

Hands-on Lab: Penetration Testing

Estimated time needed: **45** minutes

About This Lab

In this lab we will perform a port scan and a website scan using online penetration tools.

Objectives

In this hands-on lab, you will:

- Scan a port.
- Scan a website.

Important Notices about This Lab

About Lab Sessions

Lab sessions are not persisted. This means that every time you connect to this lab, a new environment is created for you. Any data or files you saved in a previous session are no longer available. To avoid losing your data, plan to complete these tasks in a single session.

About the Lab Instructions and Solutions

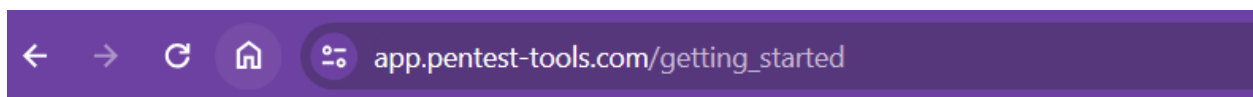
Microsoft Windows operating system features can vary based on the Windows edition. If completing these exercises on your machine, your navigation and solutions may differ from what's presented in this lab.

Exercise 1: Check Password Strength

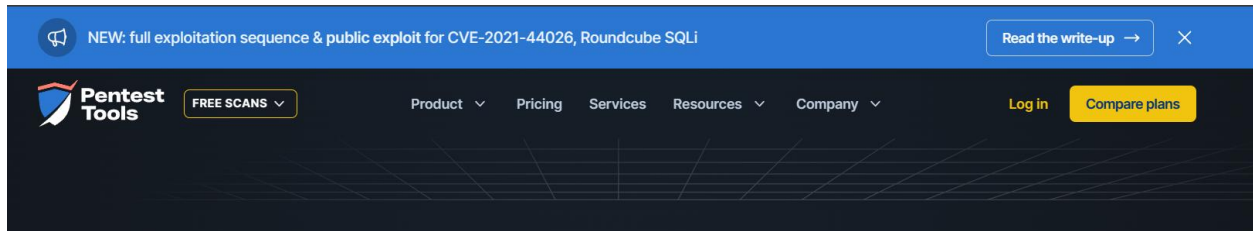
In this exercise we will perform a port scan.

1. Copy and paste the URL into the browser.
1. 1
1. pentest-tools.com

Copied!Wrap Toggled!



2. Click **Log in**.



3. Select **Sign in here** under **Need an account?**

Log in

Email

Password

[Forgot password?](#)


 


☐ Remember me

Need an account?

[Sign up here](#)

4. Accept the terms and conditions by selecting the **I agree to the Terms of Service and Privacy Policy** box and entering your email address.

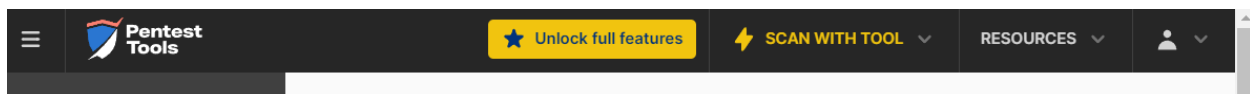
 Continue with Email

 Continue with Google

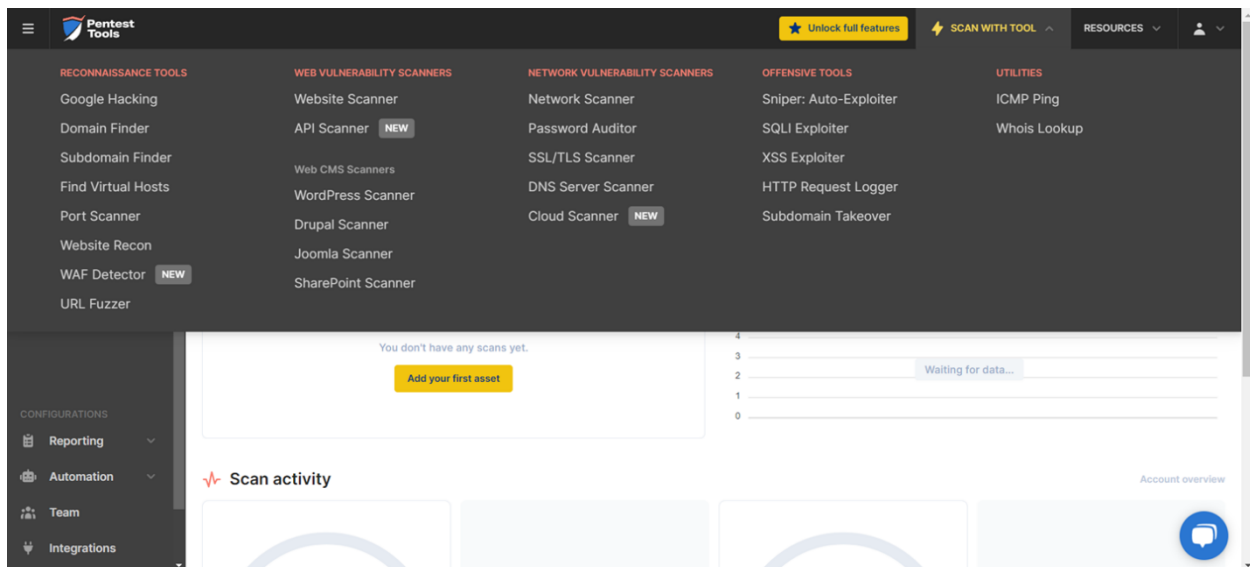
☐ I agree to the [Terms of Service](#) and [Privacy Policy](#).

Already have an account? [Sign in now](#)

5. Once logged in, select **SCAN WITH TOOL** at the top of the page.



6. Here you see a list of a few dozen tools and scanners used for reconnaissance, web vulnerability testing, network vulnerability scanning, and more active forms of penetration testing. Select **Port Scanner**.



7. Copy and paste the following URL into the IP Address text field.

1. 1

1. scanme.nmap.org

Copied!Wrap Toggled!

