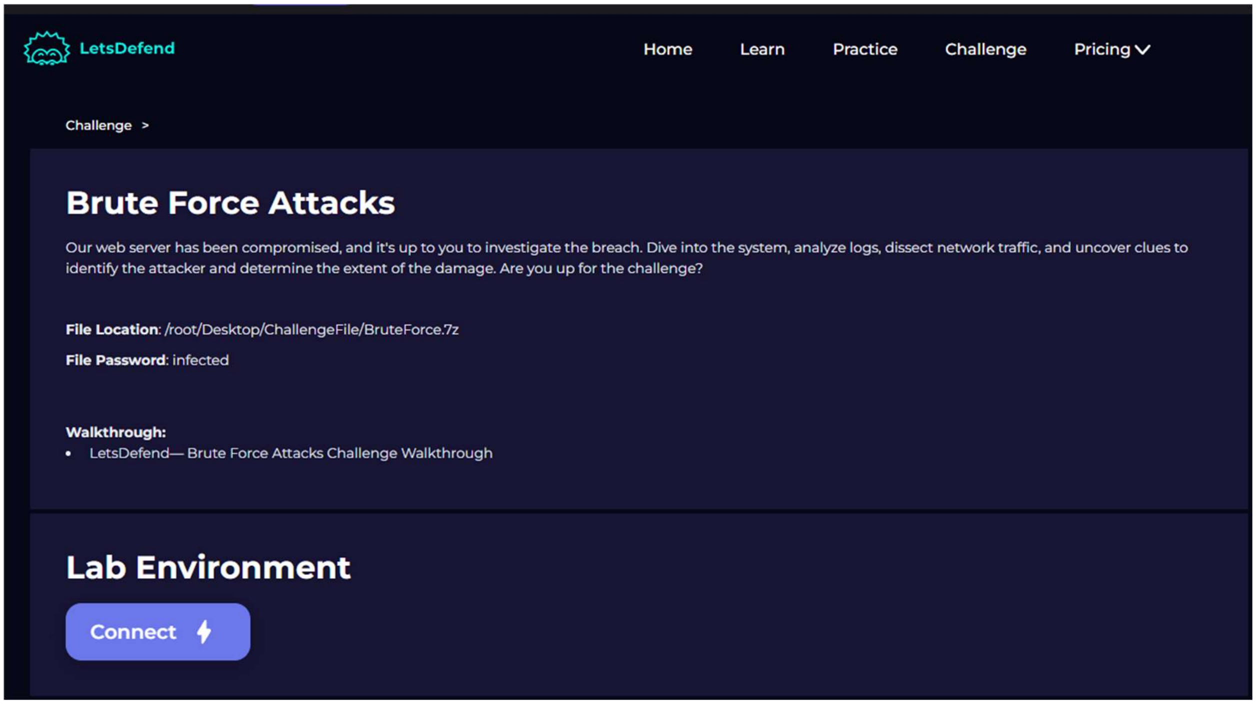
Track	Cyber Security Incident Response Analyst
Group code	DEPI_1_MNF1_ISS7_M1e
Team name	Blue_Pr0S
Members	1- Bahaa Boghdady Kamel 2- Eslam El-sayed Abdelhadi 3- Mohsen Sabry Abdelmohsen 4- Youssef Nabil Youssef
Mentor / Instructor	Eng. Marina Hany

Team_1 (Blue_Pr0S)

Incident response on Brute Force Attack

✵Introduction✵



Brute Force Attacks

Our web server has been compromised, and it's up to you to investigate the breach. Dive into the system, analyze logs, dissect network traffic, and uncover clues to identify the attacker and determine the extent of the damage. Are you up for the challenge?

File Location: /root/Desktop/ChallengeFile/BruteForce.7z

File Password: infected

Walkthrough:

- LetsDefend— Brute Force Attacks Challenge Walkthrough

Lab Environment

- Our web server has been compromised, and it's up to you to investigate the breach. Dive into the system, analyze logs, dissect network traffic, and uncover clues to identify the attacker and determine the extent of the damage.

- Now, let's delve into the intricate details of the attack technique, uncovering how it was detected, and the strategic steps taken to address and neutralize it effectively. By understanding both the discovery process and the response measures, we'll gain valuable insights into the approach used to thwart the threat.

✵Report✵

Executive Incident Summary

- ✓ This section provides an overview of the incident, including identifying a brute force attack targeting our web server and RDP access. Multiple login attempts were observed from the internal IP (192.168190.137) on our web server IP (51.116.96.181), which successfully compromised login credentials for the web interface.

- ✓ We investigated network traffic with Wireshark to determine the Threat Actor IP address and detect brute-force attacks to the web server and RDP.

- ✓ Lastly, we also investigated the authentication log to determine the threat actor's last successful logged-in attempts along with how many failed logins attempts it took before gaining a foothold on the webserver.

- ➢ A brute force attack is a method used by attackers to gain unauthorized access to a system by systematically trying all possible combinations of usernames and passwords until the correct one is found. This type of attack relies on the idea that given enough time and resources, the attacker will eventually guess the correct credentials. Brute force attacks can be carried out against various systems, including login forms, encrypted files, and network protocols like RDP or SSH.

- ➢ RDP (Remote Desktop Protocol) is a proprietary protocol developed by Microsoft that allows users to remotely connect to and control another computer over a network. It’s commonly used for accessing servers or computers in different locations, enabling administrators or users to work on a remote machine as if they were physically present at it.

- Timeline of Events

  - ✓ Feb 24 11:09:54: First signs of suspicious RDP traffic were observed.
  - ✓ Feb 24 11:11:12: Malicious web traffic targeting the login page of the server was detected.
  - ✓ Feb 24 11:14:00: RDP brute force attempts escalated.
  - ✓ Feb 24 11:20:30: Successful login was detected with the credentials “web-hacker”.
  - ✓ Feb 24 11:22:00: Incident response team engaged, and containment measures began.

- Impact

  - ✓ The brute force attack, while unsuccessful in gaining access, had a significant impact on the organization.
  - ✓ There was a temporary disruption of services due to the heightened security measures and the need for a system-wide password reset.
  - ✓ Although no data was compromised, the incident highlighted the vulnerability of weak password policies and the necessity of implementing 2FA.
  - ✓ The organization faced potential reputational damage had the attack been successful.

- Root Cause Analysis
    - ✓ The root cause of the incident was the use of weak or default credentials on the targeted systems, which allowed a brute-force attack to succeed.
    - ✓ This vulnerability stemmed from the absence of strong password policies and the lack of multi-factor authentication (MFA) on critical systems.
    - ✓ The reliance on simple, easily guessable passwords left the systems highly susceptible to attack.

- Containment and Eradication
    - ✓ Implemented IP blocking for the attacker's IP address (192.168190.137) and disabled compromised user accounts, followed by resetting their passwords.
    - ✓ Firewall rules were updated to restrict suspicious login attempts, and server-side authentication mechanisms were strengthened.
    - ✓ To contain the threat, rate-limiting was applied on the SSH service to mitigate brute-force attempts.
    - ✓ In the eradication phase, a comprehensive audit of system logs ensured no unauthorized access occurred.
    - ✓ Stronger security controls, including enforced password complexity and multi-factor authentication (MFA), were applied to further secure the system.

➢ Recovery

1- Account recovery

- ✓ Restored all compromised accounts and ensured system integrity, followed by the re-deployment of secure backups for the web server. To prevent future incidents, additional penetration testing was initiated to ensure no further vulnerabilities existed.

2- System recovery

- ✓ Recovery efforts included restoring normal operations by securing all systems, implementing enhanced monitoring, and rolling out mandatory password resets across the organization. The incident response team worked closely with the IT department to update security protocols and ensure comprehensive protection moving forward.

• Lessons Learned

1. Password Policies: Strong, complex password policies are essential in defending against brute force attacks. Regularly updating these policies and educating users on best practices is crucial.

