Sri Lanka Institute of Information Technology

Lark Technologies Bug Bounty Assignment

Web-Audit

Individual Assignment

IE2062 – Web Security

Submitted by:

Student Registration Number	Student Name
IT20620066	Weththasinghe M N G

One Drive Link - https://mysliit-my.sharepoint.com/:f:/g/personal/it20620066_my_sliit_lk/EkrTFFigncZMh5O1aWkUsGIB9LibW07PBYotuKzQe4wt7Q?e=8pLJZY

Date of Submission
06/05/2022

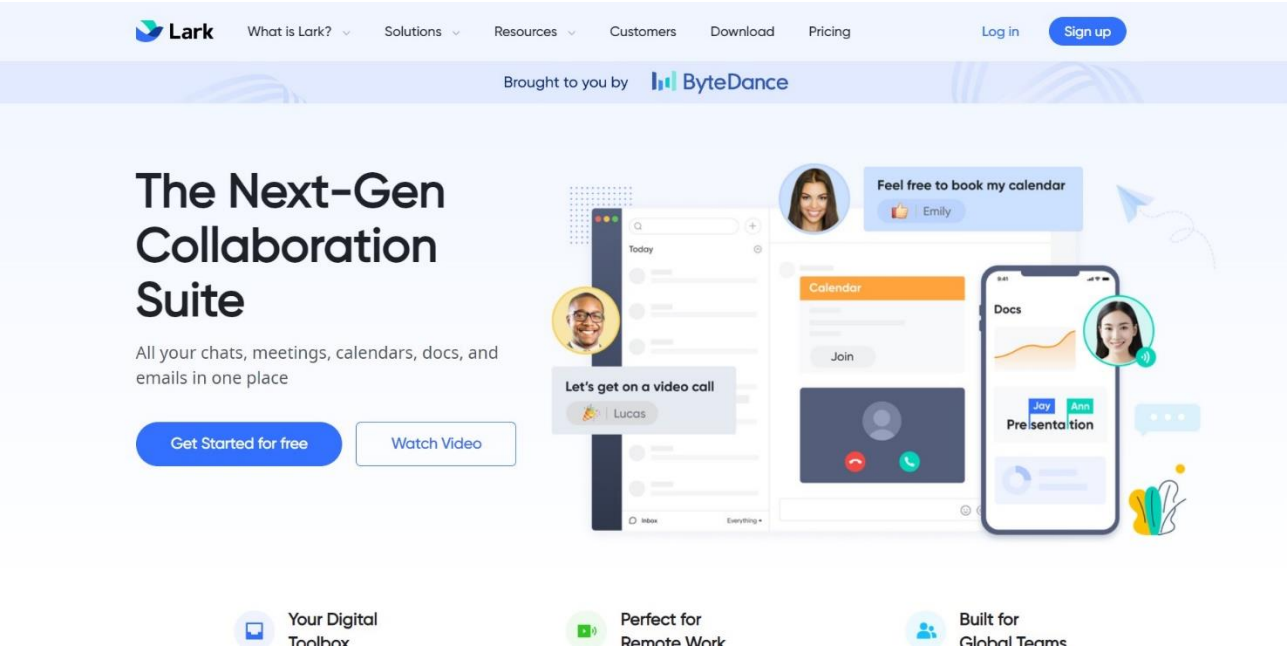
Table of Contents

Purpose

The goal of this web audit assignment is to scan the website and detect any security flaws. I'm using my academic background to detect website vulnerabilities. The website was discovered by the "Hackerone" website. www.hacker.com is a website dedicated to hackers.

Object – Lark Technologies
Name of the domain - https://www.larksuite.com

I've chosen a website to conduct my web audit on.



Scope

Lark Technologies, Inc. specializes in healthcare software. The company provides a platform for chronic disease prevention and management. Lark Technologies works with clients all across the world.

I chose 5 sub domains from that website to begin my audit, and all of the in-scope sub domains came from the HackerOne Bug Bounty website. The subdomains that were chosen are shown below.

Main Domain - https://www.larksuite.com

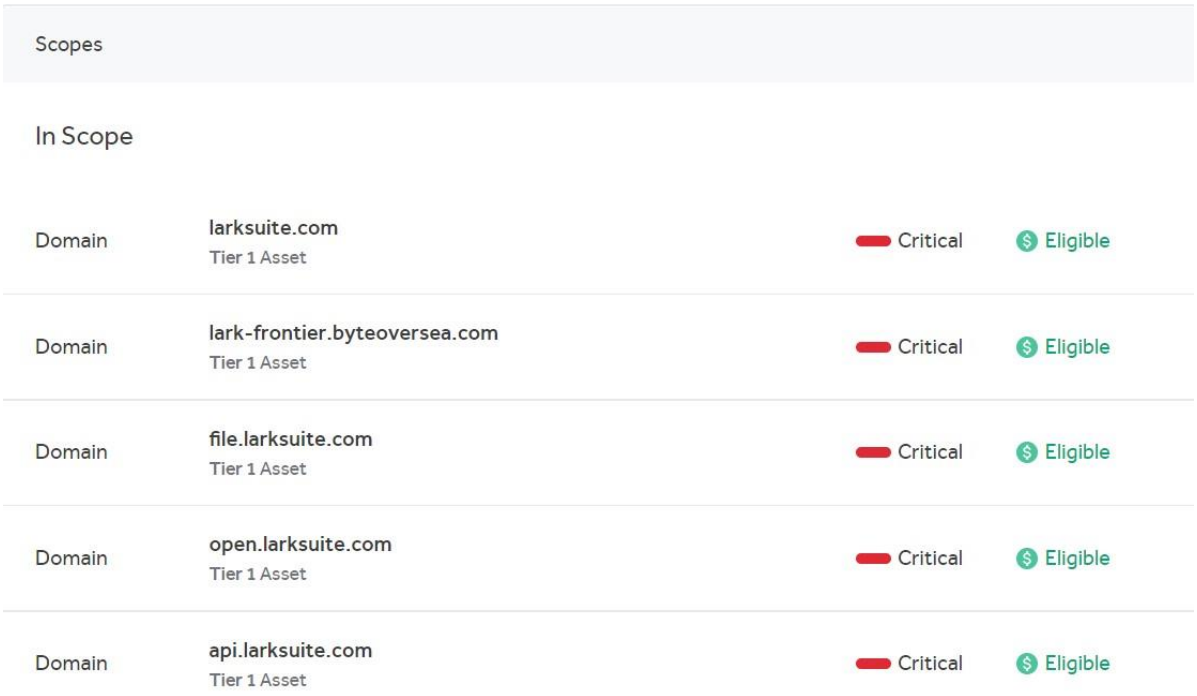
Sub Domain - larksuite.com

lark-frontier.byteoversea.com

file.larksuite.com

open.larksuite.com

api.larksuite.com



Summery

The lark technology website we have selected from the HackerOne website performs a detailed analysis of the vulnerability and vulnerability used by various software. Overall, device security is strong, with built-in medium norms, policies, and development. Although it is critical to have the web application's security tightened at all times.

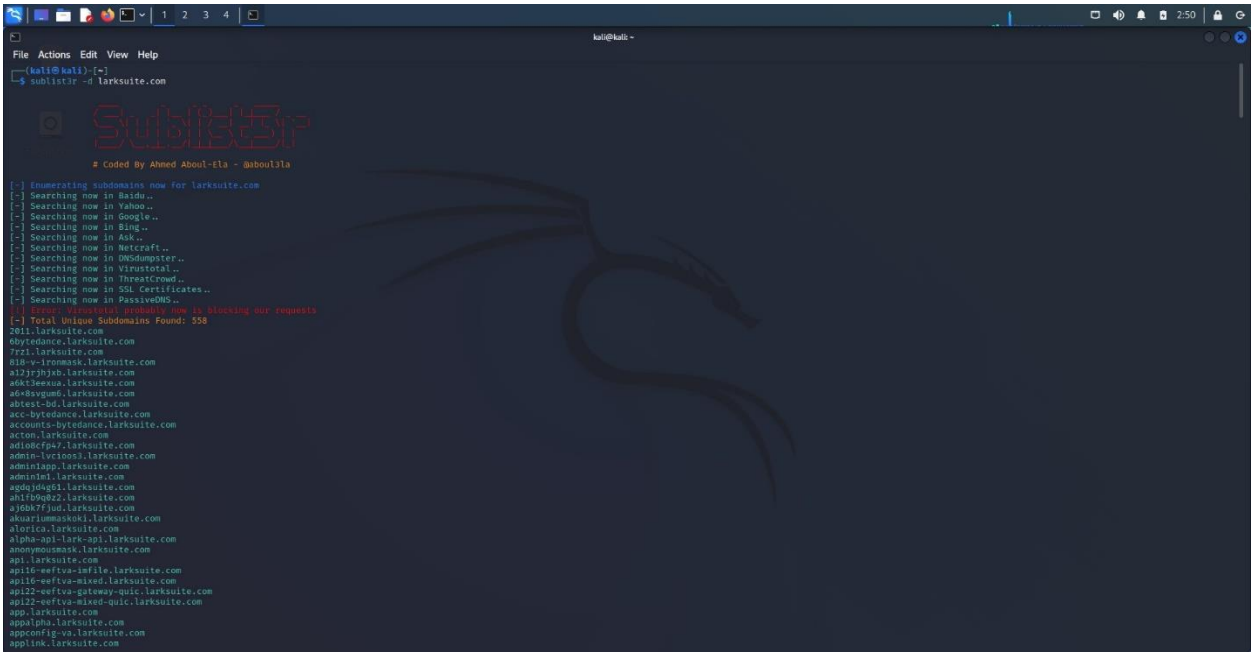
Used Tools

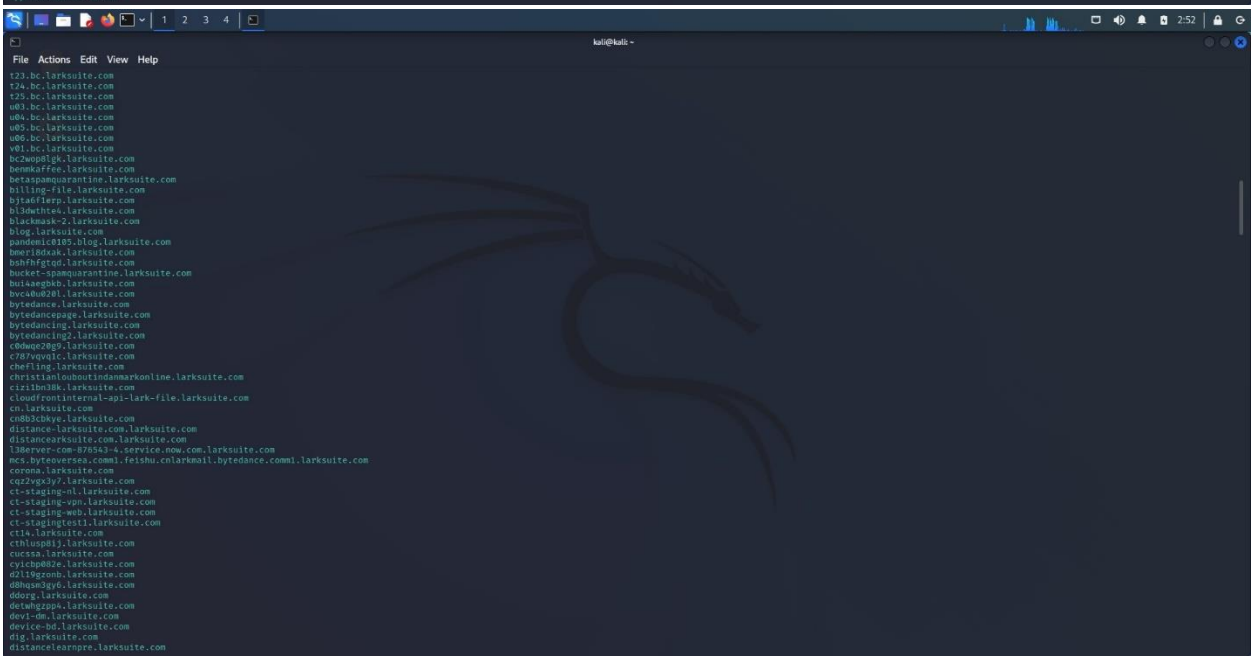
To gather information and assess hazards, I utilized certain inbuilt kali Linux tools as well as numerous tools from GitHub.

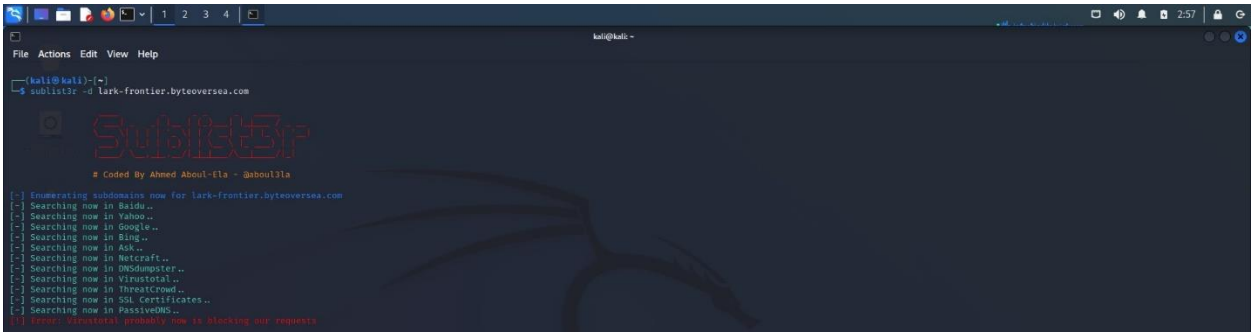
- Sublist3r
- Sub finder
- Sub domainizer
- Netsparker
- Nmap

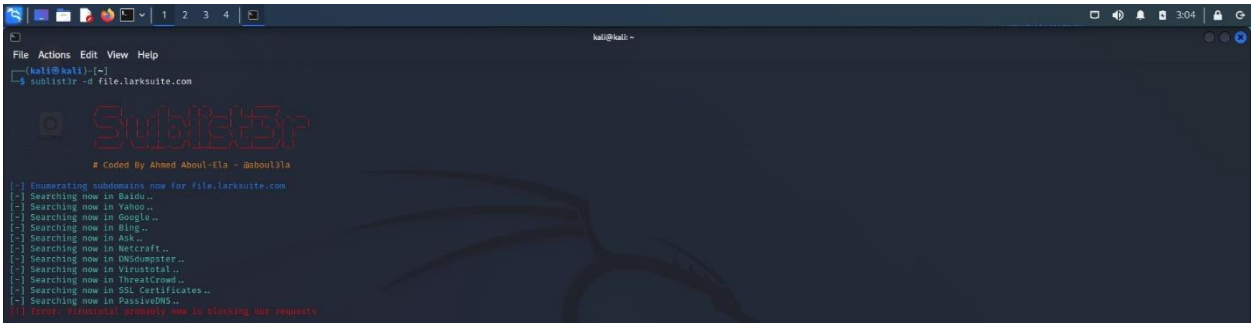
Sublist3r

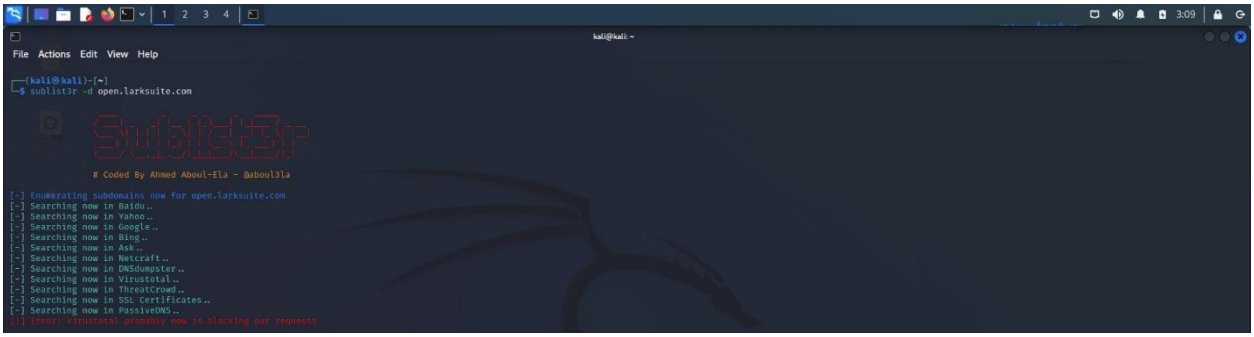
Sublist3r is a python utility that uses OSINT to enumerate website subdomains. It assists penetration testers and bug hunters in gathering and collecting subdomains for the site they are targeting. Sublist3r uses a variety of search engines to find subdomains, including Google, Yahoo, Bing, Baidu, and Ask.

