CSCD 303
Essential Computer Security
Fall 2017

Lecture 13 -
Malware – Evasion, Prevention, Detection, Removal

Reading: Chapter 6 – CompTIA Book, Links

Overview

- **Malware**
  - Techniques for Evasion

Detection/Removal

  - Antivirus/Antitrojan

Malware So Far

- So far, have looked at malware examples
  - Viruses
  - Worms
  - Trojans
  - Combination
- Have not covered
  - Rootkits – Good project topic
  - Botnets
- Answer question, How does malware hide itself?

  How do detection programs work?

  Anti-virus or trojan or other

Stealth Virus Techniques

Stealth Techniques
Virus

- Metamorphic malware is rewritten with each iteration so that each succeeding version of the code is different from the preceding one
- Code changes makes it difficult for signature-based antivirus software programs to recognize that different iterations are the same malicious program.

Metamorphic Viruses

Apparition: an early Win32 metamorphic virus

- Carries its source code (contains useless junk)
- Looks for compiler on infected machine
- Changes junk in its source and recompiles itself

New binary copy looks different!

Obfuscation and Anti-Debugging

Common in all kinds of malware

Goal: prevent code analysis and signature-based detection, foil reverse-engineering

Code obfuscation and mutation

Packed binaries, hard-to-analyze code structures

Different code in each copy of the virus

- Effect of code execution is the same, but this is difficult to detect by passive/static analysis

Mutation Techniques

Malware writes have created **obfuscation engines**

Real Permutating Engine/RPME, ADMutate, etc.

Large arsenal of obfuscation techniques

Instructions reordered, branch conditions reversed, different register names, different subroutine order

Jumps and NOPs inserted in random places

Garbage opcodes inserted in unreachable code areas

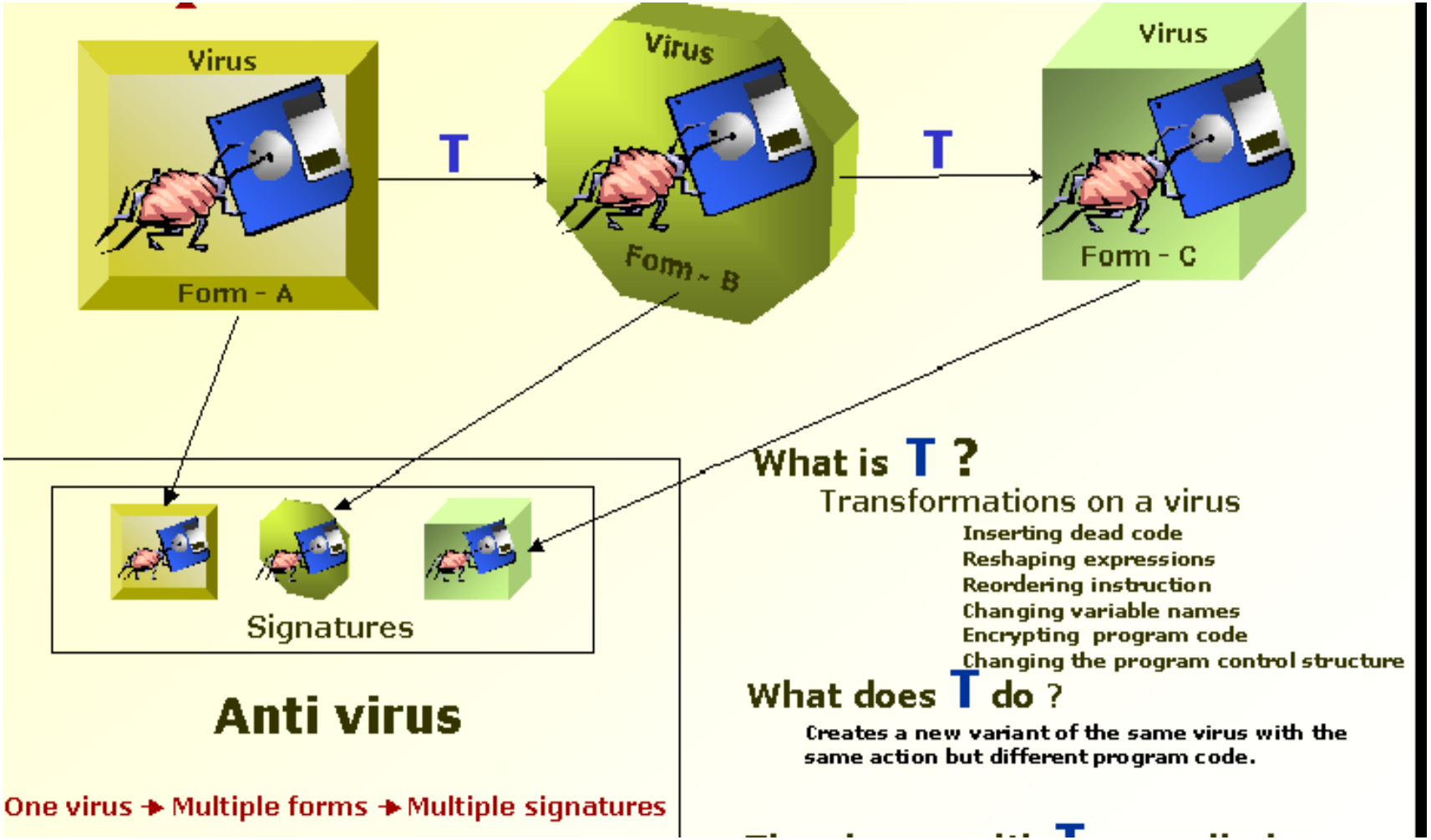
Instruction sequences replaced with other instructions that have the same effect, but different opcodes

