Chapter 3: A brief introduction to dynamic analysis

DATA SCIENCE IN SECURITY

Introduction

- Static analysis
  - focuses on what malware looks like in file form
  - it doesn’t allow us to observe malware behavior
- dynamic analysis
  - running malware in a safe, contained environment to see how it behaves
  - we can
    - get around common static analysis hurdles, such as packing and obfuscation
    - gain more direct insight into the purpose of a given malware sample

why use Dynamic analysis?

- consider the problem of packed malware
  - We could try to disassemble it using the static analysis tools
    - This is a laborious process
      - we have to find the location of the obfuscated code in the malware file.
      - we have to find the location of the deobfuscation subroutines.
      - we have to figure out how this deobfuscation procedure works
      - Only then could we begin the actual process of reverse engineering the malicious code

why use Dynamic analysis?

- consider the problem of packed malware
  - A simple yet clever alternative to this process
    - execute the malware in a sandbox: a safe, contained environment.
    - allows it to unpack itself as it would when infecting a real target.
    - we can find out
      - what servers a particular malware binary connects to
      - what system configuration parameters it changes
      - what device I/O (input/output) it attempts to perform

Dynamic analysis for Malware Data Science

- reveals what a malware sample does
  - we can compare it to other malware samples.
    - For example
      - It shows what files malware samples write to disk
        - Can connect malware samples that write similar filenames
        - help us categorize malware samples based on common traits.
        - even identifies malware samples that were authored by the same groups or are part of the same campaigns

Dynamic analysis for Malware Data Science

- is useful for building machine learning–based malware detectors
  - For example
    - observing thousands of dynamic analysis logs
    - a machine can learn that
      - msword.exe launching powershell.exe is malicious
      - msword.exe launching Internet Explorer is harmless

Typical Malware Behaviors

- Modifying the file system
  - writing a device driver to disk
  - changing system configuration files
  - adding new programs to the file system
  - modifying registry keys to ensure the program auto-starts
- Modifying the Windows registry to change the system configuration
  - changing firewall settings
- Loading device drivers
  - loading a device driver that records user keystrokes
- Network actions
  - resolving domain names
  - making HTTP requests

Basic tools for Dynamic analysis

- CuckooBox
- Malwr.ee
- Hybrid-analysis

limitations of Basic Dynamic analysis

- malware authors are aware of CuckooBox and other frameworks
- attempt to circumvent them by
  - detecting execution under CuckooBox
  - making malware fail to execute.
- CuckooBox maintainers are aware of this
  - they try to get around attempts by malware to circumvent CuckooBox

limitations of Basic Dynamic analysis

- might not reveal important malware behaviors
  - even without any circumvention attempts
  - Consider a malware that
    - connects back to a remote server upon execution
    - waits for commands to be issued.
      - look for certain kinds of files on the victim host
      - log keystrokes
      - or turn on the webcam.
    - none of these malicious behaviors will be revealed if remote server
      - sends no commands
      - or is no longer up