COMP3632 (16-17 Spring)
Assignment 1

Tristan Muratore - 20443557

March 10, 2017

Written assignment

1. **OnePass**

   (a)
   - **System :** The OnePass password management system.
   - **Asset :** Customer's Passwords.
   - **Vulnerability :** The Master password for the OnePass account giving acess to all other passwords.
   - **Attack :** Brute force guessing the Master password.
   - **System :** Enabling 2-Factor-Authentication *(or don't use OnePass)*.

   (b) Let's start of by finding the SLE. The asset value is 60$ (5$ per month), and the exposure factor is 10%. **So SLE is expected to be 6$**.

   Now onto ALE, the annualized rate of occurence is 1000 (10% of their 10,000 clients). So **ALE is expected to be 6000$** (SLE*1000).

   Since ALE is greater than the cost to implement 2FA (3000$), in terms of Quantitative Risk Analysis, 2FA **it is worth investing in 2FA**.

   (c) If the hacked clients sue OnePass for failing to protect their private information, the payout or the legal fees should also be taken into account as financial damage.

2. **Stories about malware**

   (a) **Sundown Malware**

   i. Since BitCoin mining is a CPU-intensive activity, the victim computer must be significantly slowed down. Therefore the **Availability** of the system is violated.

   ii. The Sundown malware is a **Trojan**, since the JavaScript code on the infected website tricks the user into installing the malicious code. One could argue that it is by effect on the system, a **Botnet**, since it's takes control of the computer forcing it to mine BitCoin.

   (b) **Mirai IoT DDoS**

   i. The DDoS on Brian Krebs' blog will render the website un-joinable. The **Availability** of his webpage is compromised.

ii. Since the Mirai malware takes control of IoT nodes with weak security, it is a **Botnet** by effect. Since it spreads over the network, by method of spread it is a **worm**.

(c) **Angler exploit kit**

i. Since it is used to steal personal information it violates **Confidentiality**. It also modifies legitimate websites to spread so **Integrity** is also violated.

ii. Since this is an exploit kit, it's is very versatile. It can act as a **Virus** spreading malware on a computer, it can spread through the network so it can be classified as a **Worm**. It can access all sorts of personal information on the infected hosts, using **Spyware** or **Keylogging**. If the attackers desires so, it can also take control of a network of computers, thus forming a **Botnet**.

3. **Saltzer and Schroeder's Principles of Secure Design**

(a) **Economy of Mechanism**

i. Make the system as simple as possible, so that it is easier to understand.

ii. In the TV Show, Westworld, the androids populating the attraction park have become so complex that no one really knows how they work, and on occasion some of them *"diverge"* qui spectacularly from their scripts. Despite this complexity being necessary for the androids to be realistic, if they were simpler the park technicians would have been able to understand the source of the bugs and deal with it.

(b) **Least Privilege**

i. Don't give useless extra permissions to a subject, give it only what it needs.

ii. In the 1st Matrix **only**, normal humans are not supposed to be able to do as much as the surveillance programs, *"the Agents"*. Neo proved that this principle was not respected since, he spectacularly *"overtakes"* the system with his powers defying all laws of physics. In the end, his privileges were above those of *"the Agents"*.

(c) **Separation of Privileges**

i. Depending on what each subject desires to do, he should be able to get the appropriate privileges individually/by group.

ii. The analogy with Matrix I also works here, Neo's capabilities should have been clearly defined by the simulation. However, the fine line between the imagination of the hosts fueling the world and strict separation of privileges was broken allowing Neo to exceed all limits.

(d) **Fail-safe defaults**

i. When there is an unexpected state/error, the subject should go back to a secure default.

ii. I'm going to use the Westworld example again. When a android would *"diverge"* from it's script, the park's staff would revert the android back to a "safe configuration" before putting them back in service. This shows that the androids were designed to have some sort of Fail-safe default.

(e) **Open Design**

i. Open the design of the system to everyone, to show that you have nothing to hide, or that they can help correct vulnerabilities.

ii. In **Rogue One : A Star Wars Story**, the plans of the Death Star, that the Empire was trying to keep secret, that had only been seen by the engineers in charge of the project, contained a fatal flaw to the battle station. If the plans were made public this flaw could maybe have been fixed. *(Of course if the Empire had done that, the whole Star Wars trilogy would have been, very different)*

Programming assignment

Viruses

(d) Since my code does not rely on anything random in it's content and in it's actions. A virus scanner could very easily catch my virus either by **signature** or **behavior**. To remove my virus it would be quite simple. Since most of my code is tucked in a separate class *"naughtyClass"*, one could clean the file by removing the class and it's call in *"main"*. *(Some extra cleaning might need to be done to the library imports at the beginning of the file).*