The screenshot shows the 'Port Scanner' interface of the 'Pentest Tools' application. The left sidebar contains a 'WORKSPACES' section with 'My Workspace' selected, and a 'CONFIGURATIONS' section with 'Reporting', 'Automation', 'Team', and 'Integrations'. The main area is titled 'Port Scanner' and contains the following fields and options:

- Target:** A text input field with the placeholder 'Hostname / IP Address'.
- Scan type:** Radio buttons for 'Light' (selected), 'Deep', and 'Custom'.
- Protocol type:** Radio buttons for 'TCP' (selected) and 'UDP'.
- Scan options:** A toggle switch for 'ON' (checked) with the label 'Check if host is alive before scanning'.
- Custom scan time:** A toggle switch for 'OFF'.
- Notifications:** Radio buttons for 'Workspace' (selected), 'Custom', and 'None'.

A blue information box states: 'You will receive alerts according to the notification settings configured for this workspace. You have 0 enabled notifications.' At the bottom, there is a checkbox for 'I am authorized to scan this target and I agree to the Terms of Service' and a yellow 'Start Scan' button.

8. Verify if the following options are selected:

- Scan type: Light
- Protocol type: TCP
- Scan options: ON (Check if host is alive before scanning)
- Custom scan time: OFF
- Notifications: Workspace
- I am authorized to scan this target and I agree to the Terms of Service: (checked)

Pentest Tools

WORKSPACES

- My Workspace

Dashboard

Assets 0

Scans

Findings

Attack Surface

Handlers

CONFIGURATIONS

- Reporting
- Automation
- Team
- Integrations

Port Scanner

Target:

Scan type: ☒ Light ☐ Deep ☐ Custom

Protocol type: ☒ TCP ☐ UDP

Scan options: ☒ Check if host is alive before scanning

Custom scan time:

Notifications: ☒ Workspace ☐ Custom ☐ None

You will receive alerts according to the [notification settings](#) configured for this workspace. You have [0 enabled](#) notifications.

☐ I am authorized to scan this target and I agree to the [Terms of Service](#)

Start Scan

9. Click **Start Scan**.

Pentest Tools

WORKSPACES

- My Workspace

Dashboard

Assets 0

Scans

Findings

Attack Surface

Handlers

CONFIGURATIONS

- Reporting
- Automation
- Team
- Integrations

Port Scanner

Target:

Scan type: ☒ Light ☐ Deep ☐ Custom

Protocol type: ☒ TCP ☐ UDP

Scan options: ☒ Check if host is alive before scanning

Custom scan time:

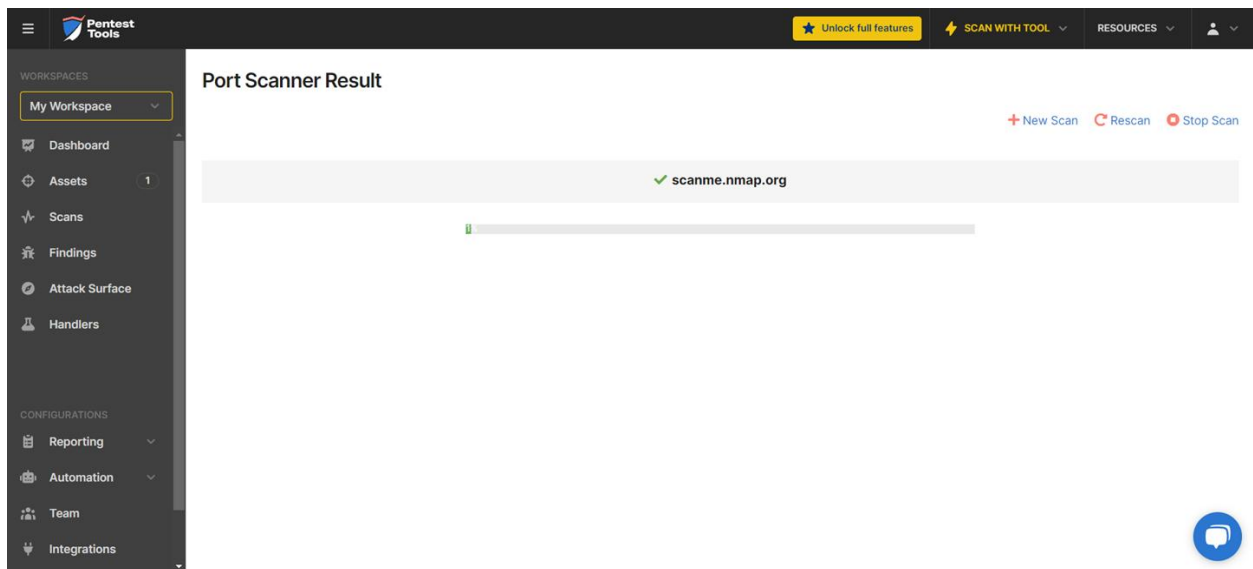
Notifications: ☒ Workspace ☐ Custom ☐ None

You will receive alerts according to the [notification settings](#) configured for this workspace. You have [0 enabled](#) notifications.

☒ I am authorized to scan this target and I agree to the [Terms of Service](#)

Start Scan

10. You will see a progress bar indicating that your scan has begun.



11. The result is displayed once the scan is complete. You will see the following:

- **Port Number:** It indicates specific services or processes identified for communication.
- **State:** It refers to the status of a port. The major states are open, closed, and filtered. An open port is where an application actively accepts TCP connections, UDP datagrams, or SCTP associations. A closed port is accessible, but there's no application listening. A filtered state means a firewall or similar software is protecting the port.
- **Service Name:** It identifies specific services running on a computer, like HTTP for web servers or SMTP for mail servers.
- **Service Product:** It refers to the specific software running the service on the port. It could be Apache, AWS, Microsoft, etc., depending on the software installed on the server.
- **Service Version:** This is the specific version of the software running the service.

