Sri Lanka Institute of Information Technology

BUG BOUNTY REPORT - 3

Web Security – IE2062

IT22362780

Jayaweera N.S

Report Details

Report #	- 03
Domain	- https://grammarly.com
Platform	-hackerone.com
Scans performed -	Recon-ng scan Nmap scan Wafw00f scan Dotdotpwn scan Manual scanning using Wapplyzer Text injection File upload vulnerability testing Command injection XSS injection Zap scan Nslookup

Nmap scan

Through nmap scan the open ports can be found.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sS -T4 grammarly.com
You requested a scan type which requires root privileges.
QUITTING!

┌──(kali㉿kali)-[~]
└─$ sudo nmap -sS -T4 grammarly.com
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-24 01:16 EDT
Nmap scan report for grammarly.com (44.212.184.83)
Host is up (0.037s latency).
Other addresses for grammarly.com (not scanned): 34.226.161.161 52.204.90.225
rDNS record for 44.212.184.83: ec2-44-212-184-83.compute-1.amazonaws.com
Not shown: 995 filtered tcp ports (no-response)
PORT     STATE  SERVICE
80/tcp   open   http
113/tcp  closed ident
443/tcp  open   https
8008/tcp open   http
8010/tcp open   xmpp

Nmap done: 1 IP address (1 host up) scanned in 18.03 seconds
```

No unusual ports found.

Nslookup

```
┌──(kali㉿kali)-[~]
└─$ nslookup grammarly.com
Server:         172.16.10.100
Address:        172.16.10.100#53

Non-authoritative answer:
Name:   grammarly.com
Address: 52.204.90.225
Name:   grammarly.com
Address: 34.226.161.161
Name:   grammarly.com
Address: 44.212.184.83
```

The ip address of Grammarly.com can be found

Wafw00f scan

Used to identify the type of WAF that is used to protect the web application.

```
┌──(kali㉿kali)-[~]
└─$ wafw00f https://grammarly.com

                    ____
                   /    \
                  (  W00f! )
                   \  ___/
                  ,,    __
            |`-.___   / /
            /"  _/  /_/
           *===*    /
          /     )__//
    /|  /     /___`
    \\`-._   \ |
     `\   /_\\_-
       `-____`-`

                                    404 Hack Not Found
                                            ___   ___
                                           \ \ \ / /
                                            \ \ _ / /   405 Not Allowed
                                             \ \ / /
                                        403 Forbidden
                                             / _ \
                                            / / \ \
                                           /_/   \_\
                  502 Bad Gateway                      500 Internal Error

                    ~ WAFW00F : v2.2.0 ~
      The Web Application Firewall Fingerprinting Toolkit
[*] Checking https://grammarly.com
[+] The site https://grammarly.com is behind AWS Elastic Load Balancer (Amazon) WAF.
[~] Number of requests: 2
```

According to the test results, “AWS Elastic load balancer(Amazon)” has been used as the firewall of the web application

Recon scan

Recon-ng will be used to find all the sub domains in the target.

```
[1] Recon modules

[recon-ng][default] > modules load hackertarget
[recon-ng][default][hackertarget] > options set SOURCE grammarly.com
SOURCE => grammarly.com
[recon-ng][default][hackertarget] > run

--------------
GRAMMARLY.COM
--------------
[*] Country: None
[*] Host: grammarly.com
[*] Ip_Address: 44.212.184.83
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------------
[*] Country: None
[*] Host: 3p-access.grammarly.com
[*] Ip_Address: 54.85.218.19
```

```
[*] Host: answers.grammarly.com
[*] Ip_Address: 18.215.32.117
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ----------------------------------------------------------
[*] Country: None
[*] Host: app.grammarly.com
[*] Ip_Address: 3.224.105.132
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ----------------------------------------------------------
[*] Country: None
[*] Host: applet-bundles.grammarly.com
[*] Ip_Address: 108.139.119.13
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ----------------------------------------------------------
[*] Country: None
[*] Host: apps.grammarly.com
[*] Ip_Address: 54.174.247.24
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ----------------------------------------------------------
[*] Country: None
```

