

Brute Force Attacks Challenge

Brute Force Attacks

Challenge Image

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?

Question 1:

What is the IP address of the server targeted by the attacker's brute-force attack?

I checked the conversation in Wireshark

| Ethernet · 6 | IPv4 · 6 | IPv6 · 2 | TCP · 429 | UDP · 7 | | | | | | | | | | |
|-----------------|-----------------|----------|-----------|---------------|-------------|---------------|-------------|------------|----------|--------------|--------------|--|--|--|
| Address A | Address B | Packets | Bytes | Packets A → B | Bytes A → B | Packets B → A | Bytes B → A | Rel Start | Duration | Bits/s A → B | Bits/s B → A | | | |
| 192.168.190.137 | 51.116.96.181 | 23,199 | 4 MB | 10,556 | 1 MB | 12,643 | 3 MB | 0.000000 | 216.9161 | 41 kbps | 122 kbps | | | |
| 192.168.190.137 | 91.189.91.96 | 10 | 856 bytes | 5 | 377 bytes | 5 | 479 bytes | 179.879228 | 0.2813 | 10 kbps | 13 kbps | | | |
| 192.168.190.137 | 162.213.33.48 | 8 | 578 bytes | 7 | 518 bytes | 1 | 60 bytes | 89.796686 | 21.0498 | 196 bits/s | 22 bits/s | | | |
| 192.168.190.137 | 192.168.190.2 | 8 | 1 kB | 4 | 378 bytes | 4 | 983 bytes | 89.694186 | 90.1824 | 33 bits/s | 87 bits/s | | | |
| 192.168.190.1 | 224.0.0.251 | 2 | 170 bytes | 2 | 170 bytes | 0 | 0 bytes | 29.366633 | 1.0124 | 1343 bits/s | 0 bits/s | | | |
| 192.168.190.1 | 192.168.190.255 | 1 | 243 bytes | 1 | 243 bytes | 0 | 0 bytes | 142.079208 | 0.0000 | | | | | |

Answer: 51.116.96.181

Question 2:

Which directory was targeted by the attacker's brute-force attempt?

I filtered for ip.src == 192.168.190.137 && ip.dst == 51.116.96.181 && http and found POST requests

| ip.src == 192.168.190.137 && ip.dst == 51.116.96.181 && http | | | | | | | | | |
|--|-------|------------|-----------------|---------------|--------|------|-----------------|----------|-------------------------------------|
| . | 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-form-urlencoded) |
| | 20959 | 177.824962 | 192.168.190.137 | 51.116.96.181 | HTTP | 85 | POST /index.php | HTTP/1.1 | (application/x-www-form-urlencoded) |
| | 20972 | 178.028310 | 192.168.190.137 | 51.116.96.181 | HTTP | 83 | POST /index.php | HTTP/1.1 | (application/x-www-form-urlencoded) |
| | 20985 | 178.232694 | 192.168.190.137 | 51.116.96.181 | HTTP | 85 | POST /index.php | HTTP/1.1 | (application/x-www-form-urlencoded) |
| | 20998 | 178.440526 | 192.168.190.137 | 51.116.96.181 | HTTP | 86 | POST /index.php | HTTP/1.1 | (application/x-www-form-urlencoded) |
| | 21011 | 178.636566 | 192.168.190.137 | 51.116.96.181 | HTTP | 88 | POST /index.php | HTTP/1.1 | (application/x-www-form-urlencoded) |
| | 21024 | 178.832298 | 192.168.190.137 | 51.116.96.181 | HTTP | 93 | POST /index.php | HTTP/1.1 | (application/x-www-form-urlencoded) |
| | 21037 | 179.026069 | 192.168.190.137 | 51.116.96.181 | HTTP | 84 | POST /index.php | HTTP/1.1 | (application/x-www-form-urlencoded) |
| | 21050 | 179.227874 | 192.168.190.137 | 51.116.96.181 | HTTP | 83 | POST /index.php | HTTP/1.1 | (application/x-www-form-urlencoded) |
| | 21064 | 179.427092 | 192.168.190.137 | 51.116.96.181 | HTTP | 83 | POST /index.php | HTTP/1.1 | (application/x-www-form-urlencoded) |
| | 21076 | 179.625902 | 192.168.190.137 | 51.116.96.181 | HTTP | 85 | POST /index.php | HTTP/1.1 | (application/x-www-form-urlencoded) |

Answer: index.php

Question 3:

Identify the correct username and password combination used for login.

