Malware

Shuai Wang

香 港 科 技 大 學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

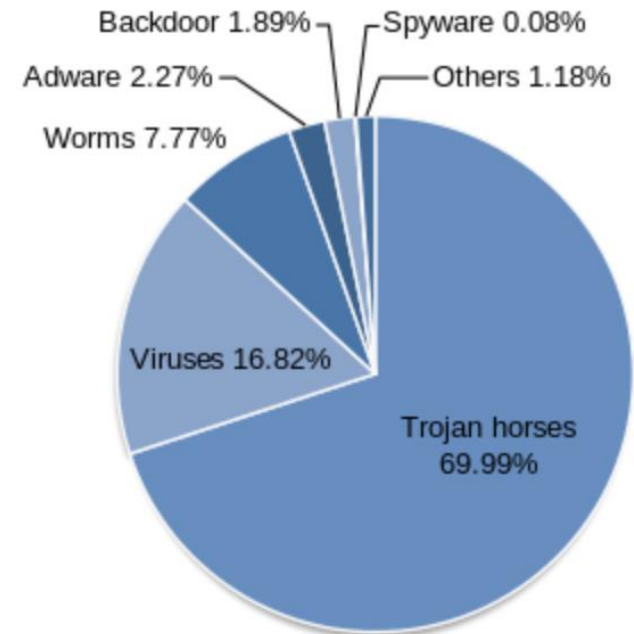Some slides are written by Mark Stamp.

Malicious Software

- Adversaries aim to get code running on your computer that performs tasks of their choosing
  - This code is often called malware
  - What can they do?
    - Steal personal information
    - Delete files
    - Click fraud
    - Steal software serial numbers
    - Use your computer as relay

Malicious Software

- Adversaries aim to get code running on your computer that performs tasks of their choosing
  - Types of malware (no **standard** definition)
    - **Worm/Virus — active or passive propagation**
    - **Trojan horse — unexpected functionality**
    - **Trapdoor/backdoor — unauthorized access**
    - …

Backdoor 1.89%
Spyware 0.08%
Adware 2.27%
Others 1.18%
Worms 7.77%
Viruses 16.82%
Trojan horses 69.99%

Morris Worms

- A worm is a self-propagating program through the Internet.

- What it tried to do
    - Exploits some vulnerability on a target host …
    - embeds itself into the host …
    - Searches for other vulnerable hosts …
   - Do it again until you take down the whole network!

Morris Worm 1988

- Robert Morris, a 23 doctoral student from Cornell
  - Wrote a small (99 line) program
  - Launched on November 3rd, 1988
  - Simply disabled the Internet

- Obtained access to machines by…
  - Exploit bugs in fingerd and sendmail (buffer overflow)
  - Guess user password (a list of common password)

- Flaws in fingerd and sendmail were well-known
  - but not widely patched

Worms

- What makes worms so dangerous is that infection grows at an exponential rate
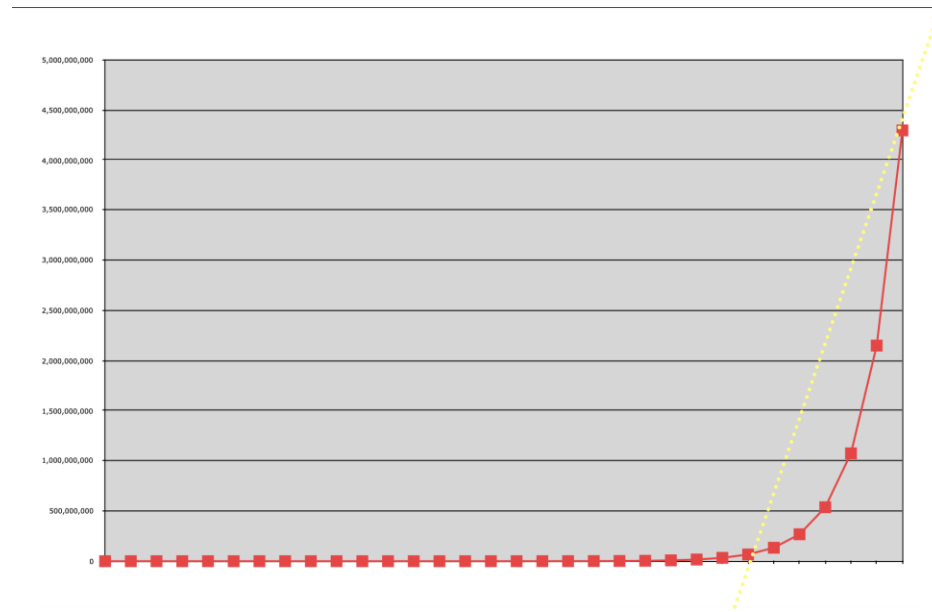
- Repeated propagation could cause machines to lose response.
 - Violation of Availability principle (our CIA principle).