MY SCAN

Port Scanner IP Finished

Target
scanme.nmap.org

Export Results New scan Rescan Schedule Report incorrect result

Summary **Results** Scan Parameters

Summary

2 open ports

Start time
Aug 05, 2025 - 06:50

Finish time
Aug 05, 2025 - 06:50

Scan duration
14 seconds

Results Show as raw output

Host
45.33.32.156
scanme.nmap.org
Unknown

Ports

Port Number	Protocol	State	Service	Product	Version	
22	TCP	open	ssh	OpenSSH	6.6.1p1 Ubuntu 2ubuntu2.13	⋮
80	TCP	open	http	Apache httpd	2.4.7	⋮

Scan Parameters

Host
scanme.nmap.org

Protocol
TCP

Scan type
Light

Ports
Top 1000 ports

Check alive
True

Detect svc version
True

Detect OS
False

Traceroute
False

Pentest Tools Unlock full features SCAN WITH TOOL RESOURCES

WORKSPACES
My Workspace

Dashboard
Assets 1
Scans
Findings
Attack Surface
Handlers

CONFIGURATIONS
Reporting
Automation
Team
Integrations

45.33.32.156
scanme.nmap.org

Port Number	State	Service Name	Service Product	Service Version	Service Extra Info	Actions
22	open	ssh	OpenSSH	6.6.1p1 Ubuntu 2ubuntu2.13	Ubuntu Linux; protocol 2.0	Scan with
80	open	http	Apache httpd	2.4.7	(Ubuntu)	Scan with

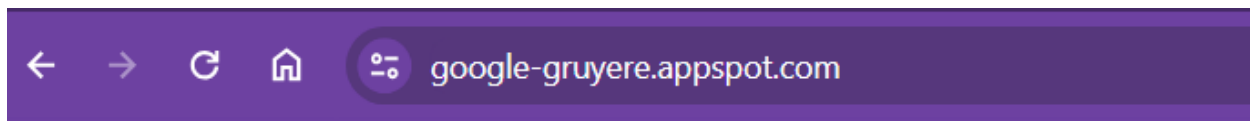
Scan parameters
Host: scanme.nmap.org
Protocol: Tcp
Scan type: Light
Ports: Top 100 ports
Check alive: True
Detect svc version: True
Detect OS: False
Traceroute: False
Scan Technique: TCP SYN

Scan information
Start time: Jan 08, 2024 / 18:23:26
Finish time: Jan 08, 2024 / 18:23:38
Scan duration: 12 sec
Scan status: Finished

Note: scanme.nmap.org is a service provided by the Nmap Security Scanner Project. It is setup for learning and testing. Anyone is authorized to scan this machine with Nmap or other port scanners. Other penetration tests are not permitted. For additional information, visit scanme.nmap.org.

Exercise 2: Perform a Website Scan

1. In a second browser window, visit google-gruyere.appspot.com.



2. Scroll down to the bottom of the page. Carefully read the warning at the bottom of the page and click **Continue**.

To get the most out of this lab, you should have some familiarity with how a web application works (e.g., general knowledge of HTML, templates, cookies, AJAX, etc.).

Gruyere

This codelab is built around **Gruyere** /gru:ˈjɑːr/ - a small, cheesy web application that allows its users to publish snippets of text and store assorted files. "Unfortunately," Gruyere has multiple security bugs ranging from cross-site scripting and cross-site request forgery, to information disclosure, denial of service, and remote code execution. The goal of this codelab is to guide you through discovering some of these bugs and learning ways to fix them both in Gruyere and in general.

The codelab is organized by types of vulnerabilities. In each section, you'll find a brief description of a vulnerability and a task to find an instance of that vulnerability in Gruyere. Your job is to play the role of a malicious hacker and find and exploit the security bugs. In this codelab, you'll use both black-box hacking and white-box hacking. In **black box hacking**, you try to find security bugs by experimenting with the application and manipulating input fields and URL parameters, trying to cause application errors, and looking at the HTTP requests and responses to guess server behavior. You do not have access to the source code, although understanding how to view source and being able to view http headers (as you can in Chrome or LiveHTTPHeaders for Firefox) is valuable. Using a web proxy like **Burp** or **ZAP** may be helpful in creating or modifying requests. In **white-box hacking**, you have access to the source code and can use automated or manual analysis to identify bugs. You can treat Gruyere as if it's open source: you can read through the source code to try to find bugs. Gruyere is written in Python, so some familiarity with Python can be helpful. However, the security vulnerabilities covered are not Python-specific and you can do most of the lab without even looking at the code. You can run a local instance of Gruyere to assist in your hacking: for example, you can create an administrator account on your local instance to learn how administrative features work and then apply that knowledge to the instance you want to hack. Security researchers use both hacking techniques, often in combination, in real life.

We'll tag each challenge to indicate which techniques are required to solve them:

- Challenges that can be solved just by using black box techniques.
- Challenges that require that you look at the Gruyere source code.
- Challenges that require some specific knowledge of Gruyere that will be given in the first hint.

WARNING: Accessing or attacking a computer system without authorization is illegal in many jurisdictions. While doing this codelab, you are specifically granted authorization to attack the Gruyere application as directed. You may not attack Gruyere in ways other than described in this codelab, nor may you attack App Engine directly or any other Google service. You should use what you learn from the codelab to make your own applications more secure. You should not use it to attack any applications other than your own, and only do that with permission from the appropriate authorities (e.g., your company's security team).

Continue >>

© Google 2017 [Terms of Service](#)
The code portions of this codelab are licensed under the Creative Commons Attribution-No Derivative Works 3.0 United States license <<https://creativecommons.org/licenses/by-nd/3.0/us/>>. Brief excerpts of the code may be used for educational or instructional purposes provided this notice is kept intact. Except as otherwise noted the remainder of this codelab is licensed under the Creative Commons Attribution 3.0 United States license <<https://creativecommons.org/licenses/by/3.0/us/>>.

3. Select the URL at the top of the page - <https://google-gruyere.appspot.com/start>.

Web Application Exploits and Defenses (Part 1)

A Codelab by Bruce Leban, Mugdha Bendre, and Parisa Tabriz

Setup

To access Gruyere, go to <https://google-gruyere.appspot.com/start>. AppEngine will start a new instance of Gruyere for you, assign it a unique id and redirect you to <https://google-gruyere.appspot.com/123/> (where 123 is your unique id). Each instance of Gruyere is "sandboxed" from the other instances so your instance won't be affected by anyone else using Gruyere. You'll need to use your unique id instead of 123 in all the examples. If you want to share your instance of Gruyere with someone else (e.g., to show them a successful attack), just share the full URL with them including your unique id.

The Gruyere source code is available online so that you can use it for white-box hacking. You can browse the source code at <https://google-gruyere.appspot.com/code/> or download all the files from <https://google-gruyere.appspot.com/gruyere-code.zip>. If you want to debug it or actually try fixing the bugs, you can download it and run it locally. You do not need to run Gruyere locally in order to do the lab.

