

**Sri Lanka Institute of Information Technology**



## **BUG BOUNTY REPORT - 2**

Web Security – IE2062

IT22362780

Jayaweera N.S

## Report Details

Report # - 02

Domain - <https://canva.com>

Platform -bugcrowd.com

Scans performed -  
Nmap scan  
Recon-ng  
Wafw00f scan  
Dotdotpwn scan  
Nikto scan  
Sqlmap scan  
Manual scanning using Wapplyzer  
Text injection testing  
File upload vulnerability testing  
Command injection testing  
XSS injection testing

## Nmap scan

Using nmap scan all the open ports in the target can be identified.

```
File Actions Edit View Help
(kali@kali)-[~]
$ nmap -sS -T4 canva.com
You requested a scan type which requires root privileges.
QUITTING!

(kali@kali)-[~]
$ sudo nmap -sS -T4 canva.com
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-22 02:27 EDT
Nmap scan report for canva.com (104.16.102.112)
Host is up (0.045s latency).
Other addresses for canva.com (not scanned): 104.16.103.112 2606:4700::6810:6670 2606:4700::6810:6770
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
443/tcp    open  https
8080/tcp   open  http-proxy
8443/tcp   open  https-alt

Nmap done: 1 IP address (1 host up) scanned in 26.01 seconds
```

According to the scan results, no unusual ports can be identified as open.

## Recon scan

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

```
[recon-ng][bb1] > modules load hackertarget
[recon-ng][bb1][hackertarget] > show options
Shows various framework items

Usage: show <companies|contacts|credentials|domains|hosts|leaks|locations|netblocks|ports|profiles|pushpins|repositories|vulnerabilities>

[recon-ng][bb1][hackertarget] > options set SOURCE canva.com
SOURCE => canva.com
[recon-ng][bb1][hackertarget] > run
```

---

CANVA.COM      Out of Range

```
[*] Country: None
[*] Host: canva.com
[*] Ip_Address: 104.16.102.112
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
```

---

```
[*] Country: None
[*] Host: ip-sc.canva.com
[*] Ip_Address: 34.83.150.148
```

```
[*] Country: None
[*] Host: ru-ru.learn.canva.com
[*] Ip_Address: 104.16.224.149
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
```

---

```
[*] Country: None
[*] Host: zh-cn.learn.canva.com
[*] Ip_Address: 104.17.239.159
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
```

---

```
[*] Country: None
[*] Host: mailer1.canva.com
[*] Ip_Address: 198.2.128.77
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
```

---

```
[*] Country: None
[*] Host: mailer2.canva.com
[*] Ip_Address: 198.2.128.98
[*] Latitude: None
[*] Longitude: None
```

### SUMMARY


```
[*] 115 total (115 new) hosts found.
[recon-ng][bb1][hackertarget] >
```

115 sub domains were found.

## WafW00f scan

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

```
(kali㉿kali)-[~]
$ wafw00f https://canva.com
```



```
~ WAFW00F : v2.2.0 ~
The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://canva.com
[+] The site https://canva.com is behind Cloudflare (Cloudflare Inc.) WAF.
[~] Number of requests: 2
```

According to the test results, “Cloudflare (Cloudflare Inc.)” is used as the firewall of the web application.

