Analyzing Modern Trends in Malware

Daniel Sun, Joseph Torres, Katherine Wang, Jessica Zhu

March 2019

1 Background

Malware is any type of ”malicious software” that is intentionally designed to cause damage to computer systems. The mechanisms and intentions behind malware construction and distribution vary greatly (1). While some types of malware aim to cause as much damage as possible, others have ulterior motives such as keyloggers which gain a user's credentials (2), rogueware which scams the user (3), or ransomware which holds a user's data hostage (4). This is by no means an exhaustive list, especially since some malware can be classified under more than one category.

The nature of malware has shifted significantly over time. Early malware was often designed with damage more in mind than financial gain. As time has passed, attack vectors have become increasingly cunning in order to manipulate victims. Recently, malware developers have been particularly interested in creating programs which either fool or scare a user into giving sensitive information and money. Although some malware is simple, others are incredibly complex and target flaws in multiple systems to accomplish some purpose, such as Conficker (5) and Stuxnet (6).

Malware analysis is the technique of analyzing malicious programs. This can be done statically (without executing the suspect binary) or dynamically (running it in a sandboxed environment). By performing analysis on malware, exploits that may be taken advantage of can be reverse engineered or examined. This can result in patches which render the malware ineffective, as well as insight into other vulnerabilities and exploits that can be fixed before potential zero-day attacks (7).

2 Objectives and Goals

The question remains, then, what direction malware in the present day is currently taking. Are there particular trends or is there somewhat of a dry spell due to the security of modern systems? We aim to address this question by sampling the present space of malware and determining key insights. The main goals of our project are:

- Perform a meta-analysis of existing trends in malware from the past up until the present (we currently intend to scope this down to the past decade, but can increase or decrease this as needed). We plan on collecting various pieces of data such as: origin country, targeted platforms, source languages, attack vectors, payloads, and vulnerabilities exploited. We are open to expanding and reducing the set of factors to record, and may do so as we perform these analyses.

- Examine a stream of blacklisted URLs in order to acquire malicious programs and content for present analysis. There exist numerous domains that collect blacklisted URLs that we can peruse and scrape for metadata (e.g. `https://zeltser.com/malicious-ip-blocklists/`). Over the course of the project, we can collect as much of this data as possible and use it to compare to past trends.

- In addition to examining metadata, we intend on working with a randomly selected subset of modern malware. We hope that through static and dynamic analysis on a sandboxed virtual machine we can extract additional, meaningful information.

- Using the data from our meta-analysis and present analysis, we hope to make meaningful predictions about the current direction and future of malware.

Should our procedure prove successful, we will have a holistic and comprehensive set of data about recent malware and modern malware alike. This information will include useful meta-data as well as detailed information about the malware. In turn, we can make well-supported predictions about the direction of malware and grant well-advised precautions as a result. If we are lucky, we may also be able to spot potential CVEs and report them.

3 Prior Work

Malware analysis is a major domain in computer security research. Naturally, there has been plenty of work done in this space. However, this analysis is imperfect and requires collaboration from numerous individuals across the world in order to create a better security framework. Despite increasingly complex security systems and a greater consideration of security, the number of malware is actually higher than ever (8). Though such analyses on previous trends (or then modern trends) cover large amounts of data, they lack descriptive features which we are interested in. Furthermore, the number of studies out there are plentiful which makes this area an excellent candidate for meta-analysis. Previous meta-analyses are growing stale, and can contribute to a more comprehensive modern one (9). We hope to perform our analysis in a way which makes it easily repeatble and extensible for subsequent usage.

Furthermore, existing research has taken advantage of various tools in order to create declarative categories and characteristics for malware (10). Such categories and insights are extracted using a certain subset of tools, which we hope to expand by considering new technologies such as Ghidra which has only recently been open-sourced.

Plenty of cyber-security firms create predictions about malware for the each upcoming year (11). Such predictions are extensive and contain various information, and our work can validate these predictions (through our own predictions). Furthermore, as we are performing actual analysis, we can corroborate our predictions (which these official predictions may sometimes lack).