Morris Worm

- Shock to the Internet community of 1988
  - Internet of 1988 *much* different than today
- Internet designed to survive nuclear war
  - Yet, brought down by one graduate student!
  - At the time, Morris' father worked at NSA…
- Could have been much worse
- Result? More security awareness

SQL Slammer -- 2003

Aggregate Scans/Second in the 12 Hours After the Initial Outbreak

- Infected **75,000 systems in 10 minutes!**
  - A tiny worm within one network package
    - One 376-byte UDP package
    - Firewall often let one package thru
  - A bug in database software
    - Although that patch has been released 6 months ago.

- At its peak, infections doubled every 8.5 seconds

- Spread "too fast" so it "burned out" available bandwidth
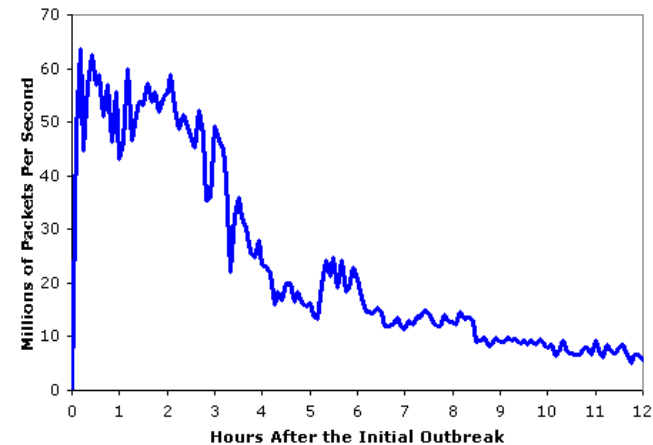
Aggregate Scans/Second in the first 5 minutes based on Incoming Connections To the WAIL Tarpit
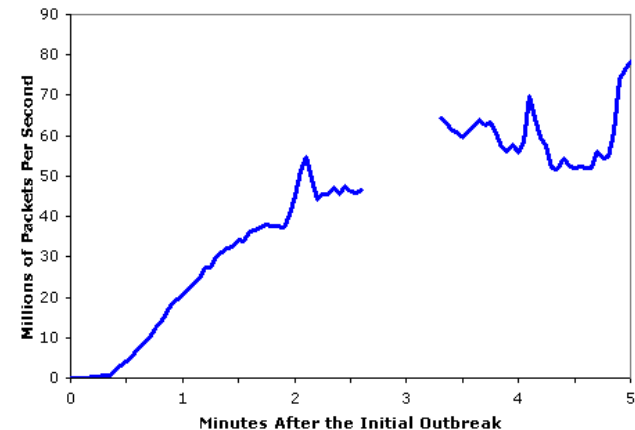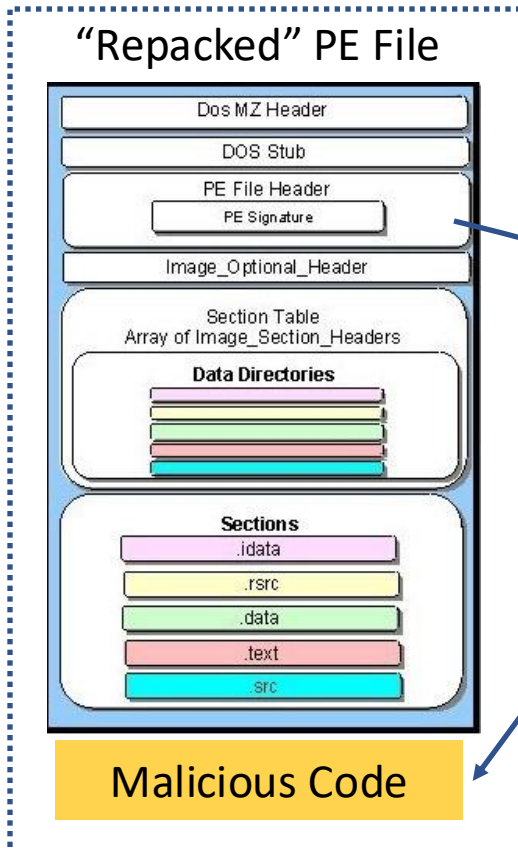
Viruses