2. Multi-Factor Authentication (MFA): Implementing MFA across all critical systems significantly reduces the risk of unauthorized access, even if passwords are compromised.

3. Monitoring and Detection: Continuous monitoring and proactive threat detection are vital in identifying and mitigating attacks early.

4. Incident Response Plan: The incident underscored the importance of having a well-defined incident response plan that can be executed efficiently to minimize damage.

5. Security Awareness Training: Regular training for all employees on recognizing and responding to security threats is necessary to enhance the overall security posture of the organization.

✵Documenting the Incident Response Process✵

➢ Description of incident

- ➢ Our web server has been compromised, and it's up to you to investigate the breach. Dive into the system, analyze logs, dissect network traffic, and uncover clues to identify the attacker and determine the extent of the damage.

✵Files machine

1- BruteForce.pcap
2- Auth.log

✵Tools & Programs

1- Wireshark :

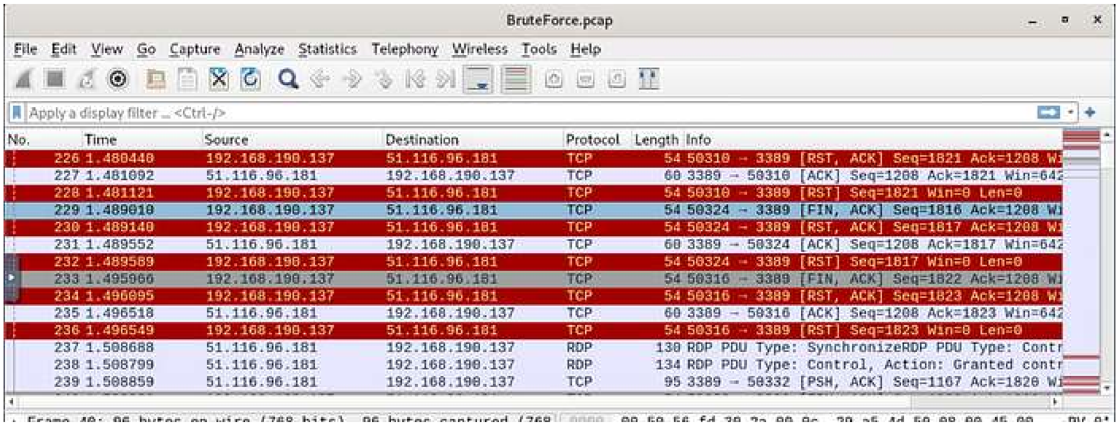
- ➢ A popular open-source network protocol analyzer used for capturing and analyzing data traffic moving across a network in real time. It allows users to inspect the details of network protocols, packets, and other information sent over a network.

2- grep :

- ➢ grep is a powerful command-line utility in Linux and Unix-based systems used for searching text within files. It stands for Global Regular Expression Print and is commonly used to search for specific patterns (strings or regular expressions) in files or output streams.

- ## Initial Detection and Analysis of the Incident

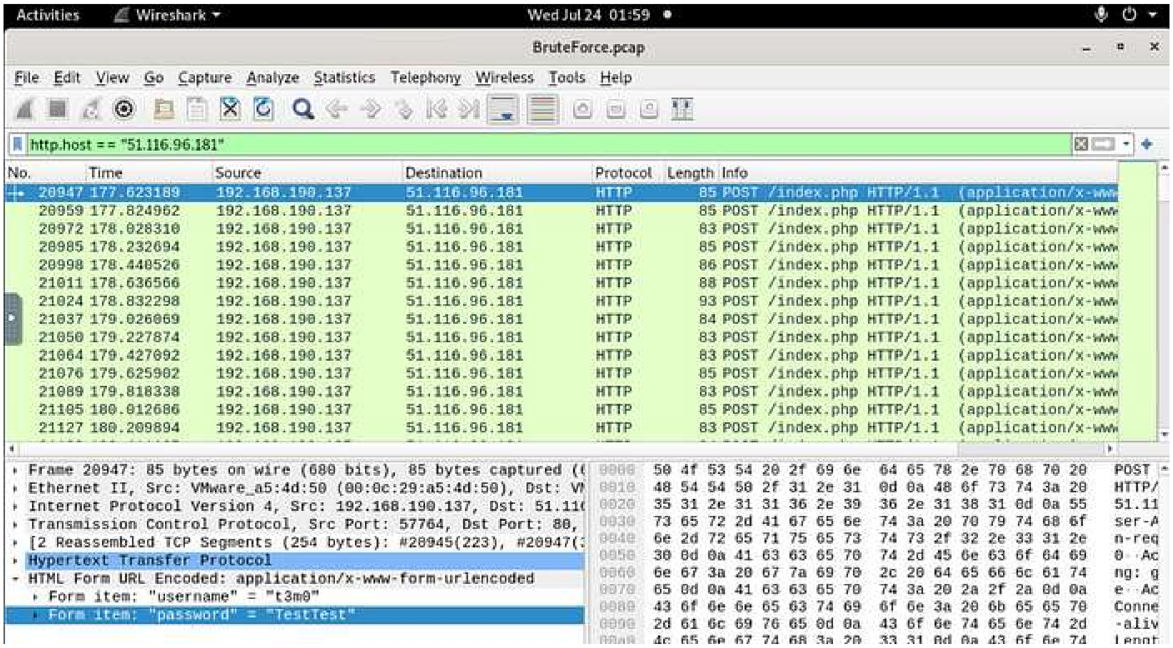
✵First, we used Wireshark for analysis BruteForce.pcap file


BruteForce.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
226	1.480440	192.168.190.137	51.116.96.181	TCP	54	50310 → 3389 [RST, ACK] Seq=1821 Ack=1208 Wi
227	1.481092	51.116.96.181	192.168.190.137	TCP	60	3389 → 50310 [ACK] Seq=1208 Ack=1821 Win=642
228	1.481121	192.168.190.137	51.116.96.181	TCP	54	50310 → 3389 [RST] Seq=1821 Win=0 Len=0
229	1.489010	192.168.190.137	51.116.96.181	TCP	54	50324 → 3389 [FIN, ACK] Seq=1816 Ack=1208 Wi
230	1.489140	192.168.190.137	51.116.96.181	TCP	54	50324 → 3389 [RST, ACK] Seq=1817 Ack=1208 Wi
231	1.489552	51.116.96.181	192.168.190.137	TCP	60	3389 → 50324 [ACK] Seq=1208 Ack=1817 Win=642
232	1.489589	192.168.190.137	51.116.96.181	TCP	54	50324 → 3389 [RST] Seq=1817 Win=0 Len=0
233	1.495966	192.168.190.137	51.116.96.181	TCP	54	50316 → 3389 [FIN, ACK] Seq=1822 Ack=1208 Wi
234	1.496095	192.168.190.137	51.116.96.181	TCP	54	50316 → 3389 [RST, ACK] Seq=1823 Ack=1208 Wi
235	1.496518	51.116.96.181	192.168.190.137	TCP	60	3389 → 50316 [ACK] Seq=1208 Ack=1823 Win=642
236	1.496549	192.168.190.137	51.116.96.181	TCP	54	50316 → 3389 [RST] Seq=1823 Win=0 Len=0
237	1.508688	51.116.96.181	192.168.190.137	RDP	130	RDP PDU Type: SynchronizeRDP PDU Type: Contr
238	1.508799	51.116.96.181	192.168.190.137	RDP	134	RDP PDU Type: Control, Action: Granted contr
239	1.508859	51.116.96.181	192.168.190.137	TCP	95	3389 → 50332 [PSH, ACK] Seq=1167 Ack=1820 Wi

Frame 40: 96 bytes on wire (768 bits), 96 bytes captured (768

- ➢ We could notice is that there is multiple RDP connection from this specific IP address(51.116.96.181) to private IP(192.168.190.137) address which should be the webserver that we got this network capture file from.