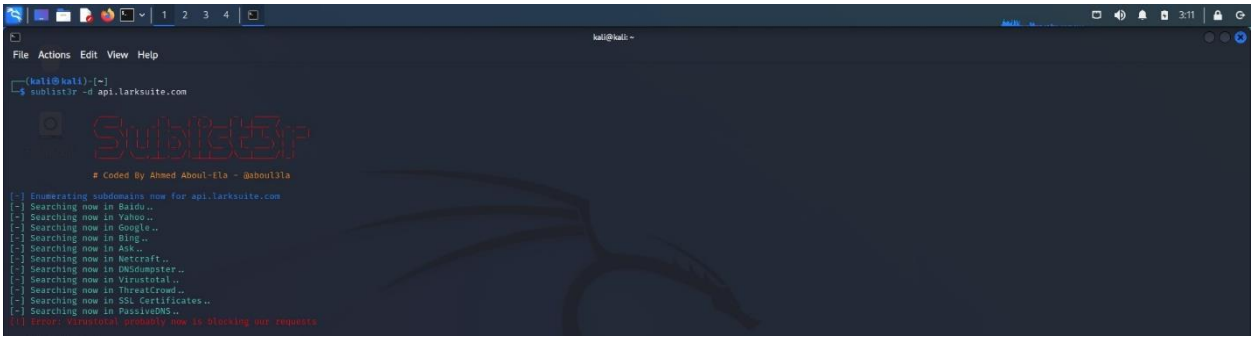

kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ sublist3r -d larksuite.com
# Coded By Ahmed Aboul-Ela - @aboul3la
[-] Enumerating subdomains now for larksuite.com
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
[!] Error: Virustotal probably now is blocking our requests
[-] Total Unique Subdomains Found: 558
2011.larksuite.com
6bytedance.larksuite.com
7rz1.larksuite.com
818-v-ironmask.larksuite.com
a12jrjbjxb.larksuite.com
a6kt3eexua.larksuite.com
a6x8svgum6.larksuite.com
abtest-bd.larksuite.com
acc-bytedance.larksuite.com
accounts-bytedance.larksuite.com
acton.larksuite.com
adio8cfp47.larksuite.com
admin-lucipos3.larksuite.com
admin1app.larksuite.com
admin1m1.larksuite.com
agdqjd4g61.larksuite.com
ah1fb9qb22.larksuite.com
aj6bk7fjud.larksuite.com
akuariummaskoki.larksuite.com
alorica.larksuite.com
alpha-ap1-lark-ap1.larksuite.com
anonymousmask.larksuite.com
api.larksuite.com
api16-eeftva-imfile.larksuite.com
api16-eeftva-mixed.larksuite.com
api22-eeftva-gateway-quic.larksuite.com
api22-eeftva-mixed-quic.larksuite.com
app.larksuite.com
appalpha.larksuite.com
appconfig-va.larksuite.com
applink.larksuite.com


kali@kali: ~
File Actions Edit View Help
t23.bc.larksuite.com
t24.bc.larksuite.com
t25.bc.larksuite.com
u03.bc.larksuite.com
u04.bc.larksuite.com
u05.bc.larksuite.com
u06.bc.larksuite.com
v01.bc.larksuite.com
bc2wxp81gk.larksuite.com
benmkaffee.larksuite.com
betaspamquarantine.larksuite.com
billing-file.larksuite.com
bytaGflerp.larksuite.com
bl3dwthte4.larksuite.com
blackmask-2.larksuite.com
blog.larksuite.com
pandemic0105.blog.larksuite.com
bmeri8dxak.larksuite.com
bshfhfgtqd.larksuite.com
bucket-spamquarantine.larksuite.com
bui4aegbkb.larksuite.com
bvc40u020l.larksuite.com
bytedance.larksuite.com
bytedancepage.larksuite.com
bytedancing.larksuite.com
bytedancing2.larksuite.com
c0dwqe20g9.larksuite.com
c787vqvq1c.larksuite.com
chefling.larksuite.com
christianlouboutindanmarkonline.larksuite.com
cizi1bn38k.larksuite.com
cloudfrontinternal-ap1-lark-file.larksuite.com
cn.larksuite.com
cn8b3cbkye.larksuite.com
distance-larksuite.com.larksuite.com
distancearksuite.com.larksuite.com
l38erver-com-876543-4.service.now.com.larksuite.com
mcs.byteoversea.comm1.feishu.cnlarkmail.bytedance.comm1.larksuite.com
corona.larksuite.com
cqz2vgx3y7.larksuite.com
ct-staging-nl.larksuite.com
ct-staging-vpn.larksuite.com
ct-staging-web.larksuite.com
ct-stagingtest1.larksuite.com
ct14.larksuite.com
cthlusp8ij.larksuite.com
cucssa.larksuite.com
cyichp0d2e.larksuite.com
d2l19gzonb.larksuite.com
d8hqsm3gy6.larksuite.com
ddorg.larksuite.com
detwhgzpp4.larksuite.com
dev1-dm.larksuite.com
device-bd.larksuite.com
dig.larksuite.com
distancelearnpre.larksuite.com


kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ sublist3r -d lark-frontier.byteoversea.com
# Coded By Ahmed Aboul-Ela - @aboul3la
[-] Enumerating subdomains now for lark-frontier.byteoversea.com
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
[!] Error: Virustotal probably now is blocking our requests





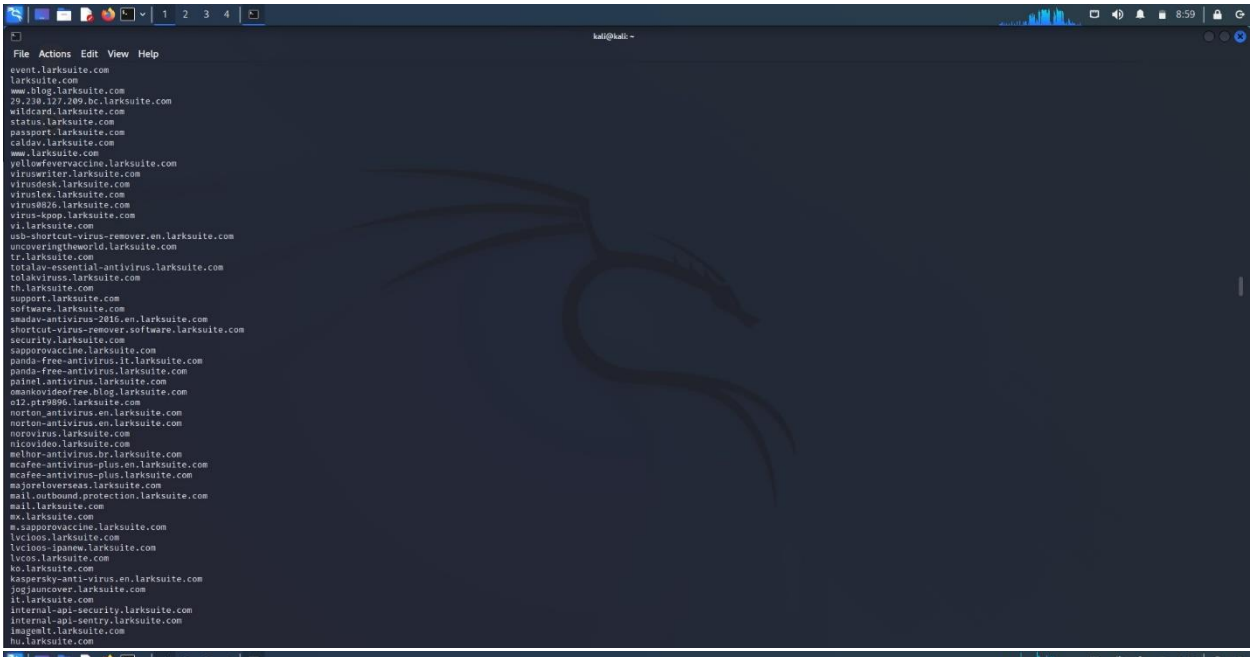


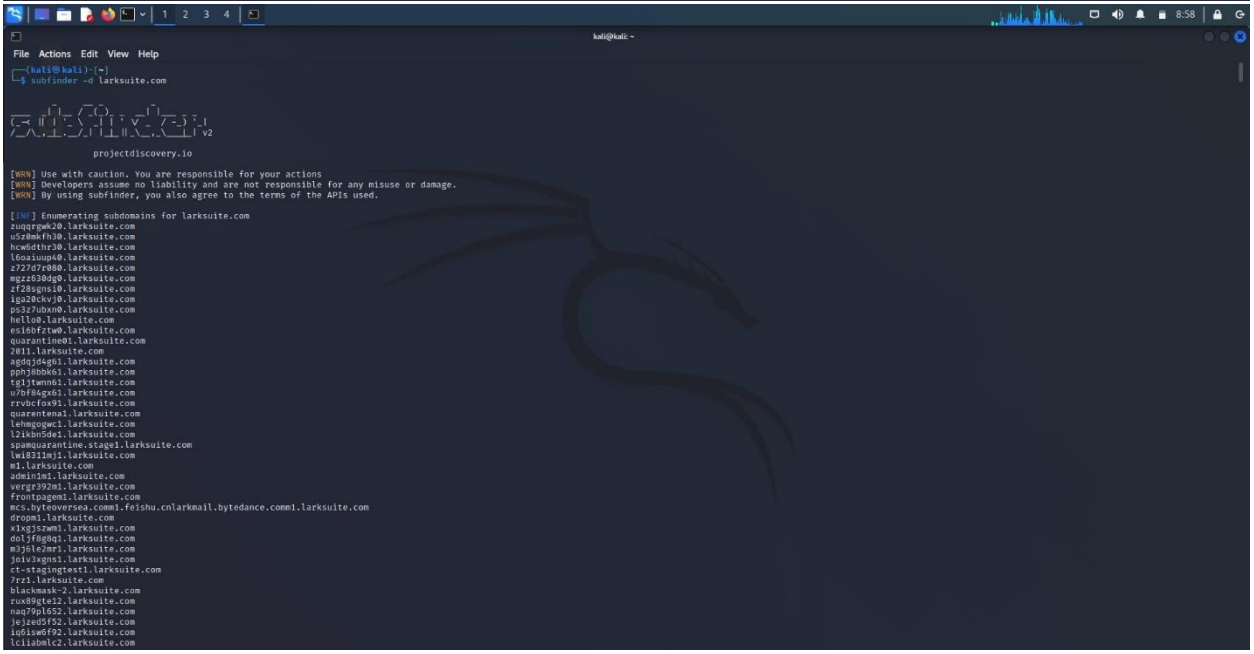
- Out of the five domains I selected, sublist3r searched for the subdomain and found 558 subdomains for only the first domain, but no subdomain was found in the other four domains.

Sub Finder

Sub finder is a subdomain discovery tool that uses passive online sources to locate acceptable subdomains for websites. It features a straightforward modular design

that is intended for speed. Sub finder was designed to accomplish one thing and one thing well: passive subdomain enumeration.

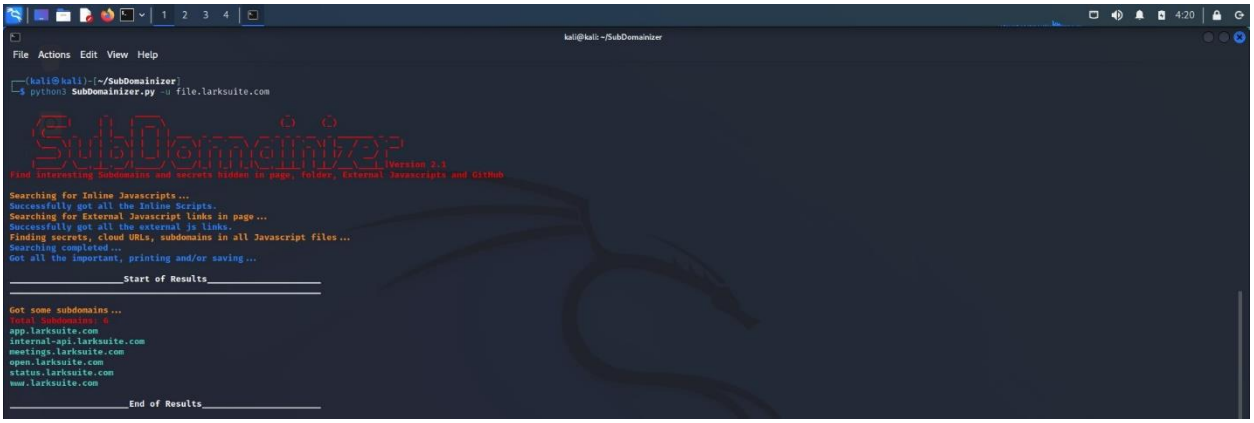


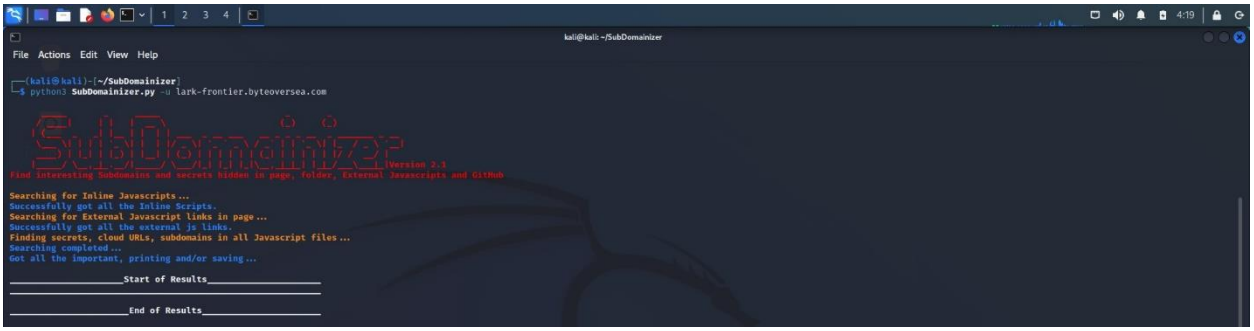


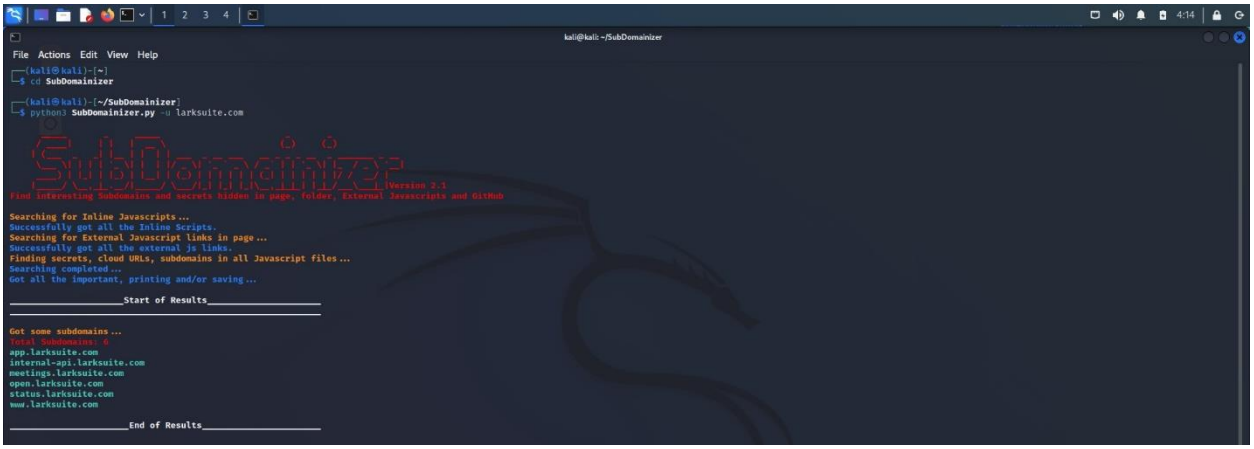
- Like sublist3r, out of the five domains I entered, only the first domain shows the subdomain and the other four domains do not show the subdomain.

