Vulnerability Scan on Chameleon website

CHAMELEON

FOR OUR SMARTER WORLD

Jason Galletti.

High Level Summary

Chameleon is a Static web application hosted in GCP on Google app engine (GAE). The below scans were conducted with automated tools and software using Kali Linux. The findings show the web application is secure, with 1 medium and 1 minor suggestion for Improvement.

Suggested Security Improvement

Enable HTTP Strict Transport Security (HSTS). Enforcing HTTP Strict Transport Security (HSTS) will ensure communication between the browser and server only allows the HTTPS protocol.

Review of Medium Findings – Security headers

Not required; Content Security Policy (CSP) mainly mitigates against XSS attacks, currently the web application does not have an attack vector as it is a static web application.

Required; Anti Clickjacking was tested and confirmed to be vulnerable using Burp Suite professional, sending the X-frame options header will control if a page can be displayed within an iframe on another site, and will mitigate Clickjacking attempts.

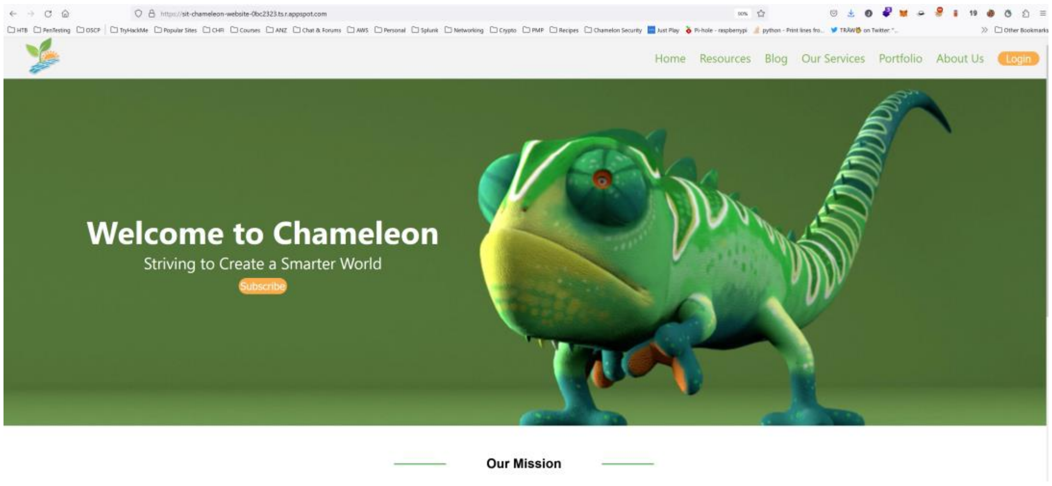
Other Learnings:

As the application is a static web application, the following attacks should be out of scope: injection attacks - A03_2021-Injection.

As the web application is hosted in Google Cloud Platform (GCP) future considerations to take into account is testing using the OWASP Cloud-Native Application Security Top 10, which was not in scope for this report.

Tools Used

Autorecon.py, Nmap, Nessus, Burp Suite Pro, OWASP ZAP, Manual Inspection.



Target: sit-chameleon-website-0bc2323.ts.r.appspot.com

Nmap scan report for sit-chameleon-website-0bc2323.ts.r.appspot.com (142.250.70.244)

rDNS record for **142.250.70.244**: mel05s02-in-f20.1e100.net

SSL-cert: Subject: commonName=*.appspot.com

Nmap Scans:

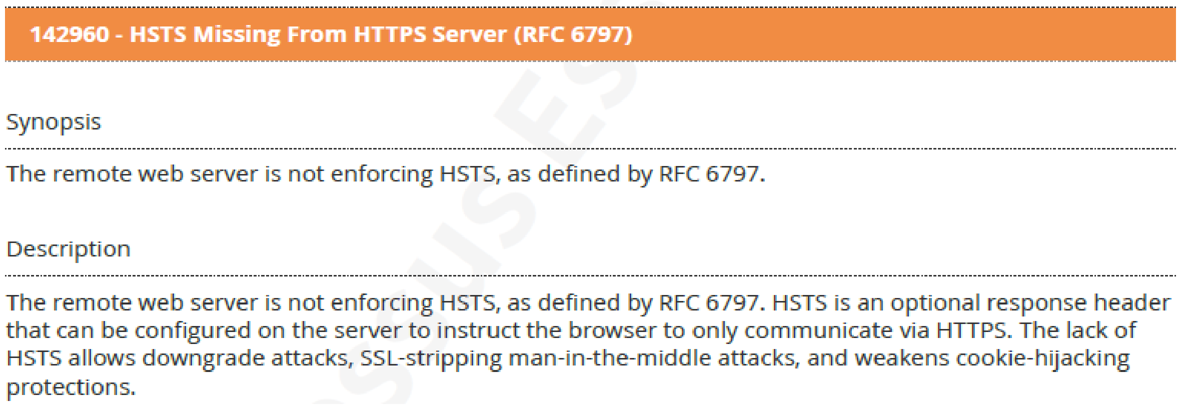
```
nmap -vv --reason -Pn -T4 -sV -sC --version-all -A --osscan-guess
```

```
nmap -vv --reason -Pn -T4 -sU -A --top-ports 100
```