I exported all the HTTP logs and then used cat on all files and searched for "Correct"

Protocol Length Info

| Protocol | Length | Info |
|----------|--------|--|
| HTTP | 86 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 87 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 86 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 86 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 86 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 86 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 90 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 89 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 86 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 87 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 90 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 90 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 85 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 85 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 83 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 85 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 86 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 88 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 93 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 84 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 83 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 83 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 85 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 83 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 85 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 83 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 84 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 83 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |
| HTTP | 83 | POST /index.php HTTP/1.1 (application/x-www-form-urlencoded) |

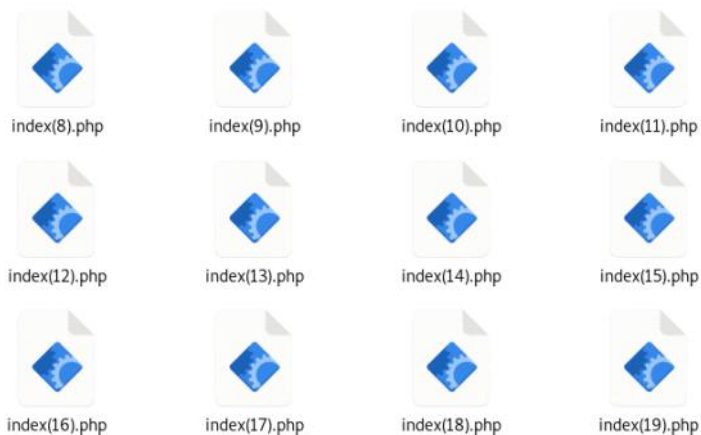
(664 bits)
VMware_f0:30:2a (00:50:56:fd:30:2a)
16.96.181
, Seq: 224, Ack: 1, Len: 29
(29)]

Wireshark · Export · HTTP object list

Text Filter: Content Type: All Content-Types

| Packet | Hostname | Content Type | Size | Filename |
|--------|---------------|-----------------------------------|-----------|-----------|
| 20947 | 51.116.96.181 | application/x-www-form-urlencoded | 31 bytes | index.php |
| 20949 | 51.116.96.181 | text/html | 472 bytes | index.php |
| 20959 | 51.116.96.181 | application/x-www-form-urlencoded | 31 bytes | index.php |
| 20962 | 51.116.96.181 | text/html | 472 bytes | index.php |
| 20972 | 51.116.96.181 | application/x-www-form-urlencoded | 29 bytes | index.php |
| 20975 | 51.116.96.181 | text/html | 472 bytes | index.php |
| 20985 | 51.116.96.181 | application/x-www-form-urlencoded | 31 bytes | index.php |
| 20988 | 51.116.96.181 | text/html | 472 bytes | index.php |
| 20998 | 51.116.96.181 | application/x-www-form-urlencoded | 32 bytes | index.php |
| 21001 | 51.116.96.181 | text/html | 472 bytes | index.php |
| 21011 | 51.116.96.181 | application/x-www-form-urlencoded | 34 bytes | index.php |
| 21014 | 51.116.96.181 | text/html | 472 bytes | index.php |
| 21024 | 51.116.96.181 | application/x-www-form-urlencoded | 39 bytes | index.php |
| 21027 | 51.116.96.181 | text/html | 472 bytes | index.php |
| 21037 | 51.116.96.181 | application/x-www-form-urlencoded | 30 bytes | index.php |
| 21040 | 51.116.96.181 | text/html | 472 bytes | index.php |
| 21050 | 51.116.96.181 | application/x-www-form-urlencoded | 29 bytes | index.php |
| 21053 | 51.116.96.181 | text/html | 472 bytes | index.php |
| 21064 | 51.116.96.181 | application/x-www-form-urlencoded | 29 bytes | index.php |
| 21066 | 51.116.96.181 | text/html | 472 bytes | index.php |
| 21076 | 51.116.96.181 | application/x-www-form-urlencoded | 31 bytes | index.php |

