CMPSC443 Final Exam Score: /100

Student Name: Matt Popovich

**Problem 1 [20%]** - Name the various types of **injection** attacks discussed throughout the course. For each attack, use a few sentences to describe the idea and the security measures that can be used to defend against it. Note that an injection attack means that an attacker is able to cause certain damage by injecting well-crafted payload to client/server side software systems.

1. Format String
   * Exploit
     + Able to view contents of memory and even write any value to a specified memory address by using a payload that specifies multiple arguments (“%x%x%x%x%s) when in reality printf is not expecting that many arguments. This causes printf to work its way up the stack, reading values of registers instead of variables on the stack. “%s” can be used to view a memory address. “%n” can be used to write to a memory address.
   * Defense
     + Have printf verify before execution that the number of format specifiers it has received is consistent with the number of arguments passed to printf. For example, bash can compare the argument $# with the amount of ‘%’ in the format strings.
2. Buffer Overflow
   * Exploit
     + Exploit is a buffer overflow, allowing us to jump to any address in memory to continue execution of our program. Uses a vulnerability in strcpy to overwrite the value in the return address register.
   * Defense
     + ExecShield protection
     + Saving a copy of the return address and restoring it before returning.
     + Enabling the stack protector “-fno-stack-protector”
     + Using the updated function, strlcpy.
3. Return to libc
   * Exploit
     + Similar to buffer overflow, except we can’t jump to any address in memory, we can jump to the C library. The C library has many functions, one being system, which will allow us to pass any command as a parameter to the function for execution.
   * Defense
     + Preventing the buffer overflow of the return address register by using strlcpy.
     + Enabling the stack protector “-fno-stack-protector”
4. SQL Injection
   * Exploit
     + An attacker can create queries to a database server that are executed as statements. These statements can read, add, modify, and delete sensitive records in a database.
   * Defense
     + Sanitize data input
     + Avoid constructing SQL queries with user input.

**Problem 2 [20%]** - Given a public IP address, describe the steps you will use to hack into the system. Your answer should explore as many as options you can think of.

For starters, we’re going to want to see if they have any available ports open with a port scanner. After this, we’re going to try and connect to the IP address and open port via telnet. If we are asked for a username and password, we can brute force it beginning the preferred login information of “username”, “password”, “admin”, etc. If we can get past this step then we are ‘into the system’. From here we can run windows commands such as “dir, cd, etc.”.

**Problem 3 [10%]** - Assume you want to send a compressed and encrypted file to a friend. Will you first do compression and then encryption, or vice versa? Justify your answer.

Compress before you encrypt.

Although both ways will work, compression before encryption will result in the smallest files and least amount of computation time. Compression works by finding patters in data. A good encryption algorithm should result in basically random data – and no patterns, which will be difficult to compress.

**Problem 4 [20%]** - Given a malware sample, how would you analyze it manually? Briefly describe each step of action you will take.

The very first thing I would do is upload the executable to virustotal.com. Then, I would use PEiD to see if the file was packed or obfuscated. If it was, my first order of business would be to de-pack/obfuscate it. Assuming it is packed, I can try to either locate the program’s actual start point in OllyDbg or hope that PEiD can find the OEP (original entry point). I will know I succeeded when PEiD tells me that the file isn’t packed/obfuscated and was instead, as an example, “Compiled with: Microsoft Visual Studio C++ 6.0”.

Assuming it is now unpacked, I can load the malware into IDA Pro. From here I should see a main function because the executable is now unpacked. In IDA Pro I can view the imports, strings, and processes to give me a better idea of what this malware is looking to achieve.

Analysis in IDA Pro will reveal what I’m looking for. I can look for major code constructs and subroutine names to follow the flow of the code. If it is difficult to debug statically, I can look into debugging the program instruction by instruction in OllyDbg for more of an insight.

**Problem 5 [20%]** - You are hired by a small company to provide a security solution for their network protection against external threats. Please provide a list of the security components (e.g., firewall) you will include in your solution, and state the reason you use them.

The first thing I would do is to make sure that all employees are informed of the basic knowledge of malware and social engineering: spam emails, phishing emails, email attachments, Nigerian Princes, phishing telephone calls, popups, verified websites via https, only plug in known devices, always lock your computer when you are away from your desk etc.

I would then look to make sure all passwords are strong and maybe require a password change every few months, possibly even two factor authentication. All computers will need a firewall active, updated anti-virus protection, all software automatically scheduled to update, administrator access granted on a case by case basis: disabled by default.

I’d look into a proxy to have all the computers connecting to, restricting access (whitelist or blacklist) from known IP addresses of malware, ransomware, etc.

I would back up all critical software and files, looking into one backup a month, one every week, and one every 12 hours. The monthly backup would remain offline and only plugged in during the night of the backup. The backups would be encrypted if necessary.

All Wi-Fi access points would be encrypted with a security protocol that required each user to have their own private key to access (WPA2 at the minimum). Any Ethernet cord would need to have a specific computer’s MAC address registered and later validated in order to connect to the network.

Only certain drivers for USB flash drives would be installed by default (see rubber ducky, LAN turtle hack).

**Problem 6 [10%]** - It is impossible to formulate laws to enforce all sorts of behaviors acceptable to society. Instead, society depends on ethics to build awareness of socially accepted behavior. Ethics are objective. Unlike laws, they cannot be forced on individuals. In fact different individuals may have different ethical beliefs. The point however, is that some sort of social standard needs to be set with regard to the use of computer resources. Unlike laws, ethics can be molded and modified to suit the situation much more easily. Thus it is the responsibility of groups, companies, organizations, service providers, and even countries to establish codes of ethical behavior that people should strive to achieve and live by [1].

Read the following article regarding the ethics aspect of information security, then answer a few questions

<http://www.secureworks.com/resources/articles/other_articles/ethics/>

a. We watched a video about blind sql injection (link: <https://www.youtube.com/watch?v=Rqt_BgG5YyI> ) in class. Is his behavior legal? Is his behavior ethical? Justify your answer.

b. You want to assess the security of Apple’s App Store. You created an App containing malicious code, and managed to bypass the App Store’s screening/vetting system. You immediately removed the App from App Store, so no one else but you can access this App. Is this behavior legal? Is it ethical? Justify your answer. Is it ethical to share your software with others?

1. His behavior is not legal. Hacking into a website is illegal. If he asked for permission, informed the website that he successfully hacked their website, and helped them (and many others by making a video) remove the vulnerability then I would consider himself a ‘white hat hacker’, and an ethical hacker. All it takes is one malicious actor to hack a site. All it takes is one ethical hacker and all websites can prevent hacking.
2. Once again, this behavior is not legal. Writing malicious code is illegal. If the App Store was notified of this beforehand and granted permission to this ‘experiment’, then I would say it is ethical. They are making the App Store safer for millions and millions of people.

Once the App Store implements a patch/fix, it is ethical to share the software with others. While it may be educating hackers with malicious intent, it’s better to give all App Stores the power to protect themselves than just a few.

**Reference**

1. Amit Raju Philip, “The Legal System and Ethics in Information Security”, SANS Security Essentials, 2002.