IP Address	Ports Open
142.250.70.244	TCP: 80, 443 | UDP 443

Vulnerability Scanning using Nessus (tenable)

The automated Nessus scan identifies **1 medium security vulnerability**.



142960 - HSTS Missing From HTTPS Server (RFC 6797)

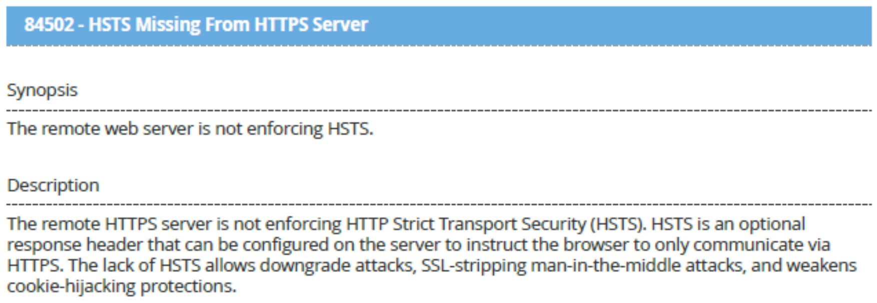
Synopsis

The remote web server is not enforcing HSTS, as defined by RFC 6797.

Description

The remote web server is not enforcing HSTS, as defined by RFC 6797. HSTS is an optional response header that can be configured on the server to instruct the browser to only communicate via HTTPS. The lack of HSTS allows downgrade attacks, SSL-stripping man-in-the-middle attacks, and weakens cookie-hijacking protections.

The automated Nessus scan also provided the following as information.



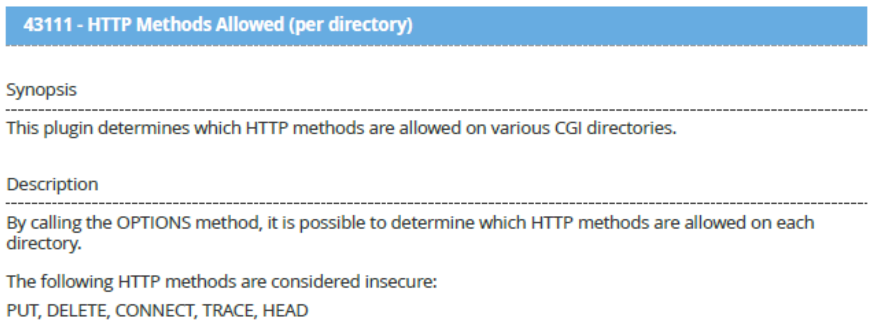
84502 - HSTS Missing From HTTPS Server

Synopsis

The remote web server is not enforcing HSTS.

Description

The remote HTTPS server is not enforcing HTTP Strict Transport Security (HSTS). HSTS is an optional response header that can be configured on the server to instruct the browser to only communicate via HTTPS. The lack of HSTS allows downgrade attacks, SSL-stripping man-in-the-middle attacks, and weakens cookie-hijacking protections.



43111 - HTTP Methods Allowed (per directory)

Synopsis

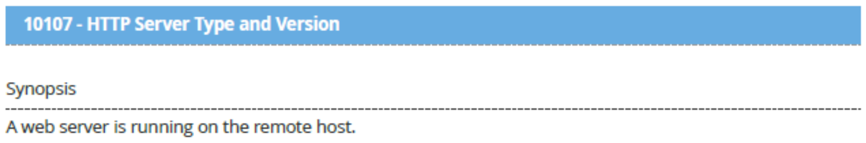
This plugin determines which HTTP methods are allowed on various CGI directories.

