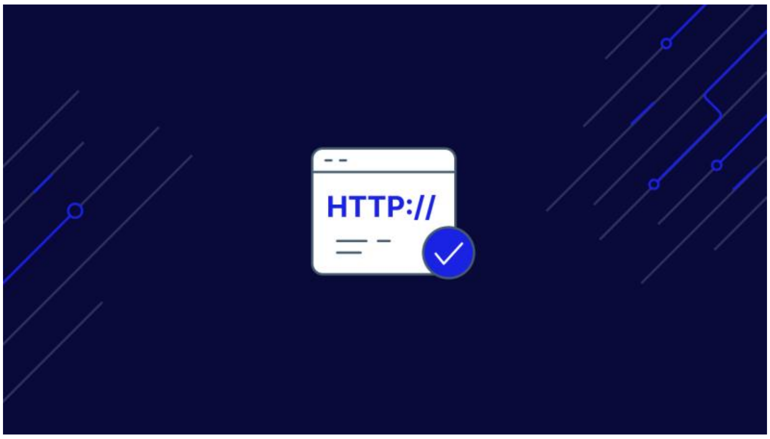
DECEMBER 2, 2023

HOST HEADER INJECTION ATTACKS

SAKTHI AYYAPPAN

HOST HEADER INJECTION ATTACKS



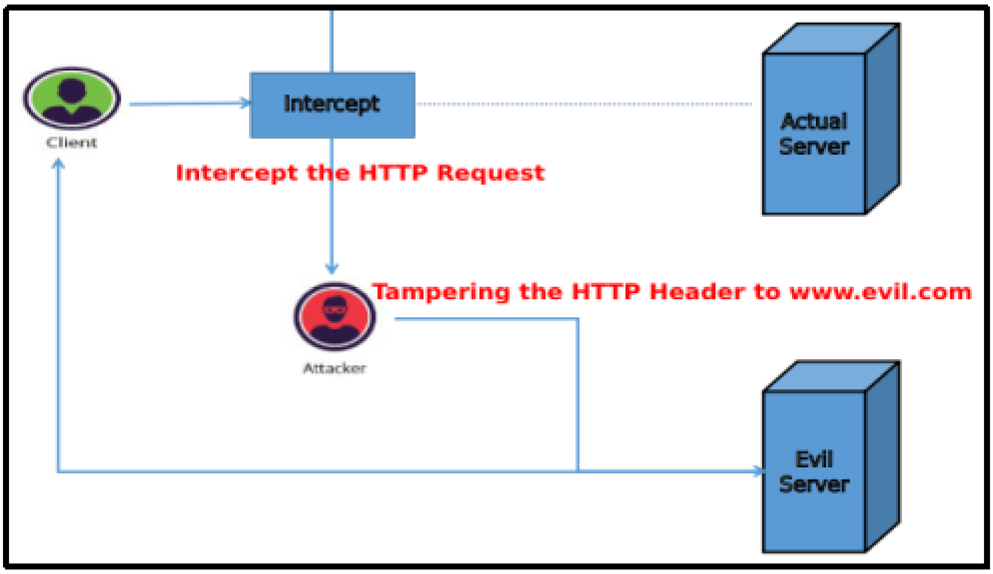
- In the context of HTTP (Hypertext Transfer Protocol), the Host header is an essential piece of information included in an HTTP request.
- It specifies the domain name or hostname of the server that the client (web browser) intends to communicate with.
- This header plays a crucial role in enabling virtual hosting, allowing multiple websites to share a single IP address and port combination.

- When a client initiates an HTTP request to a server, it includes the Host header in the request message.
- The server then utilizes this information to identify the specific website or application that should handle the request.
- This enables a single server to host multiple websites simultaneously, each with its own unique domain name

WHAT IS HOST HEADER INJECTION ?

