Sri Lanka Institute of Information Technology

IE2062

Web Security

Bug Bounty Report VII

Submitted by:

Student Registration Number	Student Name
IT21197550	Nihila Premakanthan

Date of Submission: 28.05.2023

Acknowledgement

I would like to express my special thanks to our mentor Ms. Chethana Liyanapathirana and Dr. Lakmal Rupansighe for their time and efforts she provided throughout the course, and for the Web Security lecture panel for guiding us through this semester and for helping us by giving examples, guidelines, and advice about the project. Your useful advice and suggestions were really helpful to me during the project's completion. In this aspect, I am eternally grateful to you.

Executive Summary

This report aims to provide an overview of the vulnerability identified in a particular domain. The bug bounty platform called Hackerone was used for this purpose. This report analyses the domain of Casper (http://www.casper.com/).

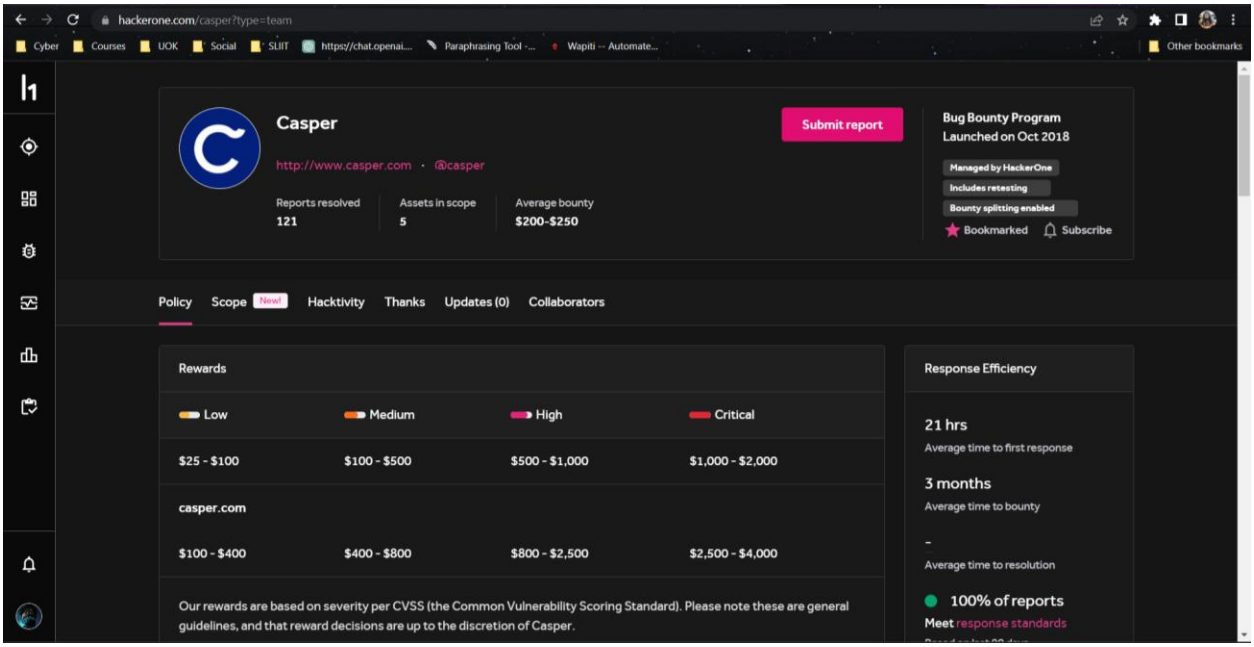
This report uses different tools to gather information detect vulnerabilities and perform penetration testing. The tool name Netsparker and Owasp Zap was mainly used to identify the vulnerability. Further this report provides the vulnerability title, vulnerability description, Affected Components, Impact Assessment, Steps to reproduce the vulnerability, proof of concept and the proposed mitigation.

By including these comprehensive details for each vulnerability, the report provides a comprehensive overview of the security weaknesses present within the system and offers actionable insights for remediation and improvement.

Contents

Introduction

Affected URL: http://www.casper.com/



Vulnerability title

HTTP Security Transport Security(HSTS) not enabled

Security Level:

Risk Level:
MEDIUM

Vulnerability description

A security tool called HTTP Strict Transport Security (HSTS) is intended to shield websites from specific attacks, especially those involving unsecured or unauthorized network connections. In order to ensure that all communication between the browser and the website is encrypted and safe, HSTS tells web browsers to only connect to the website over the HTTPS protocol when the feature is enabled on the website.