Description

By calling the OPTIONS method, it is possible to determine which HTTP methods are allowed on each directory.

The following HTTP methods are considered insecure:

PUT, DELETE, CONNECT, TRACE, HEAD

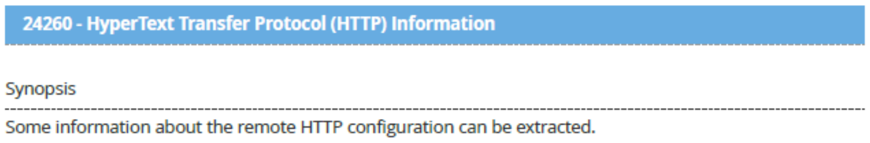


10107 - HTTP Server Type and Version

Synopsis

A web server is running on the remote host.

```
The remote web server type is : Google Frontend
```

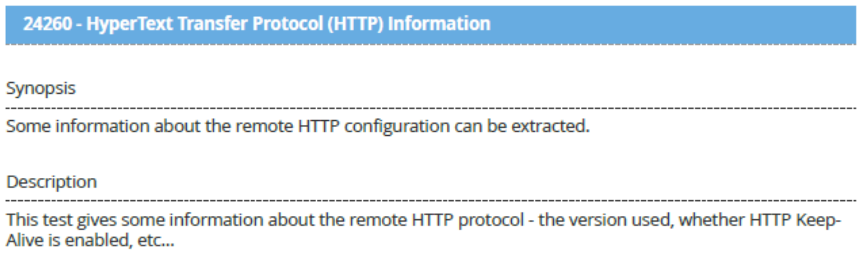


24260 - HyperText Transfer Protocol (HTTP) Information

Synopsis

Some information about the remote HTTP configuration can be extracted.

```
Response Code : HTTP/1.1 200 OK
Protocol version : HTTP/1.1
HTTP/2 TLS Support: No
HTTP/2 Cleartext Support: No
```



24260 - HyperText Transfer Protocol (HTTP) Information

Synopsis

Some information about the remote HTTP configuration can be extracted.

Description

This test gives some information about the remote HTTP protocol - the version used, whether HTTP Keep-Alive is enabled, etc...

Vulnerability Scanning using OWASP ZAP

The automated scan identifies **2 medium & 1 low security vulnerability**.

Generated on Sun, 24 Mar 2024 11:02:15

ZAP Version: 2.14.0

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	2
Low	1
Informational	2

Alerts

Name	Risk Level	Number of Instances
Content Security Policy (CSP) Header Not Set	Medium	5
Missing Anti-clickjacking Header	Medium	1
X-Content-Type-Options Header Missing	Low	3
Information Disclosure - Suspicious Comments	Informational	1
Modern Web Application	Informational	1

Medium	Content Security Policy (CSP) Header Not Set
Description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.

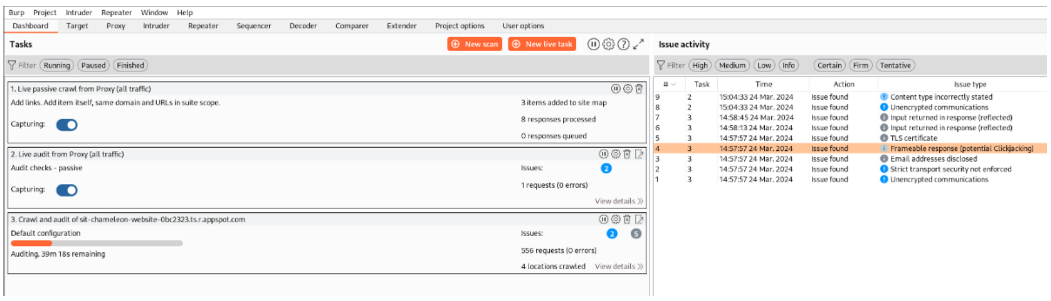
Medium	Missing Anti-clickjacking Header
Description	The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.

Low	X-Content-Type-Options Header Missing
Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.

Clickjacking Testing using Burp suite Pro.

Burp suite professional has a tool to confirm if a web application is vulnerable to clickjacking.

Following the steps in the screenshot below a user can attempt to attack UI buttons to simulate a real-world clickjacking attack.


