CMPT 403 Assignment 1

1.

(a) ENISA reports a 30% increase in crypto-jacking incidents year-on-year in 2020, and they have only increased since. Crypto-jacking uses the victim's computing resources (usually CPU) to mine cryptocurrencies for the attacker. Crypto-jacking can be done by background scripts on a webpage. Webpages that are otherwise useful to visit are particularly powerful attack vectors. The resources stolen are not large enough to be noticeable to a victim.

i. The integrity principle is being violated. Since the victim's computing resources are being used to mine cryptocurrencies for the attacker, the integrity of their computer system has been compromised to serve the attacker's objectives without the victim's consent. One might also argue that the availability principle is being violated since crypto-jacking involves utilizing system resources for the purpose of harvesting; however, since it states that the resources stolen are not large enough to be noticeable to a victim, you could also argue in this case that availability is *not* being violated.

ii. A web application flaw is being exploited that targets clients on the network, so this is being spread through a network.

iii.
   a) Users should use ad-blockers to both detect and block malicious scripts and avoid unsecured websites.
   b) Website owners should implement secure coding practices and follow web development guidelines to minimize the vulnerabilities that can be exploited. This includes regularly updating all themes, plugins, and content management systems. Website owners should also regularly scan their websites for vulnerabilities using security scanning tools.

(b) In 2013, the New York Times reported that the Dual EC DRBG random number generator has a potential backdoor. The NSA is the sole editor of this algorithm's standard. The backdoor allows an attacker to fully compromise cryptography based on this tool. RSA Security started using Dual EC DRBG as its standard RNG for some of its software after accepting $10 million from the NSA.

i. Integrity means that the data and systems remain accurate, complete, and unaltered. Since the presence of a potential backdoor exists, the integrity of the cryptographic systems that utilize the RNG has been compromised. It should also be mentioned that the fact that RSA Security started using Dual EC DRBG as its standard RNG after accepting a substantial sum of money from the NSA makes one wonder about the integrity of RSA's decision-making process.

ii. If it could be shown that the NSA intentionally planted the backdoor, it would align with the concept of planted malware. While the intention here would be hard to prove, because the NSA is the sole editor of this algorithm, it is certainly reasonable to be suspicious of the NSA having planted it. It does not closely align with network malware, removable media malware, or trojan malware.
iii. The primary issue here is that the NSA is the sole editor of this algorithm – i.e., that open design is not being utilized or implemented. Thus, requesting that the code be open, analyzable, subject to public scrutiny and/or undergoing an independent evaluation of the algorithm by trusted experts before implementation would have prevented the security breach.

(c) Pegasus is a powerful piece of malware developed by the NSO Group that has frequently been used for surveillance on high-profile targets such as politicians and human-rights activists. Attackers exploited a zero-day vulnerability in the Safari Webkit by sending a file to a victim that appears to be a GIF file, but clicking on it would cause surveillance software to be installed on their iPhone. A 2021 report by Amnesty International shows that it has been used in thousands of attacks over the three preceding years.

i. Arguably all three principles are being violated. Confidentiality, since the surveillance software attains access to information that should be accessible only to authorized individuals or entities – in this case, the iPhone's owner. Integrity, since the installation of surveillance software on the user's device without their consent or knowledge, has compromised the integrity of their cell phone. And availability since the software running on their cell phones utilizes system resources, potentially causing performance issues or disruptions to the normal functioning of the device.
ii. Having sent the victim surveillance software masquerading as a GIF, there is a clear element of trickery here; hence we can confidently classify this as a trojan.
iii. I can see two reasonable countermeasures to prevent an attack like this, and potentially a third. First, implement regular software updates. Ensuring that your OS and applications are up to date with the latest security patches and bug fixes is one way to defend. Two, deploying and maintaining reputable endpoint protection ("antivirus") software updated with the latest trojan signatures and scanning all programs before they are installed would be another way to defend. third, consider using cell phones that come from open-source communities. The nature of open-source systems allows for more eyes on the code, theoretically resulting in quicker detection and remediation of vulnerabilities (including zero-day exploits). Point three is debatable, however, since closed systems limit the number of individuals who can analyze and identify vulnerabilities since source code and design details are not publicly available.

(d) The Meris botnet broke several records for DDoS volume in 2021. It compromises MikroTik routers with a directory traversal vulnerability that allows remote attackers to steal the admin password of the device to gain full control over it. With 250,000 such routers, Cloudflare estimates that Meris targeted approximately 50 different websites a day, demanding ransoms and DDoSing websites that refused to pay.

i. All three principles of cyber security have been violated. Confidentiality, since having access to admin passwords procures sensitive information held on the devices. Integrity, since attackers had full access to and control over the devices, allowing the modification of the device's purpose, configurations, settings, and functionality. And availability since websites that refused to pay were DDoS'd.
ii. This is network-based malware. It targets vulnerable routers and gains control over them to create a network of compromised devices that can be used for malicious purposes.
iii. To stop this type of attack at the root, it is crucial to address the vulnerability in the MikroTik routers through security patches or firmware updates. A secondary solution is for websites themselves to implement network filtering in an attempt to block malicious network traffic; however, this addresses the symptom of the problem, not the source.

2.

