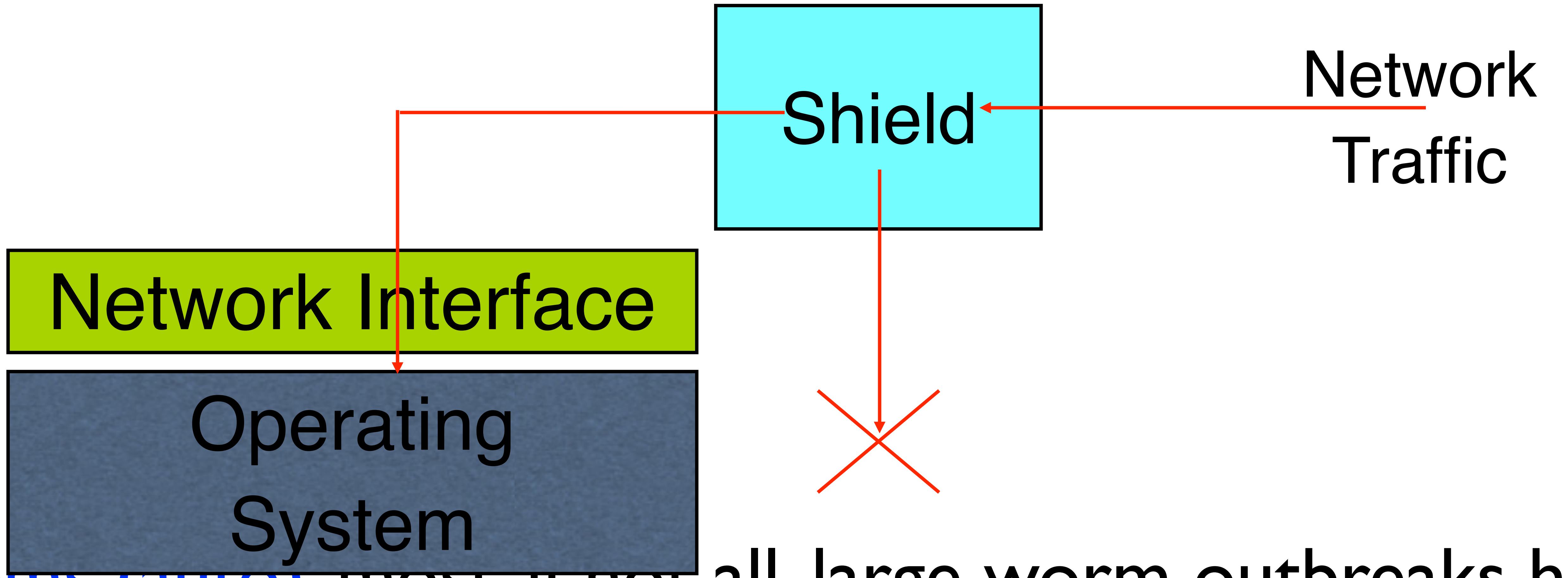
- The doomsday worm: a flash worm
  - Create a hit list of all vulnerable hosts
    - Staniford et al. argue this is feasible
    - Would contain a 48MB list
  - Do the infect and split approach
  - Use a zero-day vulnerability



