Lab - Isolate Compromised Host Using 5-Tuple

Objectives

In this lab, you will review logs that were gathered during the exploitation of a documented vulnerability to determine the compromised hosts and file.

Part 1: Review Alerts in Sguil

Part 2: Pivot to Wireshark

Part 3: Pivot to Kibana

Background / Scenario

The 5-tuple is used by IT administrators to identify requirements for creating an operational and secure network environment. The components of the 5-tuple include a source IP address and port number, destination IP address and port number, and the protocol in use in the data payload. This is the protocol field of the IP packet header.

In this lab, you will also review the logs to identify the compromised hosts and the content of the compromised file.

Required Resources

- Security Onion virtual machine

Instructions

After the attack, the users no longer have access to the file named **confidential.txt**. Now you will review the logs to determine how the file was compromised.

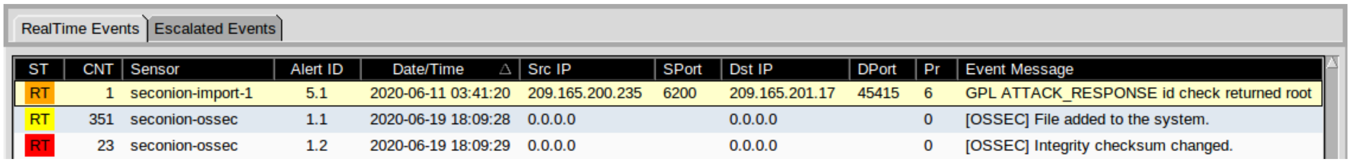
Note: If this was a production network, it is recommended that **analyst** and **root** users change their passwords and comply with the current security policy.

Part 1: Review Aerts in Sguil

a. Launch the Security Onion VM and log in. Log in with the user **analyst** and password **cyberops**

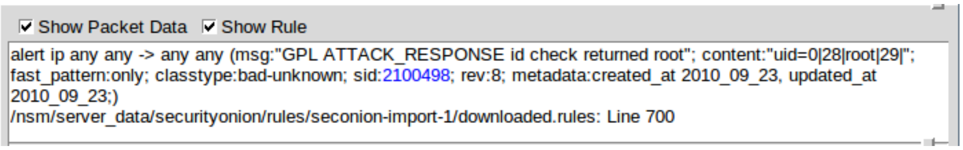
b. Open **Sguil** and log in. Click **Select All** to select the interfaces and then **Start SGUIL**.

c. Review the events listed in the Event Message column. One of these messages is **GPL ATTACK_RESPONSE id check returned root**. This message indicates that root access may have been gained during an attack. The host at 209.165.200.235 returned root access to 209.165.201.17. The alert ID **5.1** is used as an example in this lab.



ST	CNT	Sensor	Alert ID	Date/Time △	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	1	seconion-import-1	5.1	2020-06-11 03:41:20	209.165.200.235	6200	209.165.201.17	45415	6	GPL ATTACK_RESPONSE id check returned root
RT	351	seconion-ossec	1.1	2020-06-19 18:09:28	0.0.0.0		0.0.0.0		0	[OSSEC] File added to the system.
RT	23	seconion-ossec	1.2	2020-06-19 18:09:29	0.0.0.0		0.0.0.0		0	[OSSEC] Integrity checksum changed.

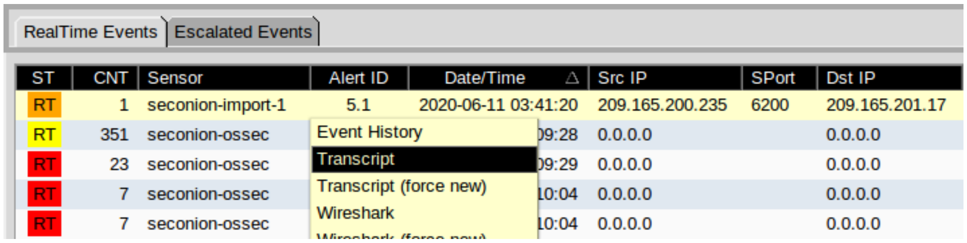
d. Select the **Show Packet Data** and **Show Rule** checkboxes to view each alert in more detail.



