Nicolas Barraclough
10/18/2020
CS373 - Defense Against the Dark Arts

Week 3 Write-Up

This week we learned about Malware Defenses. The lecturer described the typical process of how malware attacks a system. This process is divided into 4 parts: First Contact, Local Execution, Establish Presence, and Malicious Activity. The malware could have varying ways in which it follows these four parts of the process, but it will generally follow that progression. For example, in a phishing attack, the malware makes First Contact by the attacker gaining access to the victim's email and sending them a malicious message. The victim then clicks on a link that contains a piece of malware initiating Local Execution. The malware then Establishes its Presence by downloading additional malware or propagating to another system. Then the Malicious Activity commences through the execution of the malware on the intended targets.

Anti-malware software and practices can help the victims defend against the malicious intent of the attacker(s). In my example with the phishing attack, anti-spam filters on the user's email could prevent phishing emails from even entering the user's inbox. There will often be a warning issued by the browser if an untrusted domain attempts to download something on the computer and will ask the user if they are sure they want to continue with the download. These forms of defense would stop the malware from successfully completing the first stage of First Contact. If the malware does happen to reach local execution, anti-malware software on the computer could warn the user that malicious files are present on the machine along with a wide range of anti-virus defense mechanisms that the software could provide.

The lecturer also introduced Yara and pattern-matching signatures. Yara is a tool that can detect patterns inside malware. These patterns are then used to develop specific signatures that can help to identify if the suspected malware falls within a particular set of known malware. Yara can identify specific strings and binary patterns that can relate the respective file to other files which is why it is used to compare against malicious files.

Another useful tool that we learned about was Cuckoo. Researching Cuckoo, I learned that it can accept any suspicious file and Cuckoo will provide a detailed response outlining what the file did when it was executed. This can provide loads of valuable insight to what the malware was attempting to do while it executed. Cuckoo can let the user know if and what files were copied/deleted from the filesystem, as well as provide a full memory and network dump that can be used for analysis. Cuckoo behaves like an isolated environment so it is safe for the malware to execute.