Activities
Wireshark ▾
Wed Jul 24 01:59 ●
BruteForce.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
http.host == "51.116.96.181"
```

```
No.    Time        Source           Destination    Protocol Length Info
 20947 177.623189  192.168.190.137  51.116.96.181  HTTP     85 POST /index.php HTTP/1.1  (application/x-www
 20959 177.824962  192.168.190.137  51.116.96.181  HTTP     85 POST /index.php HTTP/1.1  (application/x-www
 20972 178.028310  192.168.190.137  51.116.96.181  HTTP     83 POST /index.php HTTP/1.1  (application/x-www
 20985 178.232694  192.168.190.137  51.116.96.181  HTTP     85 POST /index.php HTTP/1.1  (application/x-www
 20998 178.440526  192.168.190.137  51.116.96.181  HTTP     86 POST /index.php HTTP/1.1  (application/x-www
 21011 178.636566  192.168.190.137  51.116.96.181  HTTP     88 POST /index.php HTTP/1.1  (application/x-www
 21024 178.832298  192.168.190.137  51.116.96.181  HTTP     93 POST /index.php HTTP/1.1  (application/x-www
 21037 179.026069  192.168.190.137  51.116.96.181  HTTP     84 POST /index.php HTTP/1.1  (application/x-www
 21050 179.227874  192.168.190.137  51.116.96.181  HTTP     83 POST /index.php HTTP/1.1  (application/x-www
 21064 179.427092  192.168.190.137  51.116.96.181  HTTP     83 POST /index.php HTTP/1.1  (application/x-www
 21076 179.625902  192.168.190.137  51.116.96.181  HTTP     85 POST /index.php HTTP/1.1  (application/x-www
 21089 179.818338  192.168.190.137  51.116.96.181  HTTP     83 POST /index.php HTTP/1.1  (application/x-www
 21105 180.012686  192.168.190.137  51.116.96.181  HTTP     85 POST /index.php HTTP/1.1  (application/x-www
 21127 180.209894  192.168.190.137  51.116.96.181  HTTP     83 POST /index.php HTTP/1.1  (application/x-www
```

```
▸ Frame 20947: 85 bytes on wire (680 bits), 85 bytes captured ((
▸ Ethernet II, Src: VMware_a5:4d:50 (00:0c:29:a5:4d:50), Dst: VM
▸ Internet Protocol Version 4, Src: 192.168.190.137, Dst: 51.116
▸ Transmission Control Protocol, Src Port: 57764, Dst Port: 80,
▸ [2 Reassembled TCP Segments (254 bytes): #20945(223), #20947(:
▸ Hypertext Transfer Protocol
▾ HTML Form URL Encoded: application/x-www-form-urlencoded
   ▸ Form item: "username" = "t3m0"
   ▸ Form item: "password" = "TestTest"
```

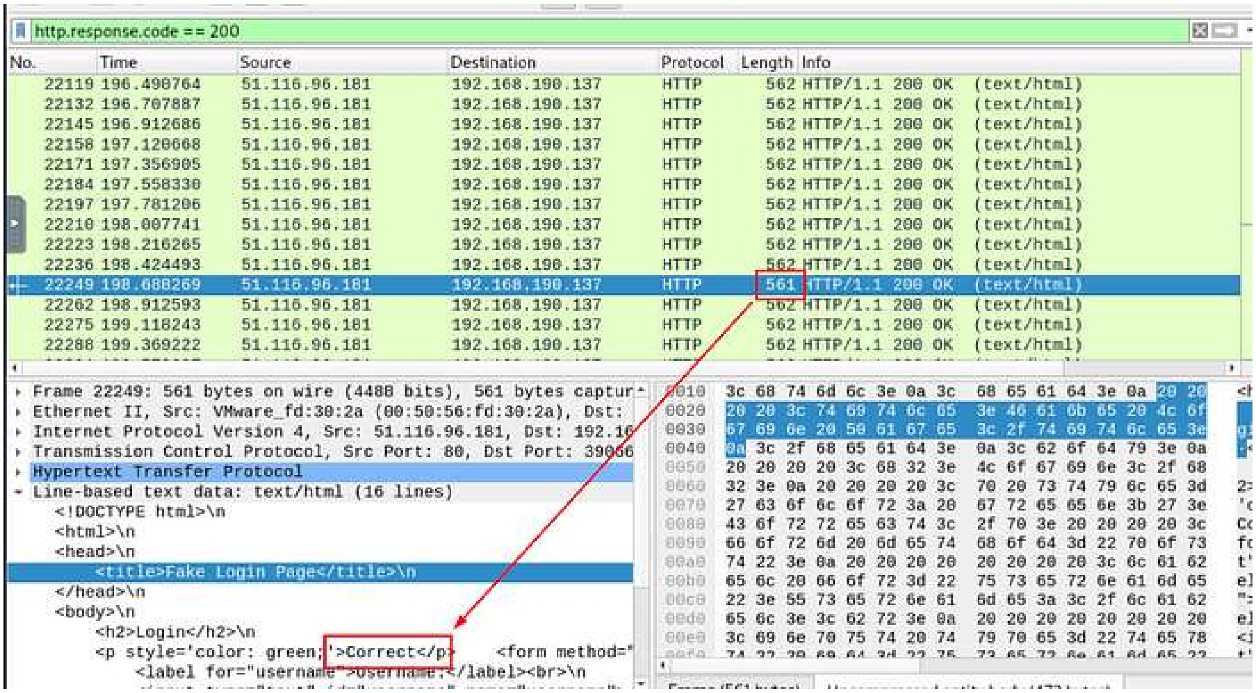
```
0000  50 4f 53 54 20 2f 69 6e  64 65 78 2e 70 68 70 20   POST
0010  48 54 54 50 2f 31 2e 31  0d 0a 48 6f 73 74 3a 20   HTTP/
0020  35 31 2e 31 31 36 2e 39  36 2e 31 38 31 0d 0a 55   51.11
0030  73 65 72 2d 41 67 65 6e  74 3a 20 70 79 74 68 6f   ser-A
0040  6e 2d 72 65 71 75 65 73  74 73 2f 32 2e 33 31 2e   n-req
0050  30 0d 0a 41 63 63 65 70  74 2d 45 6e 63 6f 64 69   0··Ac
0060  6e 67 3a 20 67 7a 69 70  2c 20 64 65 66 6c 61 74   ng: g
0070  65 0d 0a 41 63 63 65 70  74 3a 20 2a 2f 2a 0d 0a   e··Ac
0080  43 6f 6e 6e 65 63 74 69  6f 6e 3a 20 6b 65 65 70   Conne
0090  2d 61 6c 69 76 65 0d 0a  43 6f 6e 74 65 6e 74 2d   -aliv
00a0  4c 65 6e 67 74 68 3a 20  33 31 0d 0a 43 6f 6e 74   Lengt
```

- When filtered for the external IP (51.116.96.181), address which reveals another story, turns out that this external IP address is a web server, and it was also brute-forced to get access via the login page.

  - External IP Address: 51.116.96.181 - This is the IP address of the server that was targeted by the brute-force attack.

  - Internal IP Address: 192.168.100.137 - This is the private IP address from the internal network where the traffic originated.

  - (index.php) is a directory was targeted by the attacker's brute-force attempt.