A website becomes susceptible to potential security vulnerabilities if HSTS is not enabled on it. Without HSTS, a hacker may be able to intercept user communications by taking advantage of unsecured network connections. Attacks like man-in-the-middle incidents, session hijacking, and the interception of private data are all made possible by this.

The website loses out on the extra layer of security it offers by not activating HSTS. By ensuring that users always connect to the website securely, HSTS helps to guard against accidental HTTP unsafe connections. It also helps defend against downgrade attacks, in which a user's connection is attempted to be forced from HTTPS to HTTP, potentially exposing their data.

A response header must be added to the website's server configuration in order to enable HSTS. This header notifies browsers that the website should only be visited via HTTPS. This header contains the maximum amount of time, or "HSTS max-age," that the browser should keep in mind while remembering to use HTTPS whenever possible for that specific website.

Website administrators should think about configuring their servers to activate HSTS and select a suitable max-age value in order to reduce the vulnerability caused by not having HSTS enabled. They can improve the security of their website and shield customers from potential assaults brought on by unsecured network connections by doing this.

Affected components

The website's communication and security are the main aspects that are impacted when HTTP Strict Transport Security (HSTS) is not enabled. The following are the main factors that may be impacted:

- User Browser: In the absence of HSTS, the user's web browser is not expressly told to always establish a secure connection to the website. Due to the browser's propensity to permit connections using insecure HTTP protocols, this makes the user vulnerable to potential attacks. As a result, potential security concerns can now enter through the user's browser.
- Network Connections: HSTS offers defense against assaults that take place as network connections are being established. Without HSTS, an attacker may be able to leverage insecure network

connections to intercept communications between the user's browser and the website. assaults like session hijacking and man-in-the-middle assaults may result from this.

- Traffic to the website: The absence of HSTS leaves it open to the possibility of eavesdropping and interception. Without HSTS, an attacker may be able to intercept sensitive data, such as login credentials, personal information, or financial information, as it is communicated between the user's browser and the website. This compromises user data security and privacy.
- Downgrade Attacks: Attacks that seek to force a user's connection to switch from the secure HTTPS protocol to the less secure HTTP are known as "downgrade attacks," and HSTS protects against them. The website is vulnerable to such attacks without HSTS, putting users at risk of having their encrypted connections compromised and potentially disclosing their personal information.
- Website Reputation: Poor security precautions, such as the lack of HSTS, can harm a website's reputation. If a website does not require HTTPS connections using HSTS, users may view it as less reliable or secure. This may result in a decline in user engagement and trust as well as possible business repercussions.

Impact assessment

Lack of HTTP Strict Transport Security (HSTS) can have a severe negative effect on a website and increase security and privacy problems. Here are a few possible effects:

- Man-in-the-Middle (MitM) Attack Vulnerability: Without HSTS, the website is open to MitM attacks. Attackers can read, alter, or introduce malicious content into the transferred data by intercepting the communication between the user's browser and the website. This may result in illegal access to sensitive data, like login passwords, private information, or financial information.
- Session Hijacking: HSTS aids in preventing attacks that take advantage of sessions. Without HSTS, a hacker might be able to take over a user's session by taking advantage of unsecured network connections. This gives the attacker the ability to operate in the victim's place, mimic them, and perhaps get access to their accounts without authorization.
- Exposure of Sensitive Information: Without HSTS, there is a chance that sensitive information sent between a user and a website could be made public. Data delivered over insecure HTTP connections can be intercepted without the encryption offered by HTTPS, possibly exposing private and confidential information.

- Attacks that seek to force a user's connection to switch from HTTPS to HTTP are known as "downgrade attacks," which HSTS mitigates. Without HSTS, a hacker can take advantage of this flaw and degrade the connection, leaving the user's data vulnerable to possible interception and manipulation.
- User Trust and Reputation Affected: Users' confidence in the security and privacy of the website may be affected if HSTS isn't enabled. Users might think less favorably of the website and be reluctant to interact with it or disclose sensitive information. A compromised reputation brought on by inadequate security measures can also have detrimental effects on a firm, resulting in a loss of clients and potential earnings.

