Foundstone
Professional Services A DIVISION OF McAFEE

Automating file analysis

Pär Österberg Medina

6/27/2012

Agenda

- Automating file analysis
- Introducing autoMate
  - Identifying a file
  - Analyzing the file based of its file type
  - Review the report

Who am I

- Pär Österberg Medina
  - McAfee and Foundstone Professional Service
  - Worked 8 years for the Swedish CERT
  - Background in Ethical Hacking

  - http://parosterbergmedina.blogspot.com/
  - https://github.com/parosterbergmedina

  - http://blog.opensecurityresearch.com/

Forensic analysis of files

- Challenges in forensic analysis of files
  - A case may include more than 1000000 files
    - We often need to be very skilled and lucky to find the needle in the haystack
  - Not clear how to identify which files to analyze
  - Difficulties to remove files that are already known
  - No framework for automating the process
    - Unnecessary time is spent repeating the same commands over and over again

Automated file analysis

- autoMate - Design specification
  - Identify the file and classify it based of the type
    - Similar to ‘sorter’ from TSK
  - Classify the file based on matches in hash db
  - Analyze the file based of type and hash match
    - Plug-in framework to support calls to external program
  - Review the report and result of the analysis

Identifying a file

- How to identify and classify a file?
  - Looking at the file extension
    - Error prone and easy to defeat
  - Opening the file and looking at the content
    - Native Unix utility ‘file’ - using magic database
    - TrID by Marco Pontello - has it’s own database

File identification

- AutoMate - file identification
  - Uses output from external file identifier
    - Currently both ‘file’ and ‘TrID’
  - Has a database containing which output should match which file type
    - Based of regular expressions
  - Will report on any extension miss-matches found
    - A .pdf file named .tmp is however still a .pdf file

Creating a filemap

- Step one is to generate the ‘filemap’ file
  - Run the external commands on all the files
    - Processes are forked for improved speed
  - Result is stored in a file called ‘filemap’

Format of the filemap

- Internal format loosely based of xml
  - The filemap contains;
    - Path to the file on disk
    - The output of the commands
    - Size of the file

Classifying the files

- Step two is to classify the files based of which file type it belongs to
  - autoMate uses a config file to determine the file type
  - The result is multiple '.map' files containing files broken down in individual file types

Analyzing the files

- Step three is to analyze the files based of;
  - If the file is know to us before - hash lookup
    - KnownBad, KnownGood, KnownUsed
    - Supports the use of hashmap databases
  - The previous classification done in step two
    - Driven by rule sets for individual file types specifying which external commands should be run

autoMate - rules

- Rules that dictates what to do for which file types
  - Using main category and sub categories
    - exe.pe32

External commands

- autoMate uses external commands to produce various result plug-in framework with calls to external commands
  - Will load all perl modules in a specified directory
  - Passes arguments to the modules
    - File path
    - Size
    - Checksums

autoMate - Modules

- MultiAV.pm
  - Module to scan the file with multiple anti-virus engines and look for a detected virus
- OfficeMalScaner.pm
  - Look for a malicious OLE document using OfficeMalScaner

autoMate - report

- Report generator picks up the result of the analysis
  - stand alone HTML report

autoMate - future

- More modules
  - Modules need to be written and incorporated
- Testing and hunt for bugs
  - More testing needs to be done in other environments

This is the end

- Questions?