```
http.response.code == 200
```

```
No.   Time       Source        Destination      Protocol Length Info
22119 196.490764 51.116.96.181 192.168.190.137  HTTP     562 HTTP/1.1 200 OK  (text/html)
22132 196.707887 51.116.96.181 192.168.190.137  HTTP     562 HTTP/1.1 200 OK  (text/html)
22145 196.912686 51.116.96.181 192.168.190.137  HTTP     562 HTTP/1.1 200 OK  (text/html)
22158 197.120668 51.116.96.181 192.168.190.137  HTTP     562 HTTP/1.1 200 OK  (text/html)
22171 197.356905 51.116.96.181 192.168.190.137  HTTP     562 HTTP/1.1 200 OK  (text/html)
22184 197.558330 51.116.96.181 192.168.190.137  HTTP     562 HTTP/1.1 200 OK  (text/html)
22197 197.781206 51.116.96.181 192.168.190.137  HTTP     562 HTTP/1.1 200 OK  (text/html)
22210 198.007741 51.116.96.181 192.168.190.137  HTTP     562 HTTP/1.1 200 OK  (text/html)
22223 198.216265 51.116.96.181 192.168.190.137  HTTP     562 HTTP/1.1 200 OK  (text/html)
22236 198.424493 51.116.96.181 192.168.190.137  HTTP     562 HTTP/1.1 200 OK  (text/html)
22249 198.688269 51.116.96.181 192.168.190.137  HTTP     561 HTTP/1.1 200 OK  (text/html)
22262 198.912593 51.116.96.181 192.168.190.137  HTTP     562 HTTP/1.1 200 OK  (text/html)
22275 199.118243 51.116.96.181 192.168.190.137  HTTP     562 HTTP/1.1 200 OK  (text/html)
22288 199.369222 51.116.96.181 192.168.190.137  HTTP     562 HTTP/1.1 200 OK  (text/html)
```

```
▸ Frame 22249: 561 bytes on wire (4488 bits), 561 bytes captur
▸ Ethernet II, Src: VMware_fd:30:2a (00:50:56:fd:30:2a), Dst:
▸ Internet Protocol Version 4, Src: 51.116.96.181, Dst: 192.16
▸ Transmission Control Protocol, Src Port: 80, Dst Port: 39066
▸ Hypertext Transfer Protocol
▾ Line-based text data: text/html (16 lines)
    <!DOCTYPE html>\n
    <html>\n
    <head>\n
        <title>Fake Login Page</title>\n
    </head>\n
    <body>\n
        <h2>Login</h2>\n
        <p style='color: green;'>Correct</p>    <form method="
            <label for="username">Username:</label><br>\n
```

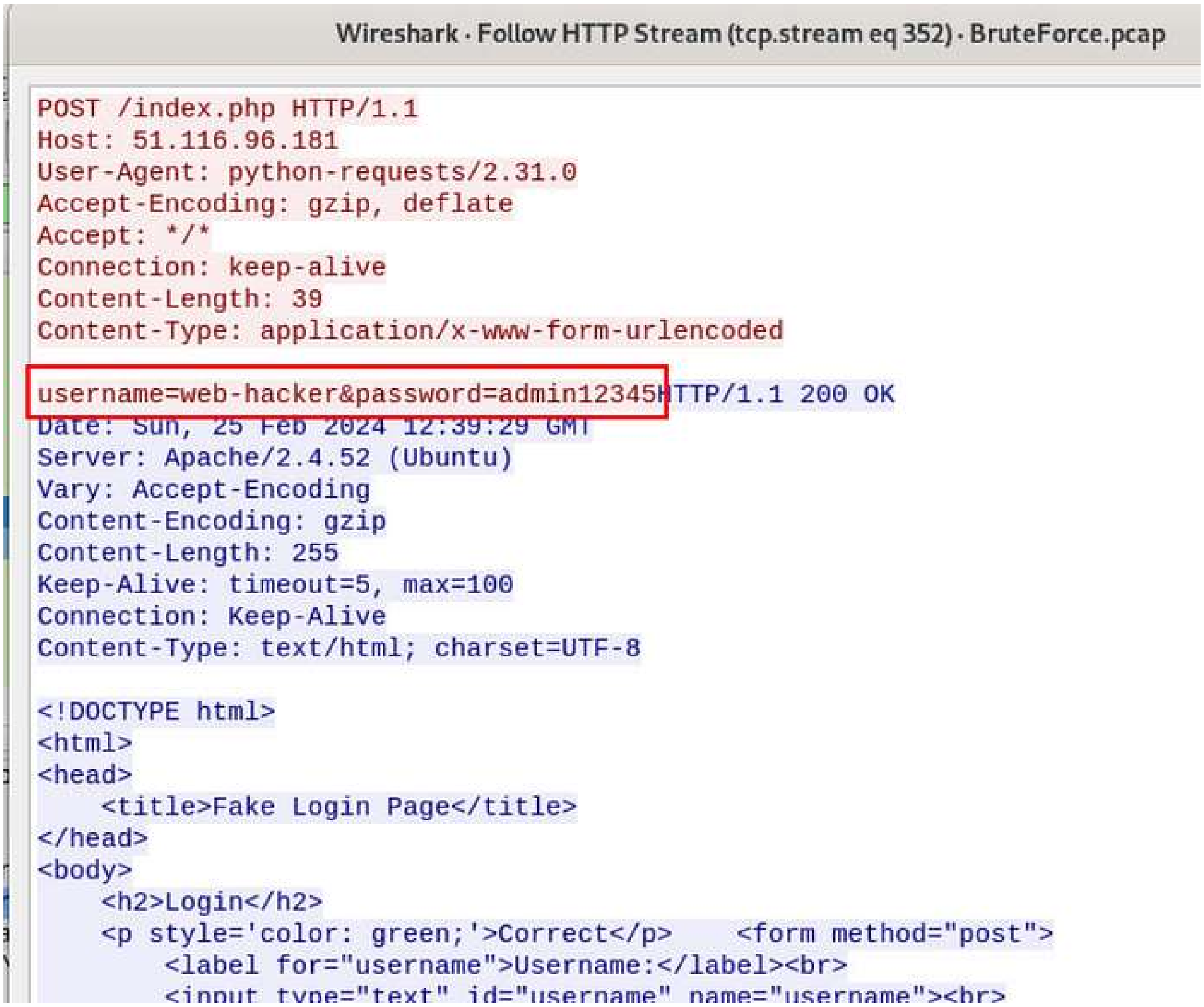
```
0010  3c 68 74 6d 6c 3e 0a 3c  68 65 61 64 3e 0a 20 20   <h
0020  20 20 3c 74 69 74 6c 65  3e 46 61 6b 65 20 4c 6f
0030  67 69 6e 20 50 61 67 65  3c 2f 74 69 74 6c 65 3e   gi
0040  0a 3c 2f 68 65 61 64 3e  0a 3c 62 6f 64 79 3e 0a
0050  20 20 20 20 3c 68 32 3e  4c 6f 67 69 6e 3c 2f 68
0060  32 3e 0a 20 20 20 20 3c  70 20 73 74 79 6c 65 3d   2>
0070  27 63 6f 6c 6f 72 3a 20  67 72 65 65 6e 3b 27 3e   'c
0080  43 6f 72 72 65 63 74 3c  2f 70 3e 20 20 20 20 3c   Co
0090  66 6f 72 6d 20 6d 65 74  68 6f 64 3d 22 70 6f 73   fo
00a0  74 22 3e 0a 20 20 20 20  20 20 20 20 3c 6c 61 62   t'
00b0  65 6c 20 66 6f 72 3d 22  75 73 65 72 6e 61 6d 65   el
00c0  22 3e 55 73 65 72 6e 61  6d 65 3a 3c 2f 6c 61 62   ":
00d0  65 6c 3e 3c 62 72 3e 0a  20 20 20 20 20 20 20 20   el
00e0  3c 69 6e 70 75 74 20 74  79 70 65 3d 22 74 65 78   <i
```

- ➢ I tried to filter for HTTP response code 302 (Redirected) which most of website will be redirected to dashboard or another webpage after successfully logged on but there is no HTTP 302 here.

- ➢ Filter for successful HTTP responses: http.response.code == 200

- ➢ So, I tried to identify which would be the different between successful logged on attempt and failed logon attempt.

- ➢ And I finally found one right there, we can see that this HTTP response got 1 less byte than the rest and when we inspected it then we could it that this attempt was successful.



Wireshark · Follow HTTP Stream (tcp.stream eq 352) · BruteForce.pcap

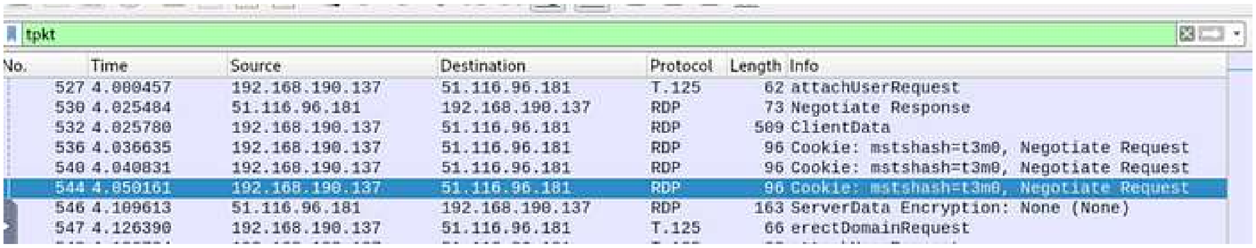
```
POST /index.php HTTP/1.1
Host: 51.116.96.181
User-Agent: python-requests/2.31.0
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
Content-Length: 39
Content-Type: application/x-www-form-urlencoded

