Malware Detection

Prof. Ravi Sandhu
Executive Director and Endowed Chair

Lecture 14

ravi.utsa@gmail.com
www.profsandhu.com
Virus detection is undecidable

Anti-virus (more generally anti-malware) is a great business model
- Need regular updates
- Infinite supply of new malware

Malware can be stealthy
Malware can be really stealthy
Malware Detection Techniques

Signature-based
- static
- dynamic
- hybrid

Anomaly-based
- static
- dynamic
- hybrid

Specification-based
- static
- dynamic
- hybrid

Nwokedi Idika and Aditya Mathur, A Survey of Malware Detection Techniques, Purdue University, Feb 2007.
Malware Detection Techniques

Misuse Detection

Signature-based
- static
- dynamic
- hybrid

Anomaly-based
- static
- dynamic
- hybrid

Behavior-Based Detection

Specification-based
- static
- dynamic
- hybrid
Signature Limitations

$U = \text{set of all malicious behavior}$

$S = \text{set of all known signatures}$

S needs regular updates
Anomaly Based

Training Phase
  - Infer patterns

Detection Phase
  - Infer specifications
**Anomaly Based Limitations**

\[ A = \text{set of all behaviors} \]
\[ V = \text{set of all valid behaviors} \]
\[ V_{\text{approx}} = \text{approximation to } V \]

Blue area is false positives
If white area extends outside blue area we have false negatives

Nwokedi Idika and Aditya Mathur, A Survey of Malware Detection Techniques, Purdue University, Feb 2007.
Defeat signature-based detection

- Encrypted malware
- Polymorphic malware
- Metamorphic malware

Rootkit can misrepresent the existence or content of executable files
Encrypted Malware

execute malware

propagate malware

Encrypted Main Body | Key | Decryptor

Cleartext Main Body

Encrypted Main Body | Key’ | Decryptor
Encrypted Malware

Encrypt Malware

Key

Decrypt Malware

execute malware

Cleartext Main Body

propagate malware

reveals signature
Polymorphic Malware

- Encrypted Main Body
- Key
- Decryptor
- Cleartext Main Body

execute
malware

propagate
malware

- Encrypted Main Body
- Key'
- Obfuscated Decryptor
Polymorphic Malware

execute malware

propagate malware

no signature

encrypted main body

key

decryptor

cleartext main body

encrypted main body

key’

obfuscated decryptor

World-Leading Research with Real-World Impact!
Polymorphic Malware

- Encrypted Main Body
- Key
- Decryptor

execute malware

- Cleartext Main Body

propagate malware

no signature

- Encrypted Main Body
- Key’
- Obfuscated Decryptor

Execute in a sandbox and detect the signature after decryption

© Ravi Sandhu

World-Leading Research with Real-World Impact!
Polymorphic Malware

- Encrypted Main Body
- Key
- Decryptor
- Cleartext Main Body

Execute malware

propagate malware

no signature

Execute in a sandbox and detect the signature after decryption

Mutation Engines automate this construction
Metamorphic Malware

Original Main Body

propagate malware

execute malware

Obfuscated Main Body

propagate malware

execute malware

Obfuscated Main Body

Obfuscated Main Body

execute malware

Obfuscated Main Body

no signature
Obfuscation Techniques

- Dead-Code Insertion
- Register Reassignment
- Subroutine Reordering
- Instruction substitution
- Code transposition
- Code Integration
Really Stealthy Malware

- Not visible in source code
- Reappears in binary code due to malware infected compiler
- In theory could reappear in binary code due to other components in binary execution workflow
  - Loader
  - Linker
  - OS
  - BIOS

Malicious Compiler Inserts a Backdoor

OS Login module

Malicious Compiler Binary

Infected Login Binary
Malicious Compiler Inserts a Backdoor

Assumption:
Malicious behavior cannot be detected in binary, but may be detectable in compiler source
Self-Compiler

Compiler source for language L

Compiler binary for language L

Compiler binary for language L
Malicious Self-Compiler in Binary and Source

- Malicious Compiler source for language L
- Compiler binary for language L
- Malicious Compiler binary for language L
Malicious Self-Compiler in Binary and Source

Source code analysis will reveal malicious behavior

- Malicious Compiler source for language L
- Compiler binary for language L
- Malicious Compiler binary for language L
Source code analysis will reveal doubly malicious behavior.
Doubly Malicious Compiler Binary Behavior

OS Login module

Doubly Malicious Compiler binary for language L

Infected Login Binary

Compiler source for language L

Doubly Malicious Compiler binary for language L

Doubly Malicious Compiler binary for language L
Doubly Malicious Compiler Binary Behavior

OS Login module

Doubly Malicious Compiler binary for language L

Infected Login Binary

Compiler source for language L

Doubly Malicious Compiler binary for language L

Doubly Malicious Compiler binary for language L

No trace of malicious behavior in source code
Malicious Self-Compiler in Binary but not in Source

Compiler source for language L

Malicious Compiler binary for language L

Malicious Compiler binary for language L

partial countermeasure