▶ **Running locally**

Reset Button

As noted above, each instance is sandboxed so it can't consume infinite resources and it can't interfere with anyone else's instance. Notwithstanding that, it is possible to put your Gruyere instance into a state where it is completely unusable. If that happens, you can push a magic "reset button" to wipe out all the data in your instance and start from scratch. To do this, visit this URL with your instance id:

<https://google-gruyere.appspot.com/resetbutton/123>

About the Code

Gruyere is small and compact. Here is a quick rundown of the application code:

- `gruyere.py` is the main Gruyere web server
- `data.py` stores the default data in the database. There is an administrator account and two default users.
- `gtl.py` is the Gruyere template language
- `sanitize.py` is the Gruyere module used for sanitizing HTML to protect the application from security holes.
- `resources/...` holds all template files, images, CSS, etc.

Table of Contents

- Beat the hackers
- Gruyere
- Set-up
 - Reset Button
 - About the Code
 - Features and Technologies
- Using Gruyere
- Cross-Site Scripting (XSS)
 - XSS Challenges
 - File Upload XSS
 - Reflected XSS
 - Stored XSS
 - Stored XSS via HTML Attribute
 - Stored XSS via AJAX
 - Reflected XSS via AJAX
 - More about XSS
- Client-State Manipulation
 - Elevation of Privilege
 - Cookie Manipulation
- Cross-Site Request Forgery (XSRF)
 - XSRF Challenge
 - More about preventing XSRF
- Cross Site Script Inclusion (XSSI)
 - XSSI Challenge
- Path Traversal
 - Information disclosure via path traversal
 - Data tampering via path traversal
- Denial of Service
 - DoS - Quit the Server
 - DoS - Overloading the Server
 - More on Denial of Service
- Code Execution

4. Read the page carefully and click **Agree & Start**.

Start Gruyere

Your Gruyere instance id is 531643138450435746918947915538943383519.

**WARNING: Gruyere is not secure.
Do not upload any personal or private data.**

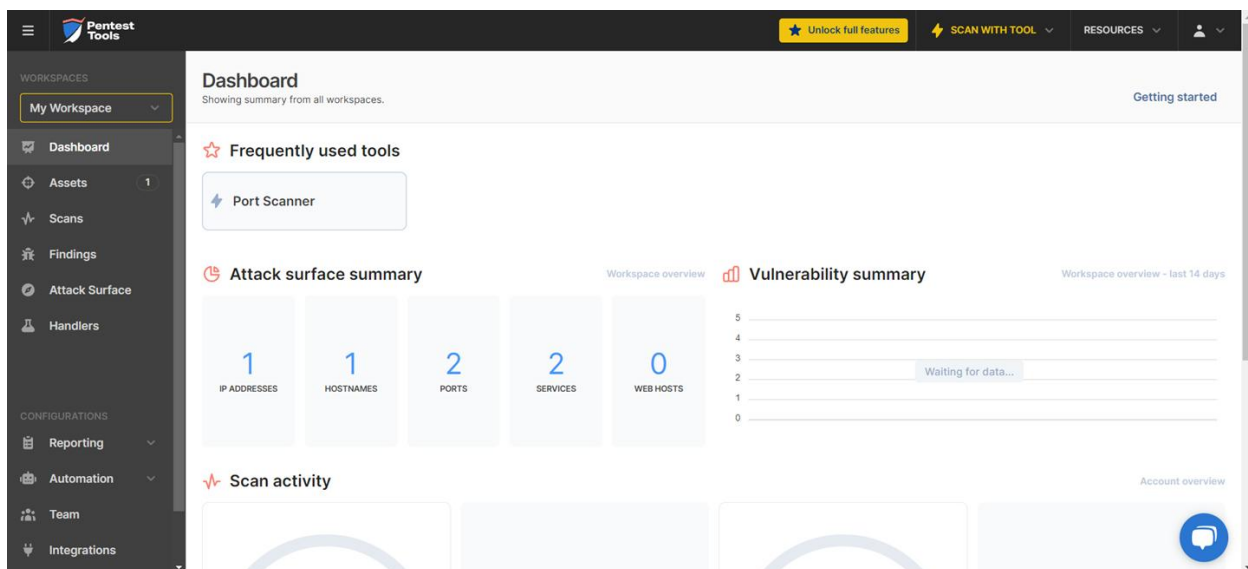
By using Gruyere you agree to the [terms of service](#).

Agree & Start

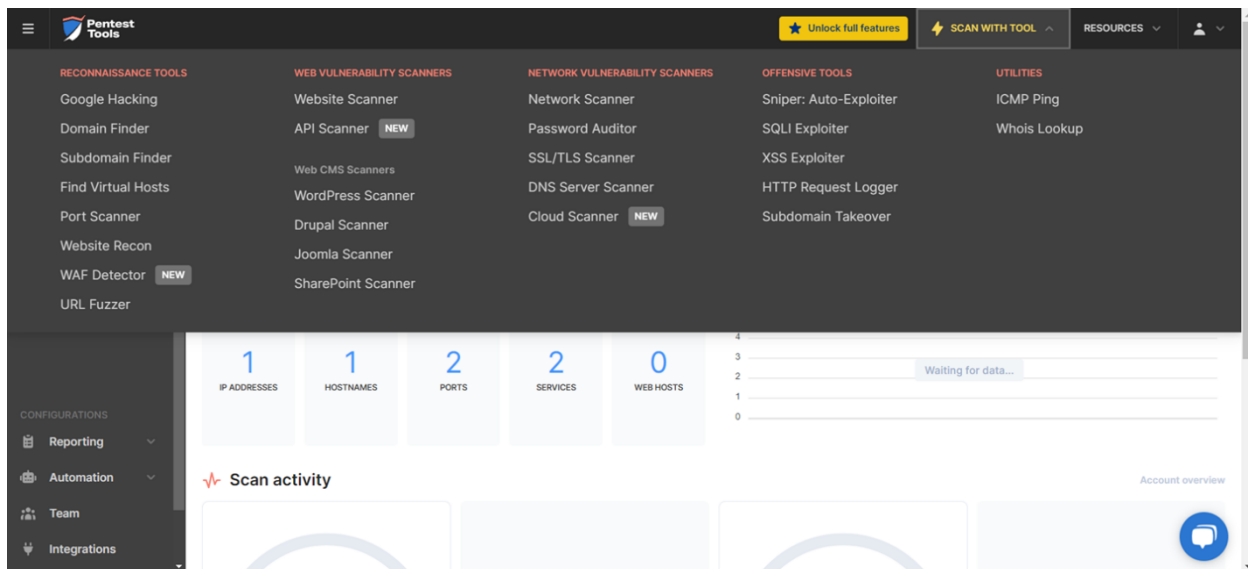
5. Copy and paste the URL from the address bar into your browser (yours will look similar to this screenshot, but the instance numbers will be differ).



