Writeup 1

Kevin Turkington

I. Malware Basics

Over this weeks lectures we learned the various differences in malware mainly pertaining to the who, when, where, why, and how. In addition to the tools, roles, and methods of an anti-malware researcher like the use of sacrificial goats (isolated machines used for malware research), as well as tools like processMonitor and flypaper. To begin malicious software can be broken down into four different categories:

- Viruses (Worms, parasitic programs)
- Trojans (backdoors, keylogging, bots)
- Malware
- Potentially Unwanted programs (typically Ad-ware, spyware, or unwanted tools)

In general the motivations behind creating malicious software is most common for destruction, spying (corporate espionage), political gain, and financial gain. For example this weeks lecturer provided samples containing malware intended for embedding windows xp machines (credit card readers) with the sole purpose of stealing unsuspected victums at the ATM or a local store. Furthermore we have encountered samples designed for keylogging utilizing an insider for installation and retrieval of the malware itself. Aside from the aforementioned, attackers infection vectors can also be dropped USB sticks outside of targets parking lots, users downloading unapproved software, as well as embedding macros within commonly used office documents. However over time these attack vectors have increasingly focused on unsuspecting employees and general users. Resulting in people themselves being the biggest vulnerability in any system.

II. Malware Analysis

Within the Analysis section of this weeks lecture we learned about a collection of web and non web based tools that can be extremely helpful for malware analysis. For example VirusTotal can be used for the analysis of suspicious files and if matched this site will return results from a collection of known anti virus scanners and details about the potential threats. Similarly malwr is used for behavioral analysis of malicious software. If these suspicious files contain potentially sensitive information like work specific documents or personal financial, etc. An MD5 hash can be generated and used as a stand in and compared to a database of known malware. To generate an MD5 hash the easiest way would be utilizing known python packages like hashlib. MD5 is a widely used algorithm typically used for cryptography.

```
#MD5 hash signature example
# Note: ~untested~
import hashlib
m = hashlib.md5()

with open('file.txt','r') as file:
    fileContents = file.read()

    m.update(fileContents)
    newSignature = m.digest()

# newSignature can be printed to a file or simply copied and
# pasted into a web interface for searching.
```

Aside from the web interfaces used for malware analysis, we learned how to analyze the affects of malware of a goat machine. to setup a goat the best practice is to use the following programs:

- AntiSpy (used for spoting the malware itself)
- FakeNet (used for spoofing get/post requests)
- Flypaper (keeps programs running and prevents exiting)
- ProcExp (reveals process data to the researcher)
- ProcMon (monitors process actions and generates a log for the researcher.)

With the aforementioned tools a researcher can conducted behavioral analysis following a malicious software actions in real time seeing what files are created and manipulated, as well as any internet traffic the malware is trying to conduct (therefore providing urls to governments to blacklist.)

Once the malware is isolated researchers can use a mixture of decompression and binary analysis tools to look into the source code. For example FileInsight can be used for anything from programming language analysis to targeting strings to decipher possible variables with in the source. This tool also allows researchers to analyze the assembly instructions directly of any given compiled binary or python script.