## Dotdotpwn scan

Dotdotpwn is a directory traversal checker.

```
(kali@kali)-[~]
$ dotdotpwn -m http -h canva.com
#####
#
# CubilFelino Access Chatsubo
# Security Research Lab and [(in)Security Dark] Labs
# chr1x.sectester.net are publicly accessible chatsubo-labs.blogspot.com
#
# Focus Areas pr0udly present:
#
# • The w... com and ...
# • The ...
# • The ...
#
# - DotDotPwn v3.0.2 -
# The Directory Traversal Fuzzer
# http://dotdotpwn.sectester.net
# dotdotpwn@sectester.net
#
# Out of scope
#
# by chr1x & nitro5
#####
[+] Report name: Reports/canva.com_04-22-2024_05-26.txt
[===== TARGET INFORMATION =====]
[+] Hostname: canva.com
[+] Protocol: http
[+] Port: 80
[===== TRAVERSAL ENGINE =====]
```

```

[===== 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 suffixes
[+] 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://canva.com:80/../../etc/passwd
[*] HTTP Status: 400 | Testing Path: http://canva.com:80/../../etc/issue
[*] HTTP Status: 400 | Testing Path: http://canva.com:80/../../../../etc/passwd
[*] HTTP Status: 400 | Testing Path: http://canva.com:80/../../../../etc/issue
[*] HTTP Status: 400 | Testing Path: http://canva.com:80/../../../../etc/passwd
[*] HTTP Status: 400 | Testing Path: http://canva.com:80/../../../../etc/issue
[*] HTTP Status: 400 | Testing Path: http://canva.com:80/../../../../etc/passwd
[*] HTTP Status: 400 | Testing Path: http://canva.com:80/../../../../etc/issue
[*] HTTP Status: 400 | Testing Path: http://canva.com:80/../../../../etc/passwd
[*] HTTP Status: 400 | Testing Path: http://canva.com:80/../../../../etc/issue
[*] HTTP Status: 400 | Testing Path: http://canva.com:80/../../../../etc/passwd
[*] HTTP Status: 400 | Testing Path: http://canva.com:80/../../../../etc/issue
[*] HTTP Status: 404 | Testing Path: http://canva.com:80/..%5Cetc%5Cpasswd
[*] HTTP Status: 404 | Testing Path: http://canva.com:80/..%5Cetc%5Cissue
[*] HTTP Status: 404 | Testing Path: http://canva.com:80/..%5C..%5Cetc%5Cpasswd
[*] HTTP Status: 404 | Testing Path: http://canva.com:80/..%5C..%5Cetc%5Cissue
[*] HTTP Status: 404 | Testing Path: http://canva.com:80/..%5C..%5C..%5Cetc%5Cpasswd
[*] HTTP Status: 404 | Testing Path: http://canva.com:80/..%5C..%5C..%5Cetc%5Cissue

```

The scan results returned status codes within the range 400 (400-499). It shows a client error.

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



## Nikto scan

```
(kali@kali) ~$ nikto -h https://canva.com
- Nikto v2.1.6

+ Target IP: 104.16.102.112
+ Target Hostname: canva.com
+ Target Port: 443

+ SSL Info: Subject: /CN=canva.com
           Ciphers: TLS_AES_256_GCM_SHA384
           Issuer: /C=US/O=Google Trust Services LLC/CN=GTS CA 1P5
+ Message: Multiple IP addresses found: 104.16.102.112, 104.16.103.112
+ Start Time: 2024-04-22 04:59:12 (GMT-4)

+ Server: cloudflare
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ Uncommon header 'nel' found, with contents: [{"success_fraction":0.01,"report_to":"cf-nel","max_age":604800}]
+ Uncommon header 'report-to' found, with contents: [{"endpoints":[{"url":"https://l/a.nel.cloudflare.com/report/v4?s=0k47QeK8n2MFayZfssQOPJl368MnEUxV0L1jotZrtIs88BuAqpx4X2F3RotS2B61zBkAo43s6BJIANJK4UxTCUyLFiAGfsU2WAVC2Rg42BwctfKfK52F90I2YpsDYlK2FwK3DN30"}],"group":"cf-nel","max_age":604800}]
+ The site uses SSL and Expect-CT header is not present.
+ Root page / redirects to: https://www.canva.com/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
-C
-C all
- STATUS: Completed 5630 requests (~82% complete, 1.2 hours left): currently in plugin 'Nikto Tests'
- STATUS: Running average: 100 requests: 135.48550 sec, 10 requests: 148.7700 sec.
- 7786 requests: 0 error(s) and 5 item(s) reported on remote host
- End Time: 2024-04-22 10:40:37 (GMT-4) (20485 seconds)

+ 1 host(s) tested
```

The above scan results the following:

- The anti-clickjacking "X-Frame-Options" header, which helps prevent clickjacking attacks, is not present.
- The "X-XSS-Protection" header is not defined, which can protect against some forms of XSS.
- The site uses SSL and Expect-CT header is not present.

The "expect-CT" header is a security feature that helps websites, and their users avoid the risks associated with incorrectly issued SSL certificates.

It supports transparency and accountability when issuing SSL certificates, which improves overall web security.



There are some issues/disadvantages occurred when the “expect-CT” header is absent:

- The protection against the mis issuing of SSL certificates will be low.
- Mismanagement of SSL certificates.
- No trust and security

But the absence of “expected-CT” header is not a huge vulnerability or a security issue in a website.