- Is an attack that modifies programs on your host

- Approach (for Windows platform)

  - Download a program …

  - Run the program …

  - Searches for executables and other code (firmware, boot sector) that it can modify …

  - Modifies these programs by adding code that the program will run

The Windows Executable File Format: Portable Executable (PE)

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

"Repacked" PE File

| Dos MZ Header |
| DOS Stub |
| PE File Header |
| PE Signature |
| Image_Optional_Header |

Section Table
Array of Image_Section_Headers

Data Directories

Sections
.idata
.rsrc
.data
.text
.src

Malicious Code

1. Modify the header
2. Attach a "malicious" code section to PE file

Redirect the execution
To malicious data section.

Where do Viruses Live?

- They live just about anywhere, such as...

- Boot sector
  - Take control before anything else

- Applications

- Library routines

- Compilers, debuggers, virus checker, etc.
  - These would be particularly nasty!

Trojan Horse

- **Trojan horse** is a malicious program that is disguised as legitimate software.
  - Like the gift horse left outside the gates of Troy by the Greeks, Trojan Horses appear to be useful or interesting, but are actually harmful.
    - But why the "horse" in the figure appears useful? I don't know ;-)
  - Conceptually like virus with some subtle differences
    - Does not duplicate itself
    - Can usually be controlled remotely
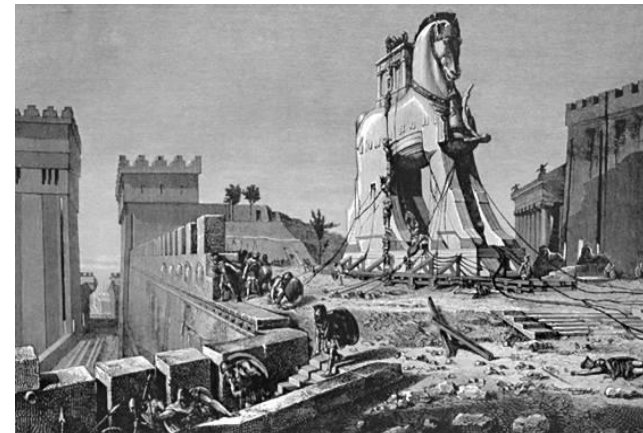
Mac Trojan

- Prototype trojan for the Mac
- File icon for freeMusic.mp3:
 - Can use other accepted file extensions (e.g. .WSF, .WSH, .HTA, .PUB files)

freeMusic.mp3

❑ For a real mp3, double click on icon

  o iTunes opens

  o Music in mp3 file plays

❑ But for freeMusic.mp3, unexpected results…

Mac Trojan

- Double click on freeMusic.mp3
    - iTunes opens (expected)
    - "Wild Laugh" (not expected)
    - Message box (not expected)

Yep, this is an application.

