Note:

As this was a penetration test performed against a real company as part of an offensive cybersecurity class, all identifying information has been redacted.

MIST 400
Nate Santiago

XXXXXX Pentest

This report focuses on the first three stages of the Penetration Testing process, which are Reconnaissance, Scanning, and Enumeration. The purpose of these stages is to gather information about the target system and identify potential vulnerabilities that can be exploited during the later stages of Penetration Testing.

Target Validation

During the reconnaissance phase, I performed some target validation to gather information about the target domain XXXXXX. The following information was obtained from the Whois command, nslookup, and dnsrecon:

- **Whois:** The whois record shows that the domain was registered on XXXXXX, and the last update was made on XXXXXX. The domain ID is XXXXXXX. The domain registrar is Amazon Registrar, Inc. The name servers are XXXXXX, XXXXXX, XXXXXX, and XXXXXX. The domain status is clientDeleteProhibited, clientTransferProhibited, and clientUpdateProhibited. The WHOIS record also indicates that DNSSEC is not configured for XXXXXX.

- **Nslookup:** nslookup shows that the IP address of XXXXXX is XXXXXX. The server used for the query was XXXXXX.

- **Dnsrecon:** The dnsrecon tool was used to perform general enumeration against the target domain. The results show that wildcard resolution is enabled on this domain, and it is resolving to XXXXXX. DNSSEC is not configured for this domain. The name servers are XXXXXX, XXXXXX, XXXXXX, and XXXXXX. The mail servers are XXXXXX and XXXXXX. The domain has one A record and two TXT records. The A record points to the IP address XXXXXX. One TXT record contains Google-site-verification information, and the other TXT record contains SPF information. There were no SRV records found during enumeration. [2]

Finding Subdomains

During the penetration testing of the website XXXXXX, I conducted reconnaissance, scanning, and enumeration to gather information about the website. One of the primary objectives of reconnaissance was to discover subdomains of the target website.

I used different tools to discover subdomains, including Dig, Sublist3r, and crt.sh. Here is a summary of our findings:

- **Dig:** I used the Dig tool to get information about the DNS records of the target domain. The Dig command "dig XXXXXX" returned the IP address XXXXXX. However, this information did not reveal any subdomains for the target website.

- **Sublist3r:** I used Sublist3r, a subdomain enumeration tool, to discover subdomains of XXXXXX. The tool searched different search engines, such as Google, Yahoo, Bing, Ask, and Netcraft, to identify subdomains. Additionally, it searched in DNSdumpster, Virustotal, ThreatCrowd, SSL Certificates, and PassiveDNS. Unfortunately, I encountered an error while using Virustotal, which might have blocked my requests. [1]

- **crt.sh:** I used crt.sh, a website that provides SSL certificate transparency logs, to search for subdomains. The website returned several SSL certificates for the domain XXXXXX, but only returned XXXXXX as the one domain. [3]

[redacted]

Fingerprinting

During the reconnaissance phase of penetration testing, I conducted fingerprinting of the website XXXXXX using Nmap and Wappalyzer tools.

- **Nmap:** The Nmap scan results revealed that the website is hosted on IP address XXXXXX and is running on the TCP protocol. The open ports found were 80/tcp and

443/tcp. Both of these ports were detected as running "tcpwrapped" service. Additionally, 998 filtered TCP ports were found that did not respond to the Nmap scan. [4]

- **Wappalyzer:** It was used to gain further insight into the website's technologies and frameworks used. According to Wappalyzer, the website is using Microsoft ASP.NET as a web framework, and the web server used is IIS on Windows Server operating system. The website is using HTTP/2 protocol and has implemented HSTS security. The analytics tool being used is Google Analytics, and Google Tag Manager is being used as a tag manager. The website is also using Google Hosted Libraries as a Content Delivery Network (CDN). The website is using jQuery for JavaScript scripting and Google Font API for font rendering. Finally, the editor used to create the website was XXXXXX. [5]

[redacted]

This information will be useful in identifying potential vulnerabilities and attack vectors that can be exploited during penetration testing. Further testing will be conducted in the next phase to identify potential vulnerabilities in the website.