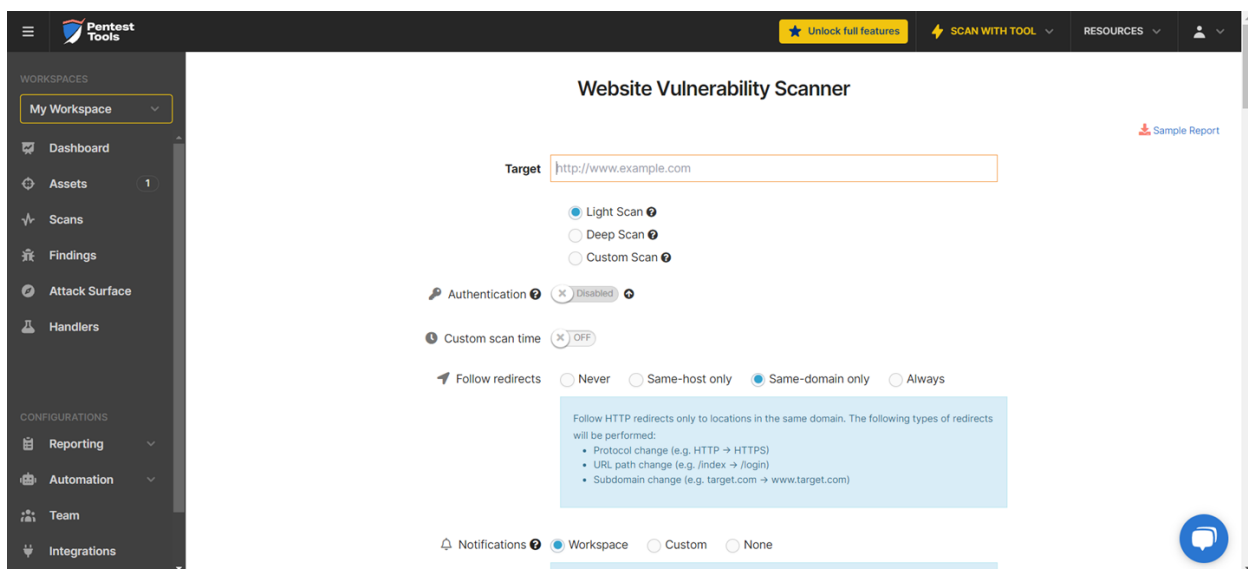
6. Return to the browser tab where you have the pentest-tools.com open. Select **SCAN WITH TOOL** at the top of the page.