Mutate SUB EAX, EAX into XOR EAX, EAX or

MOV EBP, ESP into PUSH ESP; POP EBP

There is no constant, recognizable virus body

Mutations in Action



Example:

- If the scanner were looking for the instructions

```
mov ax, 2513H
mov dx, 1307H
int 21H
```

One might modify the virus to instead execute this operation code

```
mov ax, 2513H
mov dx, 1307H
xchg ax, dx
int 21H
```

The scanner can no longer see it, and the virus can go undetected.

Some Transformations

- Insert jmps

```
#make_BIN#

MOV AX, 5
MOV BX, 10
ADD AX, BX
SUB AX, 1
HLT
```

```
#make_BIN#

MOV AX, 5
MOV BX, 10
ADD AX, BX
jmp proc_sub
proc_sub: SUB AX, 1
 HLT
```

- Add redundant labels

```
MOV AX, 1
MOV AX, 2
```

```
x1:
MOV AX, 1

x2:
MOV AX, 2
```

Another Example - Transformations

Virus Code

(from Chernobyl CIH 1.4):

```
Loop:
        pop     ecx
        jecxz   SFModMark
        mov     esi, ecx
        mov     eax, 0d601h
        pop     edx
        pop     ecx
        call    edi
        jmp     Loop
```

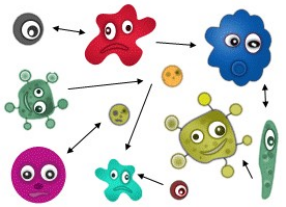
Morphed Virus Code

(from Chernobyl CIH 1.4):

```
Loop:
        pop     ecx
        nop
        jmp L1
L3:     call    edi
        xor     ebx, ebx
        beqz    N2
N2:     jmp     Loop
        jmp L4
L2:     nop
        mov     eax, 0d601h
        pop     edx
        pop     ecx
        nop
        jmp L3
L1:     jecxz   SFModMark
        xor     ebx, ebx
        beqz    N1
N1:     mov     esi, ecx
        jmp L2
L4:
```


Mihai Christodorescu mihai@cs.wisc.edu
WiSA Project http://www.cs.wisc.edu/wisa

Polymorphic Viruses



- Encrypted viruses: constant decryptor followed by the encrypted virus body
- Polymorphic viruses: each copy creates a new random encryption of the same virus body
  - Decryptor code constant and can be detected
  - Historical note: “Crypto” virus decrypted its body by brute-force key search to avoid explicit decryptor code

Examples of Polymorphic Malware

- **CryptoWall Ransomware**
  - **CryptoWall** is a polymorphic ransomware strain that encrypts files on the victim's computer and demands a ransom payment for their decryption. The polymorphic builder used in Cryptowall is used to develop what is essentially a new variant for every potential victim.
- **Storm Worm Email**
  - Email sent in 2007 with the subject "230 dead as storm batters Europe" was, at one point, responsible for as much as 8% of all global malware infections.
  - When the message's attachment is opened, the malware installs wincom32 service and a trojan onto the recipient's computer, transforming it into a bot.