(a) Some buffer overflow attacks do not overwrite any return address at all.

False. A buffer overflow occurs when a program writes to a memory address on the program's call stack outside of the intended data structure. While the canonical buffer overflow attack consists of overwriting the function's return address with a pointer to attacker-controlled data, attackers may also target function or object pointers or other critical data that can be leveraged to redirect program execution.

(b) Minimizing privileges in critical programs can help mitigate buffer overflow attacks.

True. When a program runs with elevated privileges, a successful attack can be more damaging since the attacker gains control over the system with the same privileges as the program is proferred. Thus, if the program has minimal privileges, even a successful attack will limit the attacker to those same privileges.

(c) Return-Oriented Programming is able to defeat stack canaries.

True. Blind return-oriented programming, for example, has been shown to have defeated stack canaries on 64-bit systems. In a BROP attack, the buffer overflow is carried out byte by byte, and through trial and error, the canary can be leaked. Once it has been, the return instruction pointer can be procured similarly.

(d) XSS attacks usually require the attacker to gain full control over the web server first.

False. XSS attacks only require that the website improperly interpret and store text as executable code, so when the page is loaded by unsuspecting victims, the code is executed.

(e) If there is a format string vulnerability in OpenSSL, it would be a more serious bug than Heartbleed.

False. The severity of a bug depends on more than any one factor, so it would be wrong to say that a format string vulnerability would necessarily be a more serious bug than Heartbleed, especially considering that we know the severity of Heartbleed, which affected a huge number of systems worldwide.

3.

In 2015, Ion et al. investigated the differences between security practices recommended by experts and non-experts. Experts included hacker conference attendees, professionals and researchers, while non-experts were recruited from MTurk.

i. The biggest difference in security practices between experts and nonexperts was to "update software". 35% of experts included this in their top three suggestions while only 2% of non-experts did so. Give two examples of real attacks that could have been prevented if software updates were taken more seriously.
    a. Equifax data breach. A security patch was released on March 7, 2017, after the exploit was found. The breach at Equifax began on May 12, 2017, two months later.
    b. WannaCry ransomware attack. Microsoft had released patches to close the exploit, and much of the spread was from organizations that had not applied them.

ii. The top advice from non-experts was to use antivirus software, but experts do not agree. Experts rate the effectiveness of antivirus software much lower than non-experts. Explain this by describing how malware can defeat antivirus software.
    a. One method malware can use to defeat antivirus software is through polymorphic code. Polymorphic malware changes its code structure and appearance while retaining its functionality. This evades the signature-based approach of antivirus software detection.
    b. Another way in which malware evades antivirus software detection is by way of being previously unseen. Attacks of this type are called zero-day exploits. Since they have not been seen before, a signature has not been developed for zero-day exploits, so antivirus software cannot detect them.
    c. A third approach is via fileless malware, or code that works directly within a computer's memory instead of existing on a file or directory in the file system. Fileless malware uses legitimate programs to compromise a computer and when your machine does become infected, no files are downloaded, hence the name.
    d. A fourth approach is through a rootkit installation, in which an attacker can maintain privileged access and hide their intrusion. This level of control over the

system enables an attacker to modify existing software, including antivirus software, that is designed to detect malware on the system.

e. Finally, it is also possible to generate encrypted or obfuscated payloads. This conceals the malicious nature of the code because it is concealed within obfuscated data.

iii. Experts strongly recommend using a password manager. Explain the benefit of a password manager by describing an attack vector for login compromise from the webmaster's perspective: this attack vector would work even if your web server's defenses are strong enough to resist the attacker. (Hint: Think like an attacker. If you know someone's login name but not their password, and they do not use a password manager, what could you do to obtain their password?)

Supposing that the web server's defences are strong enough to resist an attacker, there are certain things an attacker can do that a webmaster cannot defend against. These approaches include trying simple or common passwords, trying old or possibly reused passwords, implementing phishing attacks designed to steal login credentials, and probably many others. Password managers are effective means of defending users against these types of attacks:

1) Password reuse. Suppose an attacker wants to access the account of a user for whom they have the username. If the user tends to reuse certain passwords, and the attacker has or finds a way to discover them, there is no possible defence. A password manager prevents password reuse, so even the discovery of a previously used password is of no help to an attacker.
2) Simple or common passwords. Since the use of simple and common passwords is so prevalent, a common way to break into someone's account is to make a list of common and simple passwords, then run a script, or some automated tool, to systematically try each username/password combination. A password manager ensures that simple or common passwords are never used.
3) Phishing attacks. These attacks trick users into visiting websites which look authentic but are designed to steal a user's login credentials. Since password managers prompt users to input their login credentials, it can signal to the user that the website is inauthentic if the password manager has no username and password entry for a website that should have one.
4) Using hard-to-remember but easy-to-guess passwords. It is a common misconception that using a complicated password format with, say, some uncommon base word, followed by some sort of punctuation, then some numerical value creates a hard-to-guess password. Nevertheless, misconceptions like this are prevalent, causing people to make easy-to-guess but hard-to-remember passwords ad nauseum. This type of password has a relatively low degree of entropy compared to just four simple random words. Using a password manager relieves users from their misconceptions, ensuring passwords that always have a high degree of entropy.