Mofassir Ul Haque Guest Lecture Summary

Josiah Craw
35046080

August 20, 2019

In this guest lecture we covered secure software, vulnerabilities, exploitation, and tools that are used within the industry. First, Mofassir discussed what a vulnerability is, they are usually a design flaw, an implementation bug or an improper implementation. Some examples of these are, the design of the internet, as a design flaw as it was designed before hacking of internet vulnerabilities was considered. This is an issue now as the internet has expanded and vulnerabilities are a key design consideration. The next example covered an implementation bug, this is when a programmer has a code based flaw that could be exploited. The final example is an improper configuration, this is when the software is designed well and there are few to no exploitable flaws however, the end user has set the software up poorly like leaving the password as the default.

Mofassir then covered exploitations, these are when a person of party find a vulnerability and exploit it some examples are given.

- Loss of sensitive information
  - Capital One Bank lost more than 100 million records
  - The European GDPR allows a fine for a loss equal to either 10 million Euro or 4% of a companies annual turnover
    - British Airways lost 500,000 records and received a 200 million Euro fine
    - Marriot Hotel lost 3 million records and received a 110 million euro fine
- DDoS GitHub had 1.35 Terabits per second of traffic through a DDoS attack in the 1st quarter of 2019
- Firewalls, Anti-Virus, IDS, IPS are not particularly effective

The next component of the lecture was on the tools used by hackers and professionals as well as governments to access different software. Mofassir told us these can be broken down into three categories:

- Free tools
- Paid tools
- Custom tools, these are mostly used by governments

First we discussed free tools, these tools usually require some simple computer experience. These include Nmap, Nexpose, Nessus, Metasploit, Armitage, Cain & Abel, John the Ripper, Hydra, and Aircrack. These tools are effective and easy to access. The next set of tools we discussed were paid tools, these can be bought by individuals and are usually purchase for only a few thousand dollars. Usually these tools are purchased on the dark web. The final tier of tools are custom tools used by governments. These tools include:

- NSA tools
  - UNITEDRAKE - Gain total control of a computer
  - Captivated audience - Used for recording using a user microphone
  - Gum Fish - Uses users webcam to record/take pictures
  - Foggy - Used to get internet history and passwords
  - Grok - A keylogger
  - Salvage Rabbit - Used to access external media on a PC
- Pegasus - Designed by NSO
  - Used by govenments
  - Works on all operating systems
  - Is able to access data such as messages, location and accessing the webcam and microphone

The next topic we discussed was the people that exploit vulnerabilities these are: hackers, state actors, and activists. Hackers try and get access to secure systems, usually for profit. State actors are people working on behalf of a government and either attempt to secure local systems and exploit foreign systems for their government. Finally, activists these people attempt to gain access to systems in an attempt to further an agenda, eg. A more open access internet or better privacy. Examples of these activists are, Aron Swartz, Edward Snowden and Anonymous.

The next topic of discussion was finding and reporting vulnerabilities. The method described by Mofassir was through bounty programs where companies offer money in exchange for discovering software issues in their platforms. The amount of money is dependent upon the severity of the flaw and the ease at which an intruder could gain access. The next stage is to report the vulnerability to the relevant CERT organization. These organizations inform the relevant company about the flaw and give a time period before they release the flaw publicly. This is to ensure the company responds to the issue. When made public the flaw is sent on to various mailing lists, this allows users to update the software to the version where the issue is fixed.

The final discussion topic was implementation vulnerabilities these include, improper memory allocation, improper input validation, and improper protocol implementation. Improper memory allocation allows hacker to write data over the buffer end, potentially giving access beyond what is intended. Improper input validation allows hackers to inject unwanted scripts/requests into code. The final vulnerability allows the hacker to exploit a protocol that was implemented poorly such as using a TCP sync attack.

The key take away points for this lecture was to ensure your code is secure and that we should always update to the latest security patch.