CS433 Homework 2

How does the given virus work?

This program represents a basic model of a computer virus.

- The `infect-executable` subroutine is responsible for infecting executable files on the system.
- It repeatedly selects random executable files and checks if the first line of each file contains the value `1234567`.
- If the first line does not contain `1234567`, indicating that the file has not been infected before, the virus prepends
- This process continues in a loop, infecting multiple executable files across the system.
- The `trigger pulled` subroutine evaluates a certain condition.
- If the condition holds true, indicating a specific event of scenario has occurred, the virus proceeds to execute its malicious payload.
- The `do-damage` subroutine represents the malicious payload of the virus.
- Once the trigger condition is met, this subroutine is executed, and the virus carries out whatever damage or harmful actions it is programmed to perform.
- The exact nature of the damage or actions carried out by the virus is unspecified in the provided pseudocode but could inclide activites such as data deletion, system corruption or propagation to other systems.
- The `main-program` subroutine orchestrates the main execution flow of the virus.
- It initiates the infection process by calling the `infect-executable` subroutine to infect executable files.
- After infecting files, it checks if the trigger condition specified by the `trigger-pulled` subroutine is met.
- If the trigger condition is satisifed, indicating the desired scenario for executing the malicious payload, the virus calls the `do-damage` subroutine to carry out its malicious actions.

Chapter 3 Problems

4.

Here are a couple of ways that the program can be sabotaged to compute the wrong sum:

- If the source file is altered before running the program unbeknownst to the user. Such as modifying a for loop condition from `for(int i=k;i<=n;i++)` to `for(int i=3;i<=20;i++)`.

- The program can also be altered by an external process. For instance, n = 10, k = 1, And the sum has been calculated for the range of k = 1 to 5.
  Then the program is altered by an external process.
  15 (sum of 1 to 5) will be displayed instead of 55 (sum of 1 to 10).

6.

Confining this program would act as a filter between the callers and the untrusted program. A calling program would call the confining process, requesting to call the summation program. The calling program has no direct access to the summation program. Secondly, the confining program would check the resolt of the summation. This process would verify that the answer was exactly 55, the expected sum from 1 to 10. A different situation could include using a confining process to check the computation for feasibility, considering the magnitude of the given input values, values of other system variables, the name or owner of the calling progam, etc, and see if the result was reasonable. This program would act as a wrapper, wrapping the untrusted code in a trustworthy filter.

22.

a. Each bank has a verification certificate in their emails, I would need to verify that this email has that certificate. I would also need to verify the domain it came from, to make sure it is the exact same from my bank. If I have some email thread history from my bank, I could verify that the sender is exactly the same. Installing some email software that warns about unknown senders or suspicious activity could also be helpful to warn me.

b. I would send an email to the customer thanking him for providing the information and sign as the bank, and then utilize the information for whatever purpose I wanted. To ensure the bank was unaware of what was going on I could install spyware on the customer's computer to monitor their activity.

Chapter 4 problems

3.

Some steps you could take to verify the authenticity of a web page include:

- The presence of a valid HTTPS connection in the URL indiciates the connection is encrypted and secure, reducing the likelihood of tampering or interception by malicious actors
- A valid SSL certificate issued by a reputable Certificate Authority provides assurance that the website has been authenticated.
- Legitimate websites often provide contact information, including a physical address, email, phone number, and customer support options. Presence of these add credibility to the website.

- Modern web browsers may display indicators such as padlock icons, green address bars, or security warnings to indicate the security status of a website. These features can also generally be installed through browser extensions

9.

An example of how webpage framing could be used to trick a victim can be described using this scenario. Imagine you heard from your friend that your favorite shoe company is offering free vouchers to the first 100 users, so you go browsing Twitter and come across a post that appears to offer this free voucher, as it has the blue checkmark and seems legitimate. However, instead of linking to the voucher redemption page, the attacker uses webpage framing to load the redemption page in a transparent frame on a malicious webpage that the attacker can control. It is designed to look exactly like the real deal. When you click on this post, you are unknowingly interacting with the malicious webpage, but they only see the voucher. The attacker overlays deceptive elements, such as fake buttons on top of the redemption button to encourage you to click in certain areas. Unkowingly, the clicks are interacting with the underlying voucher redemption page loaded in the transparent frame. As a result, you could go as far as entering personal information or clicking the buttons to claim the voucher, and since the frame is transparent they can not see it. By using this webpage transparent frame, the attacker successfully tricks a victim into performing actions without their knowledge, potentially leading to theft of personal info or other malicious intents.

17.

A technique by which a browser could detect and block clickjacking attakcs includes frame busting. This technique works as follows:

- Websites includes a frame busting script in their web page to prevent their content from being loaded within a frame on a different domain
- The frame busting script can be implemented in JavaScript and is included in the HTML code of the webpage
- The script checks whether the webpage is being displayed in the top level window or within a frame.
- If the webpage is being displayed within a frame, the script takes action to break out of the frame and redirect the browser to the top level of the window.
- By breaking out of the frame, the frame-busting script prevents the webpage from being used in a clickjacking attack, as the attacker's malicious content loaded in the frame is no longer visible to the user.
- By implementing frame-busting scripts and enforcing security measures like CSP, browsers can effectively detect and block clickjacking attacks, protecting users from unwittingly interacting with malicious content loaded in frames.

18.

One way to communicate scripts more securely between sites is to use Content Security Policy (CSP) directives, specifically the `script-src` directive. CSP is a security feature supported by modern web browsers that allows websites to declare a set of content sources from which the browser is allowed to load resources, including scripts. This will allow website owners to effectively control the sources from which the scripts are loaded, mitigating the risk of cross-site scripting attacks. This helps in securely communicating scripts between sites while maintaining a strong security posture.

21.

The attack that a financial institution seeks to counter by asking its customers to confirm that they see their expected security picture before entering sensitive data is called a 'phishing attack' or 'phishing scam'. In a phishing attack, malicious actors attempt to deceive users into providing sensitive information, such as usernames, passwords, or financial details, by impersonating legitimate entities or websites. Phishing attacks often involve creating fake websites or email messages that closely resemble those of trusted organizations such as banks or financial institutions. To combat phishing, financial institutions imlpement additional security measures, such as displaying a unique security picture to users during the login process. This security picture serves as a visual indicator to help users verify the authenticity of the site they are interacting with. By confirming that they see their expected security picture before entering sensitive data, users can help protect themselves against phishing scams by ensuring that they are on the legitimate website of the financial institution and not on a fraudulent phishing site.