☑ Show Packet Data ☑ Show Rule

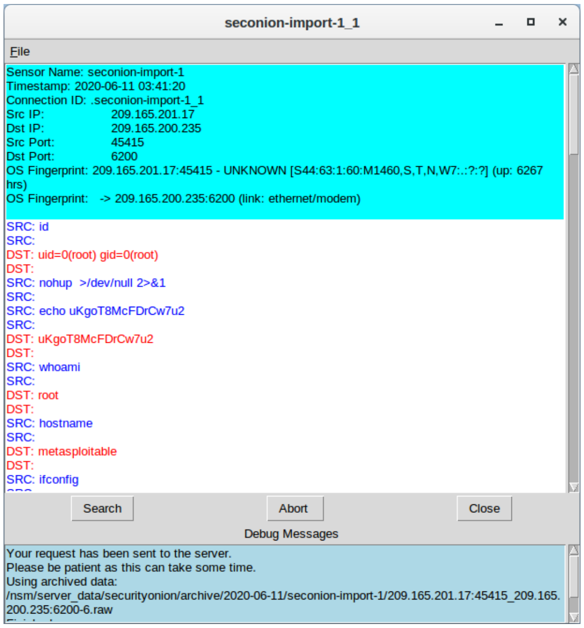
```
alert ip any any -> any any (msg:"GPL ATTACK_RESPONSE id check returned root"; content:"uid=0|28|root|29|";
fast_pattern:only; classtype:bad-unknown; sid:2100498; rev:8; metadata:created_at 2010_09_23, updated_at
2010_09_23;)
/nsm/server_data/securityonion/rules/seconion-import-1/downloaded.rules: Line 700
```

e. Right-click the alert ID 5.1 and select **Transcript**.



ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP
RT	1	seconion-import-1	5.1	2020-06-11 03:41:20	209.165.200.235	6200	209.165.201.17
RT	351	seconion-ossec		09:28	0.0.0.0		0.0.0.0
RT	23	seconion-ossec		09:29	0.0.0.0		0.0.0.0
RT	7	seconion-ossec		10:04	0.0.0.0		0.0.0.0
RT	7	seconion-ossec		10:04	0.0.0.0		0.0.0.0

