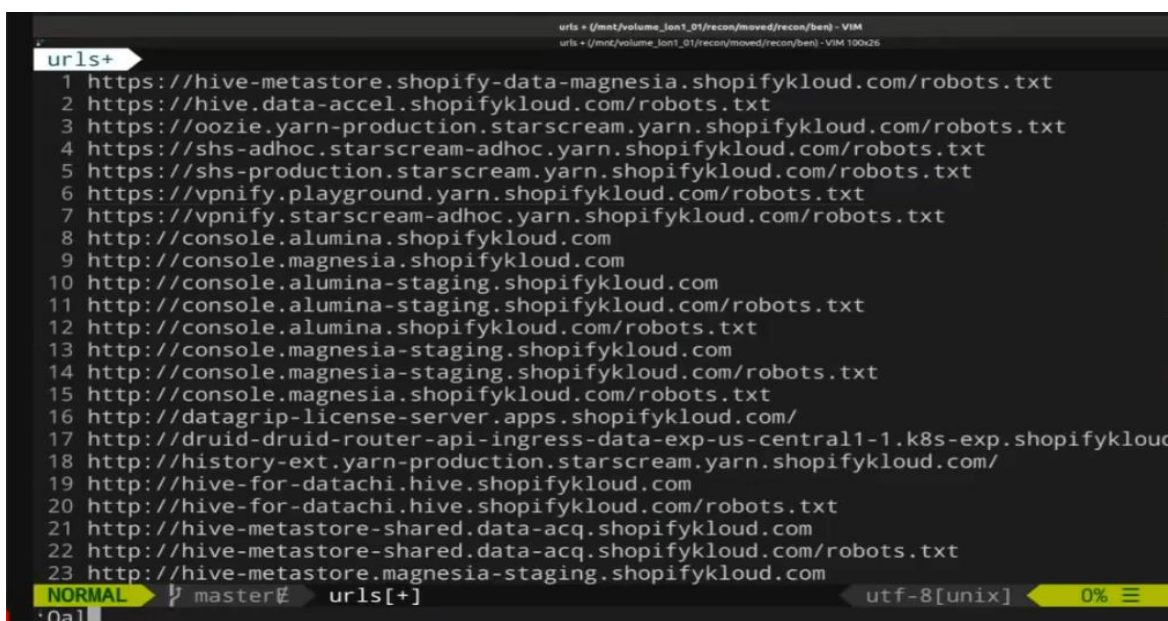


Shopify Bug Bounty Vulnerability Report

Executive Summary

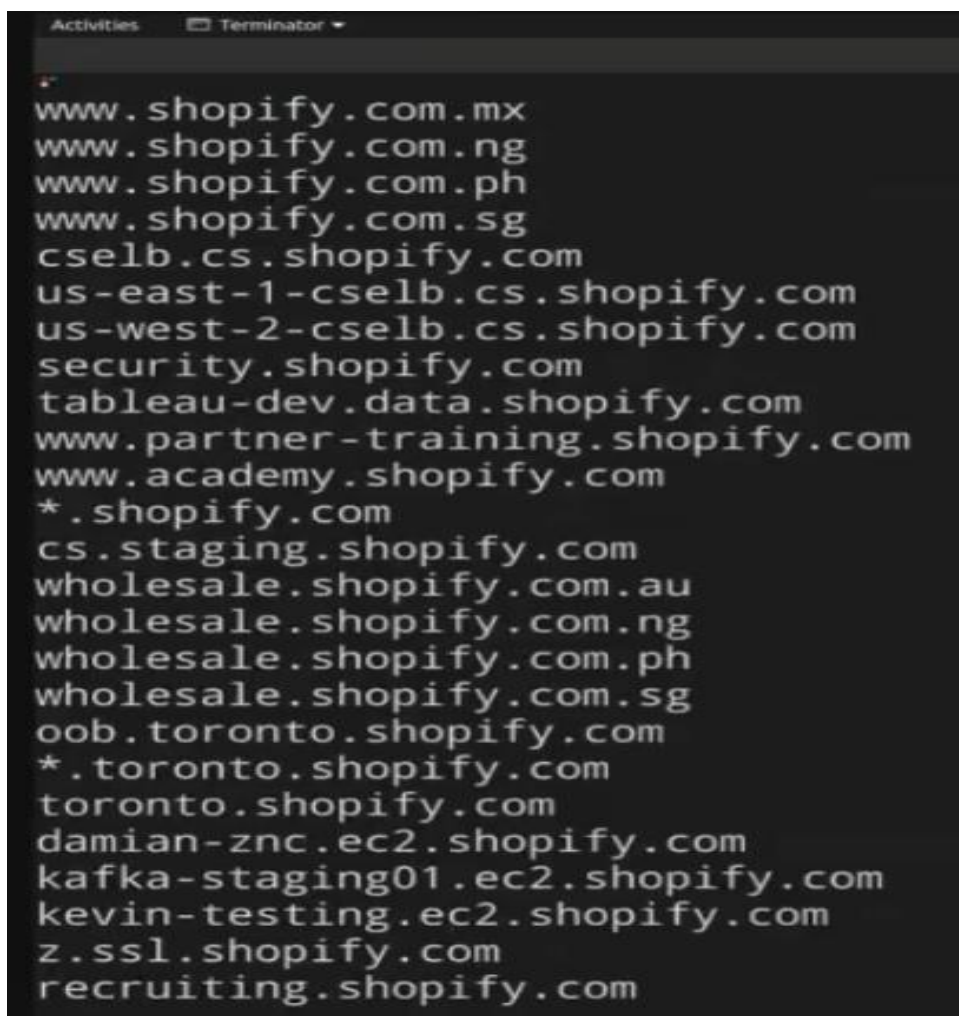
For the past three weeks, our team was tasked conducted a bug bounty exploitation as well as vulnerability assessment on shopify and all its components. Throughout the exploitation, we were tasked with discovering physical and logical security vulnerabilities within the company's digital assets that could result in compromising the confidentiality, integrity, and/or availability of the system. Furthermore, the team tested domains, website pages, APIs, authentication keys and platforms in accordance with cyber security hacking ethical guidelines. The bug bounty was carried out on Hackerone platform.



```
urls + (/mnt/volume_1on1_01/recon/moved/recon/ben) - VIM
urls + (/mnt/volume_1on1_01/recon/moved/recon/ben) - VIM 100x26

urls+
1 https://hive-metastore.shopify-data-magnesia.shopifycloud.com/robots.txt
2 https://hive.data-accel.shopifycloud.com/robots.txt
3 https://oozie.yarn-production.starscream.yarn.shopifycloud.com/robots.txt
4 https://shs-adhoc.starscream-adhoc.yarn.shopifycloud.com/robots.txt