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
Tasks
New scan
New live task
Filter Running Paused Finished

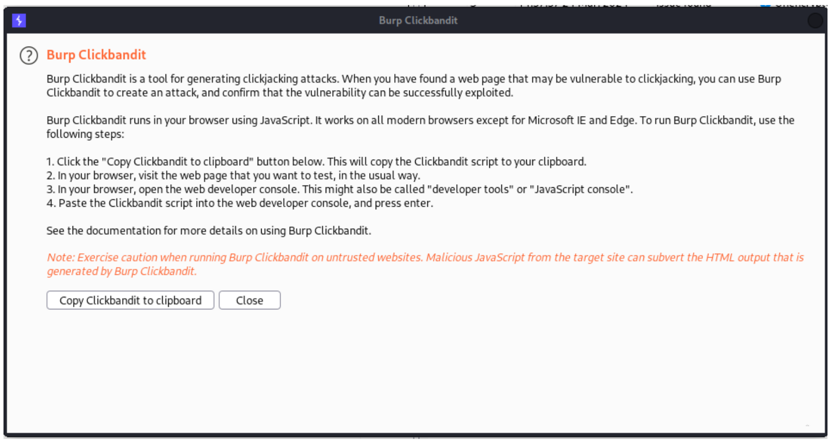
1. Live passive crawl from Proxy (all traffic)
Add links. Add item itself, same domain and URLs in suite scope.
Capturing:
3 items added to site map
8 responses processed
0 responses queued

2. Live audit from Proxy (all traffic)
Audit checks - passive
Capturing:
Issues: 2
1 requests (0 errors)
View details »

3. Crawl and audit of sit-chameleon-website-0bc2323.ts.r.appspot.com
Default configuration
Auditing. 39m 18s remaining
Issues: 2 5
556 requests (0 errors)
4 locations crawled View details »

Issue activity

#	Task	Time	Action	Issue type
9	2	15:04:33 24 Mar. 2024	Issue found	Content type incorrectly stated
8	2	15:04:33 24 Mar. 2024	Issue found	Unencrypted communications
7	3	14:58:45 24 Mar. 2024	Issue found	Input returned in response (reflected)
6	3	14:58:13 24 Mar. 2024	Issue found	Input returned in response (reflected)
5	3	14:57:57 24 Mar. 2024	Issue found	TLS certificate
4	3	14:57:57 24 Mar. 2024	Issue found	Frameable response (potential Clickjacking)
3	3	14:57:57 24 Mar. 2024	Issue found	Email addresses disclosed
2	3	14:57:57 24 Mar. 2024	Issue found	Strict transport security not enforced
1	3	14:57:57 24 Mar. 2024	Issue found	Unencrypted communications


Burp Clickbandit

Burp Clickbandit

Burp Clickbandit is a tool for generating clickjacking attacks. When you have found a web page that may be vulnerable to clickjacking, you can use Burp Clickbandit to create an attack, and confirm that the vulnerability can be successfully exploited.

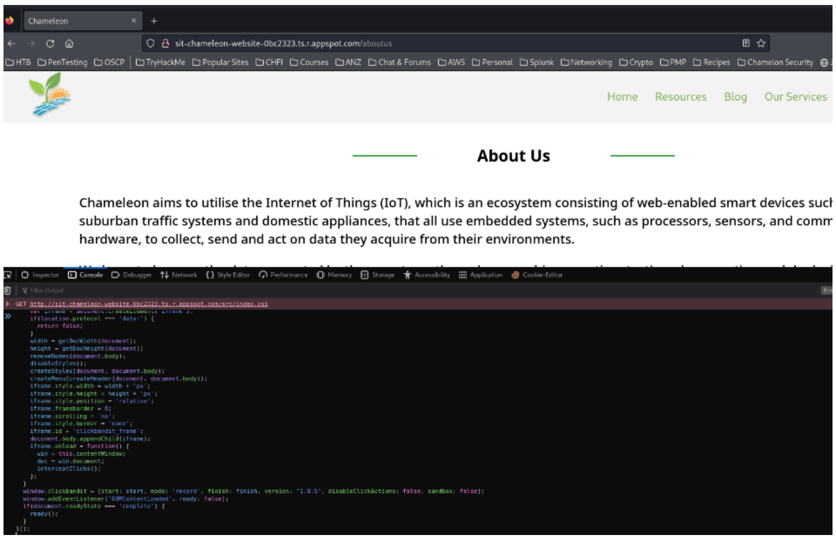
Burp Clickbandit runs in your browser using JavaScript. It works on all modern browsers except for Microsoft IE and Edge. To run Burp Clickbandit, use the following steps:

1. Click the "Copy Clickbandit to clipboard" button below. This will copy the Clickbandit script to your clipboard.
2. In your browser, visit the web page that you want to test, in the usual way.
3. In your browser, open the web developer console. This might also be called "developer tools" or "JavaScript console".
4. Paste the Clickbandit script into the web developer console, and press enter.