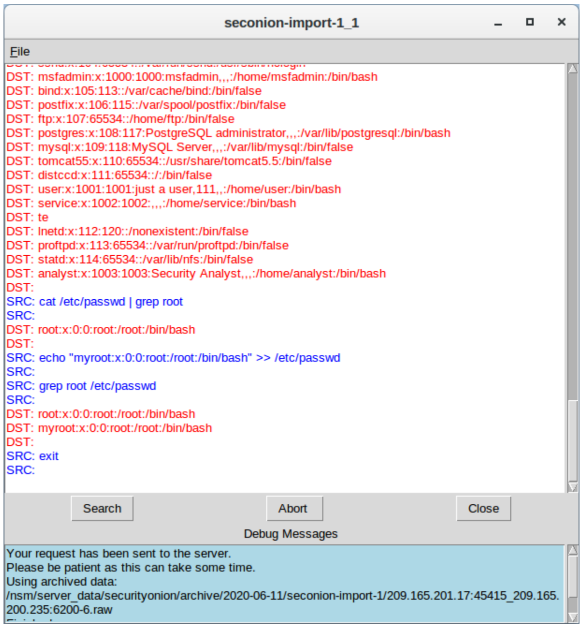
f. Review the transcripts for the alert. The transcript displays the transactions between the threat actor source (SRC) and the target (DST) during the attack. The threat actor is executing Linux commands on the target.



```
Sensor Name: seconion-import-1
Timestamp: 2020-06-11 03:41:20
Connection ID: .seconion-import-1_1
Src IP:             209.165.201.17
Dst IP:             209.165.200.235
Src Port:           45415
Dst Port:           6200
OS Fingerprint: 209.165.201.17:45415 - UNKNOWN [S44:63:1:60:M1460,S,T,N,W7:.:?:?] (up: 6267
hrs)
OS Fingerprint:   -> 209.165.200.235:6200 (link: ethernet/modem)

SRC: id
SRC:
DST: uid=0(root) gid=0(root)
DST:
SRC: nohup  >/dev/null 2>&1
SRC:
SRC: echo uKgoT8McFDrCw7u2
SRC:
DST: uKgoT8McFDrCw7u2
DST:
SRC: whoami
SRC:
DST: root
DST:
SRC: hostname
SRC:
DST: metasploitable
DST:
SRC: ifconfig
```

```
Your request has been sent to the server.
Please be patient as this can take some time.
Using archived data:
/nsm/server_data/securityonion/archive/2020-06-11/seconion-import-1/209.165.201.17:45415_209.165.
200.235:6200-6.raw
```



```
DST: msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
DST: bind:x:105:113::/var/cache/bind:/bin/false
DST: postfix:x:106:115::/var/spool/postfix:/bin/false
DST: ftp:x:107:65534::/home/ftp:/bin/false
DST: postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
DST: mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
DST: tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
DST: distccd:x:111:65534::/:/bin/false
DST: user:x:1001:1001:just a user,111,,:/home/user:/bin/bash
DST: service:x:1002:1002:,,,:/home/service:/bin/bash
DST: te
DST: lnetd:x:112:120::/nonexistent:/bin/false
DST: proftpd:x:113:65534::/var/run/proftpd:/bin/false
DST: statd:x:114:65534::/var/lib/nfs:/bin/false
DST: analyst:x:1003:1003:Security Analyst,,,:/home/analyst:/bin/bash
DST:
SRC: cat /etc/passwd | grep root
SRC:
DST: root:x:0:0:root:/root:/bin/bash
DST:
SRC: echo "myroot:x:0:0:root:/root:/bin/bash" >> /etc/passwd
SRC:
SRC: grep root /etc/passwd
SRC:
DST: root:x:0:0:root:/root:/bin/bash
DST: myroot:x:0:0:root:/root:/bin/bash
DST:
SRC: exit
SRC:
```