username=web-hacker&password=admin12345HTTP/1.1 200 OK
Date: Sun, 25 Feb 2024 12:39:29 GMT
Server: Apache/2.4.52 (Ubuntu)
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 255
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html>
<head>
    <title>Fake Login Page</title>
</head>
<body>
    <h2>Login</h2>
    <p style='color: green;'>Correct</p>    <form method="post">
        <label for="username">Username:</label><br>
        <input type="text" id="username" name="username"><br>
```

And here is the valid credentials that was accepted on this web server :

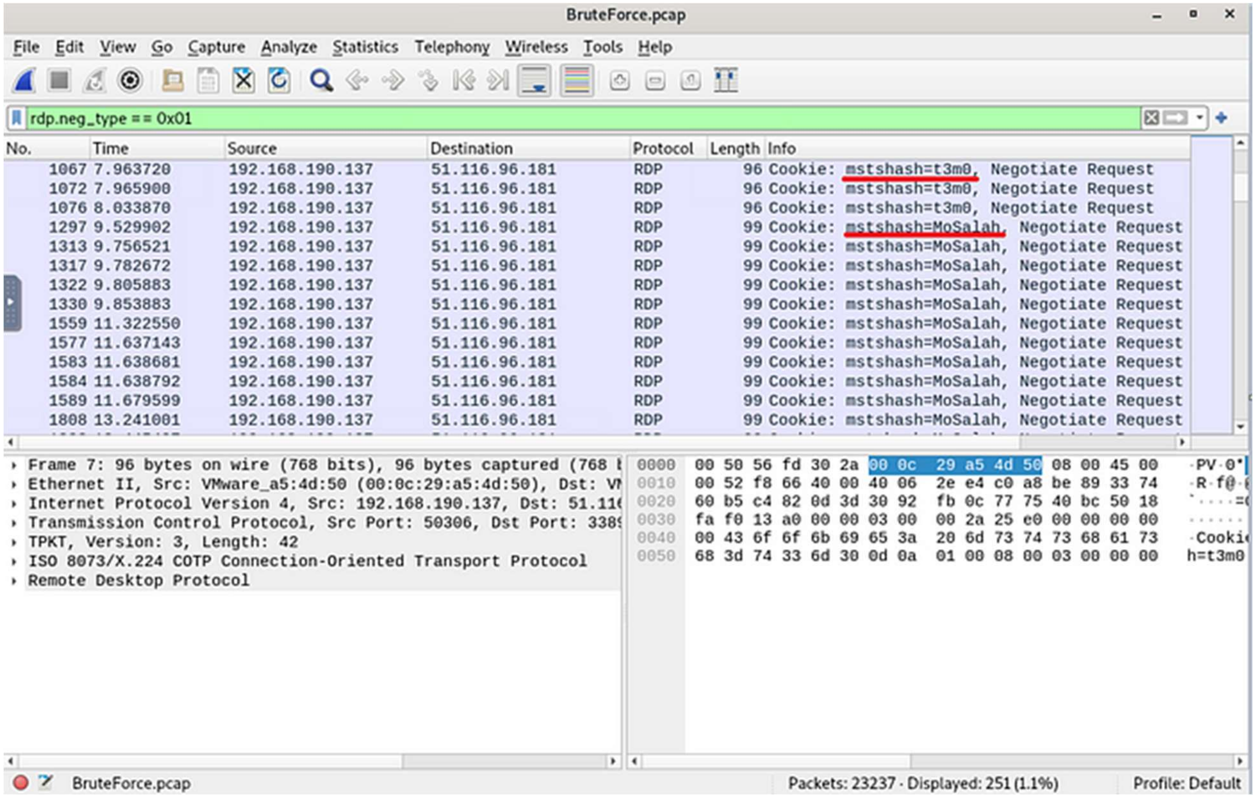
web-hacker:admin12345



```
tpkt
```

```
No.   Time       Source            Destination       Protocol Length Info
  527 4.080457   192.168.190.137   51.116.96.181     T.125       62 attachUserRequest
  530 4.025484   51.116.96.181     192.168.190.137   RDP         73 Negotiate Response
  532 4.025780   192.168.190.137   51.116.96.181     RDP        509 ClientData
  536 4.036635   192.168.190.137   51.116.96.181     RDP         96 Cookie: mstshash=t3m0, Negotiate Request
  540 4.040831   192.168.190.137   51.116.96.181     RDP         96 Cookie: mstshash=t3m0, Negotiate Request
  544 4.050161   192.168.190.137   51.116.96.181     RDP         96 Cookie: mstshash=t3m0, Negotiate Request
  546 4.109613   51.116.96.181     192.168.190.137   RDP        163 ServerData Encryption: None (None)
  547 4.126390   192.168.190.137   51.116.96.181     T.125       66 erectDomainRequest
```

- After filtered for rdp, we can see that there is user account name on Negotiation Request so we can use this information to get all users that was requested from this IP address.


BruteForce.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
rdp.neg_type == 0x01
```

```
No.   Time        Source            Destination      Protocol Length Info
 1067 7.963720    192.168.190.137   51.116.96.181    RDP         96 Cookie: mstshash=t3m0, Negotiate Request
 1072 7.965900    192.168.190.137   51.116.96.181    RDP         96 Cookie: mstshash=t3m0, Negotiate Request
 1076 8.033870    192.168.190.137   51.116.96.181    RDP         96 Cookie: mstshash=t3m0, Negotiate Request
 1297 9.529902    192.168.190.137   51.116.96.181    RDP         99 Cookie: mstshash=MoSalah, Negotiate Request
 1313 9.756521    192.168.190.137   51.116.96.181    RDP         99 Cookie: mstshash=MoSalah, Negotiate Request
 1317 9.782672    192.168.190.137   51.116.96.181    RDP         99 Cookie: mstshash=MoSalah, Negotiate Request
 1322 9.805883    192.168.190.137   51.116.96.181    RDP         99 Cookie: mstshash=MoSalah, Negotiate Request
 1330 9.853883    192.168.190.137   51.116.96.181    RDP         99 Cookie: mstshash=MoSalah, Negotiate Request
 1559 11.322550   192.168.190.137   51.116.96.181    RDP         99 Cookie: mstshash=MoSalah, Negotiate Request
 1577 11.637143   192.168.190.137   51.116.96.181    RDP         99 Cookie: mstshash=MoSalah, Negotiate Request
 1583 11.638681   192.168.190.137   51.116.96.181    RDP         99 Cookie: mstshash=MoSalah, Negotiate Request
 1584 11.638792   192.168.190.137   51.116.96.181    RDP         99 Cookie: mstshash=MoSalah, Negotiate Request
 1589 11.679599   192.168.190.137   51.116.96.181    RDP         99 Cookie: mstshash=MoSalah, Negotiate Request
 1808 13.241001   192.168.190.137   51.116.96.181    RDP         99 Cookie: mstshash=MoSalah, Negotiate Request
```

```
› Frame 7: 96 bytes on wire (768 bits), 96 bytes captured (768 b
› Ethernet II, Src: VMware_a5:4d:50 (00:0c:29:a5:4d:50), Dst: VM
› Internet Protocol Version 4, Src: 192.168.190.137, Dst: 51.116
› Transmission Control Protocol, Src Port: 50306, Dst Port: 3389
› TPKT, Version: 3, Length: 42
› ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
› Remote Desktop Protocol
```