See the documentation for more details on using Burp Clickbandit.

Note: Exercise caution when running Burp Clickbandit on untrusted websites. Malicious JavaScript from the target site can subvert the HTML output that is generated by Burp Clickbandit.

Copy Clickbandit to clipboard
Close

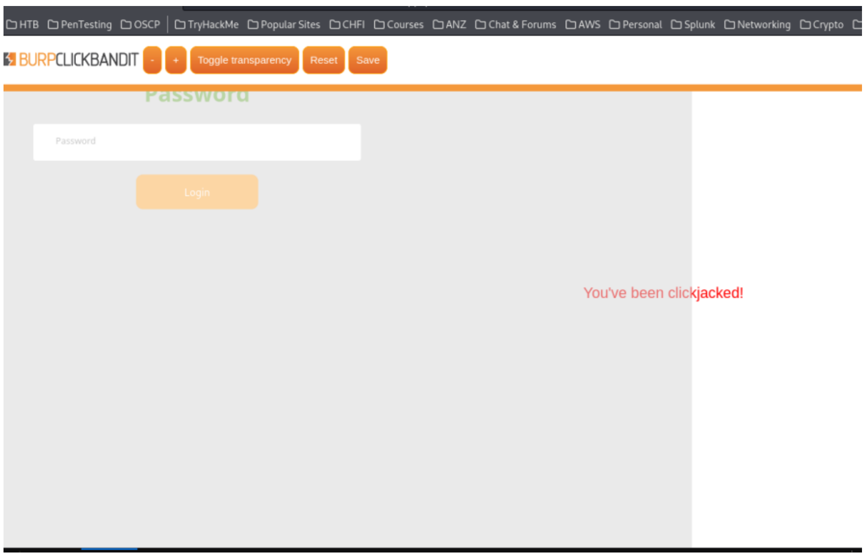

Chameleon
sit-chameleon-website-0bc2323.ts.r.appspot.com/aboutus
HTB PenTesting OSCP TryHackMe Popular Sites CHFI Courses ANZ Chat & Forums AWS Personal Splunk Networking Crypto PMP Recipes Chameleon Security
Home Resources Blog Our Services
Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application Cookie-Editor
Filter Output

About Us

Chameleon aims to utilise the Internet of Things (IoT), which is an ecosystem consisting of web-enabled smart devices suc
suburban traffic systems and domestic appliances, that all use embedded systems, such as processors, sensors, and comm
hardware, to collect, send and act on data they acquire from their environments.

```
GET http://sit-chameleon-website-0bc2323.ts.r.appspot.com/src/index.css
    if(location.protocol === 'data:') {
      return false;
    }
    width = getDocWidth(document);
    height = getDocHeight(document);
    removeNodes(document.body);
    disableStyles();
    createStyles(document, document.body);
    createMenu(createHeader(document, document.body));
    iframe.style.width = width + 'px';
    iframe.style.height = height + 'px';
    iframe.style.position = 'relative';
    iframe.frameborder = 0;
    iframe.scrolling = 'no';
    iframe.style.border = 'none';
    iframe.id = 'clickbandit_frame';
    document.body.appendChild(iframe);
    iframe.onload = function() {
      win = this.contentWindow;
      doc = win.document;
      interceptClicks();
    };
  }
  window.clickbandit = {start: start, mode: 'record', finish: finish, version: "1.0.5", disableClickActions: false, sandbox: false};
  window.addEventListener('DOMContentLoaded', ready, false);
  if(document.readyState === 'complete') {
    ready();
  }
})();
```

By adding the script to the webpage console using Mozilla developer tools this enabled burp suite click bandit simulation to start, I used the login button and was successfully able to overlay a frame from a decoy web application to confirm this vulnerability can be exploited.



References.

https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html

https://security.stackexchange.com/questions/142496/which-security-measures-make-sense-for-a-static-web-site

https://medium.com/google-cloud/web-security-headers-for-your-app-in-gcp-b4cda0197d9e

https://owasp.org/www-project-cloud-native-application-security-top-10/

https://portswigger.net/burp/documentation/desktop/testing-workflow/testing-for-clickjacking

https://portswigger.net/burp/documentation/desktop/tools/clickbandit