## Sqlmap scan

With the use of this scan, we can identify whether a sql injection can be done or not.

```
(kali@kali) ~$ sqlmap -u https://www.canva.com/search/designs?q=aksl

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 15:10:40 /2024-04-22/

[15:10:41] [INFO] testing connection to the target URL
[15:10:42] [WARNING] potential CAPTCHA protection mechanism detected
[15:10:42] [WARNING] potential permission problems detected ('Access Denied')
[15:10:43] [WARNING] the web server responded with an HTTP error code (403) which could interfere with the results of the tests
you have not declared cookie(s), while server wants to set its own (['_cf_bv=bgrrrskzsq...am5thqgW']). Do you want to use these [Y/n] y
[15:17:15] [INFO] checking if the target is protected by some kind of waf/IPS
[15:17:25] [WARNING] reflective value(s) found and filtering out
[15:17:25] [INFO] testing if the target URL content is stable
[15:17:25] [WARNING] target URL content is not stable (i.e. content differs). sqlmap will base the page comparison on a sequence matcher. If no dynamic nor injectable parameters are detected, or in case of junk results, refer to user's manual paragraph 'Page comparison'
how do you want to proceed? [(C)ontinue/((s)tring/(r)egex/(q)uit) :
[15:17:40] [INFO] testing if GET parameter 'q' is dynamic
[15:17:42] [WARNING] GET parameter 'q' does not appear to be dynamic
[15:17:42] [WARNING] heuristic (basic) test shows that GET parameter 'q' might not be injectable
[15:17:45] [INFO] testing for SQL injection on GET parameter 'q'
[15:17:45] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[15:17:52] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
```

```
[15:18:47] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[15:18:52] [WARNING] GET parameter 'q' does not seem to be injectable
[15:18:55] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level/--risk' options if you wish to perform more tests. Please retry with the switch '--text-only' (along with --technique=00) as this case looks like a perfect candidate (low textual content along with inability of comparison engine to detect at least one dynamic parameter). If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
[15:18:55] [WARNING] HTTP error codes detected during run:
403 (Forbidden) - 87 times
[15:18:55] [WARNING] it appears that the target has a maximum connections constraint
[15:18:55] [WARNING] your sqlmap version is outdated

[*] ending @ 15:18:55 /2024-04-22/
```

The above results prove that there is no injection vulnerability in the above web application.

# Manual testing -using Wappalyzer

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

The screenshot displays the Wappalyzer interface with a purple header. Below the header, there are tabs for 'TECHNOLOGIES' and 'MORE INFO', and an 'Export' button. The main content area is divided into two columns listing various technologies detected on the website.

Category	Technology	Version	Confidence
Analytics	Cloudflare Browser Insights		
	Google Analytics	GA4	
	Facebook Pixel	2.9.154	
	LinkedIn Insight Tag		
JavaScript frameworks	Next.js	12.3.4	
	React		
Issue trackers			
Web frameworks	Next.js	12.3.4	
Miscellaneous	HTTP/2		
	Module Federation		50% sure
	Open Graph		
	Webpack		50% sure
Web servers	Next.js	12.3.4	
Programming languages	Node.js		
CDN	Cloudflare		
Advertising	Microsoft Advertising		
Tag managers	web-vitals		
	MobX		
	core-js	3.0.0	
Authentication	Google Sign-in		
RUM	web-vitals		
	Cloudflare Browser Insights		

[Something wrong or missing?](#)

These are the technologies used.

Next.js 12.3.4 has been used.

it is an identified CVE (cve 2023-46729) which allows malicious actors to forge requests and responses from the user's next.js application. [1]

VULNERABILITY	VULNERABLE VERSION
<div data-bbox="240 516 277 548">M</div> <h3 data-bbox="326 520 565 548">Resource Exhaustion</h3> <p data-bbox="326 600 618 627"><code>next</code> is a react framework.</p> <p data-bbox="326 661 1154 814">Affected versions of this package are vulnerable to Resource Exhaustion via the <code>cache-control</code> header. An attacker can cause a denial of service to all users requesting the same URL via a CDN by caching empty prefetch responses.</p> <p data-bbox="326 863 695 890">How to fix Resource Exhaustion?</p> <p data-bbox="326 905 927 932">Upgrade <code>next</code> to version 13.4.20-canary.13 or higher.</p>	<13.4.20-canary.13

[1] figure from <https://security.snyk.io>

There is a resource exhaustion vulnerability in the web application and in order to mitigate the risk, update next.js to version 13.4.20-canary.12 or higher.