4 Setup

Malware analysis can be a dangerous form of security research. Therefore, we intend to keep two principles in mind while executing this project:

1. As the programs we will work with are going to be drawn from blacklisted URLs, we need not exercise caution regarding permissions when reverse engineering or tampering with source binaries. We should not run into any legal issues by experimenting with malicious software; if we ever run into suspected malware on what appears to be a legitamate platform, we will simply discard that from our sample (and perhaps note such exceptions in an additional section).

2. The nature of malware is ambiguous, especially for new malware which has not yet been analyzed. Although severe zero-day vulnerabilities are rare, we will assume that each program we run into has such capabilities. Therefore, we will only analyze code on an isolated machine through a virtual machine (VM).

To prevent our personal computers from being compromised with dangerous software, we will setup multiple VMs to isolate the malware. Furthermore, we will use a brand new VM for each program we test to prevent confounding interactions. There are numerous guides for setting up secure malware analysis labs (e.g. https://medium.com/@xNymia/malware-analysis-first-steps-creating-your-lab-21b769fb2a64), so dealing with this should not be an issue.

5 Methods

We plan to use several techniques in order to analyze malware. We break down our intended approach as twofold:

1. Scrape, analyze, and visualize various surface characteristics of malware from the last decade or so (as mentioned previously, this period is tentative and adjustable) up until and including the present.
2. Perform static and dynamic malware analysis on a subset of malware.

5.1 Malware Data Scraping

We intend to perform simple statistical analysis and visualization by backlogging malware and taking note of a handful of characteristics:

- Origin Country
- Targeted Platform (e.g. operating system, browser, application, etc.)
- Source Vector (e.g. executable, script, etc.)
- Recency (so we can cluster based on date)

Furthermore, we can append this information with anything that might come up from a procedural static analysis (discussed later) if the malware has been documented (which we can check by verifying at websites like https://www.virustotal.com/#/home/upload).

We can collect malware by simply using our sources for modern malware and looking at old records (these databases tend to date decently far back). To compensate for malware which has been taken down, we can also aggregate findings from previous researchers and collect those findings in our meta-analysis.

5.2 Malware Analysis

For more modern malware, we may not yet have in-depth details from the larger entities which perform such analyses. Furthermore, such entities may not hold consensus on whether or not a particular file is malicious. To break such ties, we intend to perform some of our own static and dynamic analysis on very recent samples.

5.2.1 Static Analysis

For static analysis, we can start by checking VirusTotal for basic information. From there, we can utilize basic techniques like those outlined in the following article: https://resources.infosecinstitute.com/malware-analysis-basics-static-analysis/. We also hope to take a look into some reverse engineering techniques, if time permits. In particular, we hope to use Ghidra (12), an open-source reverse engineering system released by the NSA in March 2019. As this is fairly new, we may wish to ramp up and include an auxiliary section which discusses its relevance and utility (testing on previously reverse engineered malware stored in various repositories (13)).

5.2.2 Dynamic Analysis

For dynamic analysis, we can follow basic protocols like those outlined in: https://resources.infosecinstitute.com/malware-analysis-basic-dynamic-techniques/. We will perform all dynamic and static analysis in a VM in order to prevent accidental infection of not only the host system, but those on the network.