5 https://shs-production.starscream.yarn.shopifycloud.com/robots.txt
6 https://vpnify.playground.yarn.shopifycloud.com/robots.txt
7 https://vpnify.starscream-adhoc.yarn.shopifycloud.com/robots.txt
8 http://console.alumina.shopifycloud.com
9 http://console.magnesia.shopifycloud.com
10 http://console.alumina-staging.shopifycloud.com
11 http://console.alumina-staging.shopifycloud.com/robots.txt
12 http://console.alumina.shopifycloud.com/robots.txt
13 http://console.magnesia-staging.shopifycloud.com
14 http://console.magnesia-staging.shopifycloud.com/robots.txt
15 http://console.magnesia.shopifycloud.com/robots.txt
16 http://datagrip-license-server.apps.shopifycloud.com/
17 http://druid-druid-router-api-ingress-data-exp-us-central1-1.k8s-exp.shopifycloud.com/
18 http://history-ext.yarn-production.starscream.yarn.shopifycloud.com/
19 http://hive-for-datachi.hive.shopifycloud.com
20 http://hive-for-datachi.hive.shopifycloud.com/robots.txt
21 http://hive-metastore-shared.data-acq.shopifycloud.com
22 http://hive-metastore-shared.data-acq.shopifycloud.com/robots.txt
23 http://hive-metastore.magnesia-staging.shopifycloud.com

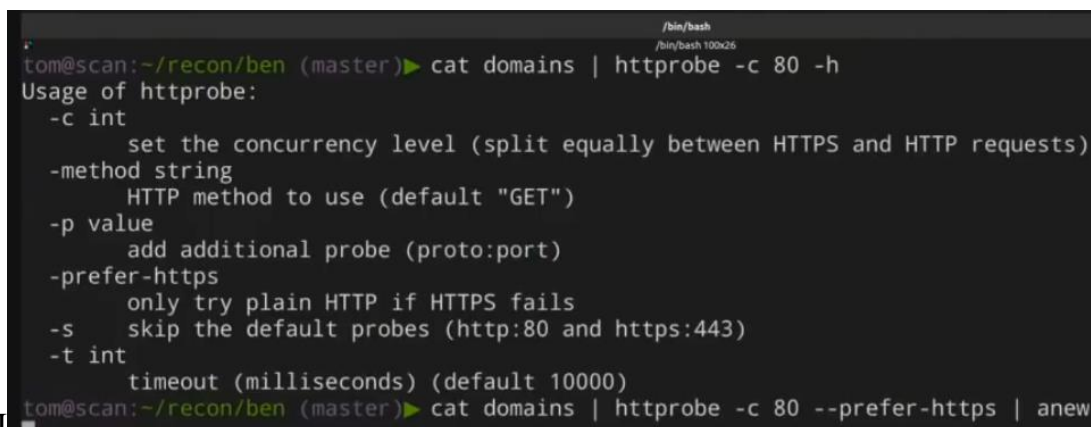
NORMAL master# urls[+] utf-8[unix] 0%
:Qal
```