7. Select **Website Scanner**.

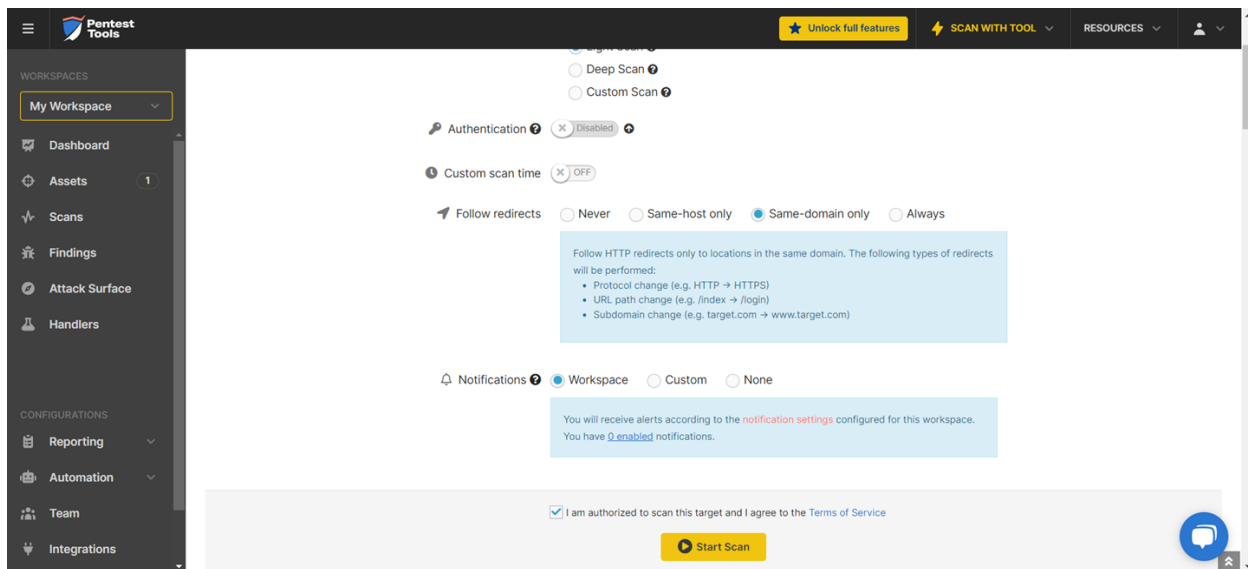


8. Paste your google gruyere URL into the target space.

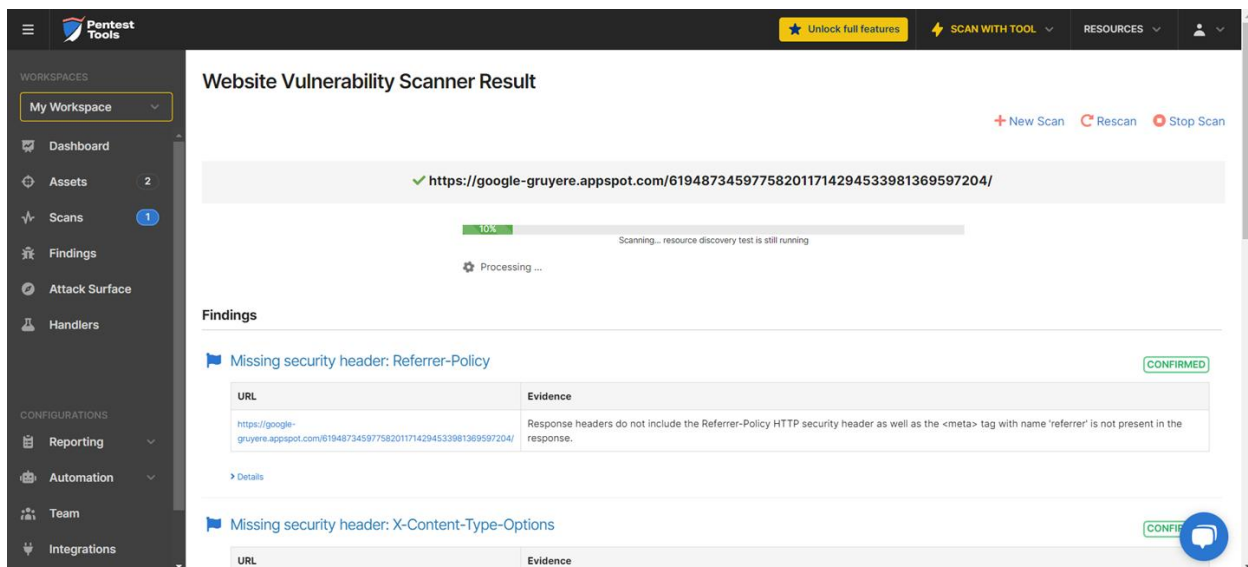


9. Verify if the following options are checked and then click **Start Scan** :

- Scan type: Light scan
- Authentication: Disabled
- Custom scan time: OFF
- Follow redirects: Same-domain only
- Notifications: Workspace
- I am authorized to scan this target and I agree to the Terms of Service: (checked)



10. You will see a progress bar showing your scanning is running.



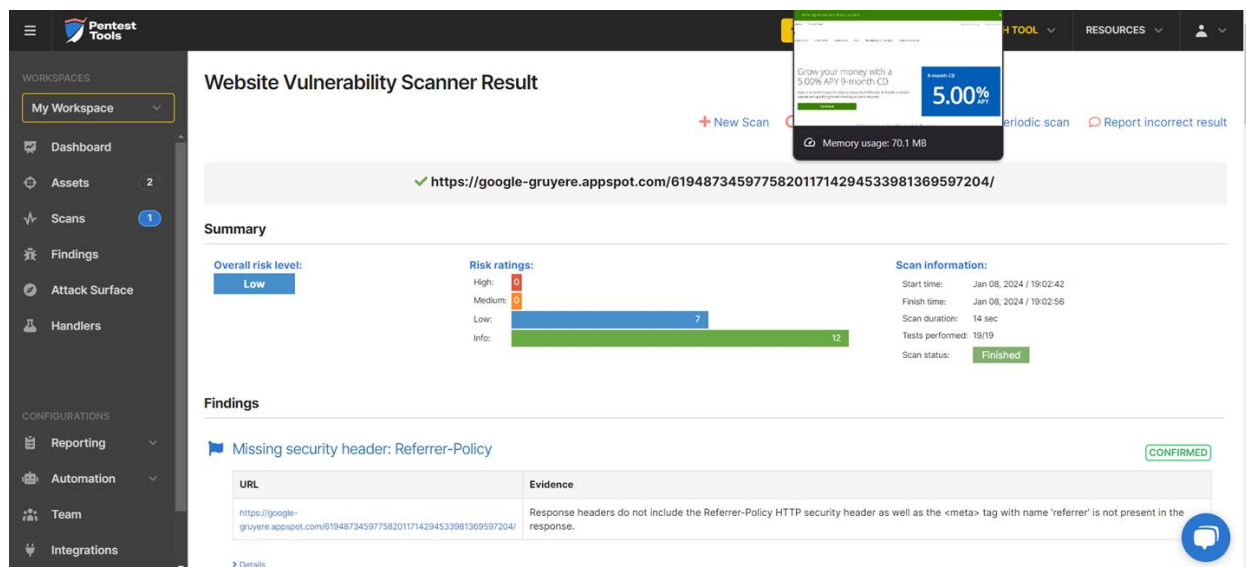
11. You will see your website vulnerability scanner results when your scan is complete:

- **Summary:** Note the high, medium, low, and info risk ratings at the top of the page. Because this is a testing site, the high and medium risks will be low, but if you were scanning a real site, you would want to be very aware of any high or medium risk ratings.
- **Findings:** Note the findings of the scan. Most likely, you will see some, if not all, of the following:
 - Missing security header: x-frame-options – The x-frame-options header indicates whether the browser should render the resource using a frame or an iframe.
 - Missing security header: content-security-policy – The Content Security Policy

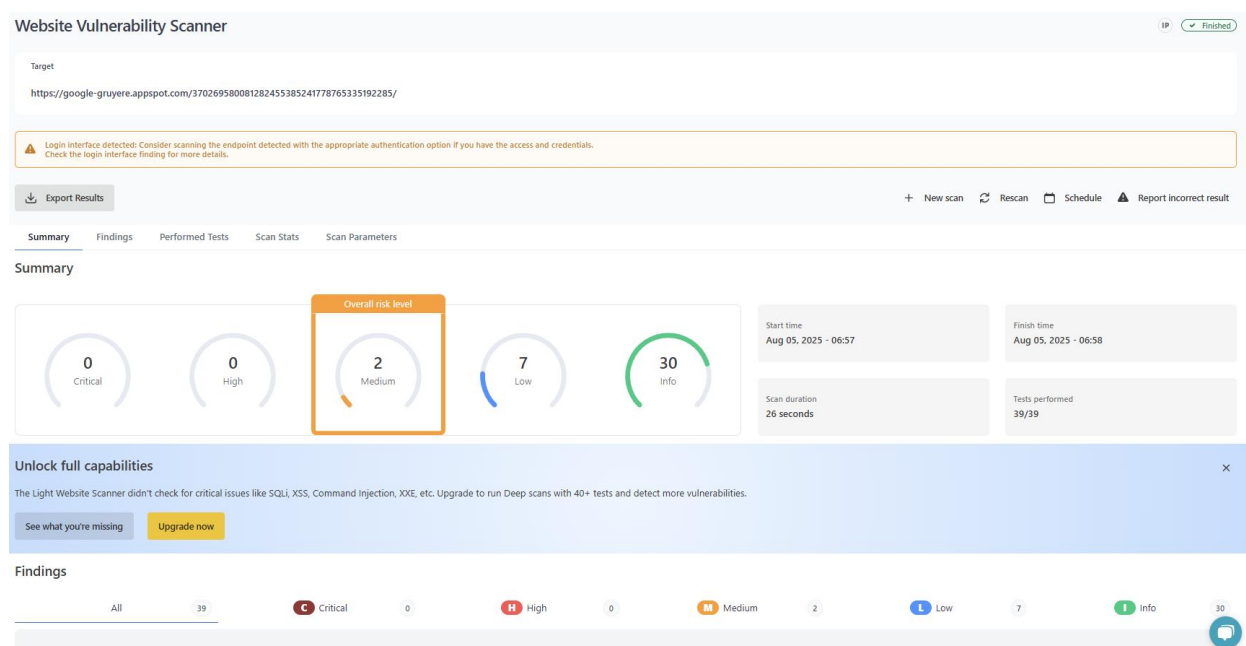
(CSP) is an HTTP header that site owners use to set security rules that the browser must follow – such as approved content sources. It's used to help mitigate cross-site scripting and clickjacking attacks.