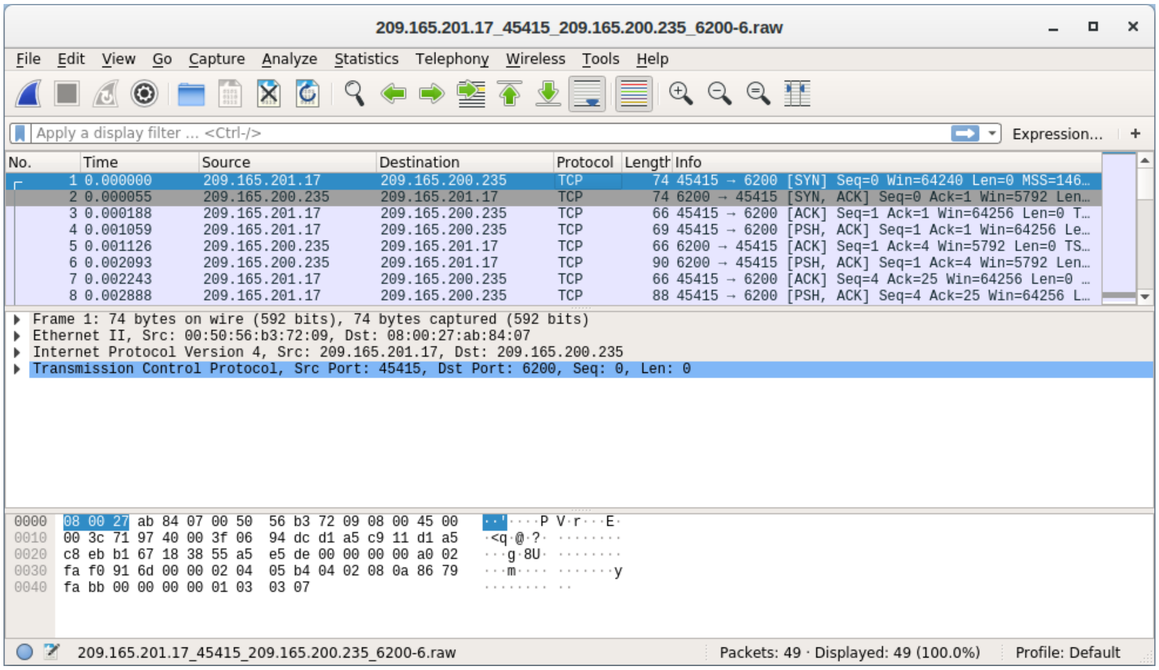
```
Your request has been sent to the server.
Please be patient as this can take some time.
Using archived data:
/nsm/server_data/securityonion/archive/2020-06-11/seconion-import-1/209.165.201.17:45415_209.165.
200.235:6200-6.raw
```

What kind of transactions occurred between the client and the server in this attack?

hacker added his credntials to /etc/shadow and shell to /etc/passwd

Part 2: Pivot to Wireshark

a. Select the alert that provided you with the transcript from the previous step. Right-click the alert ID 5.1 and select **Wireshark**. The Wireshark main window displays three views of a packet.


209.165.201.17_45415_209.165.200.235_6200-6.raw
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
+

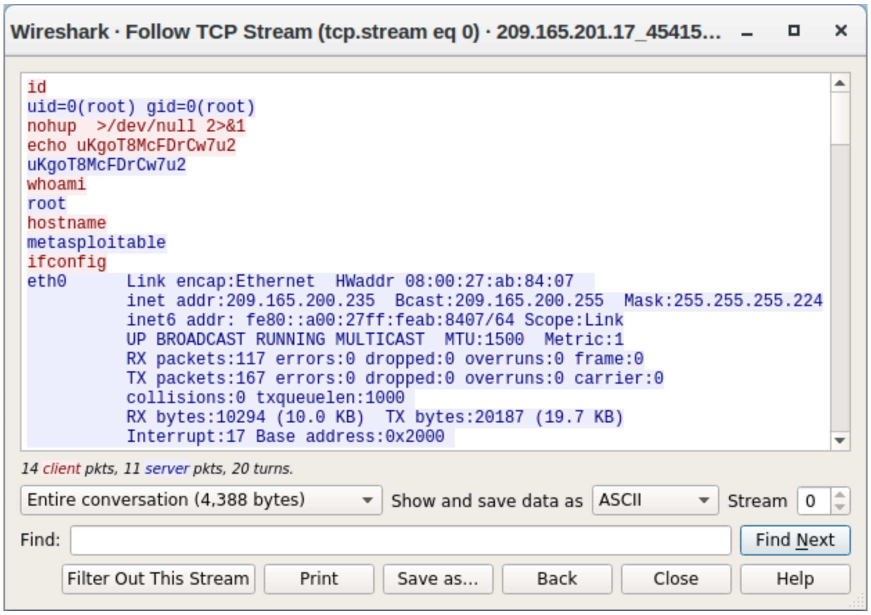
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	209.165.201.17	209.165.200.235	TCP	74	45415 → 6200 [SYN] Seq=0 Win=64240 Len=0 MSS=146…
2	0.000055	209.165.200.235	209.165.201.17	TCP	74	6200 → 45415 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len…
3	0.000188	209.165.201.17	209.165.200.235	TCP	66	45415 → 6200 [ACK] Seq=1 Ack=1 Win=64256 Len=0 T…
4	0.001059	209.165.201.17	209.165.200.235	TCP	69	45415 → 6200 [PSH, ACK] Seq=1 Ack=1 Win=64256 Le…
5	0.001126	209.165.200.235	209.165.201.17	TCP	66	6200 → 45415 [ACK] Seq=1 Ack=4 Win=5792 Len=0 TS…
6	0.002093	209.165.200.235	209.165.201.17	TCP	90	6200 → 45415 [PSH, ACK] Seq=1 Ack=4 Win=5792 Len…
7	0.002243	209.165.201.17	209.165.200.235	TCP	66	45415 → 6200 [ACK] Seq=4 Ack=25 Win=64256 Len=0 …
8	0.002888	209.165.201.17	209.165.200.235	TCP	88	45415 → 6200 [PSH, ACK] Seq=4 Ack=25 Win=64256 L…

```
▸ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
▸ Ethernet II, Src: 00:50:56:b3:72:09, Dst: 08:00:27:ab:84:07
▸ Internet Protocol Version 4, Src: 209.165.201.17, Dst: 209.165.200.235
▸ Transmission Control Protocol, Src Port: 45415, Dst Port: 6200, Seq: 0, Len: 0
```

```
0000  08 00 27 ab 84 07 00 50  56 b3 72 09 08 00 45 00   ··'····P V·r···E·
0010  00 3c 71 97 40 00 3f 06  94 dc d1 a5 c9 11 d1 a5   ·<q·@·?· ········
0020  c8 eb b1 67 18 38 55 a5  e5 de 00 00 00 00 a0 02   ···g·8U· ········
0030  fa f0 91 6d 00 00 02 04  05 b4 04 02 08 0a 86 79   ···m···· ·······y
0040  fa bb 00 00 00 00 01 03  03 07                     ········ ··
```