A screenshot of a terminal window titled 'Activities' and 'Terminator'. The terminal displays a list of domains and subdomains related to Shopify, including various country codes, regional identifiers, and specific service domains.

```
www.shopify.com.mx
www.shopify.com.ng
www.shopify.com.ph
www.shopify.com.sg
cselb.cs.shopify.com
us-east-1-cselb.cs.shopify.com
us-west-2-cselb.cs.shopify.com
security.shopify.com
tableau-dev.data.shopify.com
www.partner-training.shopify.com
www.academy.shopify.com
*.shopify.com
cs.staging.shopify.com
wholesale.shopify.com.au
wholesale.shopify.com.ng
wholesale.shopify.com.ph
wholesale.shopify.com.sg
oob.toronto.shopify.com
*.toronto.shopify.com
toronto.shopify.com
damian-znc.ec2.shopify.com
kafka-staging01.ec2.shopify.com
kevin-testing.ec2.shopify.com
z.ssl.shopify.com
recruiting.shopify.com
```

We used assetfinder GitHub repo to analyse and scan the number of open ports for connectivity and as well as the number of domains and subdomains shopify has, we developed a keen interest on shopify.com and **shopifycloud.com** to find shopify wildcards and domains. We found 6136 domains and server on the server and over 400,000 server requests on the clients. Furthermore, we employed a few techniques such as “*anew domains*” to further filter our target shopify assets. Refer to screenshots below as well as video1 shared. *However, quite interestingly a deeper look at line 17 for the Kubernetes, throws a 404 not found error, the question is why would a nginx server with no index, it should be serving something out. This can be termed as a vulnerability*

Furthermore, we scanned port 80 using the httpprobe bash command and burp suite with preference to secure https. Refer to video3 and 5 and screenshot below.



```

/bin/bash
tom@scan:~/recon/ben (master)► cat domains | httpprobe -c 80 -h
Usage of httpprobe:
  -c int
      set the concurrency level (split equally between HTTPS and HTTP requests)
  -method string
      HTTP method to use (default "GET")
  -p value
      add additional probe (proto:port)
  -prefer-https
      only try plain HTTP if HTTPS fails
  -s
      skip the default probes (http:80 and https:443)
  -t int
      timeout (milliseconds) (default 10000)
tom@scan:~/recon/ben (master)► cat domains | httpprobe -c 80 --prefer-https | anew

```

For the GraphQL, Shopify has been porting much of the functionality from REST API, over to our newer and more flexible GraphQL API. The GraphQL API is a great area to focus on for bug bounty research, as there is new code added very frequently which is very easy to exploit focused on the queries and mutations that are used commonly in Shopify GraphQL Admin. The exploit attempts to make it easier to get started testing full API, and to help you use the schema to track down bugs in lesser-known areas. The APIs are mostly implemented on checkout and billing pages and sites. Our Json source codes are implemented on the introspection endpoint *Ex.curl*

```

/bin/bash
tom@scan:~/recon/ben (master) ► findomain -f wildcards | tee -a findomain.out

Target ==> shopify.com

Searching in the Crtsh database API...
Searching in the CertSpotter API...
Searching in the Sublist3r API...
Searching in the Virustotal API...
Searching in the Facebook API...
Searching in the AnubisDB API...
Searching in the Threatminer API...
Searching in the Archive.org API...
Searching in the Threatcrowd API...
Searching in the Urlscan.io API...
Searching in the Bufferover API...
The Virustotal API has failed returning the following HTTP status: 429 Too Many Requests
The CertSpotter API has failed returning the following HTTP status: 429 Too Many Requests
✗ An error occurred while parsing the JSON obtained from the Bufferover API. Error:
west::Error { kind: Decode, source: Error("invalid type: null, expected a sequence",
n: 15) }.
The Facebook API has failed returning the following HTTP status: 403 Forbidden