Data Breaches

- **HaveIBeenpwned:** XXXXXX has not been involved in any known data breaches according to the "HaveIBeenPwned" database. This is a good sign for the website's security posture, as a data breach can result in sensitive user data being exposed, leading to potential harm to both the users and the website's reputation. [6]

However, it is important to note that just because the website is not listed on HaveIBeenPwned does not necessarily mean that it has never been involved in a data breach. It is possible that the website has been breached but the breach has not been publicly disclosed or discovered yet.

Vulnerability Scanning

- **Nessus:** I ran three separate Nessus scans to test XXXXXX and the web server XXXXXX along with tcp port 53 to check for known vulnerabilities. Two of the scans were run as basic network scans, one providing a login and port number, and the other did not. The third scan was ran as a web application test, but due to user error, the login parameters were incorrectly input, and a Nessus plugin update caused me to be unable to rerun the web application scan with the correct parameters. The web application test (incomplete) had no vulnerabilities and the two basic network scans, returned the same vulnerabilities:
    - **DNS Server Recursive Query Cache Poisoning Weakness:** This means that the remote DNS server answers to any request made, so it is possible to query the name servers of the root zone and get an answer that is bigger than the original request. By spoofing the source IP address, a remote attacker can leverage this 'amplification' to store fake data in the DNS resolver cache, allowing them to create a mass phishing or ‘pharming’ attack, spread malware, or launch a denial of service attack against a third-party host using the remote DNS server.

      The recommended solution is to restrict your DNS servers from public networks, but as you are a public facing web application, the better option is to reconfigure it to reject such malicious queries.

    - **DNS Server Spoofed Request Amplification DDoS:** This means that it is possible to query the remote name server for third-party names. If this is your internal nameserver, then the attack vector may be limited to employees or guest access when allowed. If you are probing a remote nameserver, then it allows anyone to use it to resolve third party names. This allows attackers to perform cache poisoning attacks against your nameserver and if the host allows these recursive queries via UDP, then the host can be used to 'bounce' Denial of Service attacks against another network or system.

      Restrict recursive queries to the hosts that should use this nameserver (such as those of the LAN connected to it). If you are using bind 8, you can do this by using the instruction 'allow-recursion' in the 'options' section of your named.conf. If you are using bind 9, you can define a grouping of internal addresses using the 'acl' command. Then, within the options block, you can explicitly state: 'allow-recursion { hosts_defined_in_acl }'. If you are using another name server, consult its documentation.

Bibliography

[1] Aboul-Ela, A. (2022) “Sublist3r,” *Kali*. Available at: https://www.kali.org/tools/sublist3r/ (Accessed: April 26, 2023).

[2] “dnsrecon” (2022) *Kali*. Available at: https://www.kali.org/tools/dnsrecon/ (Accessed: April 26, 2023).

[3] *Certificate search* (no date) *Certificate Search*. Available at: https://crt.sh/ (Accessed: April 27, 2023).

[4] “nmap” (2023) *Klia*. Available at: https://www.kali.org/tools/nmap/ (Accessed: April 26, 2023).

[5] no date) *Wappalyzer*. Available at: https://www.wappalyzer.com/ (Accessed: April 26, 2023).

[6] Hunt, T. (no date) “HaveIBeenPwned,” *Have I Been Pwned*. Available at: https://haveibeenpwned.com/ (Accessed: April 26, 2023).

[7] C. Espinosa, “Vulnerability assessment with Nessus,” Alpine Security, 04-Mar-2020. [Online]. Available: https://www.alpinesecurity.com/blog/vulnerability-assessment-with-nessus-home-part-1/. [Accessed: 27-Apr-2023].

[8] T. A. Nidecki, “DNS cache poisoning,” Acunetix, 23-Dec-2022. [Online]. Available: https://www.acunetix.com/blog/web-security-zone/what-is-dns-cache-poisoning/#:~:text=DNS%20cache%20poisoning%20is%20a,pharming)%20and%20to%20spread%20malware. [Accessed: 27-Apr-2023].