209.165.201.17_45415_209.165.200.235_6200-6.raw   Packets: 49 · Displayed: 49 (100.0%)   Profile: Default

b. To view all packets that are assembled in a TCP conversation, right-click any packet and select **Follow** > **TCP Stream**.


Wireshark · Follow TCP Stream (tcp.stream eq 0) · 209.165.201.17_45415…

```
id
uid=0(root) gid=0(root)
nohup  >/dev/null 2>&1
echo uKgoT8McFDrCw7u2
uKgoT8McFDrCw7u2
whoami
root
hostname
metasploitable
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:ab:84:07
          inet addr:209.165.200.235  Bcast:209.165.200.255  Mask:255.255.255.224
          inet6 addr: fe80::a00:27ff:feab:8407/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:117 errors:0 dropped:0 overruns:0 frame:0
          TX packets:167 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10294 (10.0 KB)  TX bytes:20187 (19.7 KB)
          Interrupt:17 Base address:0x2000
```

14 client pkts, 11 server pkts, 20 turns.

Entire conversation (4,388 bytes)   Show and save data as: ASCII   Stream: 0

Find:

What did you observe? What do the text colors red and blue indicate?

red - hacker, blue machine

The attacker issues the **whoami** command on the target. What does this show about the attacker role on the target computer?