Steps to reproduce

You would likely need access to the website's server configuration or administrative rights to replicate the situation when HTTP Strict Transport Security (HSTS) is not enabled. The steps to replicate this vulnerability are as follows:

- Determine the destination website: Choose the website for which you wish to check whether HSTS is turned on or off.
- Identify the type of server: Determine the kind of web server the target website is using. Apache, Nginx, IIS (Internet Information Services), and other popular web servers are listed above.
- Access server configuration: Open the web server's administrative interface or the server configuration files. This step typically calls for appropriate authorization or server administrator access.
- Find the server configuration file: Find the particular server configuration file that manages the settings for the website. Depending on the web server being utilized, the file's name and location can change. For instance, the main configuration file for Apache is frequently called "httpd.conf."
- Find the virtual host configuration: How to locate the Virtual Host configuration Find the Virtual Host configuration for the target website in the server configuration file. The website-specific parameters are described in this section.
- Check for the presence of HSTS: Look for the "Strict-Transport-Security" header in the Virtual Host configuration to see if HSTS is enabled. The website's HSTS functionality is enabled by this header. HSTS is not enabled if the header is missing or is commented out.

- Enable HSTS if necessary: You can add the "Strict-Transport-Security" header to the Virtual Host settings if it is missing or commented out. For the duration of which the browser should remember to employ HTTPS for a website, set the appropriate max-age value. For instance, the directive "Strict-Transport-Security: max-age=31536000" tells the browser to keep track of HSTS for a year.
- Save the changes, then do so: To apply the changes, save the changed server configuration file and restart the web server. As a result of this action, clients accessing the website will now receive the HSTS header from the web server.
- Check the HSTS status: Access the destination website using a web browser. Make sure the "Strict-Transport-Security" header is present and correctly specified with the desired max-age value by checking the response headers that were received from the website.

Proof of concept

1. HTTP Strict Transport Security (HSTS) Policy Not Enabled

MEDIUM | 1

Request

```
GET / HTTP/1.1
Host: casper.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 1848.809 Total Bytes Received : 158261 Body Length : 156780 Is Compressed : No


```
HTTP/1.1 200 OK
Set-Cookie: dwac_873c16fc6db9c7b4e6c58b667b=oXwQ59D082CLIau4kRi4OGCivzPAHpx87tY%3D|dw-only|||USD|false|
US%2FEastern|true; Path=/; Secure; SameSite=None
Set-Cookie: cqcid=cdm2nuafxSmj3Sr3Z9tpWGdbOL; Path=/; Secure; SameSite=None
Set-Cookie: cquid=||; Path=/; Secure; SameSite=None
Set-Cookie: sid=oXwQ59D082CLIau4kRi4OGCivzPAHpx87tY; Path=/; Secure; SameSite=None
Set-Cookie: preferredLocales=en_us; Expires=Wed, 24-May-2023 07:19:47 GMT; Path=/; Secure; SameSite=Non
e
Set-Cookie: dwanonymous_189dfebe0e99b5b8f436b5d163a82dd8=cdm2nuafxSmj3Sr3Z9tpWGdbOL; Version=1; Comment
="Demandware anonymous cookie for site Sites-casper_us-Site"; Max-Age=15552000; Expires=Mon, 20-Nov-202
3 07:04:47 GMT; Path=/; Secure; SameSite=None
Set-Cookie: __cq_dnt=0; Path=/; Secure; SameSite=None
Set-Cookie: dw_dnt=0; Path=/; Secure; SameSite=None
Set-Cookie: dwsid=Ng8zTI1kLTe5VlLw03-AavfRnJidjkIr3KEkp9-lPN11Zl5kZ8EHyjUsl0i7cUEkktYxm9carB6YXYMvk7wrl
w==; path=/; HttpOnly; Secure; SameSite=None
x-dw-request-base-id: jgkdjtsZbWQBAAB_
X-Content-Type-Options: nosniff
Server: cloudflare
Expires: Thu, 01 Dec 1994 16:00:00 GMT
Connection: keep-alive
CF-Cache-Status: DYNAMIC
Content-Security-Policy: frame-ancestors 'self'
Pragma: no-cache
vary: accept-encoding
CF-RAY: 7cc3afbdf957b2fd-CMB
Content-Type: text/html;charset=UTF-8
Transfer-Encoding: chunked
Content-Encoding:
Date: Wed, 24 May 2023 07:04:48 GMT
Cache-Control: no-cache, no-store, must-revalidate
```