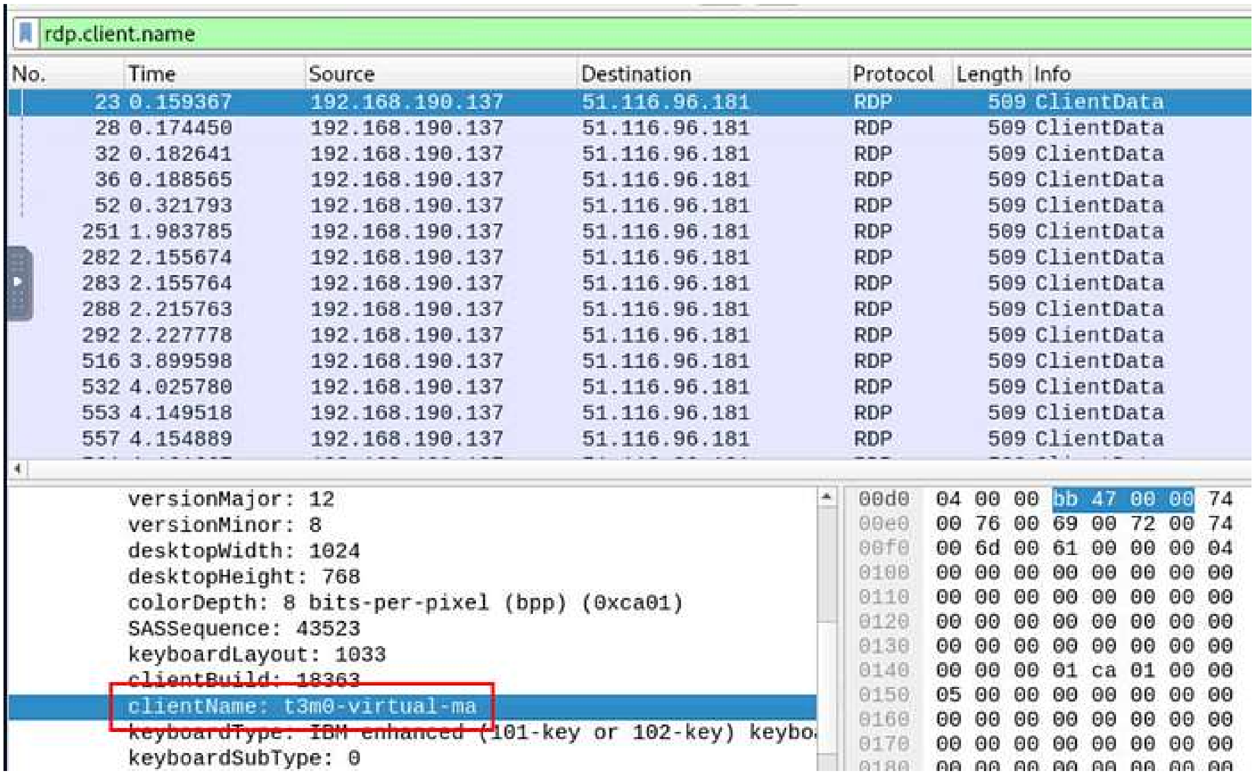
```
0000  00 50 56 fd 30 2a 00 0c  29 a5 4d 50 08 00 45 00   ·PV·0*
0010  00 52 f8 66 40 00 40 06  2e e4 c0 a8 be 89 33 74   ·R·f@·(
0020  60 b5 c4 82 0d 3d 30 92  fb 0c 77 75 40 bc 50 18   `····=(
0030  fa f0 13 a0 00 00 03 00  00 2a 25 e0 00 00 00 00   ······
0040  00 43 6f 6f 6b 69 65 3a  20 6d 73 74 73 68 61 73   ·Cookie
0050  68 3d 74 33 6d 30 0d 0a  01 00 08 00 03 00 00 00   h=t3m0
```

BruteForce.pcap   Packets: 23237 · Displayed: 251 (1.1%)   Profile: Default

- To identify RDP packets related to brute-force attempts, we can use a specific filter in Wireshark. The filter rdp.neg_type == 0x01 allows us to display only the packets that are Negotiation Requests in the RDP protocol which can indicate brute-force attempts.

- Users targeted: t3m0, MoSalah, Messi, Kareem, Mostafa, mmox, Mohamed, Ali, Mohsen, web-hacker.

- Only 7 accounts were brute-forced.



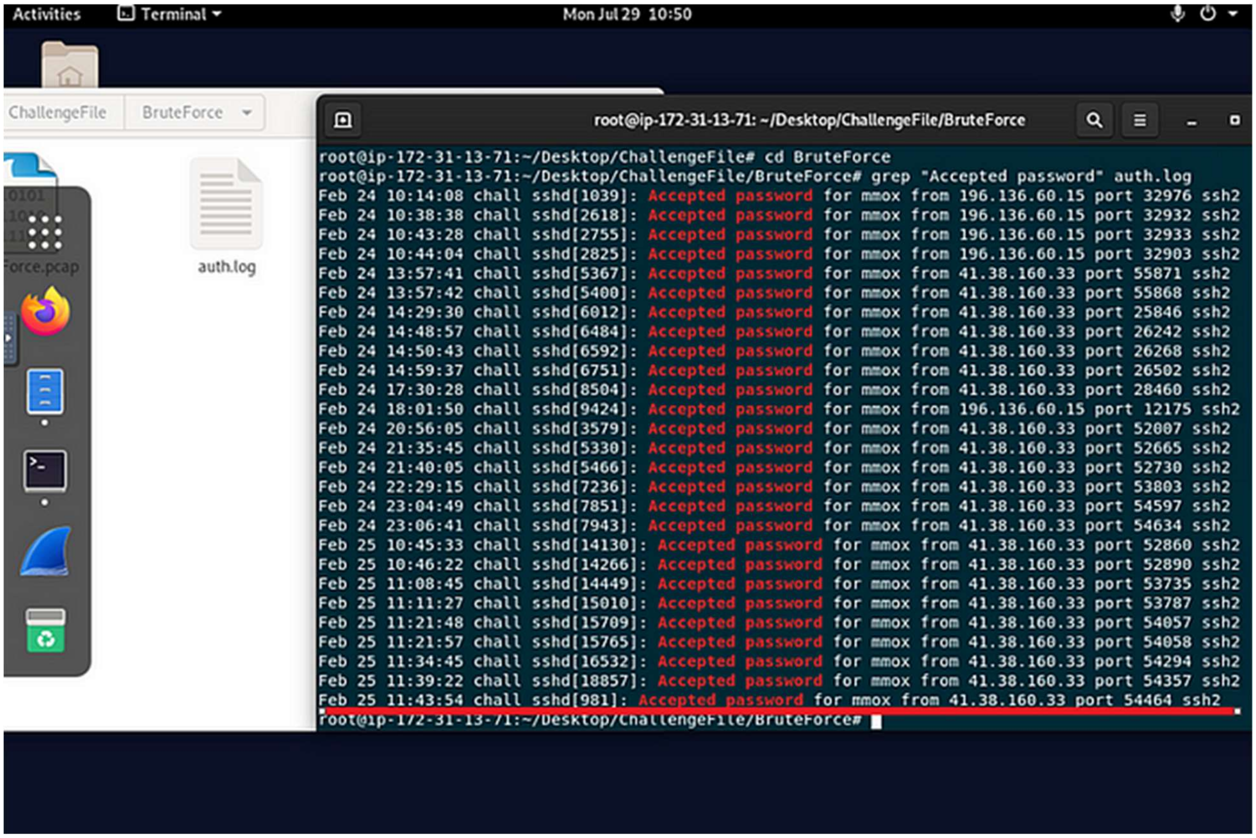
rdp.client.name

```
No.     Time      Source            Destination     Protocol  Length Info
   23 0.159367  192.168.190.137  51.116.96.181    RDP          509 ClientData
   28 0.174450  192.168.190.137  51.116.96.181    RDP          509 ClientData
   32 0.182641  192.168.190.137  51.116.96.181    RDP          509 ClientData
   36 0.188565  192.168.190.137  51.116.96.181    RDP          509 ClientData
   52 0.321793  192.168.190.137  51.116.96.181    RDP          509 ClientData
  251 1.983785  192.168.190.137  51.116.96.181    RDP          509 ClientData
  282 2.155674  192.168.190.137  51.116.96.181    RDP          509 ClientData
  283 2.155764  192.168.190.137  51.116.96.181    RDP          509 ClientData
  288 2.215763  192.168.190.137  51.116.96.181    RDP          509 ClientData
  292 2.227778  192.168.190.137  51.116.96.181    RDP          509 ClientData
  516 3.899598  192.168.190.137  51.116.96.181    RDP          509 ClientData
  532 4.025780  192.168.190.137  51.116.96.181    RDP          509 ClientData
  553 4.149518  192.168.190.137  51.116.96.181    RDP          509 ClientData
  557 4.154889  192.168.190.137  51.116.96.181    RDP          509 ClientData
```

```
    versionMajor: 12                                    00d0  04 00 00 bb 47 00 00 74
    versionMinor: 8                                     00e0  00 76 00 69 00 72 00 74
    desktopWidth: 1024                                  00f0  00 6d 00 61 00 00 00 04
    desktopHeight: 768                                  0100  00 00 00 00 00 00 00 00
    colorDepth: 8 bits-per-pixel (bpp) (0xca01)         0110  00 00 00 00 00 00 00 00
    SASSequence: 43523                                  0120  00 00 00 00 00 00 00 00
    keyboardLayout: 1033                                0130  00 00 00 00 00 00 00 00
    clientBuild: 18363                                  0140  00 00 00 01 ca 01 00 00
    clientName: t3m0-virtual-ma                         0150  05 00 00 00 00 00 00 00
    keyboardType: IBM enhanced (101-key or 102-key) keybo  0160  00 00 00 00 00 00 00 00
    keyboardSubType: 0                                  0170  00 00 00 00 00 00 00 00
```