```

We queried the shopify domain hosts and a total of more than 403,000 hosts were found. Performing XSS brute force on the type of files stored in these hosts using the type, body, and header structural format. Among the request results(4000 plus) generated nothing looked abnormal worth our further exploitation. We further explored the shopify-stage: production to see if there are any further abnormalities. We discover the canary which is contrary to our shopify production environment requests.

```

/bin/bash
tom@scan:~/recon/ben/roots (master) ► grep -hri X-Shopify-Stage | anew
< X-Shopify-Stage: production
< X-Shopify-Stage: canary
tom@scan:~/recon/ben/roots (master) ►

```

```

66df570d18d26906a73f6803711efa6a60d2cd33...recon/ben/roots/1state.shopify.com] - VIM
<f6803711efa6a60d2cd33.headers X 1/66df570d18d26906a73f6803711efa6a60d2cd33.he
1 GET https://1state.shopify.com
2
3
4 < HTTP/1.1 404 Not Found
5 < Date: Sun, 24 Jan 2021 19:46:19 GMT
6 < Content-Type: text/html; charset=utf-8
7 X-Shopify-Stage: canary
8 < Set-Cookie: _y=9147ae75-7059-4fa1-8c37-d7908158a46b; Expires=Mon, 24-Jan-22 19:
9 < Set-Cookie: _s=33f0f4f5-ed81-45d6-920d-ad0b99e3362b; Expires=Sun, 24-Jan-21 20:
10 < Set-Cookie: _shopify_y=9147ae75-7059-4fa1-8c37-d7908158a46b; Expires=Mon, 24-Ja
11 < Set-Cookie: _shopify_s=33f0f4f5-ed81-45d6-920d-ad0b99e3362b; Expires=Sun, 24-Ja
12 < Set-Cookie: _shopify_fs=2021-01-24T19%3A46%3A19Z; Expires=Mon, 24-Jan-22 19:46:
13 < X-Shopid:
14 < X-Request-Id: 6c3468a5-9021-45e7-9a7f-47766714d954
15 < Cf-Cache-Status: DYNAMIC
16 < Alt-Svc: h3-27=":443"; ma=86400, h3-28=":443"; ma=86400, h3-29=":443"; ma=86400
17 < X-Sorting-Hat-Podid: -1
18 < X-Storefront-Renderer-Rendered: 1
19 < Vary: Accept-Encoding
20 < Content-Security-Policy: block-all-mixed-content; frame-ancestors 'none'; upgra
21 < X-Shardid: -1
22 < X-Dc: gcp-us-central1,gcp-us-central1,gcp-us-central1
23 < X-Download-Options: noopen
NORMAL master <711efa6a60d2cd33.headers utf-8[unix] 22% 7/31

```

A deeper look in the canary requests generates the above results. *Thus, we propose this could be another vulnerability or loophole which can be exploited in the Https request environment.*

In the next phase we looked at shopify assets such as AWS keys, API keys, s3-buckets security, servers, URLs et cetera using the gf tool.

```

/bin/bash
tom@scan:~/recon/ben/roots (master) > gf debug-pages
tom@scan:~/recon/ben/roots (master) > gf
api-keys          firebase          json-sec          php-sources       truffle
aws-keys          fw              meg-headers      s3-buckets        upload-
base64            go-functions    php-curl         sec               urls
copyright         http-auth      php-errors       servers
cors              interesting-files php-serialized   strings
debug-pages       js-sinks       php-sinks        takeovers
tom@scan:~/recon/ben/roots (master) > gf