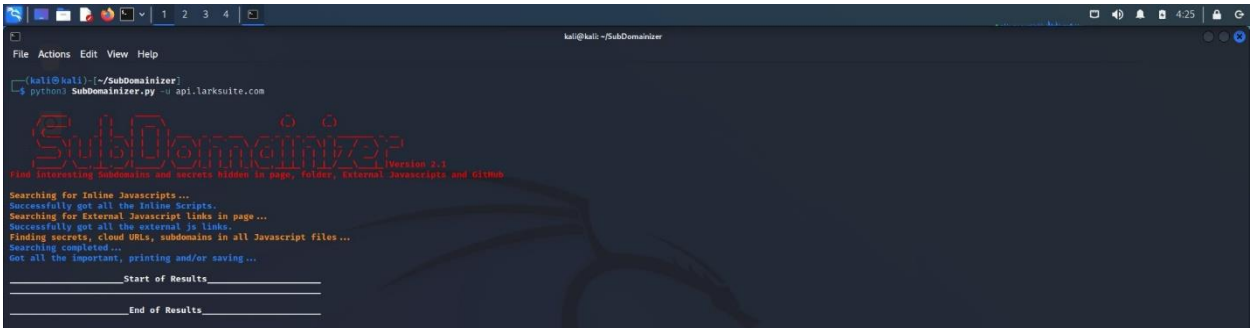
Sub domainizer

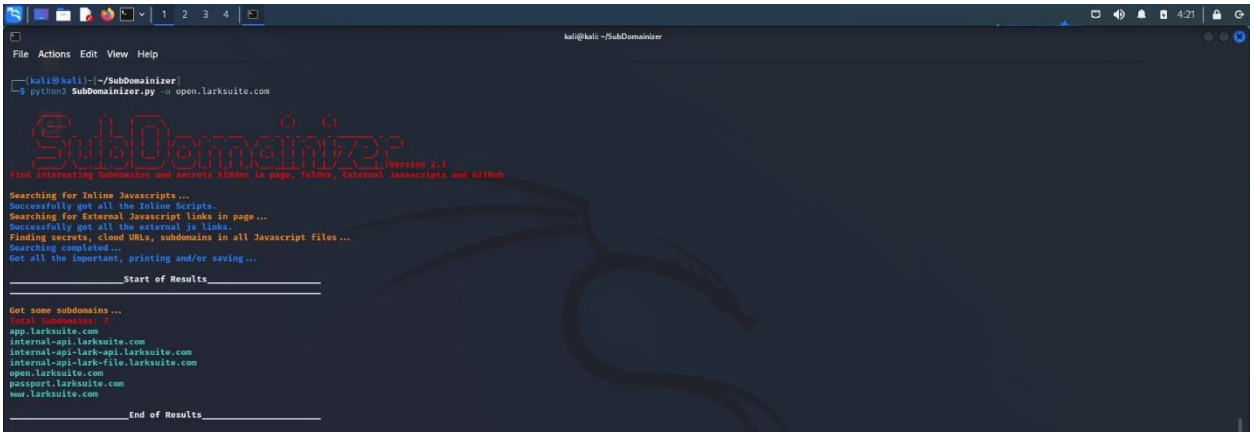
SubDomainizer is a tool for locating subdomains. SubDomainizer is used for the target's SubDomainizer. This tool is used to locate subdomains on a website or within a web application.









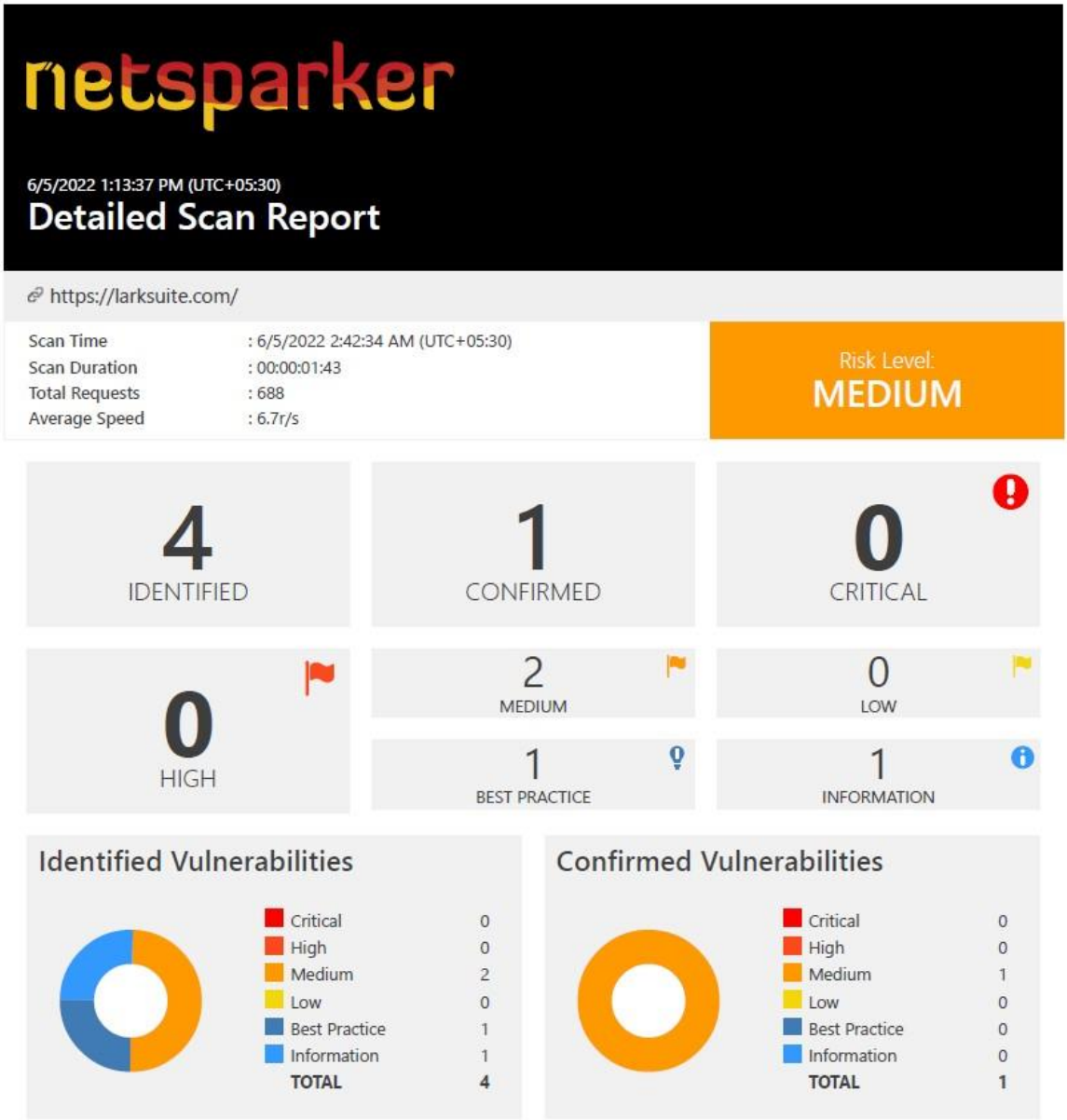


- Sublist3r and Sub finder show that only one of the five domains I entered contains a subdomain, but sub domainizer shows that three of the five domains I entered have subdomains. It also shows that there are only six subdomains in a domain that shows subdomain 558 through sublist3r and sub finder.

Netsparker

Netsparker is a web application security scanner that automatically detects security flaws in online applications, websites, and web services. It's a user-friendly and accurate program that detects SQL injections, cross-site scripting (XSS), and other major security problems.

Vulnerable subdomain: larksuite.com



Vulnerability Summary

CONFIRM	VULNERABILITY	METHOD	URL	PARAMETER
	HTTP Strict Transport Security (HSTS) Policy Not Enabled	GET	https://larksuite.com/	
	Weak Ciphers Enabled	GET	https://larksuite.com/	
	Expect-CT Not Enabled	GET	https://larksuite.com/	
	Nginx Web Server Identified	GET	https://larksuite.com/	

1. HTTP Strict Transport Security (HSTS) Policy Not Enabled

Netsparker identified that HTTP Strict Transport Security (HSTS) policy is not enabled.

The target website is being served from not only HTTPS but also HTTP and it lacks of HSTS policy implementation.

HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure (HTTPS) connections. The HSTS Policy is communicated by the server to the user agent via a HTTP response header field named "Strict-Transport-Security". HSTS Policy specifies a period of time during which the user agent shall access the server in only secure fashion.

When a web application issues HSTS Policy to user agents, conformant user agents behave as follows:

- Automatically turn any insecure (HTTP) links referencing the web application into secure (HTTPS) links. (For instance, http://example.com/some/page/ will be modified to https://example.com/some/page/ before accessing the server.)
- If the security of the connection cannot be ensured (e.g. the server's TLS certificate is self-signed), user agents show an error message and do not allow the user to access the web application.

Issue Type: Medium

Vulnerabilities

1.1. https://api.larksuite.com/

Certainty

Request

```
GET / HTTP/1.1
Host: api.larksuite.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 298.9397 Total Bytes Received : 1230 Body Length : 564 Is Compressed : No

```
HTTP/1.1 404 Not Found
Server-Timing: cdn-cache; desc=MISS, edge; dur=0, origin; dur=234
x-tt-trace-host: 0120af35ddbbf5e8c5a6162b4bf481bc96cc7a56d11f4c9cf7ebb9268659a11a3509bbb18fa82352c5ab68
ec75b97bb255b478692af752094ed316982b6df48a3cdbb5c8d6197bb913d5c45e39146b90ae
Server: nginx
X-Akamai-Request-ID: 857667f
x-tt-trace-tag: id=16;cdn-cache=miss;type=dyn
X-Cache: TCP_MISS from a222-165-168-197.deploy.akamaitechnologies.com (AkamaiGHost/10.8.2-41841244) (-)
Content-Length: 180
Connection: keep-alive
Content-Type: text/html
Content-Encoding:
Date: Sat, 04 Jun 2022 22:08:48 GMT
Vary: Accept-Encoding
X-Origin-Response-Time: 234,222.165.168.197

<html>
<head><title>404 Not Found</title></head>
<body bgcolor="white">
<center><h1>404 Not Found</h1></center>
<hr><center>nginx</center>
</body>
</html>
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
```

Remedy

Configure your webserver to redirect HTTP requests to HTTPS.

i.e. for Apache, you should have modification in the httpd.conf. For more configurations, please refer to External References section.

```
# load module
LoadModule headers_module modules/mod_headers.so

# redirect all HTTP to HTTPS (optional)
<VirtualHost *:80>
       ServerAlias *
       RewriteEngine On
       RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [redirect=301]
```

```
</VirtualHost>

# HTTPS-Host-Configuration
<VirtualHost *:443>
      # Use HTTP Strict Transport Security to force client to use secure connections only
      Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"

      # Further Configuration goes here
      [...]
</VirtualHost>
```

CLASSIFICATION

OWASP 2013	A6
OWASP 2017	A3
SANS Top 25	523
CAPEC	217
WASC	4
ISO27001	A.14.1.2

2. Weak Ciphers Enabled

Netsparker detected that weak ciphers are enabled during secure communication (SSL).

You should allow only strong ciphers on your web server to protect secure communication with your visitors.

Impact

Attackers might decrypt SSL traffic between your server and your visitors.

Issue Type: Medium

Vulnerabilities

2.1. https://api.larksuite.com/

CONFIRMED

List of Supported Weak Ciphers

- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xC028)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xC027)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xC014)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)

Request

```
[NETSPARKER] SSL Connection
```

Response

Response Time (ms) : 1 Total Bytes Received : 27 Body Length : 0 Is Compressed : No

```
[NETSPARKER] SSL Connection
```

Actions to Take

1. For Apache, you should modify the SSLCipherSuite directive in the `httpd.conf`.

```
SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4
```

2. Lighttpd:

```
ssl.honor-cipher-order = "enable"
ssl.cipher-list = "EECDH+AESGCM:EDH+AESGCM"
```

3. For Microsoft IIS, you should make some changes to the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**

   a.Click Start, click Run, type `regedt32`or type `regedit`, and then click OK.
   b.In Registry Editor, locate the following registry key: `HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders`
   c.Set "Enabled" DWORD to "0x0" for the following registry keys:

```
SCHANNEL\Ciphers\DES 56/56
SCHANNEL\Ciphers\RC4 64/128
SCHANNEL\Ciphers\RC4 40/128
SCHANNEL\Ciphers\RC2 56/128
SCHANNEL\Ciphers\RC2 40/128
SCHANNEL\Ciphers\NULL
SCHANNEL\Hashes\MD5
```

Remedy

Configure your web server to disallow using weak ciphers.

CLASSIFICATION

PCI DSS v3.2	6.5.4
OWASP 2013	A6
OWASP 2017	A3
SANS Top 25	327
CAPEC	217
WASC	4
ISO27001	A.14.1.3

CVSS 3.0 SCORE

Base	6.8 (Medium)
Temporal	6.8 (Medium)
Environmental	6.8 (Medium)

CVSS Vector String

CVSS:3.0/AV:A/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N

CVSS 3.1 SCORE

Base	6.8 (Medium)
Temporal	6.8 (Medium)
Environmental	6.8 (Medium)

CVSS Vector String

CVSS:3.1/AV:A/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N

3. Missing X-Frame-Options Header

Netsparker detected a missing `X-Frame-Options`header which means that this website could be at risk of a clickjacking attack.

The `X-Frame-Options`HTTP header field indicates a policy that specifies whether the browser should render the transmitted resource within a `frame`or an `iframe`. Servers can declare this policy in the header of their HTTP responses to prevent clickjacking attacks, which ensures that their content is not embedded into other pages or frames.

Impact

Clickjacking is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on a framed page when they were intending to click on the top level page. Thus, the attacker is "hijacking" clicks meant for their page and routing them to other another page, most likely owned by another application, domain, or both.

Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker.

Issue Type: Low

Vulnerabilities

3.1. https://api.larksuite.com/

Certainty

Request

```
GET / HTTP/1.1
Host: api.larksuite.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 407.1239 Total Bytes Received : 1230 Body Length : 564 Is Compressed : No

```
HTTP/1.1 404 Not Found
Server-Timing: cdn-cache; desc=MISS, edge; dur=0, origin; dur=234
x-tt-trace-host: 0120af35ddbbf5e8c5a6162b4bf481bc96cc7a56d11f4c9cf7ebb9268659a11a3509bbb18fa82352c5ab68
ec75b97bb255b478692af752094ed316982b6df48a3cdbb5c8d6197bb913d5c45e39146b90ae
Server: nginx
X-Akamai-Request-ID: 8574dee
x-tt-trace-tag: id=16;cdn-cache=miss;type=dyn
X-Cache: TCP_MISS from a222-165-168-197.deploy.akamaitechnologies.com (AkamaiGHost/10.8.2-41841244) (-)
Content-Length: 180
Connection: keep-alive
Content-Type: text/html
Content-Encoding:
Date: Sat, 04 Jun 2022 22:08:37 GMT
Vary: Accept-Encoding
X-Origin-Response-Time: 234,222.165.168.197

<html>
<head><title>404 Not Found</title></head>
<body bgcolor="white">
<center><h1>404 Not Found</h1></center>
<hr><center>nginx</center>
</body>
</html>
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
```