```
-------
SUMMARY
-------
[*] 115 total (115 new) hosts found.
[recon-ng][default][hackertarget] >
```

115 total sub domains found.

Dotdotpwn

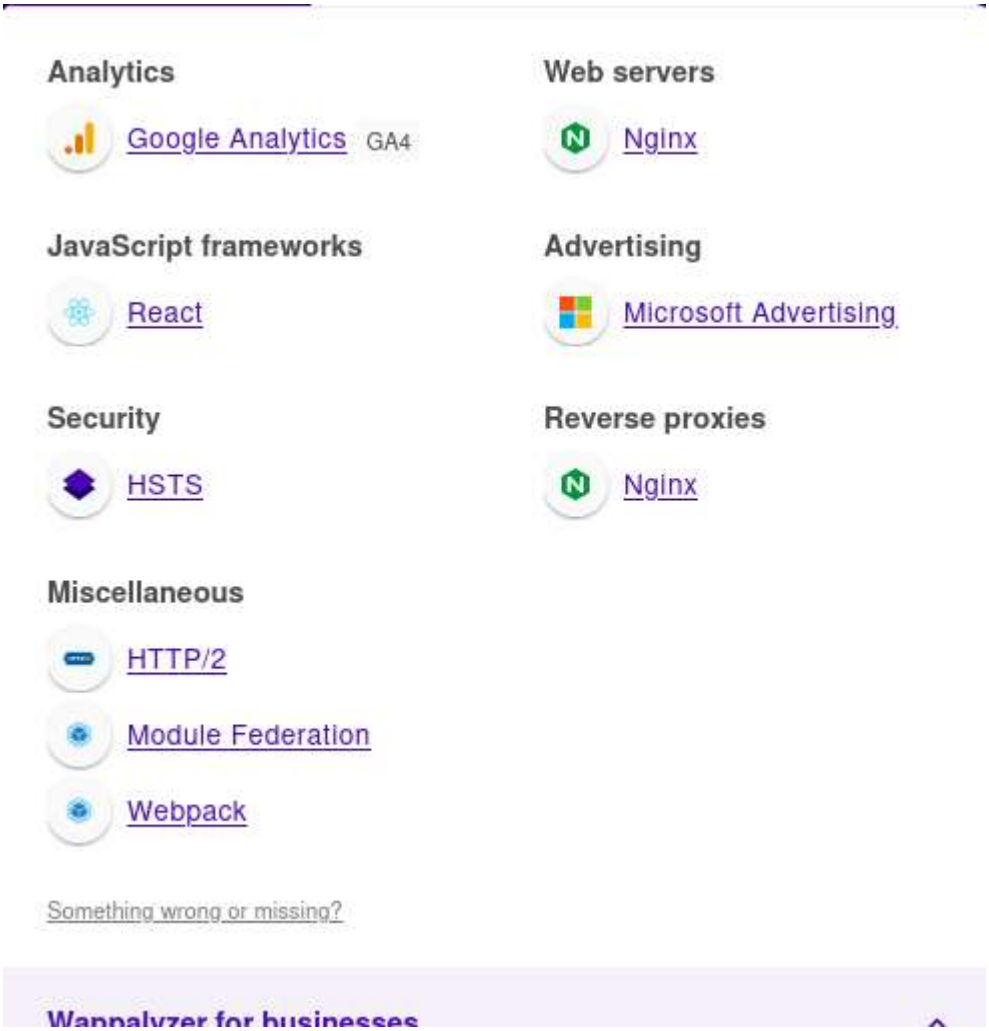
```
[+] Report name: Reports/grammarly.com_04-24-2024_02-15.txt

[========== TARGET INFORMATION ==========]
[+] Hostname: grammarly.com
[+] Protocol: http
[+] Port: 80

[========== TRAVERSAL ENGINE ==========]
[+] Creating Traversal patterns (mix of dots and slashes)
[+] Multiplying 6 times the traversal patterns (-d switch)
[+] Creating the Special Traversal patterns
[+] Translating (back)slashes in the filenames
[+] Adapting the filenames according to the OS type detected (unix)
[+] Including Special sufixes
[+] Traversal Engine DONE ! - Total traversal tests created: 11028