```


We developed interest to further exploit firebase under the pretext of user and pass@example.com which returned null value indicating good security measure from the shopify team. A further ping on the servers generated something worth concern with a timing header, further exploration indicated its docs at shopify.

```

[No Name] +
1261 < Content-Type: text/html; charset=utf-8
1262 < Date: Sun, 24 Jan 2021 19:46:19 GMT
1263 < Date: Sun, 24 Jan 2021 19:46:20 GMT
1264 < Date: Sun, 24 Jan 2021 19:46:21 GMT
1265 < Date: Sun, 24 Jan 2021 19:46:22 GMT
1266 < Date: Sun, 24 Jan 2021 19:46:24 GMT
1267 < Date: Sun, 24 Jan 2021 19:46:28 GMT
1268 < Date: Sun, 24 Jan 2021 19:46:29 GMT
1269 < Date: Sun, 24 Jan 2021 19:46:30 GMT
1270 < Expect-Ct: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-
1271 < Location: https://help.shopify.com/
1272 < Referrer-Policy: origin-when-cross-origin
1273 < Server-Timing: processing;dur=4, socket_queue;dur=0.66, edge;dur=2.057
1274 < Server: cloudflare
1275 < Set-Cookie: _shopify_fs=2021-01-24T19%3A46%3A19Z; Expires=Mon, 24-Jan-22 19:4
1276 < Set-Cookie: _shopify_fs=2021-01-24T19%3A46%3A19Z; Expires=Mon, 24-Jan-22 19:4
1277 < Set-Cookie: _shopify_fs=2021-01-24T19%3A46%3A20Z; Expires=Mon, 24-Jan-22 19:4
1278 < Set-Cookie: _shopify_fs=2021-01-24T19%3A46%3A21Z; Expires=Mon, 24-Jan-22 19:4
1279 < Set-Cookie: _shopify_fs=2021-01-24T19%3A46%3A21Z; Expires=Mon, 24-Jan-22 19:4
1280 < Set-Cookie: _shopify_fs=2021-01-24T19%3A46%3A22Z; Expires=Mon, 24-Jan-22 19:4
1281 < Set-Cookie: _shopify_fs=2021-01-24T19%3A46%3A24Z; Expires=Mon, 24-Jan-22 19:4
1282 < Set-Cookie: _shopify_fs=2021-01-24T19%3A46%3A28Z; Expires=Mon, 24-Jan-22 19:4
1283 < Set-Cookie: _shopify_fs=2021-01-24T19%3A46%3A29Z; Expires=Mon, 24-Jan-22 19:4
NORMAL master [+]
```

```

08bcc124b4c5724a067bbe271812a809d4df14d1...d/recon/ben/roots/docs.shopify.com - VIM
08bcc124b4c5724a067bbe271812a809d4df14d1...d/recon/ben/roots/docs.shopify.com - VIM 100x26
<bbe271812a809d4df14d1.headers> X <d/08bcc124b4c5724a067bbe271812a809d4df14d1.he
1 GET https://docs.shopify.com
2
3
4 < HTTP/1.1 301 Moved Permanently
5 < X-Runtime: 0.003415
6 < X-Permitted-Cross-Domain-Policies: none
7 < Server-Timing: processing;dur=4, socket_queue;dur=0.66, edge;dur=2.057
8 < Date: Sun, 24 Jan 2021 19:46:29 GMT
9 < X-Request-Id: 35bd41b50f121d90364035de5827aa5f
10 < Location: https://help.shopify.com/
11 < Cache-Control: no-cache
12 < X-Xss-Protection: 1; mode=block
13 < X-Download-Options: noopen
14 < X-Dc: gcp-us-east1,gke
15 < Content-Type: text/html
16 < Content-Length: 91
17 < Strict-Transport-Security: max-age=63072000; includeSubDomains; preload
18 < Referrer-Policy: origin-when-cross-origin
19 < X-Frame-Options: SAMEORIGIN
20 < X-Content-Type-Options: nosniff
~
~
NORMAL master <24a067bbe271812a809d4df14d1.headers> utf-8[unix] 35%
```

Thus docs.shopify.com seems a unique server for that case. It doesn't have the x shopify as headers as seen on the others which as well do not have server timing header. This gives a redirect to help.shopify.com which I further explore as seen in video4 provided. *This might be added as the third list of probable vulnerability to be further exploited for proof.*

A terminal window with a dark background. The prompt is 'tom@scan:~/recon/ben/roots (master)'. The command entered is 'find . -type f -name *.body | html-tool tags title | vim -'. The terminal shows the command being executed and the output being piped into vim.

Testing Methodology

For our penetration testing and bug bounty exploitation, we used the bash terminal as well as number of external GitHub repositories in addition to burp suite for port scanning. The exploitations were carried out within Hackerone platform. The probable aforementioned vulnerabilities were cited on the strategy of highlighting any shopify assets that look abnormal or strange from the rest of the assets.

Recommended Actions

For this bounty, the following were made as recommendations for future scope of work and exploitations;

- Further exploitation to provide proof of our proposed vulnerability highlights
- Deeper exploitations for domain redirections
- APIs versions which still exist on shopify can further be analysed for probable vulnerabilities