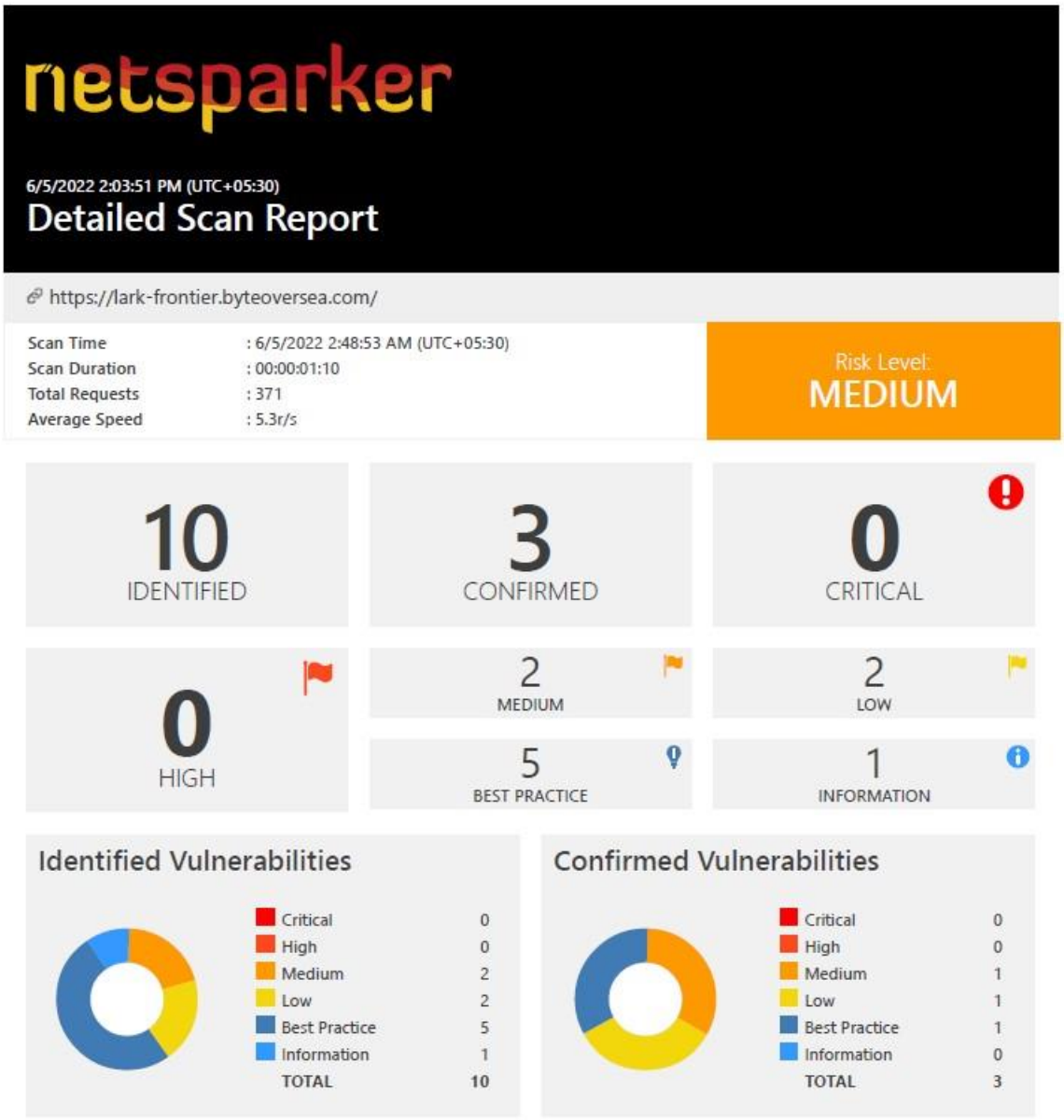
Remedy

- Sending the proper X-Frame-Options in HTTP response headers that instruct the browser to not allow framing from other domains.
    - `X-Frame-Options: DENY`It completely denies to be loaded in frame/iframe.
    - `X-Frame-Options: SAMEORIGIN`It allows only if the site which wants to load has a same origin.
    - `X-Frame-Options: ALLOW-FROM` *`URL`*It grants a specific URL to load itself in a iframe. However please pay attention to that, not all browsers support this.
- Employing defensive code in the UI to ensure that the current frame is the most top level window.

CLASSIFICATION

OWASP 2013	A5
OWASP 2017	A6
SANS Top 25	693
CAPEC	103
ISO27001	A.14.2.5

Vulnerable subdomain: lark-frontier.byteoversea.com



Vulnerability Summary

CONFIRM	VULNERABILITY	METHOD	URL	PARAMETER
	HTTP Strict Transport Security (HSTS) Policy Not Enabled	GET	https://lark-frontier.byteoversea.com/	
	Weak Ciphers Enabled	GET	https://lark-frontier.byteoversea.com/	
	Missing X-Frame-Options Header	GET	https://lark-frontier.byteoversea.com/	
	Insecure Transportation Security Protocol Supported (TLS 1.0)	GET	https://lark-frontier.byteoversea.com/	
	Content Security Policy (CSP) Not Implemented	GET	https://lark-frontier.byteoversea.com/	
	Expect-CT Not Enabled	GET	https://lark-frontier.byteoversea.com/	
	Missing X-XSS-Protection Header	GET	https://lark-frontier.byteoversea.com/	
	Referrer-Policy Not Implemented	GET	https://lark-frontier.byteoversea.com/	
	Insecure Transportation Security Protocol Supported (TLS 1.1)	GET	https://lark-frontier.byteoversea.com/	
	Nginx Web Server Identified	GET	https://lark-frontier.byteoversea.com/	

1. HTTP Strict Transport Security (HSTS) Policy Not Enabled

Issue Type: Low

Netsparker identified that HTTP Strict Transport Security (HSTS) policy is not enabled.

The target website is being served from not only HTTPS but also HTTP and it lacks of HSTS policy implementation.

HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure (HTTPS) connections. The HSTS Policy is communicated by the server to the user agent via a HTTP response header field named "Strict-Transport-Security". HSTS Policy specifies a period of time during which the user agent shall access the server in only secure fashion.

When a web application issues HSTS Policy to user agents, conformant user agents behave as follows:

- Automatically turn any insecure (HTTP) links referencing the web application into secure (HTTPS) links. (For instance, http://example.com/some/page/ will be modified to https://example.com/some/page/ before accessing the server.)
- If the security of the connection cannot be ensured (e.g. the server's TLS certificate is self-signed), user agents show an error message and do not allow the user to access the web application.

Vulnerabilities

1.1. https://lark-frontier.byteoversea.com/

Certainty

Request

```
GET / HTTP/1.1
Host: lark-frontier.byteoversea.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 232.2781 Total Bytes Received : 1018 Body Length : 564 Is Compressed : No

```
HTTP/1.1 404 Not Found
Server-Timing: cdn-cache; desc=MISS, edge; dur=1, origin; dur=226
Server: nginx
x-tt-trace-tag: id=16;cdn-cache=miss;type=dyn
Expires: Sat, 04 Jun 2022 21:19:07 GMT
Pragma: no-cache
X-Origin-Response-Time: 227,222.165.168.198
Content-Length: 180
Connection: keep-alive
Content-Type: text/html
Content-Encoding:
Date: Sat, 04 Jun 2022 21:19:07 GMT
Vary: Accept-Encoding
Cache-Control: max-age=0, no-cache, no-store

<html>
<head><title>404 Not Found</title></head>
<body bgcolor="white">
<center><h1>404 Not Found</h1></center>
<hr><center>nginx</center>
</body>
</html>
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
```

Remedy

Configure your webserver to redirect HTTP requests to HTTPS.

i.e. for Apache, you should have modification in the httpd.conf. For more configurations, please refer to External References section.

```
# load module
LoadModule headers_module modules/mod_headers.so

# redirect all HTTP to HTTPS (optional)
<VirtualHost *:80>
       ServerAlias *
       RewriteEngine On
       RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [redirect=301]
</VirtualHost>
```

```
# HTTPS-Host-Configuration
<VirtualHost *:443>
      # Use HTTP Strict Transport Security to force client to use secure connections only
      Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"

      # Further Configuration goes here
      [...]
</VirtualHost>
```

CLASSIFICATION

OWASP 2013	A6
OWASP 2017	A3
SANS Top 25	523
CAPEC	217
WASC	4
ISO27001	A.14.1.2

2. Weak Ciphers Enabled

Netsparker detected that weak ciphers are enabled during secure communication (SSL).

You should allow only strong ciphers on your web server to protect secure communication with your visitors.

Impact

Attackers might decrypt SSL traffic between your server and your visitors.

Issue Type: Medium

Vulnerabilities

2.1. https://lark-frontier.byteoversea.com/

CONFIRMED

List of Supported Weak Ciphers

- TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
- TLS_RSA_WITH_AES_128_CBC_SHA (0x002F)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xC014)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)
- TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003D)
- TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003C)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xC028)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xC027)

Request

```
[NETSPARKER] SSL Connection
```

Response

Response Time (ms) : 1 Total Bytes Received : 27 Body Length : 0 Is Compressed : No

```
[NETSPARKER] SSL Connection
```

Actions to Take

1. For Apache, you should modify the SSLCipherSuite directive in the `httpd.conf`.

```
SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4
```

2. Lighttpd:

```
ssl.honor-cipher-order = "enable"
ssl.cipher-list = "EECDH+AESGCM:EDH+AESGCM"
```

3. For Microsoft IIS, you should make some changes to the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**

   a.Click Start, click Run, type `regedt32`or type `regedit`, and then click OK.
   b.In Registry Editor, locate the following registry key: `HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders`
   c.Set "Enabled" DWORD to "0x0" for the following registry keys:

```
SCHANNEL\Ciphers\DES 56/56
SCHANNEL\Ciphers\RC4 64/128
SCHANNEL\Ciphers\RC4 40/128
SCHANNEL\Ciphers\RC2 56/128
SCHANNEL\Ciphers\RC2 40/128
SCHANNEL\Ciphers\NULL
SCHANNEL\Hashes\MD5
```

Remedy

Configure your web server to disallow using weak ciphers.

CLASSIFICATION

PCI DSS v3.2	6.5.4
OWASP 2013	A6
OWASP 2017	A3
SANS Top 25	327
CAPEC	217
WASC	4
ISO27001	A.14.1.3

CVSS 3.0 SCORE

Base	6.8 (Medium)
Temporal	6.8 (Medium)
Environmental	6.8 (Medium)

CVSS Vector String

CVSS:3.0/AV:A/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N

CVSS 3.1 SCORE

Base	6.8 (Medium)
Temporal	6.8 (Medium)
Environmental	6.8 (Medium)

CVSS Vector String
CVSS:3.1/AV:A/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N

3. Insecure Transportation Security Protocol Supported (TLS 1.0) Low

Netsparker detected that insecure transportation security protocol (TLS 1.0) is supported by your web server.

TLS 1.0 has several flaws. An attacker can cause connection failures and they can trigger the use of TLS 1.0 to exploit vulnerabilities like BEAST (Browser Exploit Against SSL/TLS).

Websites using TLS 1.0 are considered non-compliant by PCI since 30 June 2018.

Impact

Attackers can perform man-in-the-middle attacks and observe the encryption traffic between your website and its visitors.

Vulnerabilities

3.1. https://lark-frontier.byteoversea.com/

CONFIRMED

Request

```
[NETSPARKER] SSL Connection
```

Response

Response Time (ms) : 1 Total Bytes Received : 27 Body Length : 0 Is Compressed : No

```
[NETSPARKER] SSL Connection
```

Actions to Take

We recommended to disable TLS 1.0 and replace it with TLS 1.2 or higher. See Remedy section for more details.

Remedy

Configure your web server to disallow using weak ciphers. You need to restart the web server to enable changes.

- For Apache, adjust the SSLProtocol directive provided by the mod_ssl module. This directive can be set either at the server level or in a virtual host configuration.

```
SSLProtocol +TLSv1.2
```

- For Nginx, locate any use of the directive ssl_protocols in the `nginx.conf`file and remove `TLSv1`.

```
ssl_protocols TLSv1.2;
```

- For Microsoft IIS, you should make some changes on the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**
    1. Click on Start and then Run, type `regedt32`or `regedit`, and then click OK.
    2. In Registry Editor, locate the following registry key or create if it does not exist:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.0\
```

    3. Locate a key named `Server`or create if it doesn't exist.
    4. Under the `Server`key, locate a DWORD value named `Enabled`or create if it doesn't exist and set its value to "0".
- For lighttpd, put the following lines in your configuration file:

```
ssl.use-sslv2 = "disable"
ssl.use-sslv3 = "disable"
ssl.openssl.ssl-conf-cmd = ("Protocol" => "-TLSv1.1, -TLSv1, -SSLv3") # v1.4.48 or up
ssl.ec-curve = "secp384r1"
```

CLASSIFICATION

PCI DSS v3.2	6.5.4
OWASP 2013	A6
OWASP 2017	A3
SANS Top 25	326
CAPEC	217
WASC	4
HIPAA	164.306
ISO27001	A.14.1.3

4. Missing X-Frame-Options Header

Low

Netsparker detected a missing `X-Frame-Options`header which means that this website could be at risk of a clickjacking attack.

The `X-Frame-Options`HTTP header field indicates a policy that specifies whether the browser should render the transmitted resource within a `frame`or an `iframe`. Servers can declare this policy in the header of their HTTP responses to prevent clickjacking attacks, which ensures that their content is not embedded into other pages or frames.

Impact

Clickjacking is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on a framed page when they were intending to click on the top level page. Thus, the attacker is "hijacking" clicks meant for their page and routing them to other another page, most likely owned by another application, domain, or both.

Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker.