(So what is your iTunes playing right now?)

MP3

OK
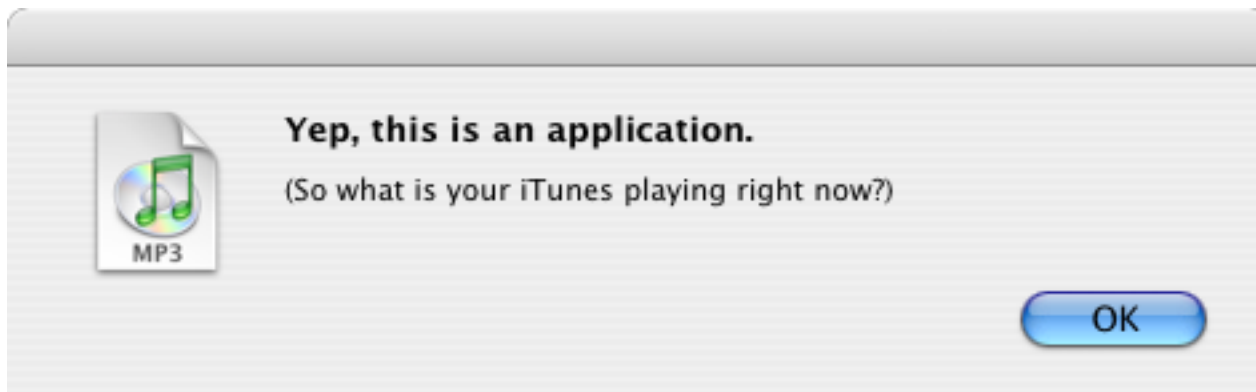
Mac Trojan

- How does freeMusic.mp3 trojan work?

- This "mp3" is an application, not data

❑ This trojan is harmless, but…

❑ …could have done anything user could do

  o   Delete files, download files, launch apps, etc.

Trojan in Android: App Repackaging Attack

- One of the most popular "modern" trojan horse...
  - Create a malicious attack code fragment
  - Download popular app and then decompile
  - Insert attack code fragment and distribute
- Don't download Android app from untrusted resource!!
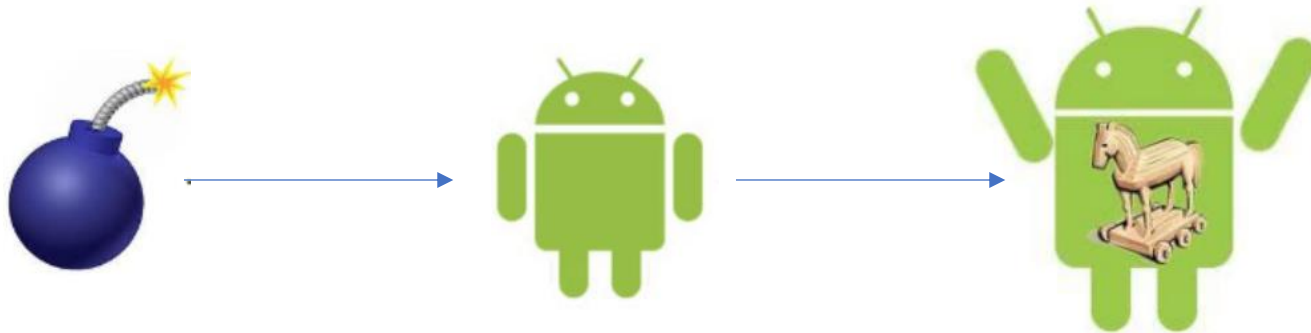
Backdoor

- Secretly bypass normal authentication in a computer system
  - Let malicious users to be remotely/locally access the computing system without being authenticated.

```
LDR          R0, =aSCtUUnSSipSDip ; ">>> %s(ct=%u, un='%s',
LDR          R1, =aAuth_admin_int ; "auth_admin_internal"
BL           sub_558F74

             ; CODE XREF: auth_admin_internal+2C↑j
ADD          R0, R5, #0x44
LDR          R1, =aSUnSU ; "<<< %s(un='%s') = %u"
BL           strcmp
CMP          R0, #0
BNE          loc_13DC78
MOV          R0, #0xFFFFFFFD
LDMDB        R11, {R4-R8,R11,SP,PC}
```