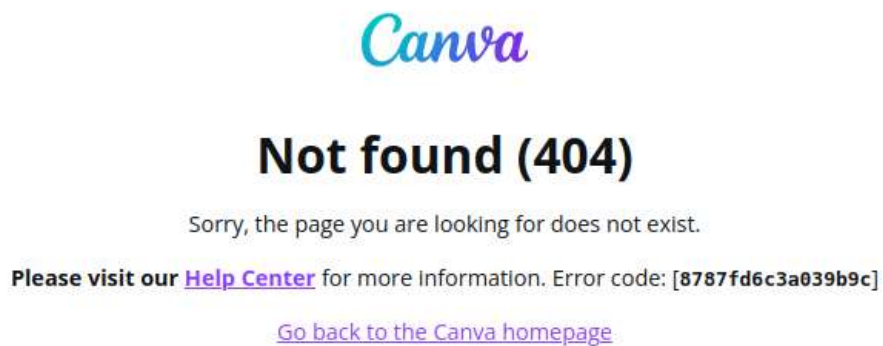
## Text injection

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.

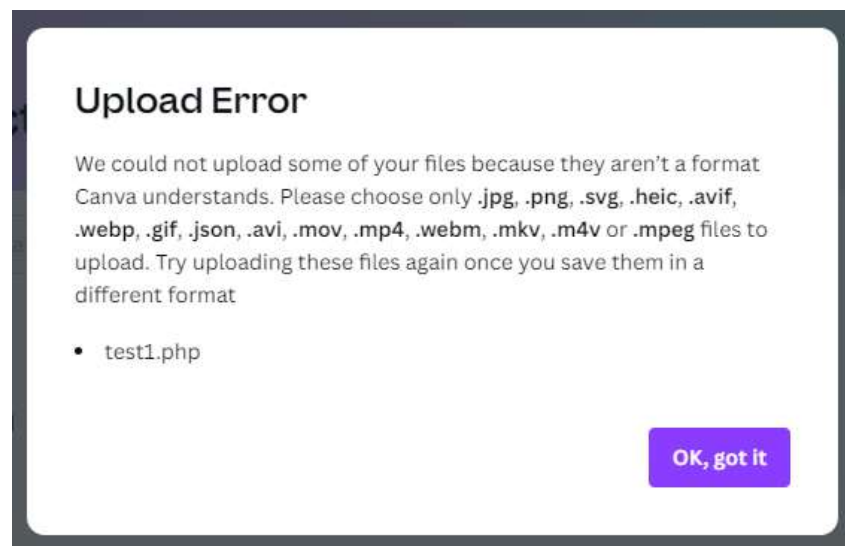
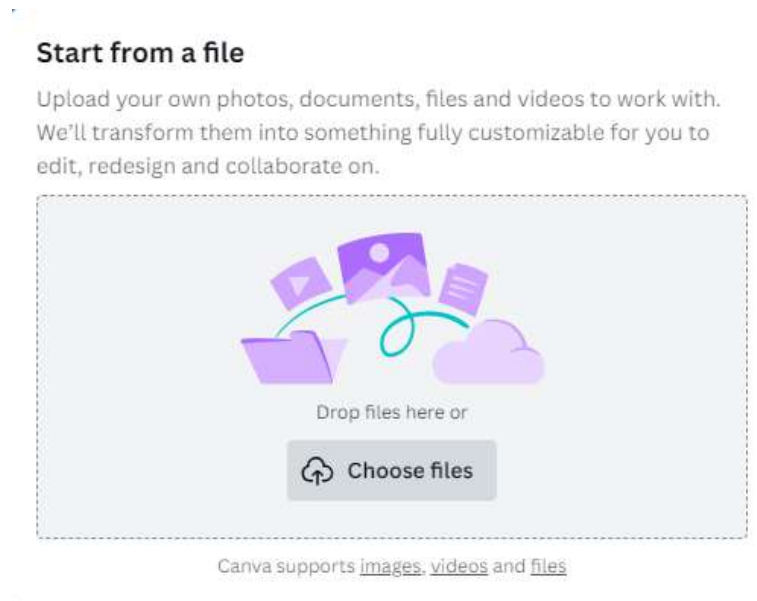
If not, the web application is safe.



No text injection vulnerability can be found.

## **File upload vulnerability testing**

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.