Vulnerabilities

4.1. https://lark-frontier.byteoversea.com/

Certainty

Request

```
GET / HTTP/1.1
Host: lark-frontier.byteoversea.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 438.5899 Total Bytes Received : 1018 Body Length : 564 Is Compressed : No

```
HTTP/1.1 404 Not Found
Server-Timing: cdn-cache; desc=MISS, edge; dur=0, origin; dur=225
Server: nginx
x-tt-trace-tag: id=16;cdn-cache=miss;type=dyn
Expires: Sat, 04 Jun 2022 21:18:56 GMT
Pragma: no-cache
X-Origin-Response-Time: 225,222.165.168.198
Content-Length: 180
Connection: keep-alive
Content-Type: text/html
Content-Encoding:
Date: Sat, 04 Jun 2022 21:18:56 GMT
Vary: Accept-Encoding
Cache-Control: max-age=0, no-cache, no-store

<html>
<head><title>404 Not Found</title></head>
<body bgcolor="white">
<center><h1>404 Not Found</h1></center>
<hr><center>nginx</center>
</body>
</html>
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
```

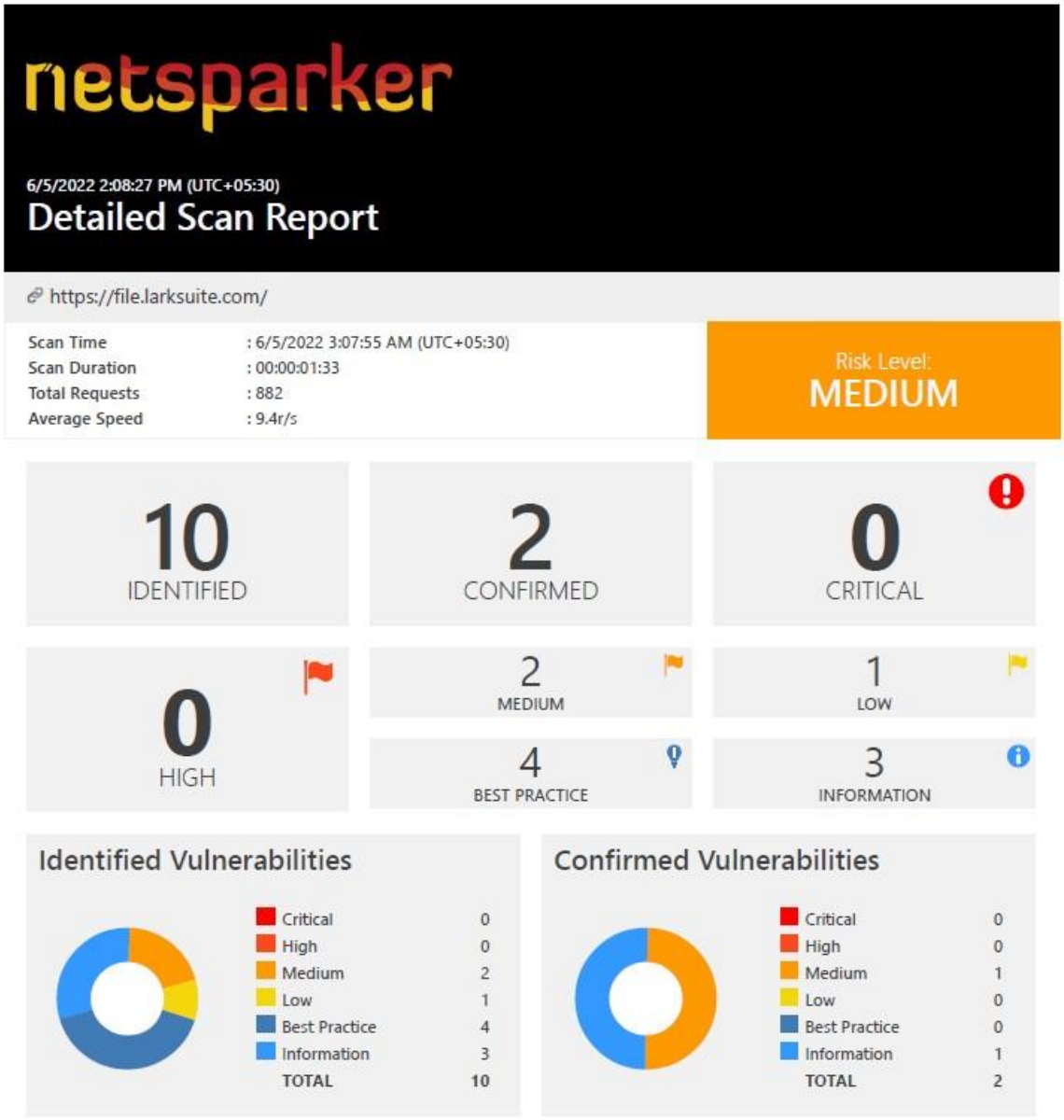
Remedy

- Sending the proper X-Frame-Options in HTTP response headers that instruct the browser to not allow framing from other domains.
    - `X-Frame-Options: DENY`It completely denies to be loaded in frame/iframe.
    - `X-Frame-Options: SAMEORIGIN`It allows only if the site which wants to load has a same origin.
    - `X-Frame-Options: ALLOW-FROM` *`URL`*It grants a specific URL to load itself in a iframe. However please pay attention to that, not all browsers support this.
- Employing defensive code in the UI to ensure that the current frame is the most top level window.

CLASSIFICATION

OWASP 2013	A5
OWASP 2017	A6
SANS Top 25	693
CAPEC	103
ISO27001	A.14.2.5

Vulnerable subdomain: file.larksuite.com



Vulnerability Summary

CONFIRM	VULNERABILITY	METHOD	URL	PARAMETER
	HTTP Strict Transport Security (HSTS) Policy Not Enabled	GET	https://file.larksuite.com/	
	Weak Ciphers Enabled	GET	https://file.larksuite.com/	
	Missing X-Frame-Options Header	GET	https://file.larksuite.com/%3Cscript%3Ealert(0)%3C/	
	Content Security Policy (CSP) Not Implemented	GET	https://file.larksuite.com/%3Cscript%3Ealert(0)%3C/	
	Expect-CT Not Enabled	GET	https://file.larksuite.com/	
	Missing X-XSS-Protection Header	GET	https://file.larksuite.com/%3Cscript%3Ealert(0)%3C/	
	Referrer-Policy Not Implemented	GET	https://file.larksuite.com/%3Cscript%3Ealert(0)%3C/	
	Nginx Web Server Identified	GET	https://file.larksuite.com/	
	Web Application Firewall Detected	GET	https://file.larksuite.com/%3Cscript%3Ealert(0)%3C/script%3E	URI-BASED
	Forbidden Resource	GET	https://file.larksuite.com/%3Cscript%3Ealert(0)%3C/	

1. HTTP Strict Transport Security (HSTS) Policy Not Enabled

Issue Type: Medium

Netsparker identified that HTTP Strict Transport Security (HSTS) policy is not enabled.

The target website is being served from not only HTTPS but also HTTP and it lacks of HSTS policy implementation.

HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure (HTTPS) connections. The HSTS Policy is communicated by the server to the user agent via a HTTP response header field named "Strict-Transport-Security". HSTS Policy specifies a period of time during which the user agent shall access the server in only secure fashion.

When a web application issues HSTS Policy to user agents, conformant user agents behave as follows:

- Automatically turn any insecure (HTTP) links referencing the web application into secure (HTTPS) links. (For instance, http://example.com/some/page/ will be modified to https://example.com/some/page/ before accessing the server.)
- If the security of the connection cannot be ensured (e.g. the server's TLS certificate is self-signed), user agents show an error message and do not allow the user to access the web application.

Vulnerabilities

1.1. https://file.larksuite.com/

Certainty

Request

```
GET / HTTP/1.1
Host: www.larksuite.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 630.0468 Total Bytes Received : 39400 Body Length : 38588 Is Compressed : No

```
HTTP/1.1 200 OK
Server-Timing: cdn-cache; desc=MISS, edge; dur=205, origin; dur=5
Server-Timing: inner; dur=2
x-tt-trace-host: 0120af35ddbbf5e8c5a6162b4bf481bc96d7148b189832a78f46441582259f18166d64dd0b2ffdf449d4f5
7974780a437fe591ddf409976303a3bfaaf8711f4dbc5aa28bc450a48c795f8d2a063d6597c386303ebc05cf0a70d90b0810307
5232c6face2b5cae9b2b281ab34a2b8c09a96a844346cb71255df163eed082b74ddb7
Server: nginx
Content-Length: 9887
x-tt-trace-tag: id=16;cdn-cache=miss;type=dyn
Connection: keep-alive
X-Frame-Options: DENY
X-Tt-Logid: 2022060421380401010000813001D592FF
X-Origin-Response-Time: 5,23.222.3.86
Content-Type: text/html; charset=utf-8
Content-Encoding:
Date: Sat, 04 Jun 2022 21:38:04 GMT
Vary: Accept-Encoding
X-Parent-Response-Time: 95,2.23.158.142
X-Parent-Response-Time: 209,104.75.84.4

<!DOCTYPE html>
<html lang="en_us">
<head>
<title>404 | Lark</title>
<meta content="initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0, width=device-width" name="viewp
ort">
<meta charset="UTF-8" />
<meta name="title" content="404 | Lark" />
<meta name="keywords" content="not found, lark" />
<meta name="description" content="The request URL was not found on this server." />
<meta name='apple-mobile-web-app-capable' content='yes' />
<meta name='full-screen' content='true' />
<meta name='x5-fullscreen' content='true' />
<meta name='360-fullscreen' content='true' />
<meta name='hera-project-version' content="1.2.419" />
<meta name="twitter:card" content="summary_large_image" /><meta name="twitter:site" content="@Larksuit
e" /><meta name="google-site-verification" content="mE6vlIsxdAT-jHj7PreF68g4nj4oEi-xnfZiHvWAzZg" /><met
a name="facebook-domain-verification" content="ysejwnm1oza639gkgnbzcbh3oj0qhn" />
<link rel="shortcut icon" href="https://p16-hera-va.ibyteimg.com/tos-useast2a-i-hn4qzgxq2n/2492c0eff2af
4cca9b9bcabe83a8ebb2~tplv-hn4qzgxq2n-image:0:0.image" type="image/png" /><link rel=" " href="https://p1
6-hera-va.ib
…
```

Remedy

Configure your webserver to redirect HTTP requests to HTTPS.

i.e. for Apache, you should have modification in the httpd.conf. For more configurations, please refer to External References section.

```
# load module
LoadModule headers_module modules/mod_headers.so

# redirect all HTTP to HTTPS (optional)
<VirtualHost *:80>
       ServerAlias *
       RewriteEngine On
       RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [redirect=301]
</VirtualHost>

# HTTPS-Host-Configuration
<VirtualHost *:443>
      # Use HTTP Strict Transport Security to force client to use secure connections only
      Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"

      # Further Configuration goes here
      [...]
</VirtualHost>
```

CLASSIFICATION

OWASP 2013	A6
OWASP 2017	A3
SANS Top 25	523
CAPEC	217
WASC	4
ISO27001	A.14.1.2

2. Weak Ciphers Enabled

Issue Type: Medium

Netsparker detected that weak ciphers are enabled during secure communication (SSL).

You should allow only strong ciphers on your web server to protect secure communication with your visitors.

Impact

Attackers might decrypt SSL traffic between your server and your visitors.

Vulnerabilities

2.1. https://file.larksuite.com/

CONFIRMED

List of Supported Weak Ciphers

- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xC028)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xC027)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xC014)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)

Request

```
[NETSPARKER] SSL Connection
```

Response

Response Time (ms) : 1 Total Bytes Received : 27 Body Length : 0 Is Compressed : No

```
[NETSPARKER] SSL Connection
```

Actions to Take

1. For Apache, you should modify the SSLCipherSuite directive in the `httpd.conf`.

```
SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4
```

2. Lighttpd:

```
ssl.honor-cipher-order = "enable"
ssl.cipher-list = "EECDH+AESGCM:EDH+AESGCM"
```

3. For Microsoft IIS, you should make some changes to the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**

   a.Click Start, click Run, type `regedt32`or type `regedit`, and then click OK.
   b.In Registry Editor, locate the following registry key: `HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders`
   c.Set "Enabled" DWORD to "0x0" for the following registry keys:

```
SCHANNEL\Ciphers\DES 56/56
SCHANNEL\Ciphers\RC4 64/128
SCHANNEL\Ciphers\RC4 40/128
SCHANNEL\Ciphers\RC2 56/128
SCHANNEL\Ciphers\RC2 40/128
SCHANNEL\Ciphers\NULL
SCHANNEL\Hashes\MD5
```

Remedy

Configure your web server to disallow using weak ciphers.

CLASSIFICATION

PCI DSS v3.2	6.5.4
OWASP 2013	A6
OWASP 2017	A3
SANS Top 25	327
CAPEC	217
WASC	4
ISO27001	A.14.1.3

CVSS 3.0 SCORE

Base	6.8 (Medium)
Temporal	6.8 (Medium)
Environmental	6.8 (Medium)

CVSS Vector String

CVSS:3.0/AV:A/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N

CVSS 3.1 SCORE

Base	6.8 (Medium)
Temporal	6.8 (Medium)
Environmental	6.8 (Medium)

CVSS Vector String

CVSS:3.1/AV:A/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N

3. Missing X-Frame-Options Header

Issue Type: Law

Netsparker detected a missing `X-Frame-Options`header which means that this website could be at risk of a clickjacking attack.

The `X-Frame-Options`HTTP header field indicates a policy that specifies whether the browser should render the transmitted resource within a `frame`or an `iframe`. Servers can declare this policy in the header of their HTTP responses to prevent clickjacking attacks, which ensures that their content is not embedded into other pages or frames.

Impact

Clickjacking is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on a framed page when they were intending to click on the top level page. Thus, the attacker is "hijacking" clicks meant for their page and routing them to other another page, most likely owned by another application, domain, or both.

Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker.

Vulnerabilities

3.1. https://file.larksuite.com/%3Cscript%3Ealert(0)%3C/

Certainty

Request

```
GET /%3Cscript%3Ealert(0)%3C/ HTTP/1.1
Host: file.larksuite.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 10.4424 Total Bytes Received : 636 Body Length : 316 Is Compressed : No

```
HTTP/1.1 403 Forbidden
Server-Timing: cdn-cache; desc=MISS, edge; dur=0, origin; dur=0
Server: AkamaiGHost
Content-Length: 316
x-tt-trace-tag: id=16;cdn-cache=miss;type=dyn
Expires: Sat, 04 Jun 2022 21:38:03 GMT
Connection: close
Mime-Version: 1.0
Content-Type: text/html
Date: Sat, 04 Jun 2022 21:38:03 GMT