50 08 00 45 00 .PV.0*..).MP..E.
a8 be 89 33 74 .E..@.@.JR...3t
df 7c 5d 50 18 '....P8/....]P.
61 6d 65 3d 6dus ername=m
64 3d 62 61 74 mox&pass word=bat
man



```
username=t3m0&password=testtestroot@1p-172-31-0-36:~# cat *.php
```

```
username=web-hacker&password=admin12345<!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>
    <label for="password">Password:</label><br>
    <input type="password" id="password" name="password"><br>
    <input type="submit" value="Login">
  </form>
```

Answer: web-hacker:admin12345

Question 4:

How many user accounts did the attacker attempt to compromise via RDP brute-force?

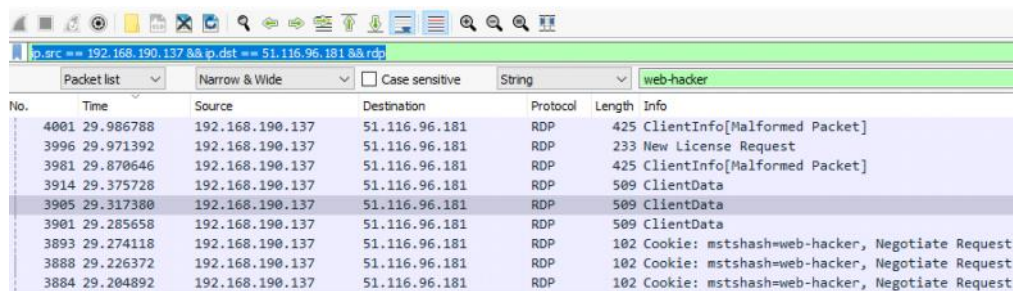
I searched for rdp and string msthash, I counted atleast 9-10 users but it was wrong. I did a brute force.

Answer: 7

Question 5:

What is the "clientName" of the attacker's machine?

I filtered for `ip.src == 192.168.190.137 && ip.dst == 51.116.96.181 && rdp` and also searched "web-hacker" by string.
Then I checked the ClientData packet