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

Worms: Defense Strategies

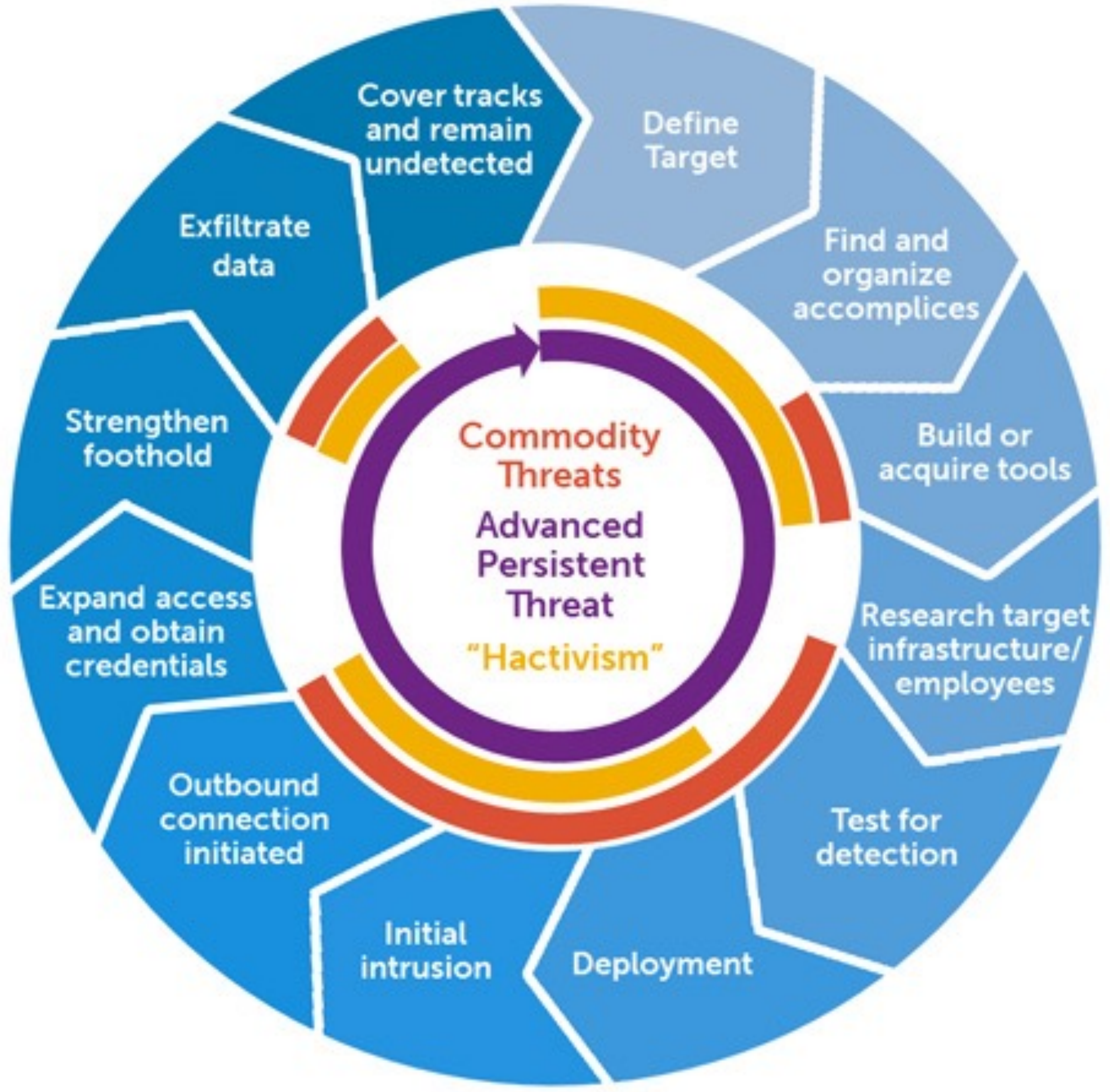
- (Network) Packet Filtering: look for unnecessary or unusual communication patterns, then drop them on the floor
  - This is the dominant method, sophisticated
- (Network) Heterogeneity: use more than one vendor for your networks



- (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 help fulfill the attack objective
    - Cover tracks to maintain access for future initiatives