```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="x-ua-compatible" content="ie=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>The Best Bed for Better Sleep | Casper</title>
<meta name="description" content="Get the sleep you've always dreamed of. Casper's award-winning mattre
sses, sheets &amp; more are quality-crafted and ethically built in the USA. Free shipping &amp; return
s!" />
<meta name="keywords" content="Casper" />
<script src="/cdn-cgi/
```

Proposed mitigation

The following methods can be used to lessen the risk caused by HTTP Strict Transport Security (HSTS) not being enabled:

- Enable HSTS: The main method of mitigation is to make HSTS available on the website. This entails setting up the web server such that it replies to client requests with the "Strict-Transport-Security" header. Set a suitable max-age value to indicate how long the browser should keep the website's HTTPS connection in mind. By doing this, you can increase security by ensuring that any upcoming connections to the website from the user's browser are immediately switched to HTTPS.
- Configure the HSTS preload list: Websites may submit their domains for inclusion in the HSTS preload lists kept up to date by the leading web browsers. Even if a user has never been to the website before, HSTS is immediately implemented for them when a domain is added to the preload list. This adds another layer of security and assists in removing the initial insecure connection.
- Redirect HTTP to HTTPS: Implement a server-side redirect so that all HTTP queries are automatically forwarded to HTTPS. This guarantees that users are automatically redirected to the secure version of the website utilizing HTTPS even if they originally type "http://" in the address bar. By using this method, it is easier to enforce secure connections and less likely that visitors will access the website through unsafe routes.
- Implement SSL/TLS certificates: To enable secure HTTPS connections, install a working SSL/TLS certificate on the web server. The certificate ensures data privacy and integrity by encrypting communications between the user's browser and the website. For HSTS to work successfully, an SSL/TLS certificate that has been properly configured is a requirement.
- Test and monitor HSTS implementation: Make sure the HSTS implementation is being used as intended by regularly testing and monitoring it. Check to see that the server is sending the "Strict-Transport-Security" header and that the max-age value is set appropriately. Additionally, keep an eye out for any potential configuration blunders or faults that can affect how HSTS is used.
- Inform website owners and developers: Inform website administrators and developers of the value of enabling HSTS and adhering to safe coding procedures. Promote the use of security best practices, such as HSTS implementation, throughout the creation and launch of websites.

External Reference

NIST

≡ NVD MENU

Information Technology Laboratory

NATIONAL VULNERABILITY DATABASE

NIST NATIONAL VULNERABILITY DATABASE NVD

VULNERABILITIES

CVE-2019-13498 Detail

Description

One Identity Cloud Access Manager 8.1.3 does not use HTTP Strict Transport Security (HSTS), which may allow man-in-the-middle (MITM) attacks. This issue is fixed in version 8.1.4.

Severity

CVSS Version 3.x | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NIST: NVD | Base Score: [illegible] | Vector: CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N

NVD Analysis use publicly available information to associate vector strings and CVSS scores. We also display any CVSS information provided within the CVE List from the CNA.

Note: NVD Analysis have published a CVSS score for this CVE based on publicly available information at the time of analysis. The CNA has not provided a score within the CVE List.

QUICK INFO

CVE Dictionary Entry:
CVE-2019-13498
NVD Published Date:
07/29/2019
NVD Last Modified:
02/28/2023
Source:
MITRE

References to Advisories, Solutions, and Tools

By selecting these links, you will be leaving NIST webspace. We have provided these links to other web sites because they may have information that would be of interest to you. No inferences should be drawn on account of other sites being referenced, or not, from this page. There may be other web sites that are more appropriate for your purpose. NIST does not necessarily endorse the views expressed, or concur with the facts presented on these sites. Further, NIST does not endorse any commercial products that may be mentioned on these sites. Please address comments about this page to nvd@nist.gov.

Hypertink	Resource
https://github.com/FurqanKhan1/CVE-2019-13498	Exploit, Third Party Advisory
https://support.oneidentity.com/technical-documents/cloud-access-manager/8.1.4/release-notes#TOPIC-1028731	Release Notes

Weakness Enumeration

CWE-ID	CWE Name	Source
CWE-319	Cleartext Transmission of Sensitive Information	NIST