- We can use rdp.client.name filter to get client Name of an attacker machine.

- t3m0-virtual-ma is client Name of an attacker machine

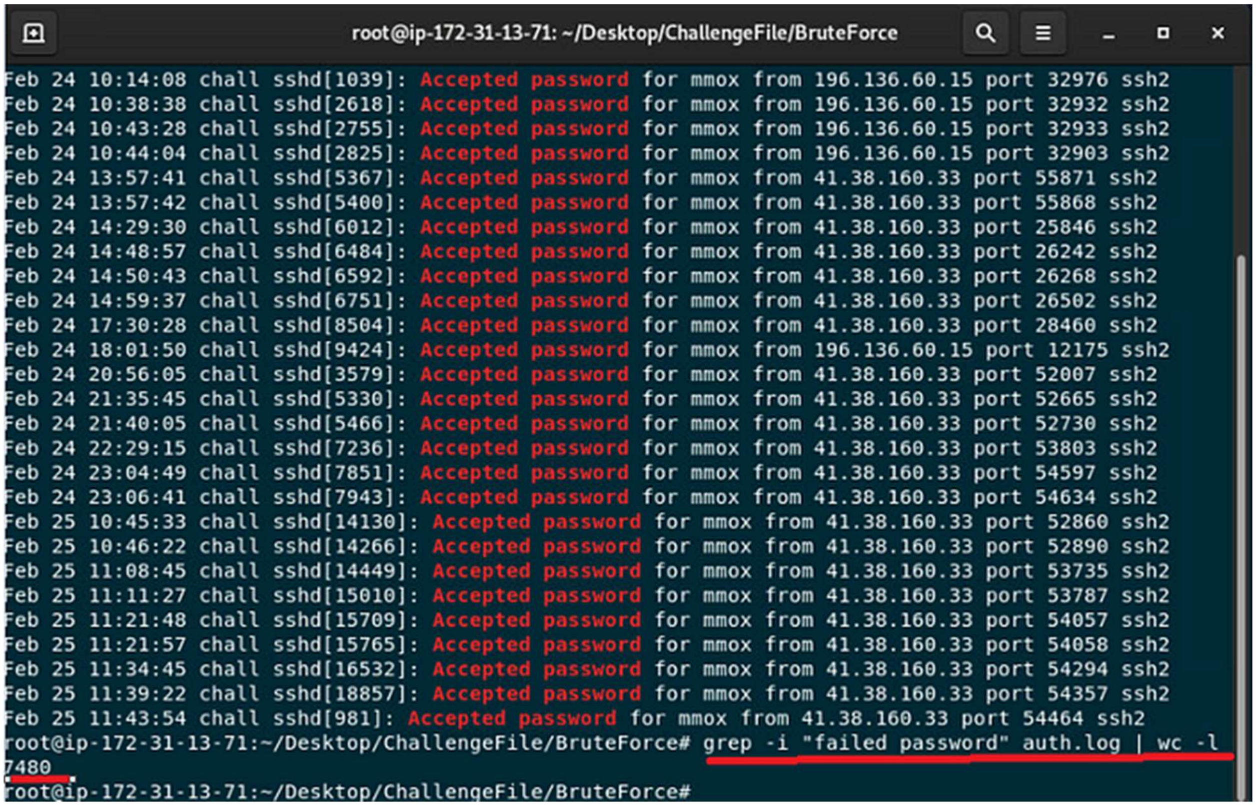


```
root@ip-172-31-13-71:~/Desktop/ChallengeFile# cd BruteForce
root@ip-172-31-13-71:~/Desktop/ChallengeFile/BruteForce# grep "Accepted password" auth.log
Feb 24 10:14:08 chall sshd[1039]: Accepted password for mmox from 196.136.60.15 port 32976 ssh2
Feb 24 10:38:38 chall sshd[2618]: Accepted password for mmox from 196.136.60.15 port 32932 ssh2
Feb 24 10:43:28 chall sshd[2755]: Accepted password for mmox from 196.136.60.15 port 32933 ssh2
Feb 24 10:44:04 chall sshd[2825]: Accepted password for mmox from 196.136.60.15 port 32903 ssh2
Feb 24 13:57:41 chall sshd[5367]: Accepted password for mmox from 41.38.160.33 port 55871 ssh2
Feb 24 13:57:42 chall sshd[5400]: Accepted password for mmox from 41.38.160.33 port 55868 ssh2
Feb 24 14:29:30 chall sshd[6012]: Accepted password for mmox from 41.38.160.33 port 25846 ssh2
Feb 24 14:48:57 chall sshd[6484]: Accepted password for mmox from 41.38.160.33 port 26242 ssh2
Feb 24 14:50:43 chall sshd[6592]: Accepted password for mmox from 41.38.160.33 port 26268 ssh2
Feb 24 14:59:37 chall sshd[6751]: Accepted password for mmox from 41.38.160.33 port 26502 ssh2
Feb 24 17:30:28 chall sshd[8504]: Accepted password for mmox from 41.38.160.33 port 28460 ssh2
Feb 24 18:01:50 chall sshd[9424]: Accepted password for mmox from 196.136.60.15 port 12175 ssh2
Feb 24 20:56:05 chall sshd[3579]: Accepted password for mmox from 41.38.160.33 port 52007 ssh2
Feb 24 21:35:45 chall sshd[5330]: Accepted password for mmox from 41.38.160.33 port 52665 ssh2
Feb 24 21:40:05 chall sshd[5466]: Accepted password for mmox from 41.38.160.33 port 52730 ssh2
Feb 24 22:29:15 chall sshd[7236]: Accepted password for mmox from 41.38.160.33 port 53803 ssh2
Feb 24 23:04:49 chall sshd[7851]: Accepted password for mmox from 41.38.160.33 port 54597 ssh2
Feb 24 23:06:41 chall sshd[7943]: Accepted password for mmox from 41.38.160.33 port 54634 ssh2
Feb 25 10:45:33 chall sshd[14130]: Accepted password for mmox from 41.38.160.33 port 52860 ssh2
Feb 25 10:46:22 chall sshd[14266]: Accepted password for mmox from 41.38.160.33 port 52890 ssh2
Feb 25 11:08:45 chall sshd[14449]: Accepted password for mmox from 41.38.160.33 port 53735 ssh2
Feb 25 11:11:27 chall sshd[15010]: Accepted password for mmox from 41.38.160.33 port 53787 ssh2
Feb 25 11:21:48 chall sshd[15709]: Accepted password for mmox from 41.38.160.33 port 54057 ssh2
Feb 25 11:21:57 chall sshd[15765]: Accepted password for mmox from 41.38.160.33 port 54058 ssh2
Feb 25 11:34:45 chall sshd[16532]: Accepted password for mmox from 41.38.160.33 port 54294 ssh2
Feb 25 11:39:22 chall sshd[18857]: Accepted password for mmox from 41.38.160.33 port 54357 ssh2
Feb 25 11:43:54 chall sshd[981]: Accepted password for mmox from 41.38.160.33 port 54464 ssh2
root@ip-172-31-13-71:~/Desktop/ChallengeFile/BruteForce#
```