One of the reasons the storm worm was so hard to detect with traditional antivirus software was the malicious code used morphed every 30 minutes !!!

Anti-virus

- Anti-virus
  - Will identify infections, viruses, trojans, worms
  - Not always able to exactly identify what got you
  - First step,
    - Detect something is wrong
    - Try to identify it - Key
  - Next step
    - Try to remove it and restore the files if possible

Static vs. Dynamic Analysis

Static Analysis

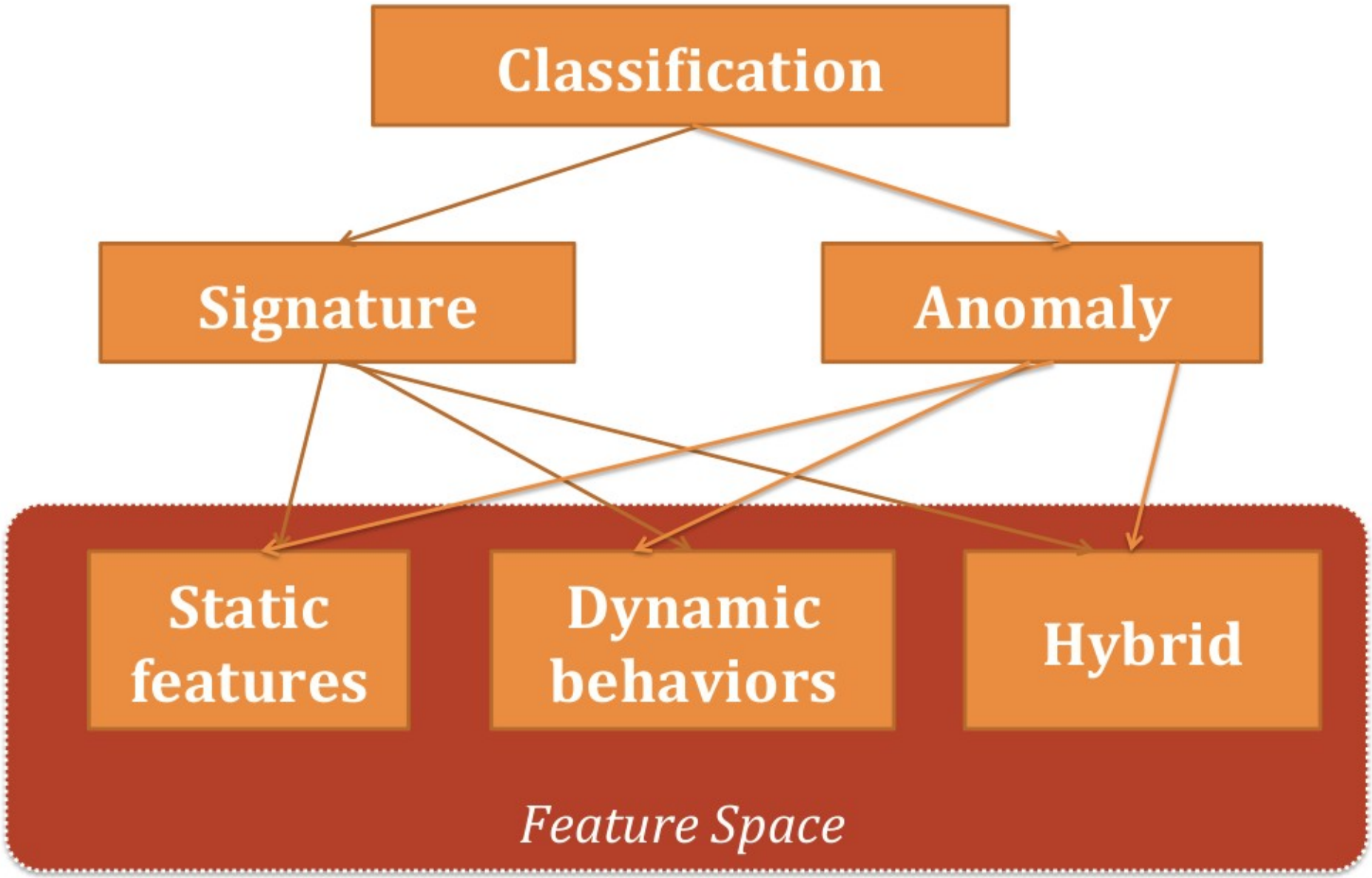
- Code is Not Executed
- Autopsy or Dissection of “Dead” Code

Dynamic Analysis

- Observing and Controlling Running (“live”) Code

The Fastest Path to the Best Answers Will Usually Involve ….