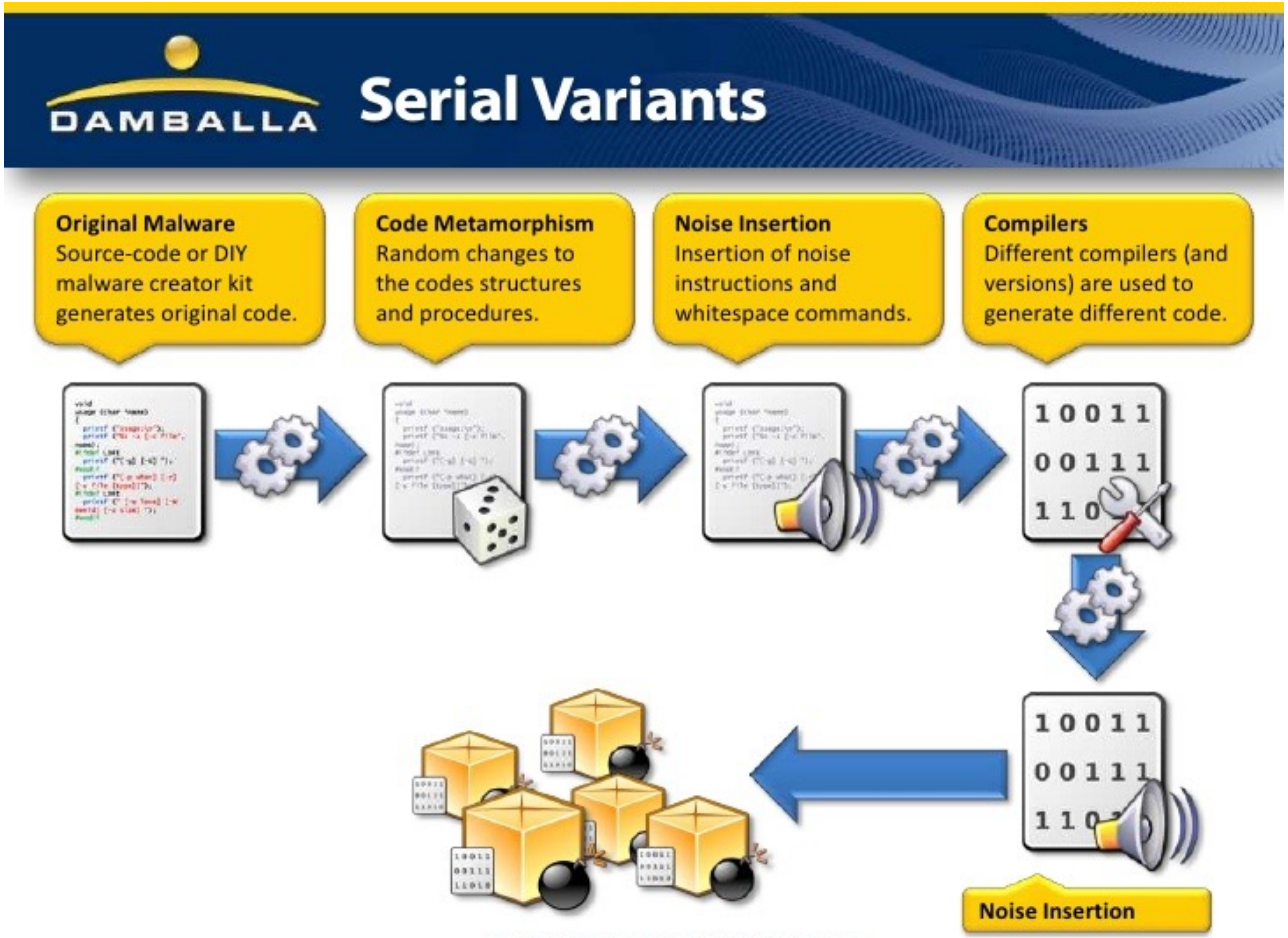


Define Target
Find and organize accomplices
Build or acquire tools
Research target infrastructure/ employees
Test for detection
Deployment
Initial intrusion
Outbound connection initiated
Expand access and obtain credentials
Strengthen foothold
Exfiltrate data
Cover tracks and remain undetected
Commodity Threats
Advanced Persistent Threat
"Hactivism"

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



Original Malware
Source-code or DIY malware creator kit generates original code.

Code Metamorphism
Random changes to the codes structures and procedures.

Noise Insertion
Insertion of noise instructions and whitespace commands.

Compilers
Different compilers (and versions) are used to generate different code.