✵Then, Analyzing auth.log provides insights into successful and unsuccessful login attempts, helping us understand the attacker's activity.

- Let's start by opening our "auth.log" file in the terminal and using grep to search for accepted passwords: grep "Accepted password" auth.log

- We are interested in latest successful login, so the latest login in this case is: mmox:11:43:54



```
Feb 24 10:14:08 chall sshd[1039]: Accepted password for mmox from 196.136.60.15 port 32976 ssh2
Feb 24 10:38:38 chall sshd[2618]: Accepted password for mmox from 196.136.60.15 port 32932 ssh2
Feb 24 10:43:28 chall sshd[2755]: Accepted password for mmox from 196.136.60.15 port 32933 ssh2
Feb 24 10:44:04 chall sshd[2825]: Accepted password for mmox from 196.136.60.15 port 32903 ssh2
Feb 24 13:57:41 chall sshd[5367]: Accepted password for mmox from 41.38.160.33 port 55871 ssh2
Feb 24 13:57:42 chall sshd[5400]: Accepted password for mmox from 41.38.160.33 port 55868 ssh2
Feb 24 14:29:30 chall sshd[6012]: Accepted password for mmox from 41.38.160.33 port 25846 ssh2
Feb 24 14:48:57 chall sshd[6484]: Accepted password for mmox from 41.38.160.33 port 26242 ssh2
Feb 24 14:50:43 chall sshd[6592]: Accepted password for mmox from 41.38.160.33 port 26268 ssh2
Feb 24 14:59:37 chall sshd[6751]: Accepted password for mmox from 41.38.160.33 port 26502 ssh2
Feb 24 17:30:28 chall sshd[8504]: Accepted password for mmox from 41.38.160.33 port 28460 ssh2
Feb 24 18:01:50 chall sshd[9424]: Accepted password for mmox from 196.136.60.15 port 12175 ssh2
Feb 24 20:56:05 chall sshd[3579]: Accepted password for mmox from 41.38.160.33 port 52007 ssh2
Feb 24 21:35:45 chall sshd[5330]: Accepted password for mmox from 41.38.160.33 port 52665 ssh2
Feb 24 21:40:05 chall sshd[5466]: Accepted password for mmox from 41.38.160.33 port 52730 ssh2
Feb 24 22:29:15 chall sshd[7236]: Accepted password for mmox from 41.38.160.33 port 53803 ssh2
Feb 24 23:04:49 chall sshd[7851]: Accepted password for mmox from 41.38.160.33 port 54597 ssh2
Feb 24 23:06:41 chall sshd[7943]: Accepted password for mmox from 41.38.160.33 port 54634 ssh2
Feb 25 10:45:33 chall sshd[14130]: Accepted password for mmox from 41.38.160.33 port 52860 ssh2
Feb 25 10:46:22 chall sshd[14266]: Accepted password for mmox from 41.38.160.33 port 52890 ssh2
Feb 25 11:08:45 chall sshd[14449]: Accepted password for mmox from 41.38.160.33 port 53735 ssh2
Feb 25 11:11:27 chall sshd[15010]: Accepted password for mmox from 41.38.160.33 port 53787 ssh2
Feb 25 11:21:48 chall sshd[15709]: Accepted password for mmox from 41.38.160.33 port 54057 ssh2
Feb 25 11:21:57 chall sshd[15765]: Accepted password for mmox from 41.38.160.33 port 54058 ssh2
Feb 25 11:34:45 chall sshd[16532]: Accepted password for mmox from 41.38.160.33 port 54294 ssh2
Feb 25 11:39:22 chall sshd[18857]: Accepted password for mmox from 41.38.160.33 port 54357 ssh2
Feb 25 11:43:54 chall sshd[981]: Accepted password for mmox from 41.38.160.33 port 54464 ssh2
root@ip-172-31-13-71:~/Desktop/ChallengeFile/BruteForce# grep -i "failed password" auth.log | wc -l
7480
root@ip-172-31-13-71:~/Desktop/ChallengeFile/BruteForce#
```

- Simple case, Use the command: grep -i “failed password” auth.log | wc -l

- We noticed that there were 7480 incorrect login attempts.

- ## Summary of the detection

1- RDP Brute Force Attack

- ✓ The attacker, originating from (192.168190.137), made multiple attempts to gain unauthorized access via RDP. The attack targeted 10 unique user accounts, including t3m0, Mosalah, and Mostafa. Seven accounts were successfully brute-forced, with web-hacker being one of the compromised sets of credentials.

2- Compromised Web Login Attempts

- ✓ The attacker also targeted the login page on the web server (index.php) through brute force attempts. A successful login with the credential’s web-hacker was identified.

3- Client Machine Details

- ✓ By analyzing the RDP negotiation requests, it was identified that the attacker's machine used the client’s name t3m0-virtual-ma.

4- SSH Activity

- ✓ The last successful SSH login was by the user mmox at 11:43:54. The attacker also attempted 7480 failed SSH login attempts before gaining access.

- ## Communication Activities with Stakeholders

  - ✓ Key stakeholders, including IT security, management, and legal teams, were immediately informed.
  - ✓ Daily briefings were held to update on the containment and eradication efforts.

- ## Containment and Eradication Procedures Implemented

  - ✓ IP (192.168190.137) was blocked.
  - ✓ Passwords for compromised accounts (e.g., "web-hacker") were reset.
  - ✓ Regular monitoring of RDP and web login attempts was implemented.
  - ✓ Two-factor authentication (2FA) was deployed across all affected systems.

- ## Recovery Efforts Undertaken

  - ✓ Restored all compromised accounts and ensured system integrity.
  - ✓ Re-deployed secure backups of the web server.
  - ✓ Initiated additional penetration testing to ensure no further vulnerabilities existed.

- ## Decisions Made Throughout the Response Process

  - ✓ Decisions focused on quick containment and thorough root cause analysis, followed by gradual system recovery.
  - ✓ The team prioritized strengthening authentication mechanisms and enhancing logging to prevent future attacks.

- ## Tools and Resources Utilized

✵Analysis & Detection

  - ✓ Wireshark: For network traffic analysis and packet capture inspection.
  - ✓ Grep: Used to filter authentication logs and identify both successful and failed login attempts.

✵ Containment & Eradication

  - ✓ Firewall: For IP blocking and traffic monitoring.
  - ✓ Penetration Testing Tools: For post-recovery security assessment.