Malware 3: Analysis

Chengyu Song

Slides modified from Vern Paxson and Dawn Song
Administrivia

- Lab1
  - Feedback
Malware detection

- Static signature based approach
  - Countermeasures from malware authors
- Dynamic behavior based approach
  - Countermeasures from malware authors
- Network based approach
  - Worm detection and botnet take down
Malware analysis

- To answer following questions
  - Is this piece of software a malware?
  - If so, what does the malware do?
    - Interesting behaviors (e.g., detection avoidance)
    - Information for repair/mitigation/takedown
    - Information about the business model
Static analysis

• Static reverse engineering
  • Disassemble, read the code, like in the lab
  • Would this work?
    • Obfuscation
    • Auto unpacking
Dynamic analysis

- Execute the malware and observe its behaviors
- Challenges
  - How to contain/recover from damages?
  - How to trigger behaviors?
Sandboxes

- A (usually) virtualized execution environment to confine host damages
  - Emulators
  - OS-level sandboxes
  - Virtual machines
Arm race

- Countermeasures from malware authors
  - Is there a way to detect you're in a virtualized environment?
    - Instructions
    - OS environment
    - Network environment
  - If we know how malware detects, can we always fix?
State-of-the-art

• Bare metal analysis platform
  • How to recover?
• Countermeasures?
  • Environment-binding malware
Okay, suppose we have a good dynamic analysis environment, how do we know what kind of behaviors the analysis target does?

Behaviors
  - Coarse-grained behaviors: OS-level behaviors
  - Fine-grained behaviors: function-level behaviors
OS-level monitoring

- OS refresh
  - Processes are isolated by OS
  - Modifications have to be done through system calls
- System call monitoring
  - Introspection
Traps and pitfalls

• Tal Garfinkel, *Traps and Pitfalls: Practical Problems in System Call Interposition Based Security Tools*
  
  • Incorrect replication/mirroring of OS state
  • Indirect paths
  • **Race conditions**
  • Incorrect subsetting of complex interfaces
  • Side-effects
Fine-grained tracing

• What kind of behaviors **cannot** be revealed at syscall level?
  • Countermeasures!!
    • Mutation engine (polymorphic/metamorphic)
    • Anti-analysis techniques
    • Domain name generation
    • etc
Fine-grained tracing (cont.)

- How?
  - Debugging
  - Emulators -> natively support
  - Hardware support
Triggers

• Malicious behaviors may only be revealed if certain preconditions are satisfied

• How to solve?
  • Decoys: typical targets of malware
  • Forced execution: not always doable
Network behaviors

• What if the malware tries to infect other machines?
  • Local network
  • Internet

• What if the malware tries to connect to C&C server?
  • How can you tell?
  • Allow or forbid?
Honeynet

- Two major components
  - Network decoys -> allow local infection
  - Gateway -> disallow Internet infection
    - Unless in whitelist
Malicious behaviors

- What kind of behaviors would cause the target to be classified as malware?
  - Replication, both locally and through network
  - Compromising the integrity of the OS
    - Autorun, rootkit, backdoor, etc
  - Leak the privacy of the users
  - Connecting to known malicious host or host of bad reputation
  - Monetization channels
    - Send spam, DDoS, premium SMS, AD fraud, fake AV, encryption, etc.
Make it scale

• Due to polymorphic and metamorphic, AV companies may collect millions of unique instances per day, how to make sure they are all analyzed?
  • Automation!!

• Limitations
  • Limited execution time
  • Only detects known malicious behaviors
By the way, how they collect samples?

- Exchange
- Client submissions
- Crawling
- Honeypot (worm-like malware)
- Honeyclient (drive-by downloads)
Infection cleanup

• Once malware detected on a system, how do we get rid of it?
• Restoring/repairing files (registry is also files)
  • Part of what AV companies sell
• Is there any guarantee?
  • What if there is a rootkit?
  • What if there is a bootkit?
  • What if the BIOS/firmware is infected?

“nuke the entire site from orbit. It’s the only way to be sure”
- Aliens
Two types of malware

- Two types of malware
  - Targeted (a.k.a. advanced persistent threat, APT), state-driven, high tech, highly stealthy
  - Large-scale infection, monetization-driven, low tech
- For the second type of malware, the most effective way to stop them is the economical way
  - **Cut their monetization channel**
- But we need to understand how they monetize first!
Understanding the underground economy

- What is their business model?
  - Where does the money come from?
  - How money flows?
- What is the criminal infrastructure?
  - Hosts, DNS provider, payment processor
- Goal: find the weakest link
Example: pay per install (PPI) ecosystem
The walled-garden model

- Why there are only a few malware on iOS devices?
  - How can you monetize on iOS?
  - How can you achieve large infection/installation?
- A healthy ecosystem matters a lot!
For next class ...

• Software security I: memory errors