Copyright © 2009-2010 Damballa, Inc. All Rights Reserved

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

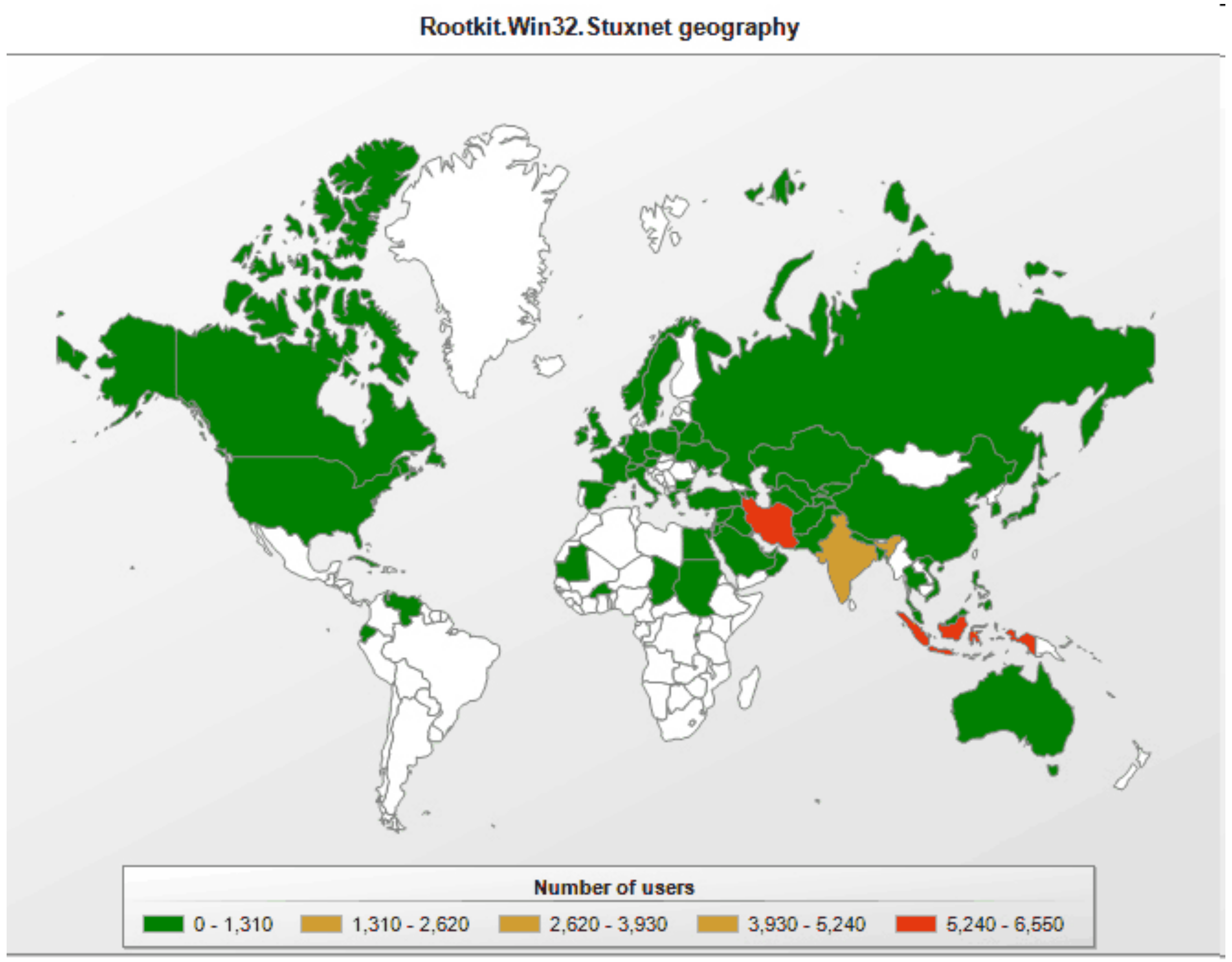
Threat

- 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: Stuxnet

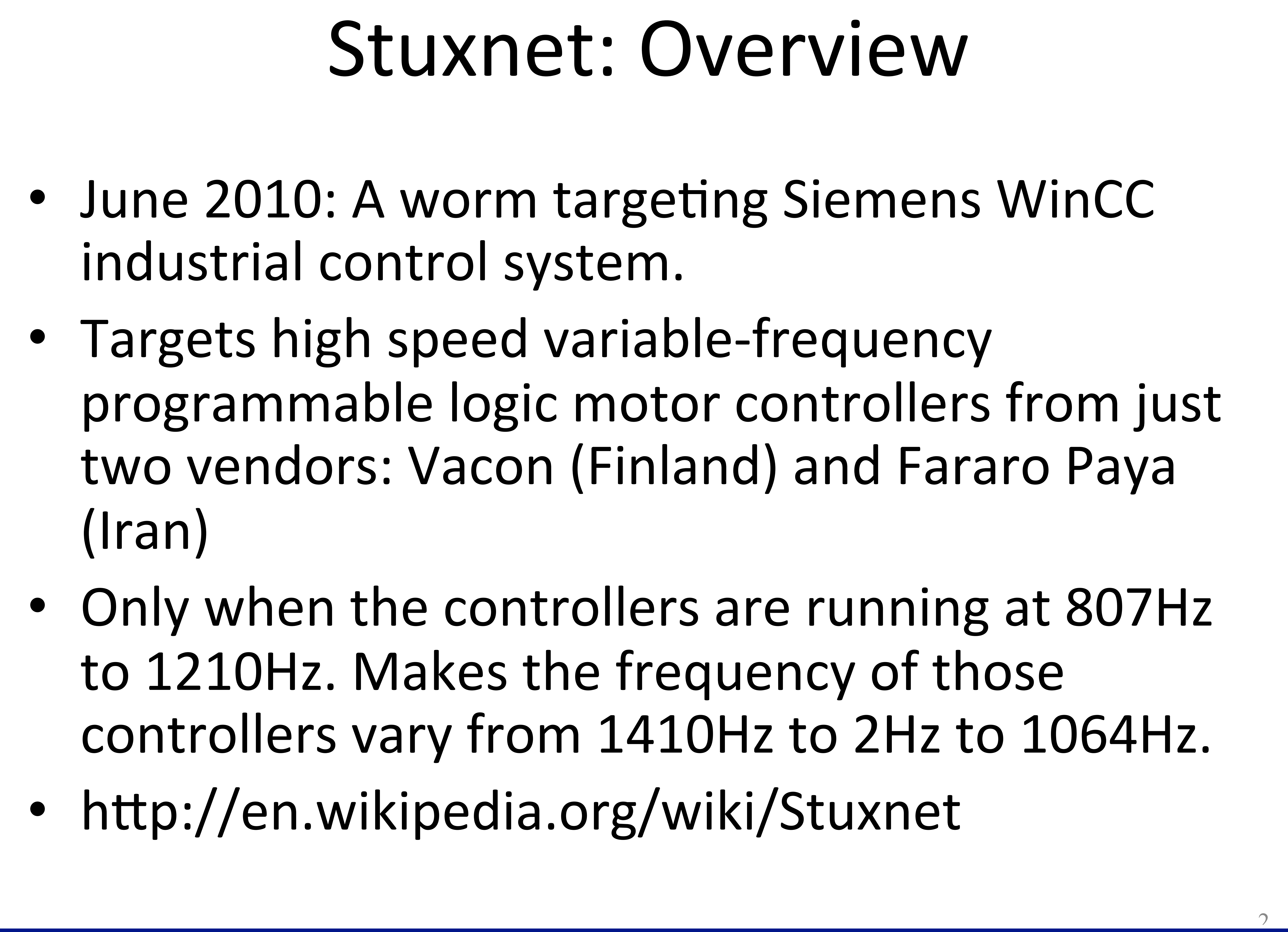
- Symantec's slides

Real world example: Stuxnet Worm



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

Example: Stuxnet



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

Example: Stuxnet

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

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

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

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*

Reading papers …

- What is the purpose of reading research papers?
  - Purpose:
    - Get paper's contributions (what?)
    - Understand the techniques (how?)
    - Critically analyze the worthiness of the paper
    - Where it fits in to the existing body of knowledge
- How do you read research papers?

Understanding what you read

- Things you should be getting out of a paper
  - (Q1) What is the central idea proposed/explored in the paper?
    - Abstract
    - Introduction
    - Conclusions

    These are the best areas to find an overview of the contribution
  - Motivation: What is the problem being addressed?
  - (Q2) How does this work fit into others in the area?
    - Related work - often a separate section, sometimes not, every paper should detail the relevant literature. Papers that do not do this or do a superficial job are almost sure to be bad ones.
    - An informed reader should be able to read the related work and understand the basic approaches in the area, and why they do not solve the problem effectively

Understanding what you read (cont.)

- (Q3) What claims do the authors make? (examine the abstract, intro, conclusion for high-level claims, the "design/analysis" section for more precise claims)
- What scientific devices are the authors using to communicate their point?
- Methodology - this is how they evaluate their solution.
  - Theoretical papers typically validate a model using mathematical arguments (e.g., proofs)