[========== TESTING RESULTS ==========]
[+] Ready to launch 3.33 traversals per second
[+] Press Enter to start the testing (You can stop it pressing Ctrl + C)

[*] HTTP Status: 400 | Testing Path: http://grammarly.com:80/../etc/passwd
[*] HTTP Status: 400 | Testing Path: http://grammarly.com:80/../etc/issue
[*] HTTP Status: 400 | Testing Path: http://grammarly.com:80/../../etc/passwd
[*] HTTP Status: 400 | Testing Path: http://grammarly.com:80/../../etc/issue
[*] HTTP Status: 400 | Testing Path: http://grammarly.com:80/../../../etc/passwd
[*] HTTP Status: 400 | Testing Path: http://grammarly.com:80/../../../etc/issue
[*] HTTP Status: 400 | Testing Path: http://grammarly.com:80/../../../../etc/passwd
[*] HTTP Status: 400 | Testing Path: http://grammarly.com:80/../../../../etc/issue
[*] HTTP Status: 400 | Testing Path: http://grammarly.com:80/../../../../../etc/passwd
[*] HTTP Status: 400 | Testing Path: http://grammarly.com:80/../../../../../etc/issue
[*] HTTP Status: 400 | Testing Path: http://grammarly.com:80/../../../../../../etc/passwd
[*] HTTP Status: 400 | Testing Path: http://grammarly.com:80/../../../../../../etc/issue
```