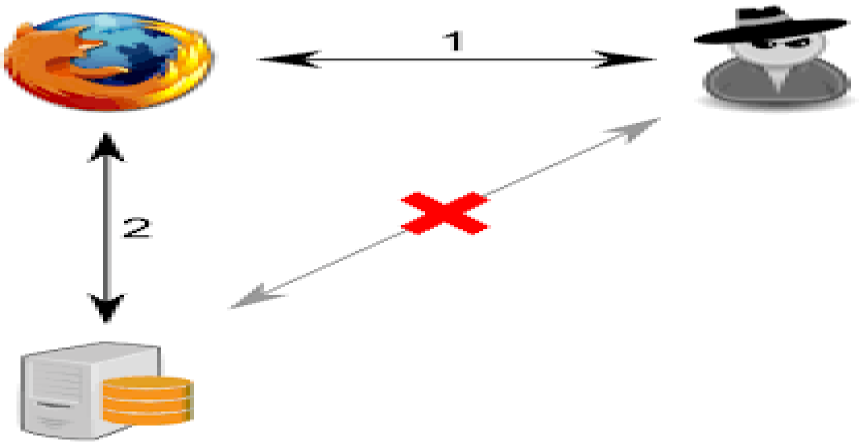
- Host header injection (HHI) is a web security vulnerability that occurs when a web application fails to properly validate or sanitize the Host header sent by an HTTP client.

- This can allow an attacker to inject malicious code into the Host header, which can then be executed by the web application.


Client
Intercept
Actual Server
Intercept the HTTP Request
Tampering the HTTP Header to www.evil.com
Attacker
Evil Server

- WORKING:The Host header is an HTTP header that specifies the domain name of the server that the client is trying to reach.
- When a client sends an HTTP request to a server, it includes the Host header in the request message.
- The server then uses this information to route the request to the correct server.

IMPACT OF HOST HEADER INJECTION



FOR EXAMPLE If an attacker is able to inject malicious code into the Host header which server the request is routed to. This can allow them to perform a variety of attacks, such as:

Bypassing security controls: If the attacker can inject the Host header of a secure website, they may be able to bypass security controls that are only applied to that website.

- For example, the attacker may be able to inject the Host header of a bank website into an HTTP request to a different website.

- This could allow them to access the bank website without having to authenticate themselves.

Denying of service (DoS) attacks:

- The attacker may be able to inject the Host header of a non-existent server into an HTTP request.
- This will cause the server to waste time trying to connect to the non-existent server, which can make the server unavailable to other users.

Content injection attacks:

- The attacker may be able to inject malicious code into the Host header that is then executed by the web application.
- This could allow the attacker to take control of the web application or to steal data from the web application.

MITIGATIONS?

Mitigating HTTP Host Header Injection attacks requires a comprehensive approach that encompasses

- input validation,
- secure coding practices,
- proper server configuration

Here are some effective strategies to safeguard against these attacks:

Input Validation:

Whitelist of Allowed Domains: Implement a whitelist of acceptable domain names that the Host header should contain.

This restricts the range of valid Host headers, preventing malicious actors from injecting unauthorized domains.

Secure Coding Practices:

Avoid Concatenating User Input: Avoid directly concatenating user input into the Host header.

Instead, use parameterized queries or prepared statements to separate user input from the Host header construction.

Escape Special Characters: Properly escape special characters in the Host header to prevent malicious code injection. Use appropriate escaping techniques to neutralize potentially harmful characters.

Validate Host Header Length: Enforce a maximum length limit for the Host header to prevent buffer overflow attacks.

Truncate overly long Host headers to a safe length.

Server Configuration:

Use Strict Host Header Parsing: Configure web servers to strictly parse the Host header, rejecting invalid or unexpected values. This helps prevent malicious actors from exploiting non-compliant Host headers.

Set Default Host Header: Define a default Host header value that the server uses if no Host header is provided. This helps prevent Host header poisoning attacks.

Enable Web Application Firewall (WAF): Implement a WAF to filter incoming HTTP requests and block those containing malicious Host headers. WAFs can provide an additional layer of defense against HHI attacks.

- In addition to these specific measures, it's crucial to follow general secure coding practices, regularly review and update web application code, and keep software components up to date with the latest security patches.

- By implementing these mitigations, organizations can significantly reduce their vulnerability to HTTP Host Header Injection attacks and enhance the overall security of their web applications.

REFERENCES:

https://portswigger.net/web-security/host-header

https://crashtest-security.com/invalid-host-header/#:~:text=A%20successful%20host%20header%20injection,brute%2Dforcing%2C%20and%20more!