Combination of Both


Classification
Signature
Anomaly
Static features
Dynamic behaviors
Hybrid
Feature Space

Modern Antivirus Software

- 1st Generation: Simple scanners
  - Require signatures to detect the behavior of known viruses
  - Look at program length often and alert the administrators if anything has changed
    - Signatures for system binaries
    - Signatures for known viruses
  - No so good for zero-day attacks

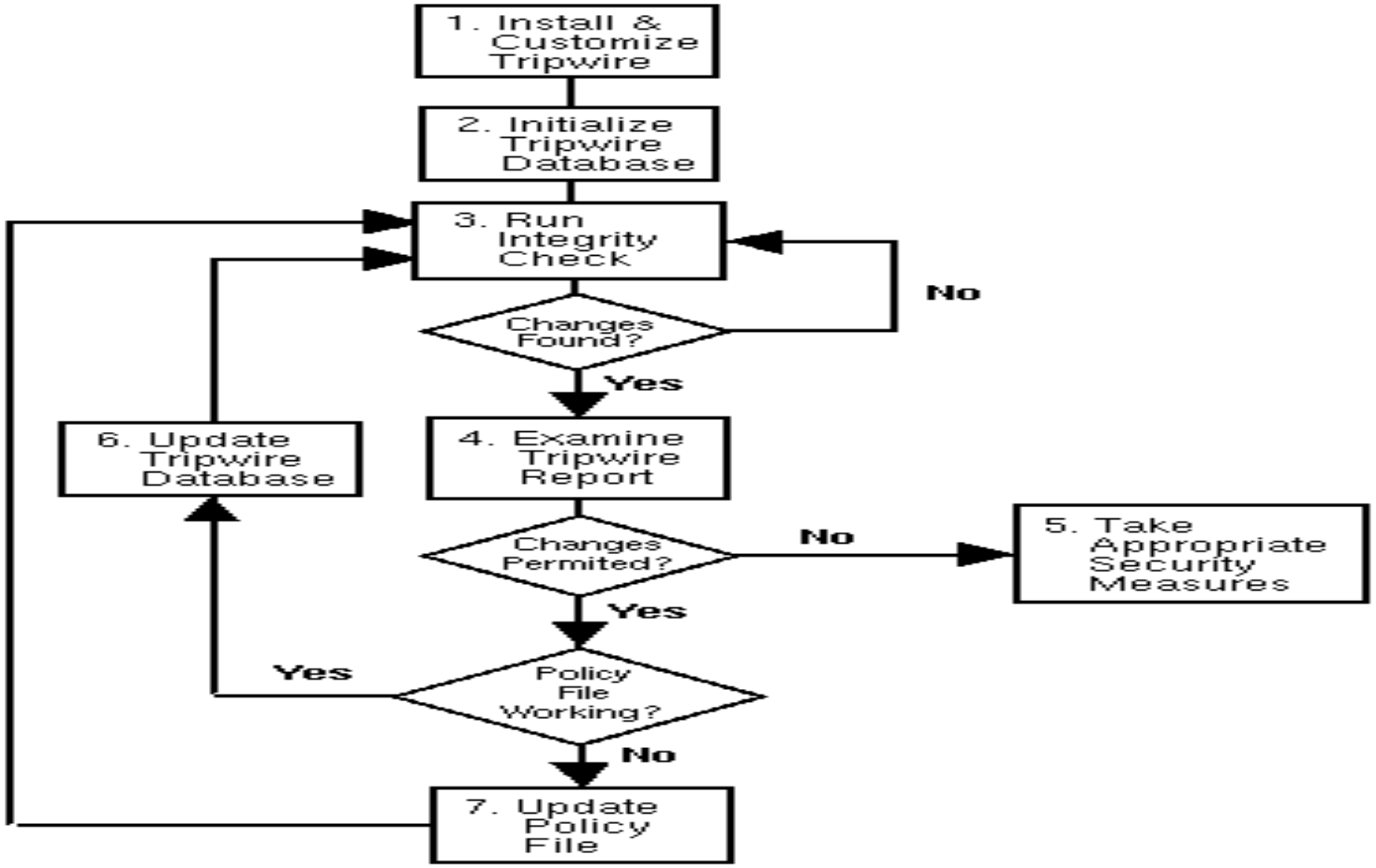
Signature File Monitoring

- File integrity monitoring (FIM) is an internal control or process that performs validation of integrity of operating system and application software files
- Uses a verification method between current file state and a known, good baseline
- This comparison method often involves calculating a known cryptographic checksum of file's original baseline and comparing to calculated checksum of current state of file

Tripwire is one example of this type of file monitoring software

https://www.tripwire.com/products/tripwire-file-integrity-manager/

Tripwire



http://original.jamesthornton.com/redhat/linux/9/Reference-Guide/figs/tripwire/tripwire.png

