Vulnerability :

Web Cache Poisoning

Website :

https://0a350060049bd07cc021561f003b00bc.web-security-academy.net/

Source :

https://portswigger.net/web-security/web-cache-poisoning/exploiting-design-flaws/lab-web-cache-poisoning-with-an-unkeyed-header

Tools Used :

Burp Suite Community Edition was used to perform the attack.

References Used For Demo:

1. https://portswigger.net/web-security/web-cache-poisoning/exploiting-design-flaws/lab-web-cache-poisoning-with-an-unkeyed-header

2. https://www.youtube.com/watch?v=ZsTxt24WqaA

The Reason For Choosing The Above Website:

After days of research, we had to choose between creating a website from scratch and designing the cache vulnerability or using a vulnerable website from an online source that had cache flaws. We opted for choosing a website from the online source as designing a website and designing cache vulnerability seemed difficult. However, we found it difficult to find any website with web cache vulnerabilities except for a few online labs that were specifically designed for carrying out web cache poisoning attacks. We were doubtful on whether to use the labs to demonstrate our attack because they already had the vulnerability designed in them. However, after reaching out to the Curtin Professor/Lecturer/Unit Coordinator (Dr Redowan Mahmud) and the local lecturer(Ms. Chethana Liyanapathirana), we confirmed that we could use the lab for demonstration. A screenshot of the email conversation regarding the confirmation of using the online lab practical for the demonstration is attached in the project folder for reference.

Steps To Perform The Attack:

1. Firstly, the GET request sent to the website have to be taken into to burp suite by turning on the intercept on the proxy tab.

2. After receiving the GET request, next step is to identify unkeyed headers. For this, the burp extension- Param Miner can be used. The request is sent to param miner by right clicking on the request and Extensions > Param Miner > guess headers. After scanning the request, param miner displays the unkeyed headers in the output section of the Extensions tab through which the unkeyed headers can be identified. For the above website, the unkeyed header identified was X-Forwarded-Host.

3. After identifying an unkeyed header, the next step is to identify the vulnerability by exploring the request to the website and the response from the website. This can be done by navigating to the Target tab in burb suit and selecting the correct request to observe both the request and the response from website. Upon close observation, a vulnerability is found where the host is reflected in the response (X-Forwarded-Host is simply a header that identifies the original host requested by the client. This header is manipulated into pointing to something else so that the manipulated response is served to clients).

4. The next step is to send the GET request to the repeater. This can be done by right clicking the request in the proxy tab > send to repeater.

5. Once the request is sent to the repeater, the request must be sent to the server to observe the response. This can be done by clicking send on the Repeater tab to forward the request. Whether or not the response comes through from cache can be identified through the response header X-cache which will be reflected on the response as either a hit or miss.

6. Before exploiting the vulnerability, a cache buster (ex : ?cb=1234) should be added to the request to make sure the response generated from that request is not served to other users requesting the same website (Though cache buster defeats the very purpose of this attack, it is used until the poisoned cache header is generated. Once the payload is successfully inserted into the header, the cache buster must be removed to successfully poison the cache).

7. Once a cache hit is received, the next step is to add a X-Forwarded-Host header to the request with a test host such as example.com and send it to the server until a cache hit is received with a response from the server reflecting the test host in the correct place.

8. Once step 7 is successfully executed, the next step is to poison the unkeyed header X-Forwarded-Host with the malicious payload and send the request with poisoned cache to the server. For that the payload must first be generated. In the demonstration, an exploit server provided by the lab (PortSwigger) is used to get the URL to which the X-Forwarded-Host is set to. The URL is simply another web page that the X-Forwarded-Host is going to redirect the client to. For this demonstration, the web page is simply going to display an alert message with the text we entered in the response header which is text – "I'm Attacked".

9. After setting X-Forwarded-Host to the generated URL, we send the request to the server and observe the response and cache. Once a cache hit is received with the injected URL, it can be noted that the cache is successfully poisoned.

10. After cache is successfully poisoned, the final step is to remove the cache buster and refresh the webpage to get the text-"I'm Attacked" in an alert message (Note: if the intercept is on during this process and burp suite is used, in order to get the alert message, the request must be forwarded by navigating back to the Proxy tab).