Automated Malware Similarity Analysis

Daniel Raygoza
General Dynamics Advanced Information Systems

Disclaimers

(they seem like a good idea)

- This project and presentation are my own personal work, and they are not a product of (or approved by) General Dynamics (GD) or its customers.

- Anything I say is my own opinion, not GD's or its customers.

Problem

- In-depth malware analysis is expensive
- Difficult to reduce duplicated effort as teams grow
- Anything not automated would probably counter benefits

Target Niche

- Teams that:
  - Routinely unpack and analyze incoming samples
  - Want to avoid duplicate analysis
  - Looking for similarity links

Initial Idea

- Use IDA for auto-analysis of all samples
- Break samples into functions
- Calculate a fuzzy hash of all functions
- Calculate the similarity between all functions in the entire system
- Weight and aggregate similarity scores between a given sample and all other samples in the system

Fuzzy Hashing In A Few Seconds

- Fuzzy hashing, created by Jesse Kornblum, based on spamsum by Andrew Tridgell
- The output hash value is tolerant of changes in the input
- Hash values compared against each other to compute similarity

On Fuzzy Hashing

- Full binary fuzzy hashing of malware doesn't perform well in many cases (packing, reordering of functions, partial code reuse, etc).

- Small files tend to be heavy on structure and light on code, causes a lot of mismatches.

- Many issues are addressed by using fuzzy hashing at the function level on unpacked binaries, though it's still not perfect.

- If we can expect similarity between any extracted byte-streams to be meaningful, we can apply fuzzy hashing effectively.

Existing Research

- Automated malware similarity analysis definitely isn't new
- There are many published papers on malware similarity analysis using a variety of techniques, some of which seem highly effective
- Very few have freely available implementations
- The ideas are good, but we need to way to practically apply them

Non-Free Tools

- Zynamics BinDiff/VXClass
- HBGary DDNA
- Various private tools

Refined Idea

- Create an open source framework to support the implementation of a variety of similarity scoring systems
- Begin with fuzzy hashing, move on to other more complex algorithms
- Make all of the similarity data available in abstract form, allowing for custom visualization

Limitations

- Automation doesn't include unpacking, I'm not nearly awesome enough to write a generic unpacker. However it should be easy enough to integrate your organizations own unpacker(s)

- Fuzzy hashing has obvious limitations, but the implementation of other algorithms should address this

- Currently relies on IDA for disassembly

- Like virtually any other implementation it can be subverted by malware authors

Limitations

- The open source framework would not be a general purpose malware classification or identification system (you may want to check out Yara)
- Not 100% automated, lacks a general purpose unpacker

Implementation

- Python (ingest, backend)
- MySQL
- PHP (frontend)
- Hopefully OS agnostic, but developed on Linux

Components

- Ingest - Uses abstracted disassembler to retrieve function blobs. Calls plugins to retrieve additional information to be stored with the sample (PE information, PEiD, strings, disassembly, decompilation, strings, etc). Packages data to be sent to database.

- Backend - Takes the general purpose data packaged by the Ingest module and applies all of the relevant similarity algorithms (implemented as plugins), stores similarity results in database.

- Frontend - Makes the database contents available via XML.

Other Ideas

- Support ingesting IDB files directly, allowing analysts to fix up the IDB prior to analysis

- Selectively null out operands that are likely to vary between instances of code, then feed this data to the similarity algorithms

Work To Be Done

- Implement additional similarity algorithms
- Find the bugs that certainly exist
- Create a prettier front-end
- Better documentation
- Installer

Some Early Results

- Stats to come...

Turbo Tool Demo...

- ... crossing fingers

Getting It

- Project homepage at http://www.raygoza.net/fuzzball/

Thank You

- Kevan, Mike, Joe, General Dynamics, and others.
- Smarter individuals from whom I've stolen/reused ideas.

References

- Yara: http://code.google.com/p/yara-project/
- Fuzzy Hashing – Jesse Kornblum: http://dfrws.org/2006/proceedings/12-Kornblum-pres.pdf
- Fuzzy Clarity – Digital Ninja: http://digitalninjitsu.com/downloads/Fuzzy_Clarity_rev1.pdf
- Spamsum – Andrew Tridgell: http://digitalninjitsu.com/downloads/Fuzzy_Clarity_rev1.pdf
- ssdeep – Jesse Kornblum: http://ssdeep.sourceforge.net/