attacker has root on device

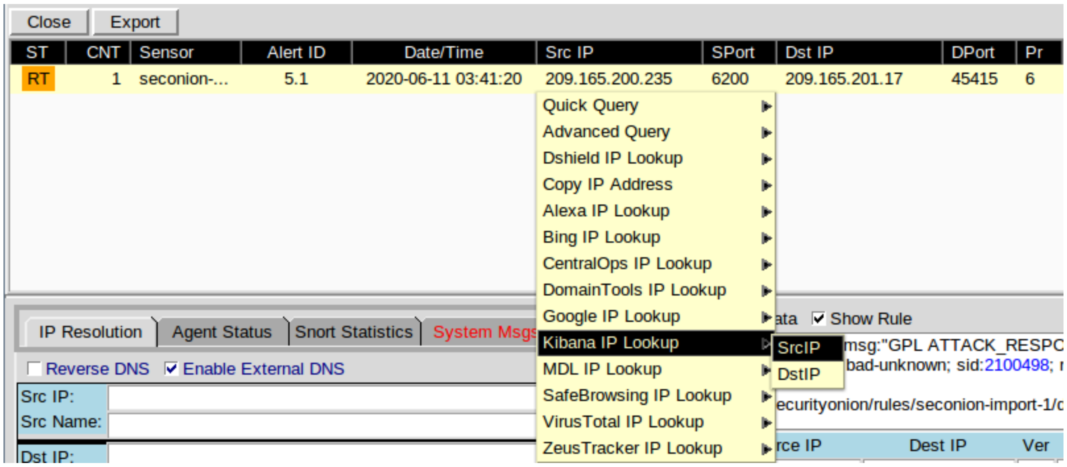
Scroll through the TCP stream. What kind of data has the threat actor been reading?

shadow and passwd file and network config

c. Exit the TCP stream window. Close **Wireshark** when you are done reviewing the information provided.

Part 3: Pivot to Kibana

a. Return to Sguil. Right-click either the source or destination IP for the alert ID 5.1 and select **Kibana IP Lookup** > **SrcIP**. Enter username **analyst** and password **cyberops** if prompted by Kibana.

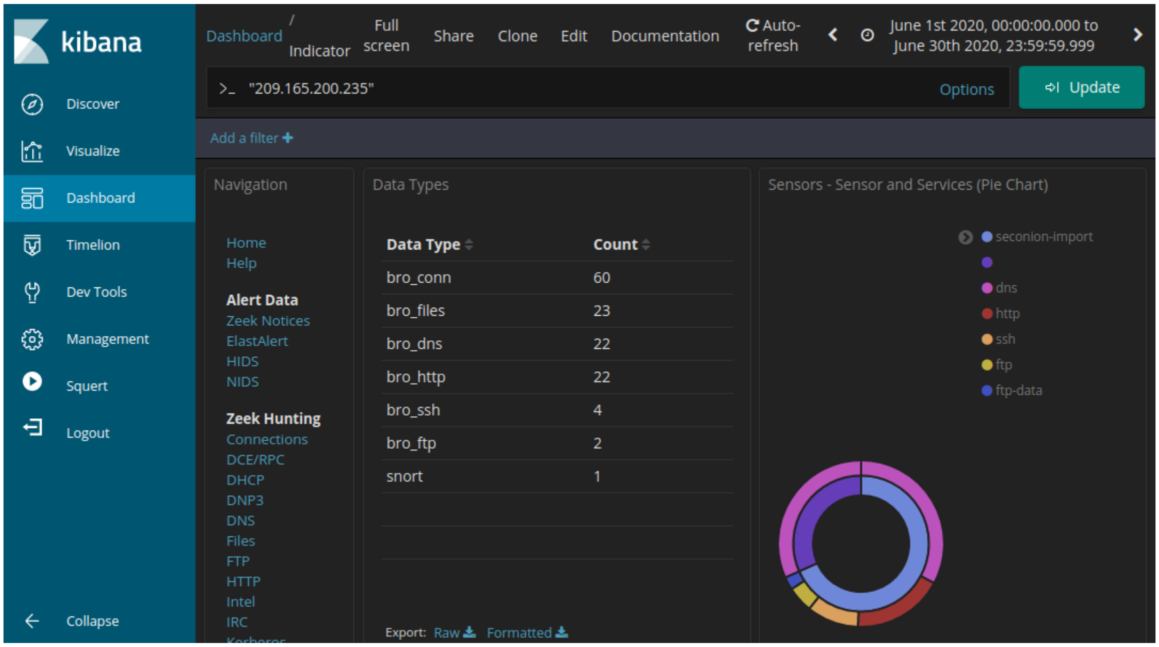


ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr
RT	1	seconion-...	5.1	2020-06-11 03:41:20	209.165.200.235	6200	209.165.201.17	45415	6