<HTML><HEAD>
<TITLE>Access Denied</TITLE>
</HEAD><BODY>
<H1>Access Denied</H1>

You don't have permission to access "http&#58;&#47;&#47;file&#46;larksuite&#46;com&#47;&#37;3Cscript&#3
7;3Ealert&#40;0&#41;&#37;3C&#47;" on this server.<P>
Reference&#32;&#35;18&#46;c6a8a5de&#46;1654378683&#46;24700fdf
</BODY>
</HTML>
```

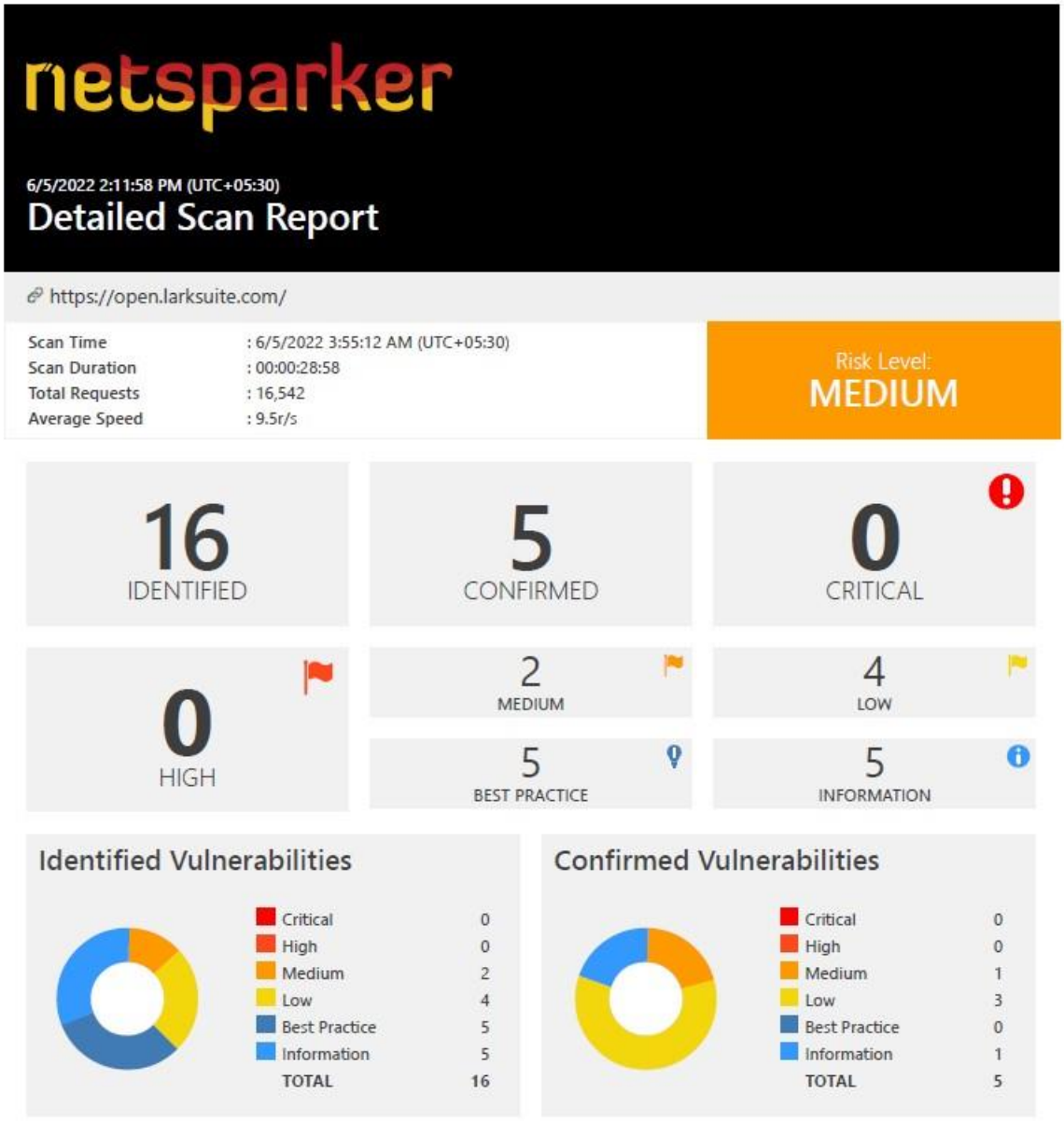
Remedy

- Sending the proper X-Frame-Options in HTTP response headers that instruct the browser to not allow framing from other domains.
    - `X-Frame-Options: DENY`It completely denies to be loaded in frame/iframe.
    - `X-Frame-Options: SAMEORIGIN`It allows only if the site which wants to load has a same origin.
    - `X-Frame-Options: ALLOW-FROM` *`URL`*It grants a specific URL to load itself in a iframe. However please pay attention to that, not all browsers support this.
- Employing defensive code in the UI to ensure that the current frame is the most top level window.

CLASSIFICATION

OWASP 2013	A5
OWASP 2017	A6
SANS Top 25	693
CAPEC	103
ISO27001	A.14.2.5

Vulnerable subdomain: open.larksuite.com



Vulnerability Summary

CONFIRM	VULNERABILITY	METHOD	URL	PARAMETER
	HTTP Strict Transport Security (HSTS) Policy Not Enabled	GET	https://open.larksuite.com/	
	Weak Ciphers Enabled	GET	https://open.larksuite.com/	
	Misconfigured Access-Control-Allow-Origin Header	GET	https://open.larksuite.com/api/tools/batch-settings	URI-BASED
	Cookie Not Marked as HttpOnly	GET	https://open.larksuite.com/	
	Cookie Not Marked as Secure	GET	https://open.larksuite.com/	
	Internal Server Error	POST	https://open.larksuite.com/napi/solutions	
	Content Security Policy (CSP) Not Implemented	GET	https://open.larksuite.com/	
	Expect-CT Not Enabled	GET	https://open.larksuite.com/	
	Missing X-XSS-Protection Header	GET	https://open.larksuite.com/api/tools/batch-settings	
	SameSite Cookie Not Implemented	GET	https://open.larksuite.com/	
	Subresource Integrity (SRI) Not Implemented	GET	https://open.larksuite.com/	
	[Possible] Internal Path Disclosure (*nix)	POST	https://open.larksuite.com/api/tools/document/search	
	Email Address Disclosure	POST	https://open.larksuite.com/api/tools/document/search	
	Nginx Web Server Identified	GET	https://open.larksuite.com/	
	Web Application Firewall Detected	GET	https://open.larksuite.com/%3Cscript%3Ealert(0)%3C/script%3E	URI-BASED

1. HTTP Strict Transport Security (HSTS) Policy Not Enabled

Issue Type: Medium

Netsparker identified that HTTP Strict Transport Security (HSTS) policy is not enabled.

The target website is being served from not only HTTPS but also HTTP and it lacks of HSTS policy implementation.

HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure (HTTPS) connections. The HSTS Policy is communicated by the server to the user agent via a HTTP response header field named "Strict-Transport-Security". HSTS Policy specifies a period of time during which the user agent shall access the server in only secure fashion.

When a web application issues HSTS Policy to user agents, conformant user agents behave as follows:

- Automatically turn any insecure (HTTP) links referencing the web application into secure (HTTPS) links. (For instance, http://example.com/some/page/ will be modified to https://example.com/some/page/ before accessing the server.)
- If the security of the connection cannot be ensured (e.g. the server's TLS certificate is self-signed), user agents show an error message and do not allow the user to access the web application.

Vulnerabilities

1.1. https://open.larksuite.com/

Certainty

Request

```
GET / HTTP/1.1
Host: open.larksuite.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 217.7616 Total Bytes Received : 5055 Body Length : 4261 Is Compressed : No

```
HTTP/1.1 200 OK
Server: nginx
x-tt-trace-host: 0120af35ddbbf5e8c5a6162b4bf481bc96387157c446f5b9f0cec5f5bfcc57713d1de361f56cdc50980dab
923ad68daf9d5fa92634c169254d4e31cf20a514f4b303d32f29c53a71eafad4cef28c2b03184446bf76f9888a928822ada125f
24b36
n-version: 1.1.1.353
x-tt-logid: 20220604222617010100008195OBD59F51
X-XSS-Protection: 1; mode=block
Connection: keep-alive
X-Download-Options: noopen
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Vary: Accept-Encoding
Content-Length: 1653
Server-Timing: inner; dur=9
Server-Timing: cdn-cache; desc=MISS, edge; dur=2, origin; dur=206
X-Origin-Response-Time: 207,222.165.168.198
Content-Type: text/html; charset=utf-8
x-tt-trace-tag: id=16;cdn-cache=miss;type=dyn
Date: Sat, 04 Jun 2022 22:26:17 GMT
Content-Encoding:

<!doctype html><html lang="en-US"><head><meta charset="UTF-8"/><meta name="viewport" content="width=dev
ice-width,initial-scale=1"/><script crossorigin="anonymous" defer="defer" src="https://sf16-va.larksuit
ecdn.com/goofy/log-sdk/collect/collect.js"></script><title>Lark Developer</title><meta name="descriptio
n" content="With its powerful open capabilities, Lark Developer enables enterprises to leverage apps an
d services to collaborate and grow."><meta name="keywords" content="With its powerful open capabilitie
s, Lark Developer enables enterprises to leverage apps and services to collaborate and grow."><meta nam
e="version" content="1.0.0.281"/><meta name="branch" content="website.1.0.0.281"/><link rel="icon" href
="//sf16-scmcdn2-va.larksuitecdn.com/lark/open/website/favicon-logo.svg"/><link rel="icon" href="//sf16
-scmcdn2-va.larksuitecdn.com/lark/open/website/favicon.png"/><script src="https://lf1-cdn-tos.bytegoof
y.com/goofy/locl/lark/external_js_sdk/h5-js-sdk-1.1.2.js"></script><script defer="defer" src="//sf16-sc
mcdn2-va.larksuitecdn.com/lark/open/website/js/runtime~app-e9304378.js" crossorigin="anonymous"></scrip
t><script defer="defer" src="//sf16-scmcdn2-va.larksuitecdn.com/lark/open/w
…
```

Remedy

Configure your webserver to redirect HTTP requests to HTTPS.

i.e. for Apache, you should have modification in the httpd.conf. For more configurations, please refer to External References section.

```
# load module
LoadModule headers module modules/mod headers.so
```

```
# redirect all HTTP to HTTPS (optional)
<VirtualHost *:80>
       ServerAlias *
       RewriteEngine On
       RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [redirect=301]
</VirtualHost>

# HTTPS-Host-Configuration
<VirtualHost *:443>
      # Use HTTP Strict Transport Security to force client to use secure connections only
      Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"

      # Further Configuration goes here
      [...]
</VirtualHost>
```

CLASSIFICATION

OWASP 2013	A6
OWASP 2017	A3
SANS Top 25	523
CAPEC	217
WASC	4
ISO27001	A.14.1.2

2. Weak Ciphers Enabled

Issue Type: Medium

Netsparker detected that weak ciphers are enabled during secure communication (SSL).

You should allow only strong ciphers on your web server to protect secure communication with your visitors.

Impact

Attackers might decrypt SSL traffic between your server and your visitors.

Vulnerabilities

2.1. https://open.larksuite.com/

CONFIRMED

List of Supported Weak Ciphers

- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xC028)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xC027)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xC014)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)

Request

```
[NETSPARKER] SSL Connection
```

Response

Response Time (ms) : 1 Total Bytes Received : 27 Body Length : 0 Is Compressed : No

```
[NETSPARKER] SSL Connection
```

Actions to Take

1. For Apache, you should modify the SSLCipherSuite directive in the `httpd.conf`.

```
SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4
```

2. Lighttpd:

```
ssl.honor-cipher-order = "enable"
ssl.cipher-list = "EECDH+AESGCM:EDH+AESGCM"
```

3. For Microsoft IIS, you should make some changes to the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**

   a.Click Start, click Run, type `regedt32`or type `regedit`, and then click OK.
   b.In Registry Editor, locate the following registry key: `HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders`
   c.Set "Enabled" DWORD to "0x0" for the following registry keys:

```
SCHANNEL\Ciphers\DES 56/56
SCHANNEL\Ciphers\RC4 64/128
SCHANNEL\Ciphers\RC4 40/128
SCHANNEL\Ciphers\RC2 56/128
SCHANNEL\Ciphers\RC2 40/128
SCHANNEL\Ciphers\NULL
SCHANNEL\Hashes\MD5
```

Remedy

Configure your web server to disallow using weak ciphers.

CLASSIFICATION

PCI DSS v3.2	6.5.4
OWASP 2013	A6
OWASP 2017	A3
SANS Top 25	327
CAPEC	217
WASC	4
ISO27001	A.14.1.3

CVSS 3.0 SCORE

Base	6.8 (Medium)
Temporal	6.8 (Medium)
Environmental	6.8 (Medium)

CVSS Vector String

CVSS:3.0/AV:A/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N

CVSS 3.1 SCORE

Base	6.8 (Medium)
Temporal	6.8 (Medium)
Environmental	6.8 (Medium)

CVSS Vector String

CVSS:3.1/AV:A/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N

3. Cookie Not Marked as HttpOnly

Issue Type: Low

Netsparker identified a cookie not marked as HTTPOnly.

HTTPOnly cookies cannot be read by client-side scripts, therefore marking a cookie as HTTPOnly can provide an additional layer of protection against cross-site scripting attacks.

Impact

During a cross-site scripting attack, an attacker might easily access cookies and hijack the victim's session.

Vulnerabilities

3.1. https://open.larksuite.com/

CONFIRMED

Identified Cookie(s)

- _tea_utm_cache_1229
- open_locale
- __tea__ug__uid
- _tea_utm_cache_1664

Cookie Source

- JavaScript

Request

```
GET / HTTP/1.1
Host: open.larksuite.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 400.1857 Total Bytes Received : 5055 Body Length : 4261 Is Compressed : No

```
HTTP/1.1 200 OK
Server: nginx
x-tt-trace-host: 0120af35ddbbf5e8c5a6162b4bf481bc96387157c446f5b9f0cec5f5bfcc57713d1de361f56cdc50980dab
923ad68daf9d246265ec761d2664511c30144bc5d60dce07a8de39f0400d2de8caf2c722cd0e48371ad4be8f429a619593f1fc0
ec7d2
n-version: 1.1.1.353
x-tt-logid: 20220604222515010100008045ODD3DDA6
X-XSS-Protection: 1; mode=block
Connection: keep-alive
X-Download-Options: noopen
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Vary: Accept-Encoding
Content-Length: 1655
Server-Timing: inner; dur=9
Server-Timing: cdn-cache; desc=MISS, edge; dur=1, origin; dur=244
X-Origin-Response-Time: 244,222.165.168.198
Content-Type: text/html; charset=utf-8
x-tt-trace-tag: id=16;cdn-cache=miss;type=dyn
Date: Sat, 04 Jun 2022 22:25:15 GMT
Content-Encoding:

<!doctype html><html lang="en-US"><head><meta charset="UTF-8"/><meta name="viewport" content="width=dev
ice-width,initial-scale=1"/><script crossorigin="anonymous" defer="defer" src="https://sf16-va.larksuit
ecdn.com/goofy/log-sdk/collect/collect.js"></script><title>Lark Developer</title><meta name="descriptio
n" content="With its powerful open capabilities, Lark Developer enables enterprises to leverage apps an
d services to collaborate and grow."><meta name="keywords" content="With its powerful open capabilitie
s, Lark Developer enables enterprises to leverage apps and services to collaborate and grow."><meta nam
e="version" content="1.0.0.281"/><meta name="branch" content="website.1.0.0.281"/><link rel="icon" href
="//sf16-scmcdn2-va.larksuitecdn.com/lark/open/website/favicon-logo.svg"/><link rel="icon" href="//sf16
-scmcdn2-va.larksuitecdn.com/lark/open/website/favicon.png"/><script src="https://lf1-cdn-tos.bytegoof
y.com/goofy/locl/lark/external_js_sdk/h5-js-sdk-1.1.2.js"></script><script defer="defer" src="//sf16-sc
mcdn2-va.larksuitecdn.com/lark/open/website/js/runtime~app-e9304378.js" crossorigin="anonymous"></scrip
t><script defer="defer" src="//sf16-scmcdn2-va.larksuitecdn.com/lark/open/w
…
```

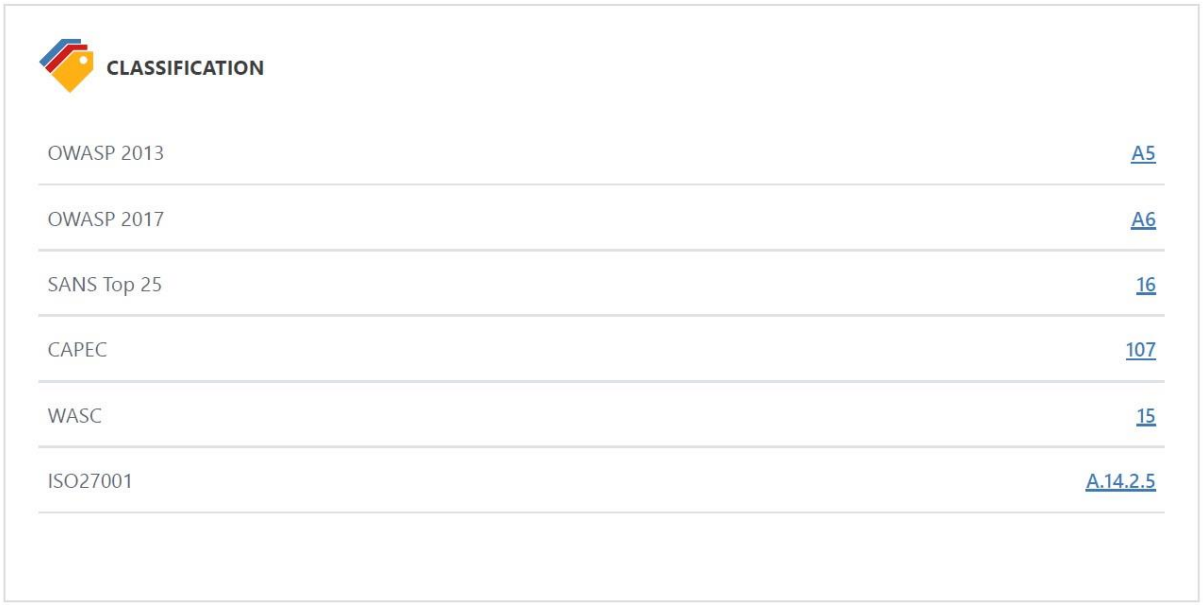
Actions to Take

1. See the remedy for solution.
2. Consider marking all of the cookies used by the application as HTTPOnly. (*After these changes javascript code will not be able to read cookies.*)

Remedy

Mark the cookie as HTTPOnly. This will be an extra layer of defense against XSS. However this is not a silver bullet and will not protect

the system against cross-site scripting attacks. An attacker can use a tool such as XSS Tunnelto bypass HTTPOnly protection.



4. Cookie Not Marked as Secure

Issue Type: Low

Netsparker identified a cookie not marked as secure, and transmitted over HTTPS.

This means the cookie could potentially be stolen by an attacker who can successfully intercept and decrypt the traffic, or following a successful man-in-the-middle attack.

Impact

This cookie will be transmitted over a HTTP connection, therefore if this cookie is important (*such as a session cookie*), an attacker might intercept it and hijack a victim's session. If the attacker can carry out a man-in-the-middle attack, he/she can force the victim to make an HTTP request to steal the cookie.

Vulnerabilities

4.1. https://open.larksuite.com/

CONFIRMED

Identified Cookie(s)

- _tea_utm_cache_1229
- open_locale
- __tea__ug__uid
- _tea_utm_cache_1664

Cookie Source

- JavaScript

Request