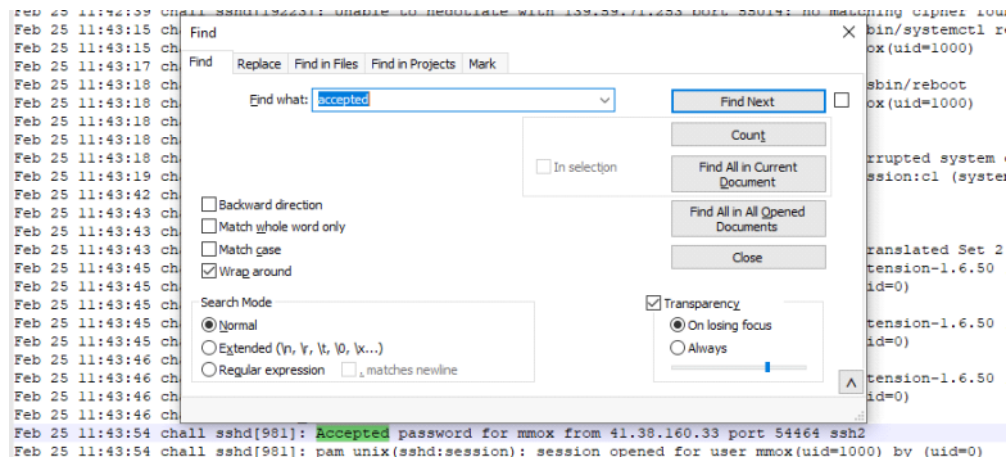
| No. | Time | Source | Destination | Protocol | Length | Info |
|------|-----------|-----------------|---------------|----------|--------|--|
| 4001 | 29.986788 | 192.168.190.137 | 51.116.96.181 | RDP | 425 | ClientInfo[Malformed Packet] |
| 3996 | 29.971392 | 192.168.190.137 | 51.116.96.181 | RDP | 233 | New License Request |
| 3981 | 29.870646 | 192.168.190.137 | 51.116.96.181 | RDP | 425 | ClientInfo[Malformed Packet] |
| 3914 | 29.375728 | 192.168.190.137 | 51.116.96.181 | RDP | 509 | ClientData |
| 3905 | 29.317380 | 192.168.190.137 | 51.116.96.181 | RDP | 509 | ClientData |
| 3901 | 29.285658 | 192.168.190.137 | 51.116.96.181 | RDP | 509 | ClientData |
| 3893 | 29.274118 | 192.168.190.137 | 51.116.96.181 | RDP | 102 | Cookie: mstshash=web-hacker, Negotiate Request |
| 3888 | 29.226372 | 192.168.190.137 | 51.116.96.181 | RDP | 102 | Cookie: mstshash=web-hacker, Negotiate Request |
| 3884 | 29.204892 | 192.168.190.137 | 51.116.96.181 | RDP | 102 | Cookie: mstshash=web-hacker, Negotiate Request |

```
> MULTIPOINT-COMMUNICATION-SERVICE T.125
> GENERIC-CONFERENCE-CONTROL T.124
> Remote Desktop Protocol
  > ClientData
    > clientCoreData
      headerType: clientCoreData (0xc001)
      headerLength: 234
      versionMajor: 12
      versionMinor: 8
      desktopWidth: 1024
      desktopHeight: 768
      colorDepth: 8 bits-per-pixel (bpp) (0xca01)
      SASSequence: 43523
      keyboardLayout: 1033
      clientBuild: 18363
      clientName: t3m0-virtual-ma
      keyboardType: IBM enhanced (101-key or 102-key) keyboard (4)
      keyboardSubType: 0
      keyboardFunctionKey: 12
```

Answer: t3m0-virtual-ma

Question 6:
When did the user last successfully log in via SSH, and who was it?

I searched in the auth.log file for "accepted"



```
Feb 25 11:43:54 ch... bin/systemctl r...
Feb 25 11:43:54 ch... ox(uid=1000)
Feb 25 11:43:17 ch...
Feb 25 11:43:18 ch... sbin/reboot
Feb 25 11:43:18 ch... ox(uid=1000)
Feb 25 11:43:18 ch...
Feb 25 11:43:18 ch... rrupted system
Feb 25 11:43:19 ch... ssion:cl (system
Feb 25 11:43:42 ch...
Feb 25 11:43:43 ch...
Feb 25 11:43:43 ch...
Feb 25 11:43:45 ch...
Feb 25 11:43:45 ch...
Feb 25 11:43:45 ch...
Feb 25 11:43:45 ch...
Feb 25 11:43:45 ch...
Feb 25 11:43:46 ch...
Feb 25 11:43:46 ch...
Feb 25 11:43:46 ch...
Feb 25 11:43:46 ch...
Feb 25 11:43:46 ch...
Feb 25 11:43:54 ch... Accepted password for mmox from 41.38.160.33 port 54464 ssh2
Feb 25 11:43:54 ch... pam_unix(sshd:session): session opened for user mmox(uid=1000) by (uid=0)
```

Answer: mmox:11:43:54

Question 7:
How many unsuccessful SSH connection attempts were made by the attacker?