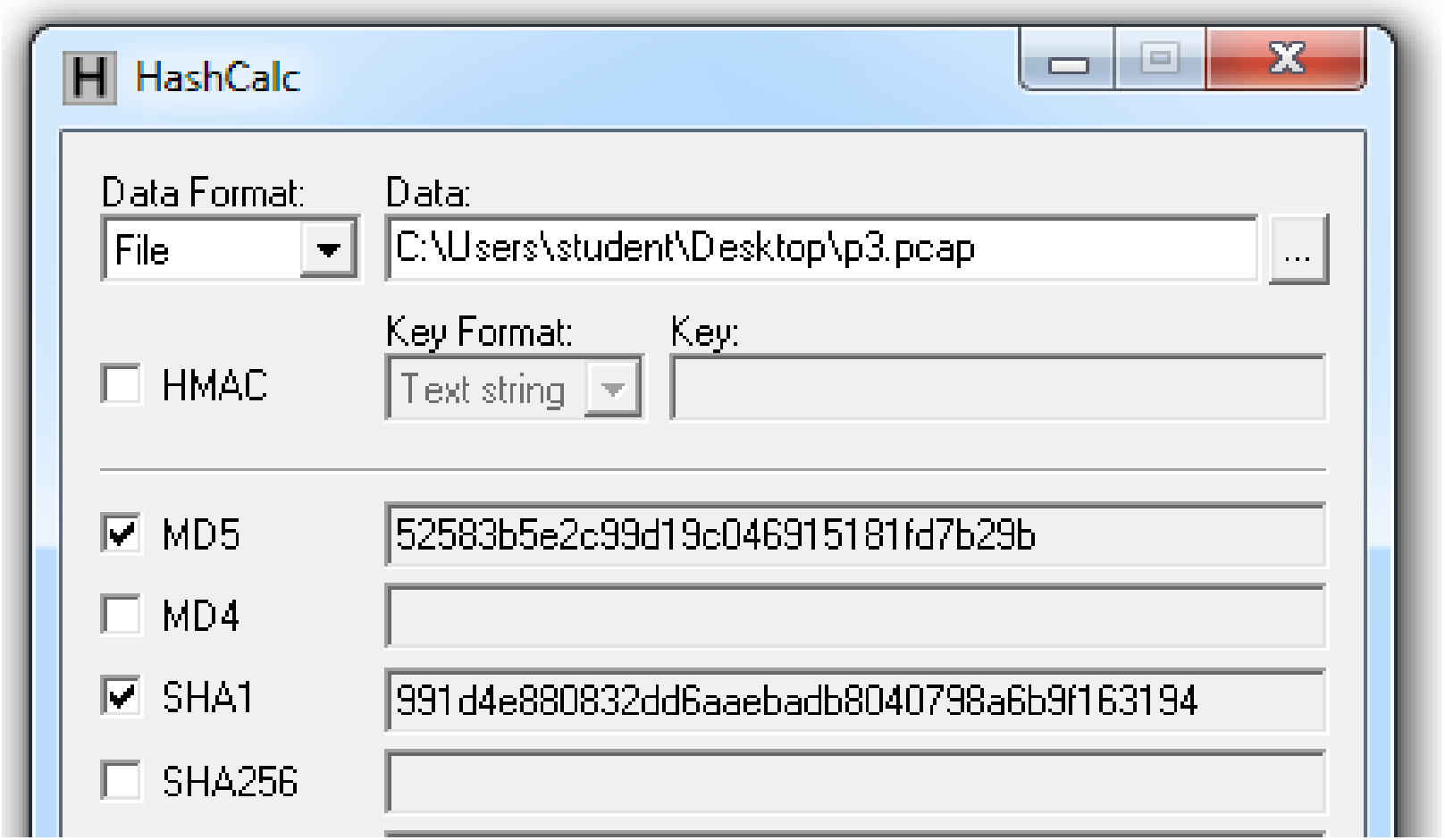
Signatures for Malware

- Malware signatures first generation anti-malware programs
- Use Hashes of the entire file or fragments of known malware
- Store in a database, use for suspected malware identification

Hashes

- MD5 or SHA-1
- Condenses a file of any size down to a fixed-length fingerprint
- Uniquely identifies a file well in practice
  - There are MD5 collisions but they are not common
  - Collision: two different files with the same hash

HashCalc


HashCalc
Data Format:
File
Data:
C:\Users\student\Desktop\p3.pcap
...
HMAC
Key Format:
Text string
Key:
MD5
52583b5e2c99d19c046915181fd7b29b
MD4
SHA1
991d4e880832dd6aaebadb8040798a6b9f163194
SHA256

Malware Hash Uses

- Label a malware file
- Use in signature based malware programs
- Share the hash with other analysts to identify malware
- Search the hash online to see if someone else has already identified the file
- Problems with Signature based approach?

Example of Virus Signature Hashes

Abraxas-1200=

cd21b43c33c9ba9e00cd21b74093ba0001b9b004cd21c3b4

Abraxas-1214=

cd21b43c33c9ba9e00cd21b74093ba0001b9be04cd21c3b4

Abraxas-15xx=

b90200b44ebaa80190cd21b8023c33c9ba9e00cd21b74093

Acid #2=

99cd212d0300c606ae02e9a3af02b440b9a20299c

d21b800422bc9cd21b440b91a00baae02cd21b8

Acid-670=

e800005d81ed0300b8ffa02bdbcd210681fbffa07

458b82135cd21899e9e028c86a0028cd8488ec026

803e00005a757c26832e03002e26832e12002e26a

11200

Ada #2=

480200740f80fc41741b80fc1374163d004b74069d2eff2e