- txt file found - Any .txt files stored on the site are listed.

- Security.txt file not found: Although there is no actual risk associated with not having a security.txt file, many cybersecurity experts recommend that every site have one. The file is designed to make it easier for ethical hackers or researchers to reach your organization when they want to report a security vulnerability.



MY SCAN



Evidence

URL

https://google-gruyere.appspot.com/370269580081282455385241778765335192285/saveprofile

Cookie Name

GRUYERE

The server responded with Set-Cookie header(s) that does not specify the HttpOnly flag:
Set-Cookie: GRUYERE=31391908|1d3d2d231d2dd4|author

Request / Response

7

8 > HTTP/1.1 200 OK

9 > cache-control: no-cache

10 > content-type: text/html

11 > pragma: no-cache

12 > set-cookie: GRUYERE=31391908|1d3d2d231d2dd4|author; path=/370269580081282455385241778765335192285

13 > x-xss-protection: 0

14 > x-cloud-trace-context: 6b0419ccf4a424d87ecfdbcf5ae7f8d8

15 > date: Tue, 05 Aug 2025 13:58:06 GMT

16 > server: Google Frontend

17 > Content-Length: 2224

Expand

+ Details

M

Insecure cookie setting: missing Secure flag

Confirmed

443 / TCP

M

Insecure cookie setting: missing Secure flag

Confirmed

443 / TCP

Evidence

URL

https://google-gruyere.appspot.com/370269580081282455385241778765335192285/saveprofile

Cookie Name

GRUYERE

Set-Cookie: GRUYERE=31391908|1d3d2d231d2dd4|author

Request / Response

7

8 > HTTP/1.1 200 OK

9 > cache-control: no-cache

10 > content-type: text/html

11 > pragma: no-cache

12 > set-cookie: GRUYERE=31391908|1d3d2d231d2dd4|author; path=/370269580081282455385241778765335192285

13 > x-xss-protection: 0

14 > x-cloud-trace-context: 6b0419ccf4a424d87ecfdbcf5ae7f8d8

15 > date: Tue, 05 Aug 2025 13:58:06 GMT

16 > server: Google Frontend

17 > Content-Length: 2224

Expand

+ Details

I

Missing security header: X-Content-Type-Options

Confirmed

Missing security header: X-Content-Type-Options

Confirmed

443 / TCP

Evidence

URL
https://google-gruyere.appspot.com/370269580081282455385241778765335192285/

Response headers do not include the X-Content-Type-Options HTTP security header

Request / Response

1 Missing X-Content Header Passive Scan Request&Response

2

3 < GET /370269580081282455385241778765335192285/ HTTP/1.1

4 < Host: google-gruyere.appspot.com

5 < User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36

6 <

7

8 > HTTP/1.1 200 OK

9 > cache-control: no-cache

10 > content-type: text/html

Expand

+ Details

Missing security header: Referrer-Policy

Confirmed

443 / TCP

Missing security header: Referrer-Policy

Confirmed

443 / TCP

Evidence

URL
https://google-gruyere.appspot.com/370269580081282455385241778765335192285/

Response headers do not include the Referrer-Policy HTTP security header as well as the <meta> tag with name 'referrer' is not present in the response.

Request / Response

1 Missing Referrer Header Passive Scan Request&Response

2

3 < GET /370269580081282455385241778765335192285/ HTTP/1.1

4 < Host: google-gruyere.appspot.com

5 < User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36

6 <

7

8 > HTTP/1.1 200 OK

9 > cache-control: no-cache

10 > content-type: text/html

Expand

+ Details

Missing security header: Strict-Transport-Security

Confirmed

443 / TCP

Missing security header: Strict-Transport-Security

Confirmed

443 / TCP

Evidence

URL
<https://google-gruyere.appspot.com/370269580081282455385241778765335192285/>

Response headers do not include the HTTP Strict-Transport-Security header

Request / Response

```
1 Missing HTS Header Passive Scan Request&Response
2
3 < GET /370269580081282455385241778765335192285/ HTTP/1.1
4 < Host: google-gruyere.appspot.com
5 < User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36
6 <
7
8 > HTTP/1.1 200 OK
9 > cache-control: no-cache
10 > content-type: text/html
```

Expand

+ Details

Missing security header: Content-Security-Policy

Confirmed

Missing security header: Content-Security-Policy

Confirmed

443 / TCP

Evidence

URL
<https://google-gruyere.appspot.com/370269580081282455385241778765335192285/>

Response does not include the HTTP Content-Security-Policy security header or meta tag

Request / Response

```
1 Missing CSP Header Passive Scan Request&Response
2
3 < GET /370269580081282455385241778765335192285/ HTTP/1.1
4 < Host: google-gruyere.appspot.com
5 < User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36
6 <
7
8 > HTTP/1.1 200 OK
9 > cache-control: no-cache
10 > content-type: text/html
```

Expand

+ Details

Robots.txt file found

Confirmed

🔍

Robots.txt file found

Confirmed

⬆️

👤

443 / TCP

Evidence

URL

https://google-gruyere.appspot.com/robots.txt

+ Details

🔍

Password Submitted in URL

Unconfirmed

?