Local password authentication

Backdoor listens on port ABC

Network

Remote access

"Reflections on Trusting Trust"

- The "root trust"
  - Ken Thompson on his 1984, Turing Award Lecture
    - Pioneer; (one) creator of UNIX and C language
  - *To what extent should one trust a statement that a program is free of Trojan horses? ← here "Trajan horses" and "backdoor" is inter-changeable.*

- Malicious compiler yields software with backdoor.
  - Happens in Apple XcodeGhost in 2015.
  - And not just compilers…

- "developers looked for local copies of Xcode"

Crypto Ransomware

Malware Detection

- Common detection methods
  - Static signature detection
  - Dynamic behavior detection
  - *Malware clustering (not a detection methods, but can speed up the detection a lot)*
- We briefly discuss each of these
  - And consider advantages…
  - …and disadvantages

Signature Detection

- A **signature** may be a string of bits in exe
  - Might also use wildcards, hash values, etc.
- For example, W32/Beast virus has signature

  83EB 0274 EB0E 740A 81EB 0301 0000
  - That is, this string of bits appears in virus
- We can search for this signature in all files
- If string found, have we found W32/Beast?
  - Very likely, since software is not "random" sequences of bytes.

Ransomware Detection

- Lightweight symmetric key crypto is used to encrypt victim's data
  - TEA; twoflish; blowfish; …

TEA Encryption

Assuming 32 rounds:

$(K[0], K[1], K[2], K[3])$ = 128 bit key

(L,R) = plaintext (64-bit block)

delta = 0x9e3779b9 ← Magic number!

sum = 0

for i = 1 to 32

   sum += delta

   L += ((R<<4)+K[0]) ⊕ (R+sum) ⊕ ((R>>5)+K[1])

   R += ((L<<4)+K[2]) ⊕ (L+sum) ⊕ ((L>>5)+K[3])

ciphertext = (L,R)

```
000005c0: 0055 4889 e574 0c48 8b3d 3a0a 2000 e80d   .UH..t.H.=:. ...
000005d0: ffff ffe8 48ff ffff c605 310a 2000 015d   ....H.....1. ..]
000005e0: c30f 1f80 0000 0000 f3c3 660f 1f44 0000   ..........f..D..
000005f0: 5548 89e5 5de9 66ff ffff 5548 89e5 c745   UH..].f...UH...E
00000600: fcb9 7937 9e90 5dc3 0f1f 8400 0000 0000   ..y7..].........
00000610: 4157 4156 4989 d741 5541 544c 8d25 ce07   AWAVI..AUATL.%..
00000620: 2000 5548 8d2d ce07 2000 5341 89fd 4989    .UH.-.. .SA..I.
00000630: f64c 29e5 4883 ec08 48c1 fd03 e877 feff   .L).H...H....w..
00000640: ff48 85ed 7420 31db 0f1f 8400 0000 0000   .H..t 1.........
```

Pattern matching delta

Static Signature Detection

- Advantages
  - Effective on "ordinary" malware
  - Minimal burden for users/administrators

- Disadvantages
  - Signature file can be **large** (10s of thousands)…
  - …making scanning slow
  - Signature files must be kept up to date
  - ***Cannot detect viruses if no specific signature is on hand***
  - Cannot detect some advanced types of malware

