OREGON STATE UNIVERSITY

CS 373

WINTER 2019

Week 3

Author:
Thomas Noelcke

Instructor:
D. Kevin McGrath

I. Overview

This week we are diving deeper into malware defense. This week we will dive in to topics such as malware defense, infiltration vectors and opportunities where we can defend against attacks. Below I will discuss topics I found interesting during lecture. I will also discuss the lab and what I learned during the lab. Finally, I will list out new terms that I learned this week along with a definition in my own words.

II. Lecture Topics

A. How computers get infected?

We started off the lecture by talking about how people get infected. Often times the point of contact is through Email, IM, Malvertising, poisoned search results. It's also possible to get infected by searching for software that you are trying to get but search results take you some where else. It's also possible that you get a infected through a watering whole. This is where some sort of interesting content targeted at certain people and they go there and look at the material. Physical access is also another way that people can become infected.

B. How to do get the code to run?

Once you have access to the computer how to do you get the code that you want to be run to run? Once common way this can be done with with auto runs. These are programs that are run automatically when you insert some sort of media. Another possible way this might happen is through an installer that the user is installing another piece of software. Document exploits are one of the main ways that code is run on users machines. This is commonly called an exploit. Exploitation has a wide array of different exploits that can be used. Another way this can be done is through a redirect where the exploit is not hosted at parent site but a redirect is used to take the user to the compromised site. Finally, It's also possible to trick the user into running the code you want to be run. This can be done in a variety of ways through a bunch of different sources. Outdated software is also another way in which a machine may become infected.

C. Once we are in what do we do?

Once installed the malware will likely try to replicate or blend in. This is often done by installing something in system 32 or mimicking the svchost process. These programs might also change time stamps so that it doesn't look like it was just installed. Bootkits and Rootkits are another way that malware might try to blend in. This method is more of a way to hide rather than blend in. Malware will also try to persist it self. This can be done by the malware adding it self to the start up list. There are many ways to force a program to run on start up. Side load attacks are another way to exploit a system. This is done by replacing dll's that go with legitimate installers. This causes the installer to call the wrong code. Using the proxy settings to run the user through your proxy instead of the site they intended to go. Ultimately, the goal is to use the malware to collect information like passwords or documents.

D. First Contact and Defense

Now that we know how computers are infected and how attacks are carried it we can start to look at this problem through the lens of what can we do to defend against these attacks? The best thing we can probably do is to prevent first contact with the user. There are many things that your browser already does to try to help with this. For instance the browser has a list of blacklisted domains and sites that will cause a warning to pop up if you try to visit one of these sites. Another way we can help protect the user is to prevent the user from taking actions that they shouldn't be. For instance if the user, such as a teller, shouldn't be allowed to use a USB stick then don't give the user physical access to a drive.

It also helps to polices in place that help prevent the user from doing things they shouldn't be doing. Script blockers are another way that we can prevent the user from being exposed to malicious scripts. It should be known that this may hurt the user experience. Firewalls are another method by which the user can be protected. Its important to know that the tools that are highly effective will lead to poor user experience.

E. The Solution

So if the ways that are effective hurt our users, what do we do to better protect our users with out destroying their experience? The real solution is to use a layered approach. This means that we will use some of the above methods in less extreme forms. This layered method will make it much more difficult for hackers to exploit users. It's also important to note that there are things we can do after first contact to minimize the risk. An example of this is security system might quarantine a file that it may suspect is bad. It's also possible that your security system may just delete a a file that is bad.

F. Behavioral Security

Today, file scanning is one way that we are protecting users. This is by scanning the files and flagging files that are suspicious. This approach does a lot of good for users but some times this is not enough. It is possible to take software and get it to run in memory with out having to physically save a file to disk. This means that the AV scanning with not catch this attack. Another way to catch this attack is to focus on the behavior of these processes. If a process is from a known source and is from a file we pretty much leave it alone. However, If there is code running in memory that does not come from a file we need to pay close attention to what that process is doing and shut it down if it does any thing suspicious.

G. Layers of Security

Below is a list of the layers of security that are employed today to protect users from being attacked.

- Network Firewall
- Network Intrusion Prevention
- Message Reputation
- Network Reputation
- Web Reputation
- Host Firewall
- Host IPS
- Access Control
- Anti-Malware

Its important to note that all of these systems require the users to actually use these systems in order for them to work. One example is ransomware attacks. The most common way to defend against this is to do regular backups. How often do people actually do this? It's important to note that often these systems are often from different vendors and usually don't communicate with each other. This is going to increasingly important as time goes on. All of these systems are content driven and require updates. This means that by the time you get the updates the threat has already evolved.

H. Layers of a Security System

Below is a list of common components of a security system.

- Management Server - This is a server that push updates and manages different parts of the security systems.

- Scanner Core - This the part of the system that is responsible for the actual scanning of files.
- Engine - The engine is responsible for consuming the content.
- Content - This is the content that the system is looking for or other information that allows it to detect a threat.

Below is a list common features that a security suite might provide.

- Traditional File Scanning (OAS, ODS)
- Registry and Cookies
- Cloud Scanning
- memory Scanning
- Scripts
- Heuristics
- Decomposition
- Configuration: Exclusions, sensitivity, reporting, ect.

I. YARA

This is the swiss army knife for malware researchers. This allows you to express different patters as expressions. This allows people to write their own content for the purposes of defending against malware. Strings are expressed as strings as you would expect in most languages. Byte patterns are also used in YARA. There are a verity of different logic operators that allow you to do some different things with the byte patterns.

Good YARA signatures are still generally written by humans. Machine learning is starting to play a part in writing signatures but often times they are still hand crafted. This is because we want to make the signatures small and make sure that they have meaning. We can use hashes of files along with fuzzy hashes to find malicious files. Its also important to note that these signatures can also be applied to memory scans.

J. Software Signing

We also talked about code signing. The purpose of code signing is to verify the binary that you are looking at came from a given source. This doesn't verify that the code isn't malicious. This only verifies that this code was not tampered with between the source sending it and you receiving it. In today's world we are really past using just the signature alone to determine that software is good or bad. We are now on to signature reputation. This means we are looking at who issued the signed software and using the reputation of that entity to determine if that software is worth trusting.

K. automation

There are many reasons why we want to automate some of the AV signature generation and processing of malware samples. When there are millions of samples being generated in a day it's hard not to use automation to process all that information. Some advantages of automation are that the computers make less mistakes, computers can handle more volume and computers can run 24/7 365. Machine learning is currently in use to help determine weather AV rules are good or bad. It is also used to help determine weather or not a new rule should be pushed to the cloud. However, there are also some problems with machine learning. The machine learning approach really fails to deal with the white code rather than the malicious code. The automation may also write cumbersome rules and may cause results that are complicated or that don't make sense. Another problem with automation is that it is out of context. Meaning that if a piece of malware is run in an automated way this is not the real world.

L. cuckoo

This is another tool that allows us to spoof the internet and serve files while closely monitoring what is happening. This is done by running cuckoo on a host machine that then monitors guest machines that then get run through a virtual network run by Cuckoo. The guest machines that are running the malware also report the behavior of the malware to Cuckoo. Below is a list of some items that cuckoo can collect for us.

- Traces of win32 api calls
- Files being created, deleted and downloaded during execution.
- Memory dumps of the malware processes.
- Network traffic and traces.
- screen shots of windows desktop taken during the execution.
- full memory dumps of the machines.

III. Terms

- **First Contact:** The first contact that hacker has with a user that causes them to become infected with some sort of malware.
- **IPS:** Intrusion Prevention System.