INDEX

Summary:-

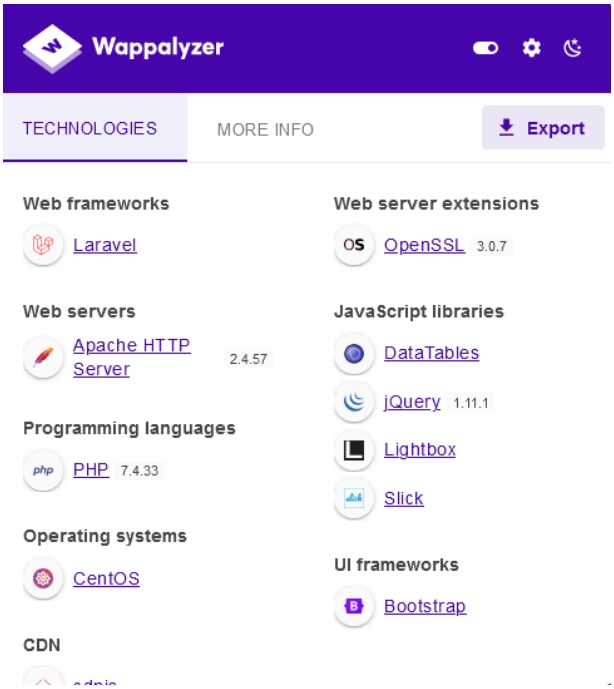
On 25th August 2024, I executed a prank on a friend's laptop by deploying a custom-built ransomware, which encrypted all of his important work-related documents. Notably, this attack was carried out without the use of phishing or social engineering techniques, yet it successfully bypassed the laptop's security measures. The ransomware was designed to be stealthy, avoiding detection by antivirus software and other security protocols. After two days, I provided the decryption key to my friend, revealing the incident as a prank and restoring access to the encrypted files. This exercise demonstrated the ability to bypass security defenses and highlighted potential vulnerabilities in the system, underscoring the importance of comprehensive security measures even in the absence of traditional attack vectors.

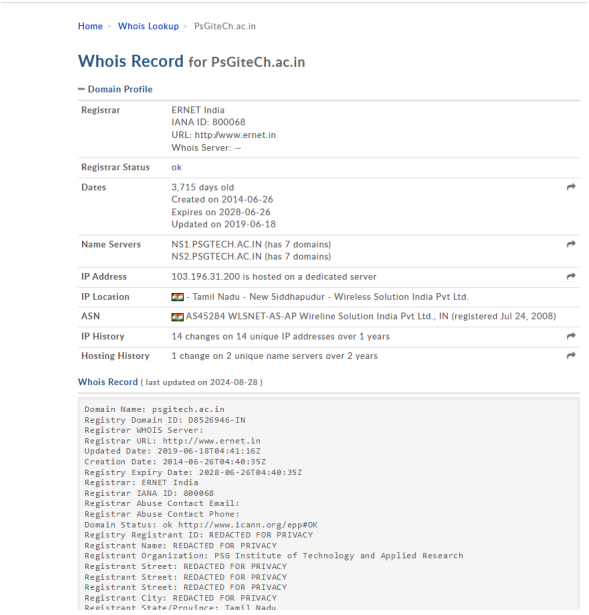
Reconnaissance:-

As a security engineer in my friend's company, I requested his device's IP address, which he shared without hesitation. I initially attempted an OS detection scan using Nmap, including a ping scan, but the results were not as expected. To proceed, I utilized tools like Airodump-ng and Bettercap to position myself as his default gateway, enabling me to directly scan his IP. Upon scanning, I determined that his device runs on Linux. I then scanned for open ports and identified ports 443 (HTTPS) and 20 (FTP) as active. Although I used Nmap scripts to search for vulnerabilities, nothing significant was found. Subsequently, I employed Dirb and Gobuster to scan

for open files or directories on his website, successfully uncovering numerous open directories.

After discovering that a website was hosted on his IP, I gathered additional information about his domain, such as the server version, database version, and other relevant details using online OSINT tools like Wappalyzer and Whois. It's important to note that the server was hosted on a single device. These findings were collected during the reconnaissance phase.





(example images on a website)

Execution:-

During the reconnaissance process, I discovered that the server hosting the website was an Apache HTTP Server version 2.4.57, using PHP as the programming language. Additionally, I found that the common user upload directory was inaccessible. I attempted an SQL injection to gain root access to the website but was unsuccessful. However, I noticed that the website allowed image uploads, so I crafted an image file named `up.png.php` containing malicious PHP code: `<?phpsystem($_GET["cmd"]);?>`. After uploading the image, I executed the code by navigating to `/upload/up.png.php`. This granted me access to the command shell, although not with root privileges. To run an .exe we must be a root user thus privilege escalation must be done. One way to find the root user password is navigating /etc/psswd/. But it is a difficult process. Instead we can use the

time_stamp of the sudo command to become a root user. After that we downloaded the ransomware from the internet and executed it as a root user. Note that the ransomware was written in rust which is an system level programming language which uses memory effectienly.