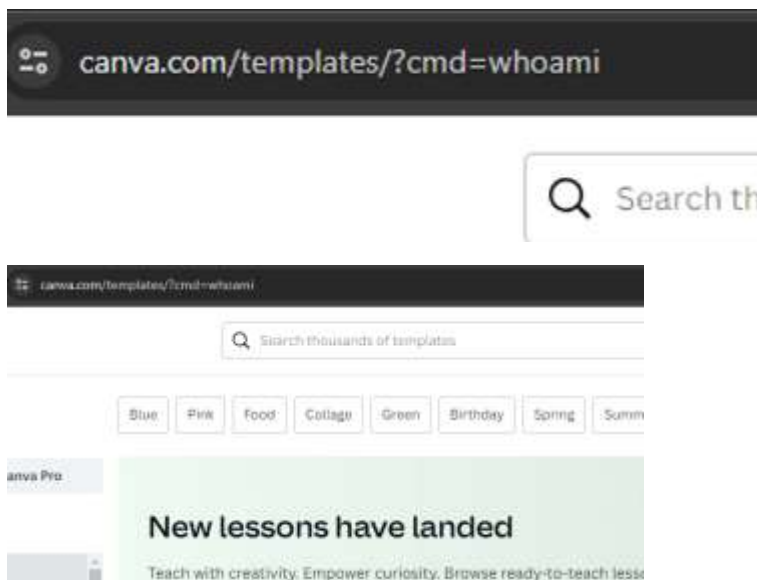
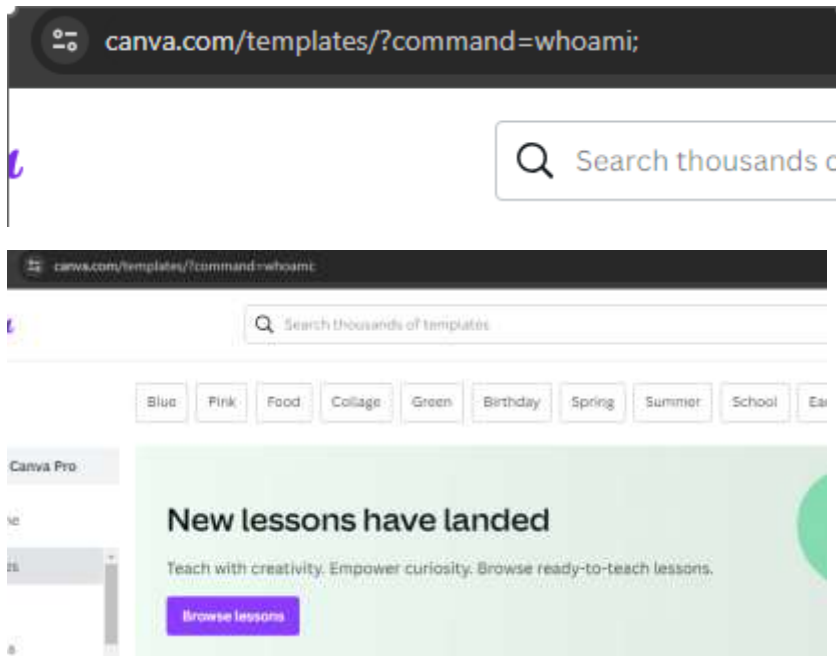


No vulnerability found.

## Command injection testing

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

The “whoami” command is appended to the url.

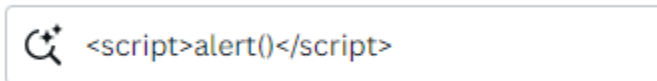
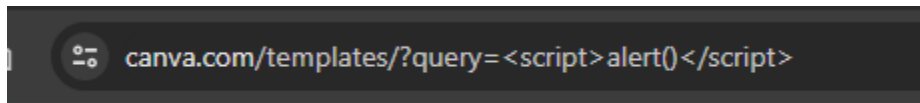


No command injection vulnerability can be found.



## XSS injection testing

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



Home > Templates > <script>alert()</script>

## <script>alert()</script> templates

Browse high quality <script>alert()</script> templates for your next design



1,905,271 templates

Input sanitization is there as the script is considered as text So no xss vulnerability is present.

## References

- [1] "security.snyk.io." Version 12.3.4. Accessed: Apr. 23, 2024. [Online]. Available: <https://security.snyk.io/package/npm/next/12.3.4>
- [2] Sentry, "Next.js SDK Security Advisory - CVE-2023-46729," Sentry Blog, Nov. 9, 2023. [Online]. Available: <https://blog.sentry.io/next-js-sdk-security-advisory-cve-2023-46729/#tldr>