⬆️

👤

443 / TCP

Evidence

+ URL

https://google-gruyere.appspot.com/370269580081282455385241778765335192285/editprofile.gtl

Method
GET

Parameters
Headers: User-Agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36

+ URL

https://google-gruyere.appspot.com/370269580081282455385241778765335192285/login

Method
GET

Parameters
Headers: User-Agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36

+ URL

https://google-gruyere.appspot.com/370269580081282455385241778765335192285/newaccount.gtl

Method
GET

Parameters
Headers: User-Agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36

💬

🔍

Server software and technology found

Unconfirmed

?

⬆️

👤

443 / TCP

Evidence

Software / Version	Category
Google Cloud	IaaS
Google Cloud Trace	Performance
HTTP/3	Miscellaneous

+ Details

🔍

File Upload

Confirmed

⬆️

👤

443 / TCP

Evidence

- URL

https://google-gruyere.appspot.com/370269580081282455385241778765335192285/upload.gtl

Method
GET

Parameters
Headers: User-Agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36

Evidence

URL

https://google-gruyere.appspot.com/370269580081282455385241778765335192285/upload.gtl

Method

GET

Parameters

Headers: User-Agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36

The following form allows file upload:

```
<form action="/370269580081282455385241778765335192285/upload0" enctype="multipart/form-data" method="post">
<table>
<tr>
<td>
<input name="upload_file" size="50" type="file"/>
</td>
</tr>
<tr>
<td align="center">
<input type="submit" value="Upload"/>
</td>
</tr>
</table>
</form>
```

Request / Response

1

File Upload Passive Scan Request&Response

2

3

4 < GET /370269580081282455385241778765335192285/upload.gtl HTTP/1.1

5 < Host: google-gruyere.appspot.com

6 < User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36

7 < Cookie: GRUYERE=

8 <

9 > HTTP/1.1 200 OK

10 > cache-control: no-cache

Expand

HTTP OPTIONS enabled

443 / TCP

Evidence

URL	Method	Summary
https://google-gruyere.appspot.com/370269580081282455385241778765335192285/	OPTIONS	We did a HTTP OPTIONS request. The server responded with a 405 status code and the header: Allow: GET, POST Request / Response

+ Details

Login Interface Found

443 / TCP

Evidence

URL

https://google-gruyere.appspot.com/370269580081282455385241778765335192285/newaccount.gtl

<input maxLength="50" name="uid" type="text" value=""/>

<input name="pw" type="password"/>

<input type="submit" value="Create account"/>

Request / Response

Screenshot

```
<input maxLength="50" name="uid" type="text" value="" />
<input name="pw" type="password" />
<input type="submit" value="Create account" />
```

Request / Response Screenshot



+ Details

Security.txt file is missing

Confirmed

443 / TCP

Evidence

URL

Missing: <https://google-gruyere.appspot.com/well-known/security.txt>

I +26 other informational findings

- Website is accessible.
- Nothing was found for vulnerabilities of server-side software.
- Nothing was found for client access policies.
- Nothing was found for use of untrusted certificates.
- Nothing was found for enabled HTTP debug methods.
- Nothing was found for secure communication.
- Nothing was found for directory listing.
- Nothing was found for passwords submitted unencrypted.
- Nothing was found for error messages.
- Nothing was found for debug messages.
- Nothing was found for code comments.
- Nothing was found for domain too loose set for cookies.
- Nothing was found for mixed content between HTTP and HTTPS.
- Nothing was found for cross domain file inclusion.
- Nothing was found for internal error code.
- Nothing was found for secure password submission.
- Nothing was found for sensitive data.
- Nothing was found for unsafe HTTP header Content Security Policy.
- Nothing was found for OpenAPI files.
- Nothing was found for SQL statement in request parameter.
- Nothing was found for password returned in later response.
- Nothing was found for Path Disclosure.
- Nothing was found for Session Token in URL.
- Nothing was found for API endpoints.

[Summary](#)[Findings](#)[Performed Tests](#)[Scan Stats](#)[Scan Parameters](#)

- Nothing was found for API endpoints.
- Nothing was found for emails.
- Nothing was found for missing HTTP header - Rate Limit.

Performed Tests (39/39)

- ✓ Scanned for HttpOnly flag of cookie
- ✓ Scanned for Secure flag of cookie
- ✓ Scanned for missing HTTP header - X-Content-Type-Options
- ✓ Scanned for missing HTTP header - Referrer
- ✓ Scanned for missing HTTP header - Strict-Transport-Security
- ✓ Scanned for missing HTTP header - Content Security Policy
- ✓ Scanned for robots.txt file
- ✓ Scanned for passwords submitted in URLs
- ✓ Scanned for website technologies
- ✓ Scanned for file upload
- ✓ Scanned for enabled HTTP OPTIONS method
- ✓ Scanned for login interfaces
- ✓ Scanned for absence of the security.txt file
- ✓ Test initial connection
- ✓ Scanned for version-based vulnerabilities of server-side software
- ✓ Scanned for client access policies
- ✓ Scanned for use of untrusted certificates
- ✓ Scanned for enabled HTTP debug methods
- ✓ Scanned for secure communication
- ✓ Scanned for directory listing
- ✓ Scanned for passwords submitted unencrypted
- ✓ Scanned for error messages
- ✓ Scanned for debug messages
- ✓ Scanned for code comments
- ✓ Scanned for domain too loose set for cookies
- ✓ Scanned for mixed content between HTTP and HTTPS

- ✓ Scanned for error messages
- ✓ Scanned for debug messages
- ✓ Scanned for code comments
- ✓ Scanned for domain too loose set for cookies
- ✓ Scanned for mixed content between HTTP and HTTPS
- ✓ Scanned for cross domain file inclusion
- ✓ Scanned for internal error code
- ✓ Scanned for secure password submission
- ✓ Scanned for sensitive data
- ✓ Scanned for unsafe HTTP header Content Security Policy
- ✓ Scanned for OpenAPI files
- ✓ Scanned for SQL statement in request parameter
- ✓ Scanned for password returned in later response
- ✓ Scanned for Path Disclosure
- ✓ Scanned for Session Token in URL
- ✓ Scanned for API endpoints
- ✓ Scanned for emails
- ✓ Scanned for missing HTTP header - Rate Limit

Scan Stats

Injection points	URLs spidered	Requests	Average requests time
11	10	20	331ms

Scan Parameters

Target	Scan type	Authentication	Source IP
https://google-gruyere.appspot.com/370269580081282455385241778765335192285/	Light	False	172.232.203.201