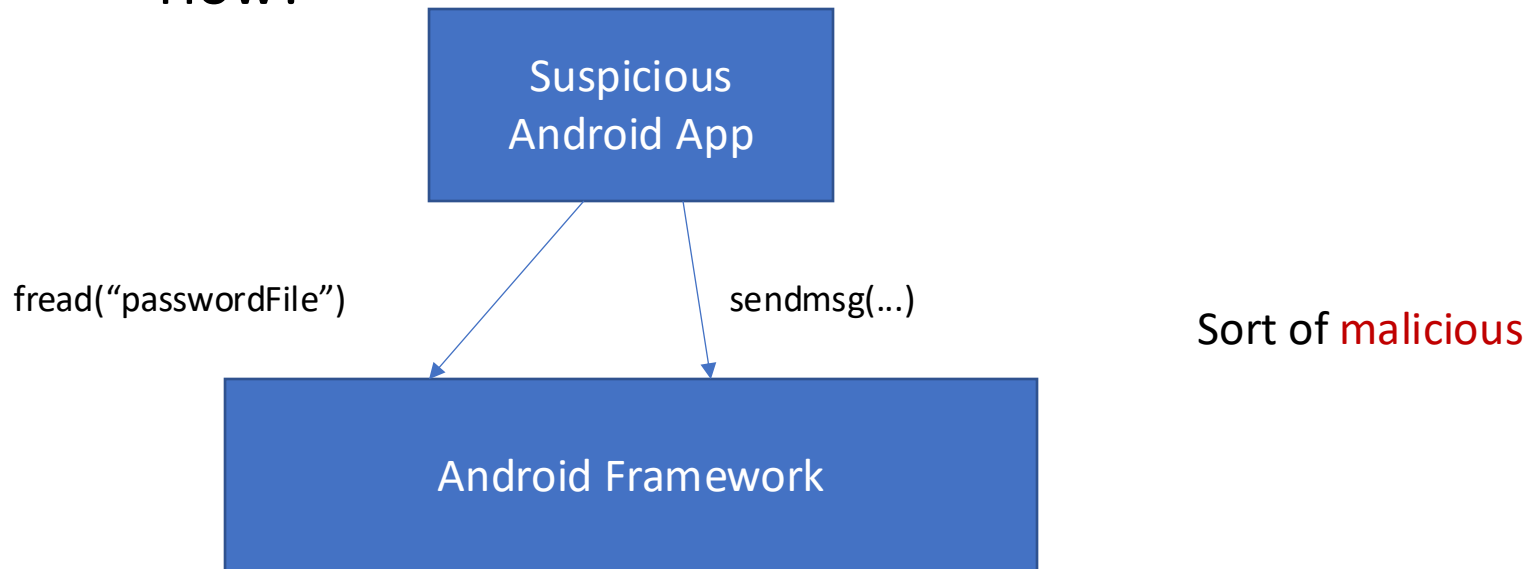- The most popular detection method
  - And also the easiest

Dynamic Behavior Detection

- Monitor system for anything "unusual" or "virus-like" or "potentially malicious" or …

- Examples of "unusual" things
  - Files access in some unexpected way
  - System misbehaves in some way
  - Unexpected network activity

- But we must first define "normal" behavior
  - And normal can (and must) change over time

Malware in Android
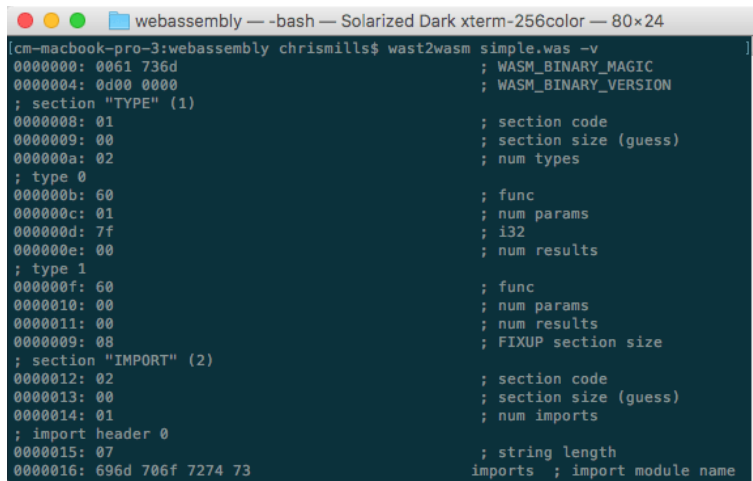
- Log the system call sequences as a way to reflect "malicious" behaviors.
  - How?