The scan results returned status codes within the range 400 (400-404). It shows a client error.

Therefore, we can conclude that the tested destinations are not vulnerable to a directory traversal.

Manual testing using wapplyzer

The Wapplyzer is used to identify the technologies used in the web application.



These are the technologies used.

There are no vulnerable versions.

Zap scan

With the use of the “active scan”, some potential vulnerabilities can be found.

Alerts

Risk=High, Confidence=Low (1)

http://grammarly.com (1)

Cloud Metadata Potentially Exposed (1)

▸ GET http://grammarly.com/latest/meta-data/

Risk=Medium, Confidence=Low (1)

http://grammarly.com (1)

Hidden File Found (1)

▸ GET http://grammarly.com/.hg

Risk=Medium, Confidence=High (3)

http://grammarly.com (3)

CSP: Wildcard Directive (1)

▸ GET http://grammarly.com

CSP: script-src unsafe-inline (1)

▸ GET http://grammarly.com

CSP: style-src unsafe-inline (1)

▸ GET http://grammarly.com

Risk=Low, Confidence=High (1)

http://grammarly.com (1)

Server Leaks Version Information via "Server" HTTP Response Header Field (1)

- GET http://grammarly.com/sitemap.xml

Risk=Low, Confidence=Medium (4)

http://grammarly.com (4)

Cookie No HttpOnly Flag (1)

- GET http://grammarly.com

Cookie with SameSite Attribute None (1)

- GET http://grammarly.com

Cookie without SameSite Attribute (1)

- GET http://grammarly.com

Cross-Domain JavaScript Source File Inclusion (1)

- GET http://grammarly.com

Risk=Informational, Confidence=Medium (3)

http://grammarly.com (3)

Modern Web Application (1)

- GET http://grammarly.com

Session Management Response Identified (1)

- GET http://grammarly.com

User Agent Fuzzer (1)

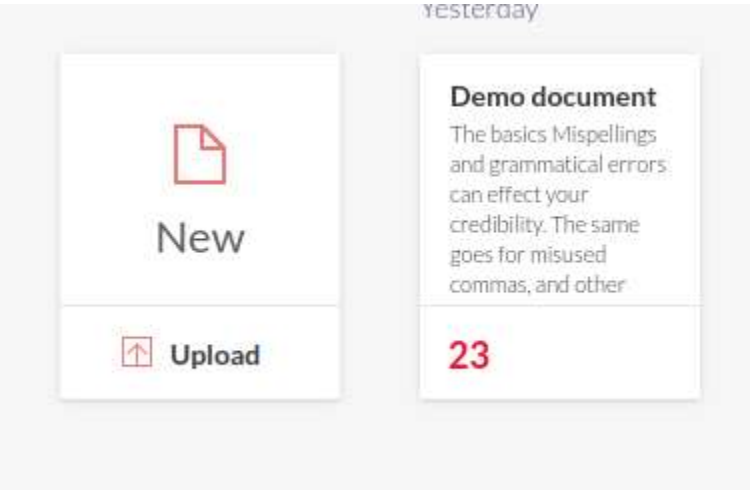
- GET http://grammarly.com/robots.txt

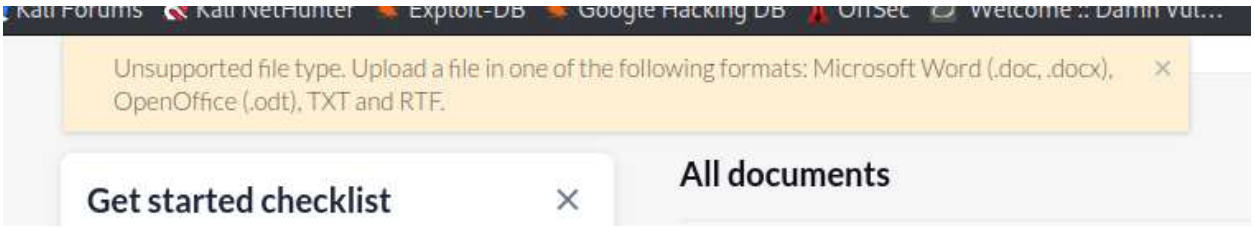
According to the above results the following can be found:

- Cloud metadata potentially exposed
- Hidden file found - .hg
- CSP allow wildcard sources in following directives : style-src, img-src, connect-src, frame-src, frame-ancestors, font-src, media-src, object-src, manifest-src, form-action
- The web application is leaking information via the server" HTTP response header.
- There is no HttpOly flag - high possibility of xss attacks and session highjacking.
