DEVELOPING MALWARE FOR REAL-TIME NETWORK MONITORING AND KEYSTROKE CAPTURE TO SECURE USER CREDENTIALS

Prepared by: Rahul Joshi
Admission No: 24je0678
IIT (ISM) Dhanbad

PROJECT DESCRIPTION

This project focuses on the development of a sophisticated malware system capable of real-time network monitoring and keystroke capture. The malware will target user interactions on specific websites to gather login credentials while evading detection by modern security tools. The administrator would be able to access the information through a web-based user friendly interface.

FEATURES

- **Network Monitoring:** Continuously monitor network traffic and send data to a remote server.
- **Using Proxy:** The malware will obscure the IP address of the remote server by routing traffic through multiple proxies.
- **Keylogger:** Activating the keylogger on specific sites and storing the inputted data in a database.
- **Screen-Capture:** A feature to capture the screenshot of the victim’s screen.
- **Web-based Admin Interface:** A web-based interface for the admin to access the information of the victim.
- **False Windows-Renewal messages:** The malware will randomly open a window on the users screen asking for his credentials to renew the Windows OS.
- **Opening Websites:** The malware will open websites on the victim’s computer and activate the keylogger to collect personal details.
- **Clearing stored passwords:** A mechanism to clear the stored passwords of the user so that the he is prompted to re enter the password.
- **Anti-detection Mechanism:** A mechanism to avoid detection by windows security tools.

TECHNOLOGY STACK

- Python (+Libraries suck as pillow, pyautogui, sqlite3, tkinter, selenium etc.)
- HTML/CSS/JavaScript
- WebSocket
- HTTP and other protocols
- Flask
- SQLite
- Linux Commands (Bash Commands)

IMPLEMENTATION DETAILS

- **Bypassing Windows Virus Detection:** To bypass Windows virus detection mechanisms, I will compile the Python script using the **Nuitka** compiler, which converts Python code into optimized C code, making it more difficult for security software to identify as a Python-based malware.

- **Establishing Remote Connection:** The malware will create a **WebSocket** connection between the victim's PC and a remote server. This will allow real-time communication, enabling the remote server to receive continuous updates from the victim’s machine.

- **Keylogging Functionality:** A keylogger will be developed in **Python** that activates when the victim visits specific websites, identified using **Selenium** for web automation. The keylogger will capture keystrokes entered on these sites. The logged data will then be securely stored in an **SQLite** database for future retrieval.

- **Opening Desired Websites:** The malware will automatically launch targeted websites on the victim's browser. Upon visiting these sites, the keylogger will be activated to monitor and record the victim's keystrokes, allowing it to capture sensitive personal information such as usernames, passwords, and other data entered on these websites. This would be done using **Selenium**.

- **Web-based Admin Interface:** The malware will include a web-based admin interface built using **HTML**, **CSS**, **JavaScript**, and **Flask** for the backend. Admins can access the victim’s information by entering their password. The interface will also include functionality to remotely capture the victim's screen, using the **Pillow** and **PyAutoGUI** Python libraries.

- **Erasing Stored Credentials:** To remove traces of the victim's stored credentials, the malware will delete the file where the browser stores login information, forcing the user to re-enter their credentials the next time they attempt to log in. This would be done using is **OS** library in Python.

- **Obscuring Remote Server IP:** To hide the identity of the remote server, the malware will use **ProxyChains** to route the traffic through multiple proxies, ensuring that the remote server's IP address remains undetected.

- **Displaying False Windows-Renewal Warnings:** The malware will display a fake Windows-renewal message using the **Tkinter** Python library. This pop-up will prompt the victim to enter their username and password in order to "renew" Windows, tricking the user into providing their credentials.

WEEK WISE TIMELINE

Week 1: Preparation and Core Development

- **Python Library Exploration:** Focus on understanding and experimenting with key Python libraries essential for the project, including Selenium for web automation, PyAutoGUI for screen capture, Pillow for image manipulation, Tkinter for GUI creation, and SQLite for database management.
- **Keylogger Development:** Begin implementing the keylogger in Python, integrating it with Selenium to activate on specific websites and log keystrokes efficiently.
- **Networking and Protocols:** Gain foundational knowledge in networking concepts, including TCP/IP protocols, WebSockets, and proxy configurations, ensuring a strong understanding of how data can be transmitted securely and stealthily.

Week 2: Web Interface and Networking Features

- **Web-based User Interface Design:** Design and develop a secure, intuitive web interface using HTML, CSS, and JavaScript, with Flask as the backend framework. Learn and implement SQLite for seamless database integration.
- **Screen Capture Integration:** Implement the screen capture feature using PyAutoGUI and Pillow, allowing real-time screenshot capture and storage.
- **ProxyChains Configuration:** Integrate ProxyChains into the malware to obscure the IP address by routing traffic through multiple proxies, enhancing anonymity and security.

Week 3: Advanced Features and Security Measures

- **Clearing Stored Passwords:** Implement a feature to identify and delete browser-stored credential files, forcing the user to re-enter their login information.
- **Automated Website Launch:** Develop functionality to automatically open target websites in the victim’s browser and trigger the keylogger to capture sensitive information.
- **Fake Windows Renewal:** Design a fake Windows-renewal pop-up using Tkinter, prompting users to enter their credentials under the guise of renewing their Windows OS.
- **Anti-Detection Mechanism:** Implement strategies to evade detection by antivirus software, including obfuscation techniques and Nuitka compilation.

Week 4: Finalization and Testing

- **Feature Tweaks:** Refine and optimize all implemented features based on functionality and stealth. Address any performance issues or feature gaps identified during development.
- **Testing and Validation:** Conduct thorough testing on a virtual machine to assess the malware’s performance, stealth capabilities, and overall stability. Document the testing process and make necessary adjustments.

ABOUT ME

I’m an 18-year-old, First Year Computer Science student at IIT (ISM) Dhanbad, passionate about coding and technology. I love exploring new areas like cybersecurity, networking, and software development.

With experience in Python, C, HTML, and CSS, I enjoy taking on challenging projects that push me to learn more. This project is a great opportunity for me to dive deeper into network protocols, malware development, and web interfaces while sharpening my skills and growing my knowledge.

Github: https://www.github.com/crypticsaiyan/

Why I Should Be Selected for This Project:

I am passionate about technology and eager to expand my knowledge through hands-on experience. With a solid background in HTML, CSS, and Python, I bring relevant skills that will contribute to the success of this project. My strong interest in information security and computer networking drives me to take on new challenges and continuously improve. I am committed to putting in my best effort to not only complete the project but also to learn and grow throughout the process.

Do you have any other commitments during the program?

I have a few commitments during the program, as I will be traveling for a few days in both Week 1 and Week 3. However, I will ensure to manage my time effectively around those days. Additionally, I am eager to enhance my knowledge of C++ and improve my competitive programming skills, which I plan to work on alongside the project.

On average, I aim to dedicate around 5 hours per day to this project. I believe this will allow me to stay on track and meet the objectives. However, if I find that I am not meeting the daily or weekly targets, I am committed to increasing the number of hours I spend to ensure I stay on schedule and meet the deadlines.