```
GET / HTTP/1.1
Host: open.larksuite.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 400.1857 Total Bytes Received : 5055 Body Length : 4261 Is Compressed : No

```
HTTP/1.1 200 OK
Server: nginx
x-tt-trace-host: 0120af35ddbbf5e8c5a6162b4bf481bc96387157c446f5b9f0cec5f5bfcc57713d1de361f56cdc50980dab
923ad68daf9d246265ec761d2664511c30144bc5d60dce07a8de39f0400d2de8caf2c722cd0e48371ad4be8f429a619593f1fc0
ec7d2
n-version: 1.1.1.353
x-tt-logid: 20220604222515010100008045ODD3DDA6
X-XSS-Protection: 1; mode=block
Connection: keep-alive
X-Download-Options: noopen
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Vary: Accept-Encoding
Content-Length: 1655
Server-Timing: inner; dur=9
Server-Timing: cdn-cache; desc=MISS, edge; dur=1, origin; dur=244
X-Origin-Response-Time: 244,222.165.168.198
Content-Type: text/html; charset=utf-8
x-tt-trace-tag: id=16;cdn-cache=miss;type=dyn
Date: Sat, 04 Jun 2022 22:25:15 GMT
Content-Encoding:

<!doctype html><html lang="en-US"><head><meta charset="UTF-8"/><meta name="viewport" content="width=dev
ice-width,initial-scale=1"/><script crossorigin="anonymous" defer="defer" src="https://sf16-va.larksuit
ecdn.com/goofy/log-sdk/collect/collect.js"></script><title>Lark Developer</title><meta name="descriptio
n" content="With its powerful open capabilities, Lark Developer enables enterprises to leverage apps an
d services to collaborate and grow."><meta name="keywords" content="With its powerful open capabilitie
s, Lark Developer enables enterprises to leverage apps and services to collaborate and grow."><meta nam
e="version" content="1.0.0.281"/><meta name="branch" content="website.1.0.0.281"/><link rel="icon" href
="//sf16-scmcdn2-va.larksuitecdn.com/lark/open/website/favicon-logo.svg"/><link rel="icon" href="//sf16
-scmcdn2-va.larksuitecdn.com/lark/open/website/favicon.png"/><script src="https://lf1-cdn-tos.bytegoof
y.com/goofy/locl/lark/external_js_sdk/h5-js-sdk-1.1.2.js"></script><script defer="defer" src="//sf16-sc
mcdn2-va.larksuitecdn.com/lark/open/website/js/runtime~app-e9304378.js" crossorigin="anonymous"></scrip
t><script defer="defer" src="//sf16-scmcdn2-va.larksuitecdn.com/lark/open/w
```

Actions to Take

1. See the remedy for solution.
2. Mark all cookies used within the application as secure. *(If the cookie is not related to authentication or does not carry any personal information, you do not have to mark it as secure.)*

Remedy

Mark all cookies used within the application as secure.

Required Skills for Successful Exploitation

To exploit this issue, the attacker needs to be able to intercept traffic. This generally requires local access to the web server or to the victim's network. Attackers need to be understand layer 2, have physical access to systems either as waypoints for the traffic, or have locally gained access to to a system between the victim and the web server.

CLASSIFICATION

PCI DSS v3.2	6.5.10
OWASP 2013	A6
OWASP 2017	A3
SANS Top 25	614
CAPEC	102
WASC	15
ISO27001	A.14.1.2

CVSS 3.0 SCORE

Base	2 (Low)
Temporal	2 (Low)
Environmental	2 (Low)

CVSS Vector String

CVSS:3.0/AV:P/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N

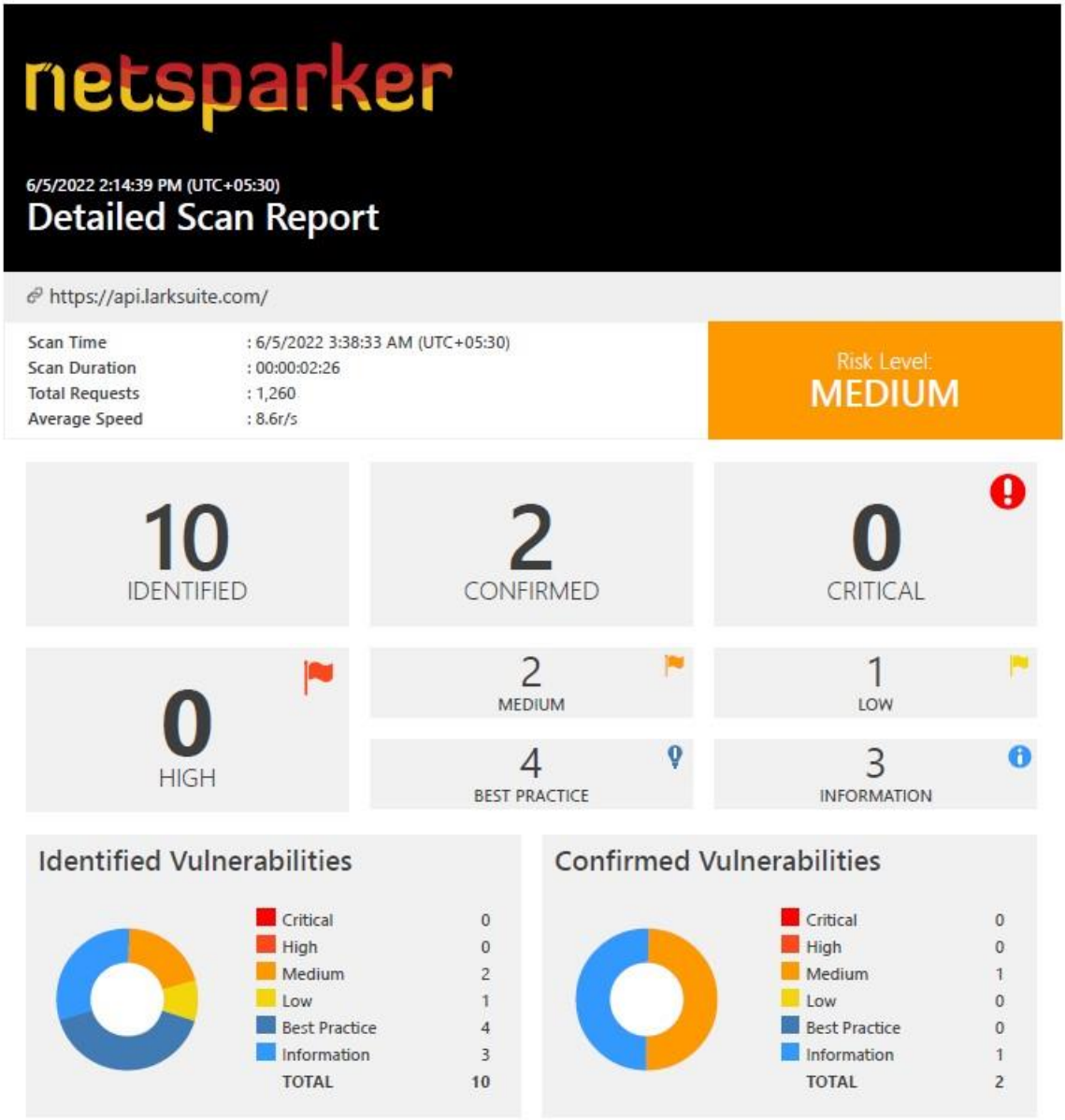
CVSS 3.1 SCORE

Base	2 (Low)
Temporal	2 (Low)
Environmental	2 (Low)

CVSS Vector String

CVSS:3.1/AV:P/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N

Vulnerable subdomain: api.larksuite.com



Vulnerability Summary

CONFIRM	VULNERABILITY	METHOD	URL	PARAMETER
	HTTP Strict Transport Security (HSTS) Policy Not Enabled	GET	https://api.larksuite.com/	
	Weak Ciphers Enabled	GET	https://api.larksuite.com/	
	Missing X-Frame-Options Header	GET	https://api.larksuite.com/	
	Content Security Policy (CSP) Not Implemented	GET	https://api.larksuite.com/	
	Expect-CT Not Enabled	GET	https://api.larksuite.com/	
	Missing X-XSS-Protection Header	GET	https://api.larksuite.com/	
	Referrer-Policy Not Implemented	GET	https://api.larksuite.com/	
	Nginx Web Server Identified	GET	https://api.larksuite.com/	
	Web Application Firewall Detected	GET	https://api.larksuite.com/%3Cscript%3Ealert(0)%3C/script%3E	URI-BASED
	Forbidden Resource	GET	https://api.larksuite.com/%3Cscript%3Ealert(0)%3C/	

1. HTTP Strict Transport Security (HSTS) Policy Not Enabled

Issue Type: Medium

Netsparker identified that HTTP Strict Transport Security (HSTS) policy is not enabled.

The target website is being served from not only HTTPS but also HTTP and it lacks of HSTS policy implementation.

HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure (HTTPS) connections. The HSTS Policy is communicated by the server to the user agent via a HTTP response header field named "Strict-Transport-Security". HSTS Policy specifies a period of time during which the user agent shall access the server in only secure fashion.

When a web application issues HSTS Policy to user agents, conformant user agents behave as follows:

- Automatically turn any insecure (HTTP) links referencing the web application into secure (HTTPS) links. (For instance, http://example.com/some/page/ will be modified to https://example.com/some/page/ before accessing the server.)
- If the security of the connection cannot be ensured (e.g. the server's TLS certificate is self-signed), user agents show an error message and do not allow the user to access the web application.

Vulnerabilities

1.1. https://api.larksuite.com/

Certainty

Request

```
GET / HTTP/1.1
Host: api.larksuite.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 298.9397 Total Bytes Received : 1230 Body Length : 564 Is Compressed : No

```
HTTP/1.1 404 Not Found
Server-Timing: cdn-cache; desc=MISS, edge; dur=0, origin; dur=234
x-tt-trace-host: 0120af35ddbbf5e8c5a6162b4bf481bc96cc7a56d11f4c9cf7ebb9268659a11a3509bbb18fa82352c5ab68
ec75b97bb255b478692af752094ed316982b6df48a3cdbb5c8d6197bb913d5c45e39146b90ae
Server: nginx
X-Akamai-Request-ID: 857667f
x-tt-trace-tag: id=16;cdn-cache=miss;type=dyn
X-Cache: TCP_MISS from a222-165-168-197.deploy.akamaitechnologies.com (AkamaiGHost/10.8.2-41841244) (-)
Content-Length: 180
Connection: keep-alive
Content-Type: text/html
Content-Encoding:
Date: Sat, 04 Jun 2022 22:08:48 GMT
Vary: Accept-Encoding
X-Origin-Response-Time: 234,222.165.168.197

<html>
<head><title>404 Not Found</title></head>
<body bgcolor="white">
<center><h1>404 Not Found</h1></center>
<hr><center>nginx</center>
</body>
</html>
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
```

Remedy

Configure your webserver to redirect HTTP requests to HTTPS.

i.e. for Apache, you should have modification in the httpd.conf. For more configurations, please refer to External References section.

```
# load module
LoadModule headers_module modules/mod_headers.so

# redirect all HTTP to HTTPS (optional)
<VirtualHost *:80>
       ServerAlias *
       RewriteEngine On
       RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [redirect=301]
```

```
</VirtualHost>

# HTTPS-Host-Configuration
<VirtualHost *:443>
      # Use HTTP Strict Transport Security to force client to use secure connections only
      Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"

      # Further Configuration goes here
      [...]
</VirtualHost>
```

2. Weak Ciphers Enabled

Issue Type: Medium

Netsparker detected that weak ciphers are enabled during secure communication (SSL).

You should allow only strong ciphers on your web server to protect secure communication with your visitors.

Impact

Attackers might decrypt SSL traffic between your server and your visitors.

Vulnerabilities

2.1. https://api.larksuite.com/

CONFIRMED

List of Supported Weak Ciphers

- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xC028)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xC027)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xC014)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)

Request

```
[NETSPARKER] SSL Connection
```

Response

Response Time (ms) : 1 Total Bytes Received : 27 Body Length : 0 Is Compressed : No

```
[NETSPARKER] SSL Connection
```

Actions to Take

1. For Apache, you should modify the SSLCipherSuite directive in the `httpd.conf`.

```
SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4
```

2. Lighttpd:

```
ssl.honor-cipher-order = "enable"
ssl.cipher-list = "EECDH+AESGCM:EDH+AESGCM"
```

3. For Microsoft IIS, you should make some changes to the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**

   a.Click Start, click Run, type `regedt32`or type `regedit`, and then click OK.
   b.In Registry Editor, locate the following registry key: `HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders`
   c.Set "Enabled" DWORD to "0x0" for the following registry keys:

```
SCHANNEL\Ciphers\DES 56/56
SCHANNEL\Ciphers\RC4 64/128
SCHANNEL\Ciphers\RC4 40/128
SCHANNEL\Ciphers\RC2 56/128
SCHANNEL\Ciphers\RC2 40/128
SCHANNEL\Ciphers\NULL
SCHANNEL\Hashes\MD5
```

Remedy

Configure your web server to disallow using weak ciphers.

CLASSIFICATION

PCI DSS v3.2	6.5.4
OWASP 2013	A6
OWASP 2017	A3
SANS Top 25	327
CAPEC	217
WASC	4
ISO27001	A.14.1.3

CVSS 3.0 SCORE

Base	6.8 (Medium)
Temporal	6.8 (Medium)
Environmental	6.8 (Medium)

CVSS Vector String

CVSS:3.0/AV:A/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N

CVSS 3.1 SCORE

Base	6.8 (Medium)
Temporal	6.8 (Medium)
Environmental	6.8 (Medium)

CVSS Vector String

CVSS:3.1/AV:A/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N

3. Missing X-Frame-Options Header

Issue Type: Low

Netsparker detected a missing `X-Frame-Options`header which means that this website could be at risk of a clickjacking attack.

The `X-Frame-Options`HTTP header field indicates a policy that specifies whether the browser should render the transmitted resource within a `frame`or an `iframe`. Servers can declare this policy in the header of their HTTP responses to prevent clickjacking attacks, which ensures that their content is not embedded into other pages or frames.

Impact

Clickjacking is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on a framed page when they were intending to click on the top level page. Thus, the attacker is "hijacking" clicks meant for their page and routing them to other another page, most likely owned by another application, domain, or both.

Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker.

Vulnerabilities

3.1. https://api.larksuite.com/

Certainty

Request

```
GET / HTTP/1.1
Host: api.larksuite.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 407.1239 Total Bytes Received : 1230 Body Length : 564 Is Compressed : No

```
HTTP/1.1 404 Not Found
Server-Timing: cdn-cache; desc=MISS, edge; dur=0, origin; dur=234
x-tt-trace-host: 0120af35ddbbf5e8c5a6162b4bf481bc96cc7a56d11f4c9cf7ebb9268659a11a3509bbb18fa82352c5ab68
ec75b97bb255b478692af752094ed316982b6df48a3cdbb5c8d6197bb913d5c45e39146b90ae
Server: nginx
X-Akamai-Request-ID: 8574dee
x-tt-trace-tag: id=16;cdn-cache=miss;type=dyn
X-Cache: TCP_MISS from a222-165-168-197.deploy.akamaitechnologies.com (AkamaiGHost/10.8.2-41841244) (-)
Content-Length: 180
Connection: keep-alive
Content-Type: text/html
Content-Encoding:
Date: Sat, 04 Jun 2022 22:08:37 GMT
Vary: Accept-Encoding
X-Origin-Response-Time: 234,222.165.168.197

<html>
<head><title>404 Not Found</title></head>
<body bgcolor="white">
<center><h1>404 Not Found</h1></center>
<hr><center>nginx</center>
</body>
</html>
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
```