- There is an absence of "Samesite" attribute -might be vulnerable to XSS injection and CSRF attacks.

File upload vulnerability

If a .php file can be uploaded from the file uploading facility, there is a possibility to upload and execute a reverse shell php code.

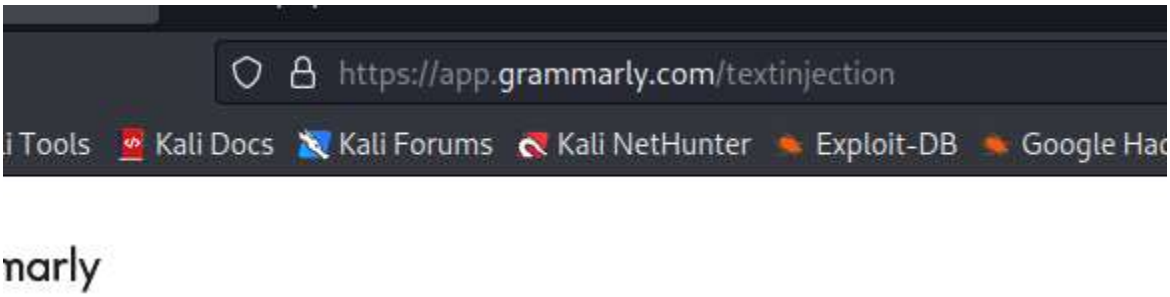




And there is no vulnerability.

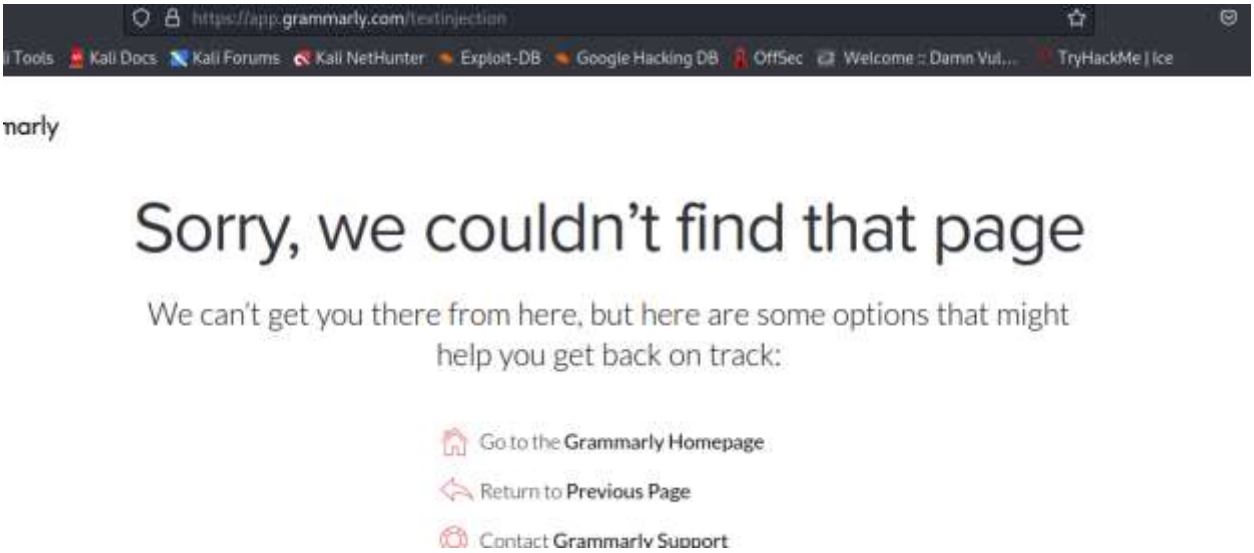
Text injection

An arbitrary string value is appended to the URL to see whether the web application is vulnerable towards a text injection.



If the entered text is reflected on the error response of the web page, there is a possibility to inject malicious content.

If not, the web application is safe.

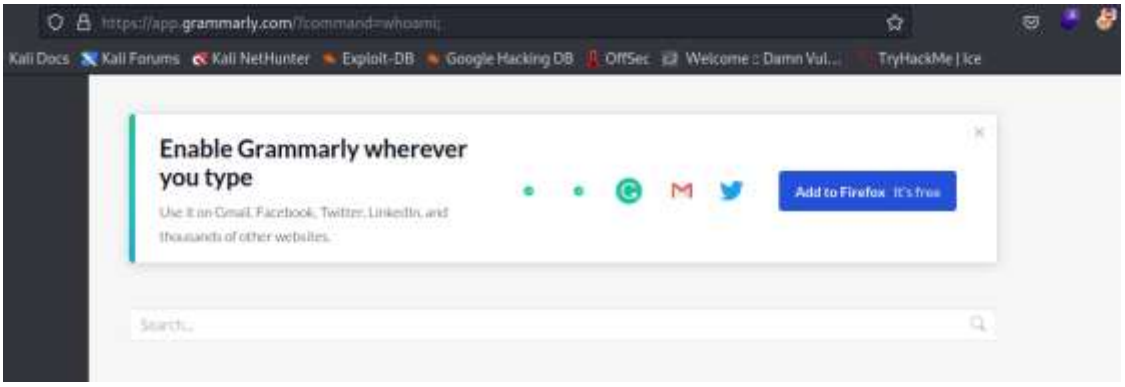


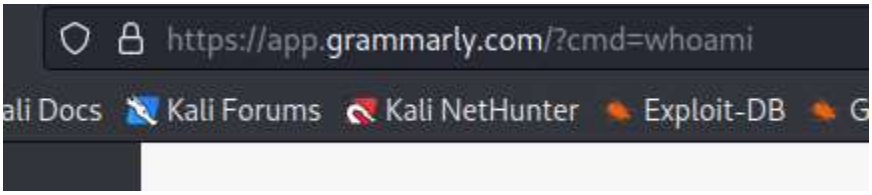
No vulnerability found.

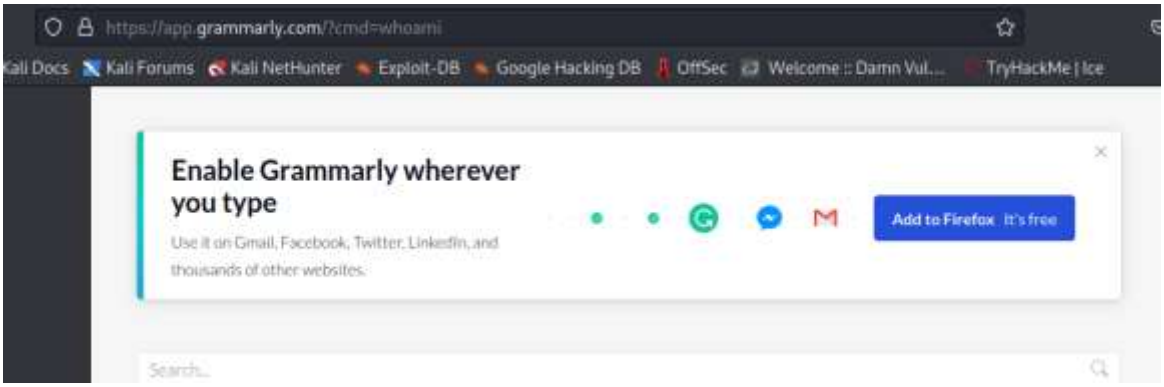
command injection

The query that is used for searching is used against this vulnerability.

The “whoami” command is appended to the url.



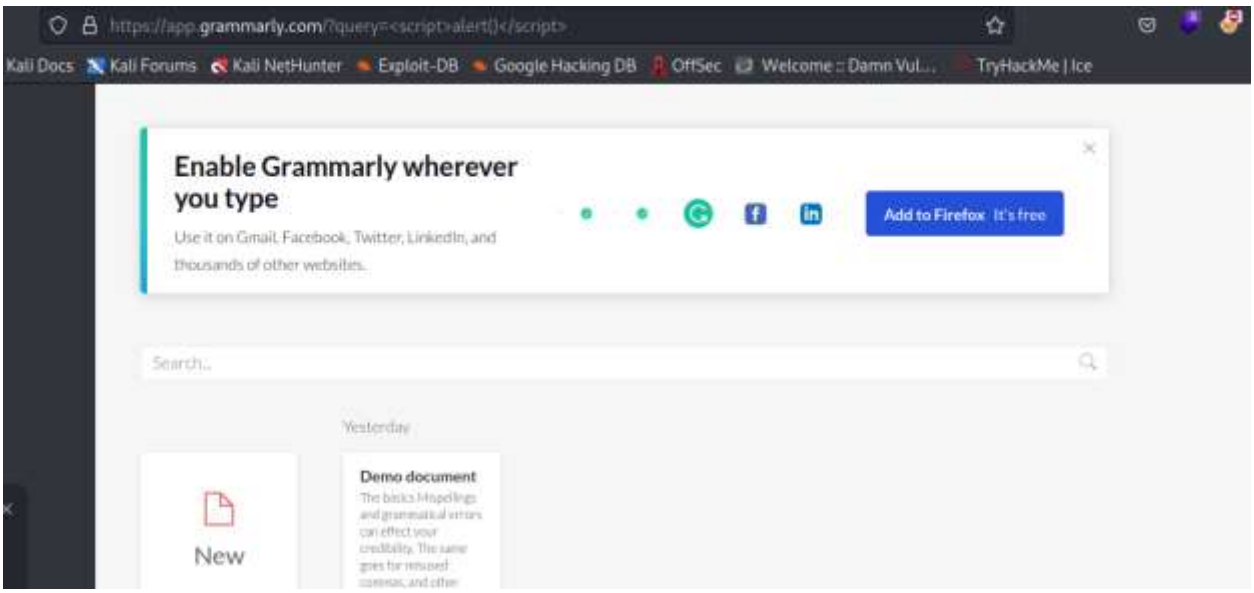




No command injection vulnerability can be found.

XSS injection

A payload is apended to the url to test against xss injection.



A xss injection cannot be done.