I asked the ChatGPT to build me a python script to count the unsuccessful SSH connections

```
# Define the path to the auth.log file using a raw string
file_path = r'C:\Users\Bubble\Desktop\BruteForce\auth.log'
```

```
# Open and read the contents of the auth.log file
with open(file_path, 'r') as file:
    auth_log_content = file.readlines()
```

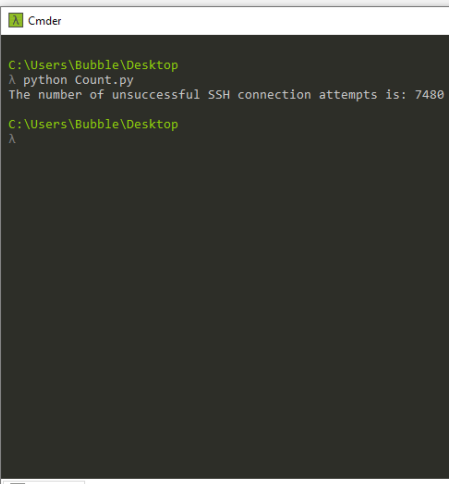
```
# Filter the log lines to count the number of unsuccessful SSH connection attempts
failed_ssh_attempts = [line for line in auth_log_content if 'Failed password' in line]
```

```
# Count the number of failed SSH attempts
num_failed_ssh_attempts = len(failed_ssh_attempts)
```

Print the number of failed SSH attempts

print(f"The number of unsuccessful SSH connection attempts is: {num_failed_ssh_attempts}")

```
pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=180.184.50.168 user=root
Failed password for root from 180.184.50.168 port 22340 ssh2
Connection closed by authenticating user root 180.184.50.168 port 22340 [preauth]
pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=180.184.50.168 user=root
Failed password for root from 180.184.50.168 port 22356 ssh2
Connection closed by authenticating user root 180.184.50.168 port 22356 [preauth]
pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=180.184.50.168 user=root
Failed password for root from 180.184.50.168 port 50540 ssh2
Connection closed by authenticating user root 180.184.50.168 port 50540 [preauth]
pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=180.184.50.168 user=root
Failed password for root from 180.184.50.168 port 50542 ssh2
Connection closed by authenticating user root 180.184.50.168 port 50542 [preauth]
pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=180.184.50.168 user=root
Failed password for root from 180.184.50.168 port 52976 ssh2
Connection closed by authenticating user root 180.184.50.168 port 52976 [preauth]
pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=180.184.50.168 user=root
Failed password for root from 180.184.50.168 port 52984 ssh2
Connection closed by authenticating user root 180.184.50.168 port 52984 [preauth]
pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=180.184.50.168 user=root
Failed password for root from 180.184.50.168 port 52996 ssh2
Connection closed by authenticating user root 180.184.50.168 port 52996 [preauth]
pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=180.184.50.168 user=root
Failed password for root from 180.184.50.168 port 59178 ssh2
Connection closed by authenticating user root 180.184.50.168 port 59178 [preauth]
pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=180.184.50.168 user=root
Failed password for root from 180.184.50.168 port 59190 ssh2
Connection closed by authenticating user root 180.184.50.168 port 59190 [preauth]
pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=180.184.50.168 user=root
Failed password for root from 180.184.50.168 port 48372 ssh2
Connection closed by authenticating user root 180.184.50.168 port 48372 [preauth]
pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=180.184.50.168 user=root
pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=218.92.0.97 user=root
Failed password for root from 180.184.50.168 port 48382 ssh2
Failed password for root from 218.92.0.97 port 56877 ssh2
Connection closed by authenticating user root 180.184.50.168 port 48382 [preauth]
pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=180.184.50.168 user=root
Failed password for root from 218.92.0.97 port 56877 ssh2
Failed password for root from 180.184.50.168 port 53932 ssh2
Connection closed by authenticating user root 180.184.50.168 port 53932 [preauth]
```



```
C:\Users\Bubble\Desktop
λ python Count.py
The number of unsuccessful SSH connection attempts is: 7480

C:\Users\Bubble\Desktop
λ
```

Answer: 7480

Question 8:

What technique is used to gain access?

I asked ChatGPT

For the technique used to gain access, you can refer to the MITRE ATT&CK framework. Common techniques used for brute-force attacks are often listed under Initial Access or Credential Access tactics.

Based on the provided information, if the attacker `mmox` used a brute-force attack to gain access via SSH, the technique might be:

- **T1110: Brute Force**

The answer format requires the MitreID, which would be:

plaintext

Copy code

T1110

So, the answer for the technique used to gain access would be `T1110`.

Answer: T1110