Remedy

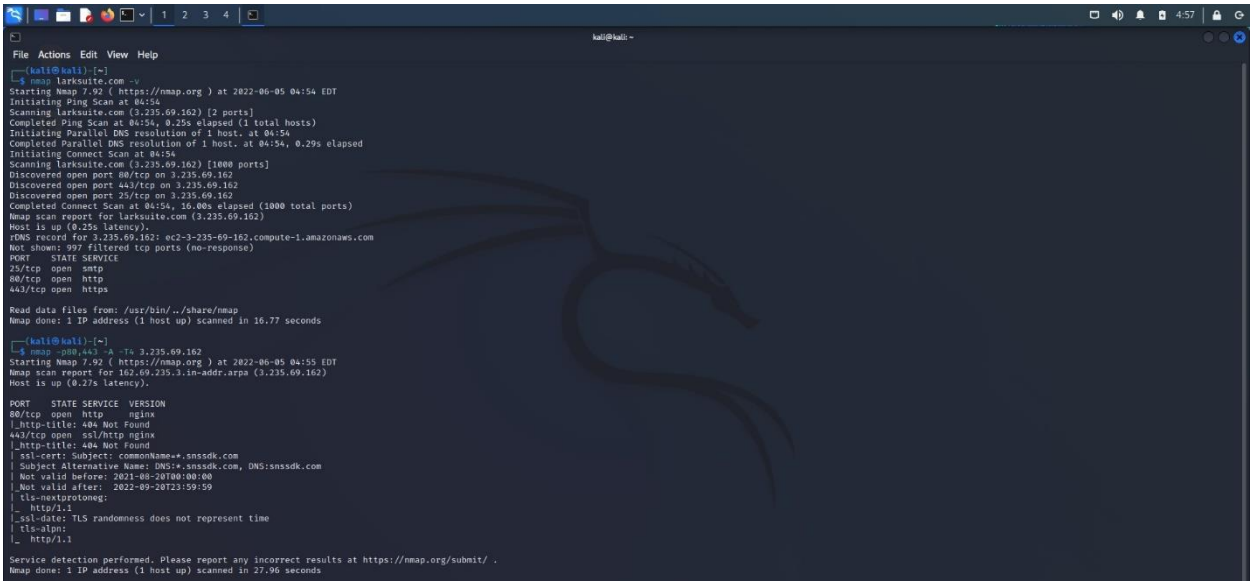
- Sending the proper X-Frame-Options in HTTP response headers that instruct the browser to not allow framing from other domains.
    - `X-Frame-Options: DENY` It completely denies to be loaded in frame/iframe.
    - `X-Frame-Options: SAMEORIGIN` It allows only if the site which wants to load has a same origin.
    - `X-Frame-Options: ALLOW-FROM` *URL* It grants a specific URL to load itself in a iframe. However please pay attention to that, not all browsers support this.
- Employing defensive code in the UI to ensure that the current frame is the most top level window.

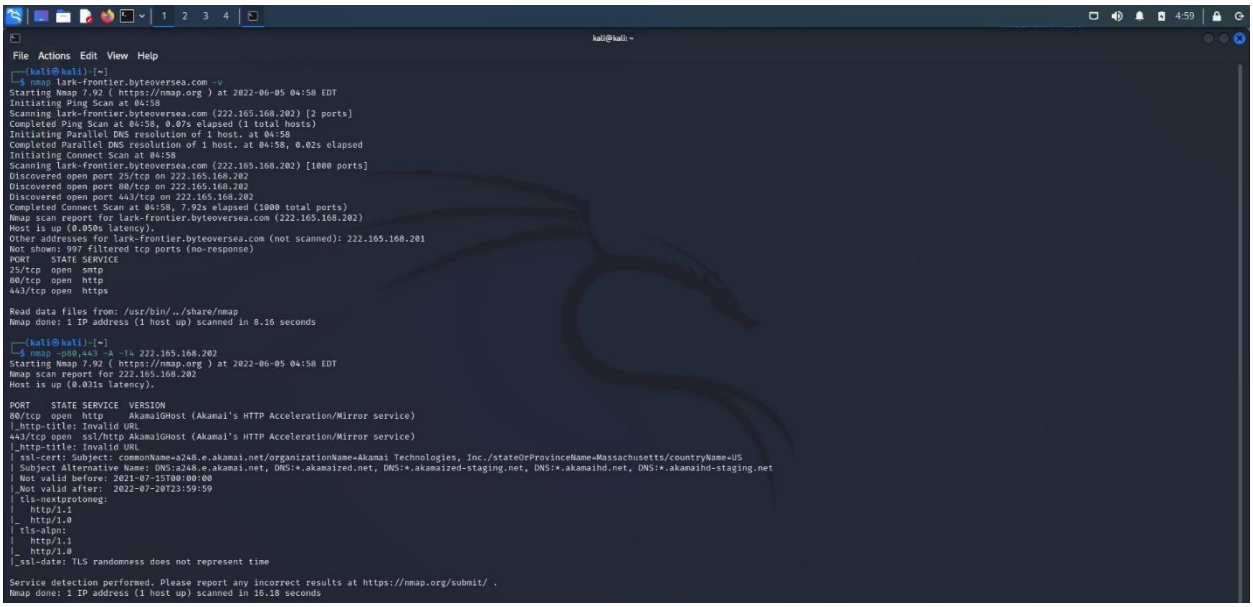
CLASSIFICATION

OWASP 2013	A5
OWASP 2017	A6
SANS Top 25	693
CAPEC	103
ISO27001	A.14.2.5

Nmap

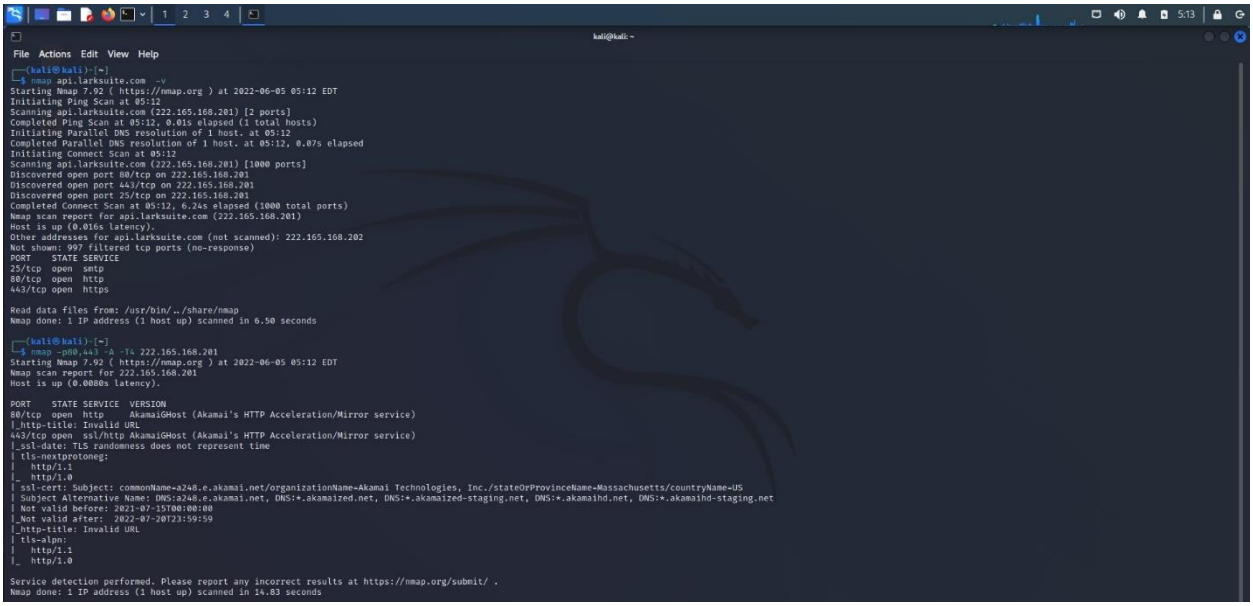
Nmap Wikipedia is a free online encyclopedia. Gordon Lyon designed Nmap (Network Mapper), a network scanner (also known by his pseudonym Fyodor Vaskovich). Nmap is a program that sends packets and analyzes the answers to find hosts and services on a computer network.






kali@kali: ~
File Actions Edit View Help
┌──(kali㉿kali)-[~]
└─$ nmap file.larksuite.com -v
Starting Nmap 7.92 (https://nmap.org) at 2022-06-05 05:10 EDT
Initiating Ping Scan at 05:10
Scanning file.larksuite.com (222.165.168.201) [2 ports]
Completed Ping Scan at 05:10, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 05:10
Completed Parallel DNS resolution of 1 host. at 05:10, 4.03s elapsed
Initiating Connect Scan at 05:10
Scanning file.larksuite.com (222.165.168.201) [1000 ports]
Discovered open port 443/tcp on 222.165.168.201
Discovered open port 25/tcp on 222.165.168.201
Discovered open port 80/tcp on 222.165.168.201
Completed Connect Scan at 05:10, 4.03s elapsed (1000 total ports)
Nmap scan report for file.larksuite.com (222.165.168.201)
Host is up (0.031s latency).
Other addresses for file.larksuite.com (not scanned): 222.165.168.202
Not shown: 997 filtered tcp ports (no-response)
PORT STATE SERVICE
25/tcp open smtp
80/tcp open http
443/tcp open https
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 8.50 seconds
┌──(kali㉿kali)-[~]
└─$ nmap -p80,443 -A -T4 222.165.168.201
Starting Nmap 7.92 (https://nmap.org) at 2022-06-05 05:10 EDT
Nmap scan report for 222.165.168.201
Host is up (0.026s latency).
PORT STATE SERVICE VERSION
80/tcp open http AkamaiGHost (Akamai's HTTP Acceleration/Mirror service)
|_http-title: Invalid URL
443/tcp open ssl/http AkamaiGHost (Akamai's HTTP Acceleration/Mirror service)
|_ssl-date: TLS randomness does not represent time
|_http-title: Invalid URL
| tls-alpn:
| http/1.1
|_ http/1.0
| tls-nextprotoneg:
| http/1.1
|_ http/1.0
| ssl-cert: Subject: commonName=a248.e.akamai.net/organizationName=Akamai Technologies, Inc./stateOrProvinceName=Massachusetts/countryName=US
| Subject Alternative Name: DNS:a248.e.akamai.net, DNS:*.akamaized.net, DNS:*.akamaized-staging.net, DNS:*.akamaihd.net, DNS:*.akamaihd-staging.net
| Not valid before: 2021-07-15T00:00:00
|_Not valid after: 2022-07-20T23:59:59
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.83 seconds


kali@kali: ~
File Actions Edit View Help
┌──(kali㉿kali)-[~]
└─$ nmap open.larksuite.com -v
Starting Nmap 7.92 (https://nmap.org) at 2022-06-05 05:07 EDT
Initiating Ping Scan at 05:07
Scanning open.larksuite.com (104.75.84.11) [2 ports]
Completed Ping Scan at 05:07, 2.01s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 05:07
Completed Parallel DNS resolution of 1 host. at 05:07, 0.01s elapsed
Initiating Connect Scan at 05:07
Scanning open.larksuite.com (104.75.84.11) [1000 ports]
Discovered open port 25/tcp on 104.75.84.11
Completed Connect Scan at 05:07, 10.52s elapsed (1000 total ports)
Nmap scan report for open.larksuite.com (104.75.84.11)
Host is up (0.013s latency).
Other addresses for open.larksuite.com (not scanned): 104.75.84.8
rDNS record for 104.75.84.11: a104-75-84-11.deploy.static.akamaitechnologies.com
Not shown: 999 filtered tcp ports (no-response)
PORT STATE SERVICE
25/tcp open smtp
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 12.95 seconds
┌──(kali㉿kali)-[~]
└─$ nmap -p80,443 -A -T4 104.75.84.11
Starting Nmap 7.92 (https://nmap.org) at 2022-06-05 05:08 EDT
Nmap scan report for 11.84.75.104.in-addr.arpa (104.75.84.11)
Host is up (0.030s latency).
PORT STATE SERVICE VERSION
80/tcp open http AkamaiGHost (Akamai's HTTP Acceleration/Mirror service)
|_http-title: Invalid URL
443/tcp open ssl/http AkamaiGHost (Akamai's HTTP Acceleration/Mirror service)
|_http-title: Invalid URL
| ssl-cert: Subject: commonName=a248.e.akamai.net/organizationName=Akamai Technologies, Inc./stateOrProvinceName=Massachusetts/countryName=US
| Subject Alternative Name: DNS:a248.e.akamai.net, DNS:*.akamaized.net, DNS:*.akamaized-staging.net, DNS:*.akamaihd.net, DNS:*.akamaihd-staging.net
| Not valid before: 2021-07-15T00:00:00
|_Not valid after: 2022-07-20T23:59:59
| tls-nextprotoneg:
| http/1.1
|_ http/1.0
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
| http/1.1
|_ http/1.0
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.72 seconds


kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ nmap api.larksuite.com -v
Starting Nmap 7.92 (https://nmap.org) at 2022-06-05 05:12 EDT
Initiating Ping Scan at 05:12
Scanning api.larksuite.com (222.165.168.201) [2 ports]
Completed Ping Scan at 05:12, 0.01s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 05:12
Completed Parallel DNS resolution of 1 host. at 05:12, 0.07s elapsed
Initiating Connect Scan at 05:12
Scanning api.larksuite.com (222.165.168.201) [1000 ports]
Discovered open port 80/tcp on 222.165.168.201
Discovered open port 443/tcp on 222.165.168.201
Discovered open port 25/tcp on 222.165.168.201
Completed Connect Scan at 05:12, 6.24s elapsed (1000 total ports)
Nmap scan report for api.larksuite.com (222.165.168.201)
Host is up (0.016s latency).
Other addresses for api.larksuite.com (not scanned): 222.165.168.202
Not shown: 997 filtered tcp ports (no-response)
PORT STATE SERVICE
25/tcp open smtp
80/tcp open http
443/tcp open https
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 6.50 seconds
(kali㉿kali)-[~]
$ nmap -p80,443 -A -T4 222.165.168.201
Starting Nmap 7.92 (https://nmap.org) at 2022-06-05 05:12 EDT
Nmap scan report for 222.165.168.201
Host is up (0.0088s latency).
PORT STATE SERVICE VERSION
80/tcp open http AkamaiGHost (Akamai's HTTP Acceleration/Mirror service)
|_http-title: Invalid URL
443/tcp open ssl/http AkamaiGHost (Akamai's HTTP Acceleration/Mirror service)
|_ssl-date: TLS randomness does not represent time
| tls-nextprotoneg:
| http/1.1
|_ http/1.0
| ssl-cert: Subject: commonName=a248.e.akamai.net/organizationName=Akamai Technologies, Inc./stateOrProvinceName=Massachusetts/countryName=US
| Subject Alternative Name: DNS:a248.e.akamai.net, DNS:*.akamaized.net, DNS:*.akamaized-staging.net, DNS:*.akamaihd.net, DNS:*.akamaihd-staging.net
| Not valid before: 2021-07-15T00:00:00
|_Not valid after: 2022-07-20T23:59:59
|_http-title: Invalid URL
| tls-alpn:
| http/1.1
|_ http/1.0
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.83 seconds

Conclusion

With the exception of a few loose ends, the basic protection of the application was not effectively designed and implemented at the completion of the evaluation. Overall, the web application's dependability and trustworthiness are well-structured thanks to the use of security techniques and protocols.