6 Schedule and Milestones

Below is our projected schedule and project milestones:

1. **April 5th**:
    - Set up any necessary virtual machines and tools that we want to use in static and dynamic analysis. This might include testing out such tools to make sure we know they work and how they work.
    - We will also begin collecting historic data and data from present URLs.
2. **April 12th**:
    - Have a simple script for scraping data from all of the different blacklist URL aggregate websites. This allows us to categorize the data into respective categories.
    - Begin clustering and analysis of historic data combined with present data.
    - Develop a manner for randomly sampling malware from the blacklists. Begin using static and dynamic analysis programs to acquire preliminary data.
3. **April 19th**:
    - Continue aggregating historic data of malware and scraping modern malware data.
    - Develop a prototype of a pipeline which translates data online to locally stored data which contributes to some kind of visualization (e.g. graphs, simple statistics, etc.).
    - Continue process of static and dynamic analysis of randomly sampled malware. If we exhaust our methods early, we can sample more malware.
    - Consider creating a prototype of an automated pipeline which performs this analysis on sampled malware.
    - Begin experimenting with Ghidra.
4. **April 26th**:

- Continue aggregating historic data of malware and scraping modern malware data.
- Continue working on pipeline which translates malware data to visualization (and continue developing ways to present and interpret the data in novel ways).
- Compare existing results of static and dynamic malware analysis to those generated by Ghidra. Continue using Ghidra on randomly sampled malware and begin grouping the malware on vectors and targeted platforms.

5. **May 3rd**:
    - Continue aggregating historic data of malware and scraping modern malware data.
    - Begin wrapping up malware URL to visualization pipeline. Create extensions to this pipeline which can process the data generated from the malware analysis that is occurring in parallel.
    - Continue static and dynamic analysis, especially usage with Ghidra.
    - Begin using all existing data to create predictions and extrapolations.
6. **May 10th**:
    - Finalize aggregation of historic data of malware and scraping of modern malware data.
    - Finalize pipeline and any extensions needed to handle malware analysis data.
    - Wrap up all malware analysis and information gathered from it. Communicate this information to our data processing and visualization pipeline.
    - Finalize extrapolations and predictions based on the gathered data.
7. **May 15th**:
    - Finalize the paper and any code or methods that were used.
    - Tear down malware analysis lab environments.

References

[1] N. DuPaul, “Common malware types: Cybersecurity 101.” [Online]. Available: https://www.veracode.com/blog/2012/10/common-malware-types-cybersecurity-101

[2] Wikipedia, “Keystroke logging.” [Online]. Available: https://en.wikipedia.org/wiki/Keystroke_logging

[3] ——, “Rogue security software.” [Online]. Available: https://en.wikipedia.org/wiki/Rogue_security_software

[4] ——, “Ransomware.” [Online]. Available: https://en.wikipedia.org/wiki/Ransomware

[5] Microsoft, “Virus alert about the win32/conficker worm.” [Online]. Available: https://support.microsoft.com/en-us/help/962007/virus-alert-about-the-win32-conficker-worm

[6] K. Zetter, “An unprecedented look at stuxnet, the world’s first digital weapon.” [Online]. Available: https://www.wired.com/2014/11/countdown-to-zero-day-stuxnet/

[7] Symantec, “Zero-day vulnerability: What it is, and how it works.” [Online]. Available: https://us.norton.com/internetsecurity-emerging-threats-how-do-zero-day-vulnerabilities-work-30sectech.html

[8] AV-TEST, “Malware.” [Online]. Available: https://www.av-test.org/en/statistics/malware/

[9] A. Usanov, “Assessing cyber security: A meta-analysis of threats, trends, and responses to cyber attacks.” [Online]. Available: https://www.researchgate.net/publication/319677972_Assessing_Cyber_Security_A_Meta-analysis_of_Threats_Trends_and_Responses_to_Cyber_Attacks

[10] NCCIC, “Malware trends.” [Online]. Available: https://ics-cert.us-cert.gov/sites/default/files/documents/NCCIC_ICS-CERT_AAL_Malware_Trends_Paper_S508C.pdf

[11] ZDNet, “Cybercrime and malware, 2019 predictions.” [Online]. Available: https://www.zdnet.com/pictures/cybercrime-and-malware-2019-predictions/

[12] NSA, “Ghidra software reverse engineering framework github.” [Online]. Available: https://github.com/NationalSecurityAgency/ghidra

[13] Y. Nativ, “thezoo github.” [Online]. Available: https://github.com/ytisf/theZoo