  - Experimental papers evaluate results based on a design of a test apparatus (e.g., measurements, data mining, synthetic workload simulation, trace-based simulation).
    - Empirical research evaluates by measurement.
  - Some papers have no evaluation at all, but argue the merits of the solution in prose (e.g., paper design papers)

Understanding what you read (cont.)

- What do the authors claim?
  - Results - statement of new scientific discovery.
    - Typically some abbreviated form of the results will be present in the abstract, introduction, and/or conclusions.
    - Note: just because a result was accepted into a conference or journal does necessarily not mean that it is true. Always be circumspect.
- What should you remember about this paper?
  - Take away - what general lesson or fact should you take away from the paper.
  - Note that really good papers will have take-aways that are more general than the paper topic.

Summarize Thompson Article

- Contribution
- Motivation
- Related work
- Methodology
- Results
- Take away

A Sample Summary

- Contribution: Ken Thompson shows how hard it is to trust the security of software in this paper. He describes an approach whereby he can embed a Trojan horse in a compiler that can insert malicious code on a trigger (e.g., recognizing a login program).
- Motivation: People need to recognize the security limitations of programming.
- Related Work: This approach is an example of a Trojan horse program. A Trojan horse is a program that serves a legitimate purpose on the surface, but includes malicious code that will be executed with it. Examples include the Sony/BMG rootkit: the program provided music legitimately, but also installed spyware.
- Methodology: The approach works by generating a malicious binary that is used to compile compilers. Since the compiler code looks OK and the malice is in the binary compiler compiler, it is difficult to detect.
- Results: The system identifies construction of login programs and miscompiles the command to accept a particular password known to the attacker.
- Take away: *What is the transcendent truth?????* (see next slide)