Ada #3= 8c4f0cb8004bbab012cd21b402b207cd

Identifying Abraxas-1214 Virus Signature in File

73746172747570 5c77696e7269702e626174220d0

a40646972202f73202f62202f6c20633a5c77696e

7a697033322e657865207c207365742077 7a3d0d0

a40464f52202f4620222f73202f62202f6c20633a

5c2a2e7a697027680401000600204000a5a5a5a**c**

d21b43c33c9ba9e00cd21b74093ba0001b9be04cd

21c3b431010000ebef68d8244000683f000f006a0

0681020400068020000 80e8320100000bc075266a

Updating the Signatures

- Anti-virus companies must release new signatures each time a new virus is discovered
  - A virus's spread is unimpeded for a while…
  - According to Andreas Marx of AV-Test.org,
    - Took Symantec 25 hours to release an updated signature file in response to W32/Sober.C worm attack

Modern Antivirus Software

- 2nd Generation: Heuristics scanners
  - Don't really rely on the signatures as much, but use "rules of recognition"
  - They look for odd behavior, or code fragments that are often associated with viruses, but again, they don't have specific signatures of every virus it can handle
  - Next slide shows possible behaviors signal malware

Static Heuristics Detection

Possible Heuristics

- Junk code
- Decryption loops
- Self-modifying code
- Use of undocumented API
- Manipulation of interrupt vectors
- Unusual instructions, especially those not emitted by a compiler
- Strings containing obscenities or “virus”
- Difference between entry point and end of file
- Spectral analysis
    - Frequency analysis of instructions