Suspicious
Android App

fread("passwordFile")

sendmsg(...)

Sort of malicious

Android Framework

However, it's difficult to figure out the content being sent out with sendmsg => we will talk about that in security static analysis.

Malware in-browser mining

- Stealthily mining bitcoin in your browsers in the era of webassembly
  - But we pinpoint a large amount of "mining" operations during runtime (crypto hashing computation).

Webassembly

Practical Issues

- But, how do we actually get the "behavior" of a suspicious software?
  - Wait until damages happened?
  - Do it earlier but how to run a malicious software, observe its "malicious behavior" without being damaged?

- Isolation the execution of suspicious software!
  - Virtual machine
  - Sandbox
  - Hardware supports

- Then, can malware evade such detection?
  - Yes! Check if you are in a virtual environment
  - "sandbox evading"

The Matrix's Red Pill or Blue Pill

Dynamic Behavior Detection

- Advantages
  - Chance of detecting unknown malware with no static signatures (e.g., a hash) on hand

- Disadvantages
  - Accuracy?
    - Attacker can make "abnormal" behavior looks normal (being slow to attack)
  - Usually combine with another method (e.g., signature detection)

Malware Clustering

- People has accumulated a large amount of "known" malware samples
  - E.g., VirusTotal, the largest malware database
  - You can even get a membership of this database
- Then, given you a "unknown" software
  - Instead of knowing nothing and analyzing from scratch
  - We try to find whether it is similar to existing malware families.
  - Speedup when analyzing millions of unknown samples..

Malware Family

- People has accumulated a large amount of "known" malware samples
  - And many malwares are written by the same author or using the same malware toolkit.

TABLE III.   12 MOST FREQUENT MALWARE FAMILIES

The family name from Kaspersky	Counts
Backdoor.Win32.Hlux	27
HEUR:Trojan-Downloader.Win32.Generic	41
HEUR:Trojan.Win32.Generic	578
HEUR:Trojan.Win32.Invader	53
Packed.Win32.Tpyn	20
Trojan-Clicker.Win32.Agent	39
Trojan-Downloader.Win32.Agent	41
Trojan-Dropper.Win32.Injector	18
Trojan.Win32.Agent	196
Trojan.Win32.AntiFW	1480
Trojan.Win32.Buzus	46
Trojan.Win32.Inject	28

Malware Clustering

- Reuse software engineering/data mining techniques to identify the similarity of suspicious software
  - Similarity; clustering; embedding…

Future of Malware

- Recent trends
  - Encrypted, polymorphic, metamorphic malware
  - Obfuscation/diversification (talk later)

- The future
  - It has become an "ecosystem" with steady revenue
  - Malware development as a "software engineering practice"..

The endless arms race: encrypted malware

- Malware writers know **signature detection** used

- So, how to evade signature detection?

- Encryption is a common approach
    - Ciphertext looks like random bits
    - Different key, then different "random" bits
    - So, different copies have no common signature

- Encryption often used in malware today

Encrypted Malware

- How to detect encrypted malware?

- Scan for the decryptor code
  - More-or-less standard signature detection

- Dump code content during runtime
  - Because the malware needs to decrypt itself before executing.

Metamorphic Malware

- A metamorphic malware mutates before infecting a new system

- Such a malware can, in principle, evade signature-based detection

- Mutated malware must function the same
  - And be "different enough" to avoid detection

- Detection is a bit more difficult
  - How ?

Metamorphic Malware

- One approach to metamorphic replication…

  - The malware disassemble itself

  - Random variations inserted into code (permute the code, insert dead code, etc.) ← we will talk about that.

  - Assemble the resulting code

- Result is a malware with same functionality as original, but different signature