University of Victoria, Software Engineering

Case Study 7: Ethical Hacking

Jordan Ell
V00660306

March 2, 2013

This report will outline five weaknesses or vulnerabilities seen or discussed in the etchical hacking guest lecture given n SENG460 on Friday March 1st 2013. Each of the five sections below will outline a weakness or vulnerability, how the weakness or vulnerability could be exploited, and a countermeasure to ensure protection agains the weakness.

1 Social Engineering

It is often said that social engineering working not because the attackers are smart, but because the targets are so dumb. Social engineering is the act of using people, not nesisarily through technology, to manipulate standard procedures or thinking in order to gain access to protected or secure information or other material. The weakness of people is often exploited by engineering situations where people behaive more trusting than they normally would. The weakness can be exploited in many ways such as finding personelle in online directories and impersonating an authratative role to them over the phone in order to manipulate them. There was also the case of simply taking advantage of people's laziness by sneaking into a factory during a shift change when a high volume of employees pass through security unchecked. Technology can also be used by send around fake emails asking for people to use authentication credentials on fake websites (this takes advantage of people's trusting demenor).

The major countermeasure to social engineering is simply awareness to social engineering attacks and proper training on technical procedures. For example, a company should know that IT support will never ask for their password directly or any form of extra authentication to websites. Employees can be trained to be less trusting so that if they get a phone call from an impersonator, proper steps can be taken to insure that not volatile information if given out.

2 Phising

Phising is a type of exploit once again that preys on people's tendency to be trusting in nature. Phising is a way of using fake or impostered website and emails in an attempt to fool a user into giving up authentication credentials. An example of this may be to create the website favebook.com, a mispelling of facebook, then create a website that looks exactly like the login page to facebook in an attempt to get users to sign in and thus give up their credentials to this fake website. A real world example of how phising is used to exploit people is the UVic banking information issues. About a year ago UVic had a security breach in which bannking information was given out, phising attackers took this opportunity to send out emails from Canadian banks asking users to sign in on a phising website in our to insure their information was secure. Once the user signed into this fake site, their accounts would be compromised.

Again, the real countermeasure to this attack is end user knowledge and training. A user should know right off the bat that a Bank or other instituion which requires authentication will never ask you to sign in to secure your credentials. Emails and other messaging services provide links all the time, however, links can be manipulated in HTML in order to look like a secure path. This being said, URLs should also be inspected when fraud is a possibility. The user should have a sense for what a phising attemp will look like compared to the legitimate website.

3 Baiting

Baiting is yet another weakness in people that is exploited through the use of technology. Baiting, in a sense, is to offer a user something curious or extravagent in order to have them be manipulated and preform some service for the attacker. There have been a couple real world scenarios where baiting has been used. A bank employee was trying to get his or her boss to understand the dangers of USB keys used by other employees. To help show the danger, the employee simply created multiple USB sticks with auto running malware. Once the USB stick was inserted into a machine, the malware would go to work. Next, the employee simply left a bunch of the USB sticks out and scattered around the bank. The next day several employees, out of curiosity, plugges the sticks into their machines. This is a perfect example of baiting by taking advantage of curiosity. The next example is that of a conference with a QR code. A, in this case pretend, attacker simply put up a QR code around a conference and waited for people to scan it. Once scanned the user would be taken to a website which attempted to execute malicious software. The QR code worked and baited in several conference goers. This is another example of real world baiting techniques that rely on people's curiosity.

A countermeasure towards baiting would involve corporate policiy for an orginization or just personal control over curiosity. A corporation should have policy in place to prevent unknown devices from being plugged into their systems or networks as well as random suggested websites from being used (or QR codes from our example) with company equipment. As for personal use, end users simply need to understand the dangers in what seems like abandonded technology. Once a device is inserted into a machine, there is no telling what could be run from a malicious coder.

4 Web Application Vulernabilities

With the internet becoming more of an accessibility tool for more and more applications, web base applications can often be target for their security holes and potential to be linked to database or other system components, since by their nature these applications tend to be the link between front and back end systems. A recent study by SANS indicates that more than 60% of all attacks on the internet target web applications. Tools such as websploit can be used to scan and find these vulernabilities in order to generate custom attacking code for the targetted website. Examples of these types of weaknesses and vulnerabilities being exploited include the following. SQL injections, by manipulating URLS or input boxes on a website, an attacker can send malicious SQL queries to a database which enables access to otherwise secure information being stored. Cross-site scripting allows for the running of scripts in the user's browser which can lead to the stealing of cookies or redirects to other websites which are used to take over a user's session.

While each web application vulnerability may require unique processes or countermeasure in order to stop malaicious attackers from gaining access to unwanted information, there are a few general purpose steps that can be taken. Always insure input from and user given command is sanatized. Many attacks begin with passing malicious code inside of user given commands or required inputs. Stay up to date on new vulnerabilities such as Java (as many web apps use Java as a backend) in order to avoid new attacks through patching of your middleware and backend facilities.

5 Sniffing / MITM / ARP Spoofing

Sniffing, man in the middle attacks, and ARP spoofing all rely on an attacker getting access to your network and packet flow. Sniffing involves the actual detection and reading of packets as they are sent over a network. Man in the middle attacks involve an attacked coming between a user and their destination while exploiting weaknesses such as ARP spoofing and MAC flooding. An example of potential sniffing exploits would be to sniff out VOIP packets on a network in order to record user's internet base phone calls. The attacker could listen to what was said and find damning information. An example man in the middle attack could involve ARP spoofing in order to redirect a user to a malicious website which appears to be their end destination, such as creating a banking phising website and using redirects to make it appear like the legitimate website orgiinally requested by the end user. Sniffing and other man in the middle attacks are extremely dangers as the tools to use them are often easily availible and user friendly enough for a basic understanding to go a long way in attacking someone. (Wireshark, ettercap, etc..)

There are a couple of ways to deter man in the middle based attacks or packet sniffing. For one, access to the network is required for these types of attacks and thus, access should only be given to trusted employees or users of the network. This will stop third party users from coming onto the network in the first place in attempting to sniff packets without permission. The second way to stop these types of attacks is with secure network connections with encryption. Packet sniffers often cannot unencrypt network packets which are encrypted from source to destination. This ensures that even if an attacker got access to the network, the packets being sniffed would be unreadable for their content.