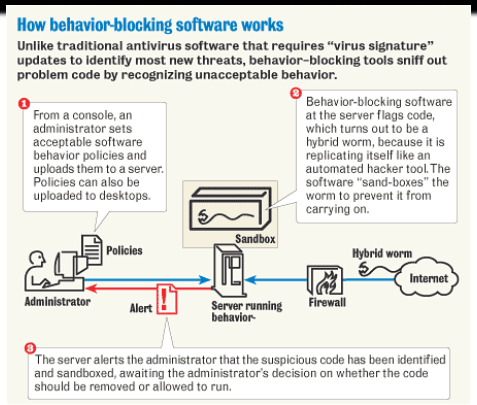
Ex. Heuristic Detection Pykeylogger

- Uses the SetWindowsHookEx API in Win32
  - Specifically the WH_KEYBOARD and WH_KEYBOARD_LL
- Commonly used APIs, but not in background
- Simple heuristic rule:
  - In general, don't allow keyboard strokes to be captured in the background

Modern Antivirus Software

- 3rd Generation: Activity traps/Emulation
  - More like the anomaly detection scheme, where this program just combs memory and looks for actions that are a threat to security rather than structures in the program code in memory
  - This has the distinct advantage of being able to prevent actions proactively rather than be responding retroactively
  - Also uses Emulation or Sandboxing to analyze malware
- 4th Generation: full-featured scanners
  - All of these tools combined and used simultaneously

Sandboxing



- Antivirus program will take suspicious code and run it in a "virtual machine" to see purpose code and how code works
- After the program is terminated the software analyzes the sandbox for any changes, which might indicate a virus

Virus: Antivirus Techniques
Dynamic Methods

Emulation

Analyze code before letting it run

Emulation uses dynamic heuristics

Similar to static heuristics, looking for patterns of behavior

Emulator can also run signature searches some time into run-time of emulated code

Since its not actually running on real machine, can take more time to figure out its true purpose

Modern Antivirus Software

- The differences:
  - Older software scanned once a day, etc.

  Now they are working constantly to prevent infection
  - Norton, McAfee: all had original versions that did scheduled scans or on-boot scans based on signatures
  - Progress adds features as malware authors find exploits

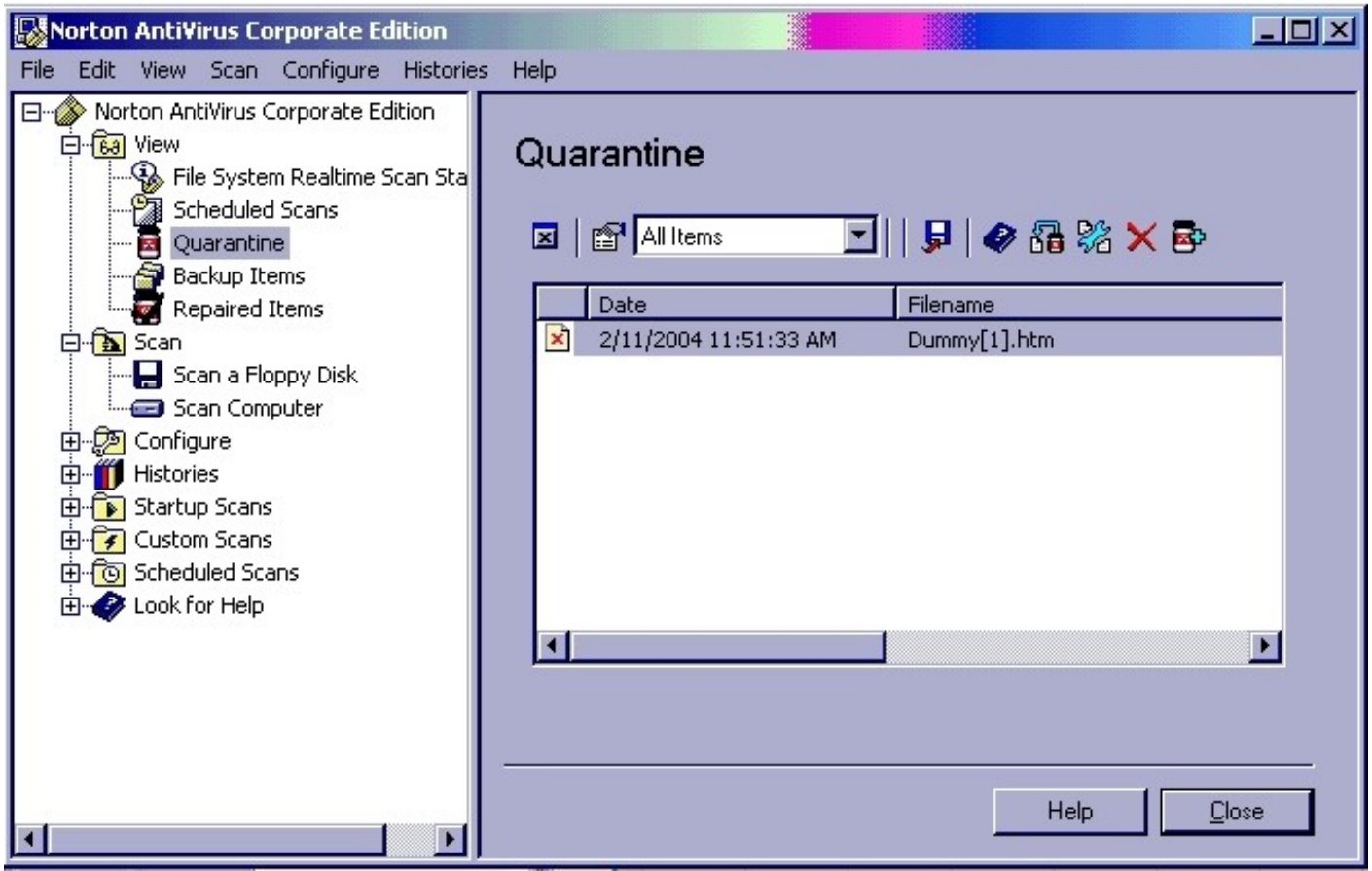
- Commercial Examples
  - Norton 2006 (13.0) introduced Internet Explorer and host file protection
  - Panda Antivirus is award winning
    - Detects **all** strange behavior, very good anomaly detection
    - Balance between good and annoying

Anti-virus

- Two main ways – Treating Infection
  - Quarantine
  - Disinfect

Anti Virus Software

- Quarantine
  - Only temporary until user decides how to handle it, user asked to make a decision



Anti Virus Software

- Why do Anti-Virus Programs Quarantine?
  - Virus detection was generic, can't determine how to clean it off of system
  - Wants user, you, to make a decision
  - Quarantine Actions
    - Copy infected file to quarantine directory
    - Remove original infected file
    - Disable file permissions so user can't accidentally transfer it out of directory

Anti Virus Software

- Disinfect Files
- a. Disinfection by Specific Virus
  - Multiple ways to disinfect files
  - Depends on the type of virus
  - From virus DB, get file executable start address
    - Run generic clean-up routine with start address
  - Can derive this information by running virus in test lab, recording information from infected file
  - Store this information for specific virus

Anti Virus Software

- b. Disinfect by Virus Behavior
  - Disinfect based on assumptions from virus behavior
    - Prepend or Appended viruses
    - Restore original program header
    - Move original byte contents back to original location
  - Can store in advance for each executable file on an uninfected system, system file
    - Program header, file length, checksum of executable file contents, which is a computed check of the file contents
    - Compute various checksums until you get the exact checksum of the file, can be tricky need to figure out which part of the file is original, look for checksum match

Best Recommended Free Antivirus Programs 2017

- A number of recommended programs are free to help keep your computer malware free
  - Bitdefender Antivirus Free
  - Avira Antivirus
  - Avast Free Antivirus
  - AVG Free Antivirus
  - Kaspersky Lab Internet Security 2017
  - Sophos Home Free Antivirus

https://fossbytes.com/10-best-free-antivirus-software-list-2017/

Best Recommended Antivirus Programs 2017

- These recommended programs are not free but hightly recommended
  - Bitdefender Antivirus Plus
  - Kapersky Internet Security
  - Kapersky Total Security
  - Norton
  - Avast
  - McAffe
  - ESET

https://antivirusprotection.reviews/best-antivirus/

Test Your Virus Scanner

- Good to test your anti-virus software to see how well it does
  - There is test file you can use to test your anti-virus software
  - **–The Anti-Virus or Anti-Malware test file**
    - From European Expert Group for IT Security, www.eicar.org
    - Run this file against your virus scanner to determine its effectiveness

    http://www.eicar.org/anti_virus_test_file.htm

Summary

Malware and anti-malware arms race

Who is winniner?

- Let you decide.

The end.

No lab this week

Take-home midterm