Note: If you received the message "Your connection is not private", click **ADVANCED > Proceed to localhost (unsafe)** to continue.

b. If the time range is the last 24 hours, change it to June 2020 so June 11 is included in the time range. Use the **Absolute** tab to change the time range.

c. In the displayed results, there is a list of different data types. You were told that the file **confidential.txt** is no longer accessible. In the Sensors - Sensors and Services (Pie Chart), ftp and ftp-data are present in the list, as shown in the figure. We will determine if FTP was used to steal the file.

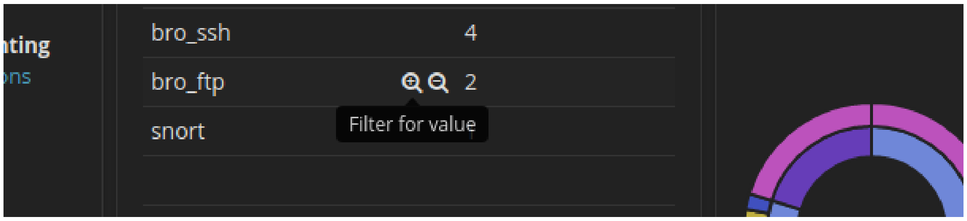


>_ "209.165.200.235"

Data Types

Data Type	Count
bro_conn	60
bro_files	23
bro_dns	22
bro_http	22
bro_ssh	4
bro_ftp	2
snort	1

d. Let's filter for **bro_ftp**. Hover over the empty space next to the count of bro_ftp data types. Select **+** to filter for only FTP related traffic as shown in the figure.



bro_ssh	4
bro_ftp	2
snort	

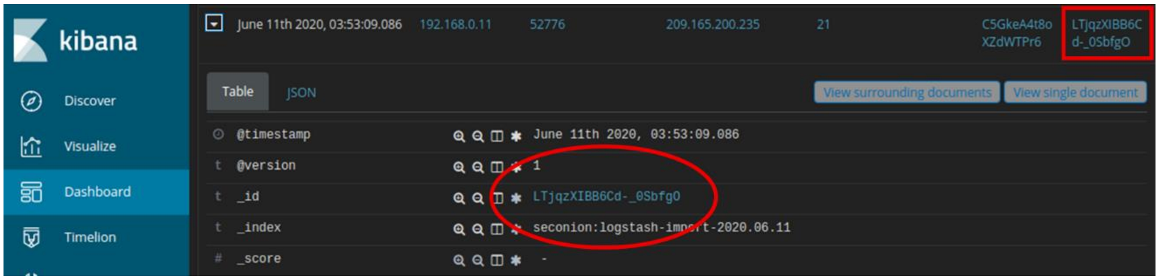
e. Scroll down to the **All Logs** section. There are two entries listed.

What are the source and destination IP addresses and port numbers for the FTP traffic?

Src - Dst
192.68.0.11 209.165.200.235
52776 21

f. Expand and review both log entries. In one of these entries, the ftp_argument has an entry of ftp://209.165.200.235/./confidential.txt. Also review the message in the log entry to learn more about this event.

g. Within the same log entry, scroll up back to the alert **_id** field and click the link.



```
June 11th 2020, 03:53:09.086   192.168.0.11   52776   209.165.200.235   21   C5GkeA4t8o XZdWTPr6   LTjqzXIBB6C d-_0SbfgO

@timestamp   June 11th 2020, 03:53:09.086
@version     1
_id          LTjqzXIBB6Cd-_0SbfgO
_index       seconion:logstash-import-2020.06.11
_score       -
```

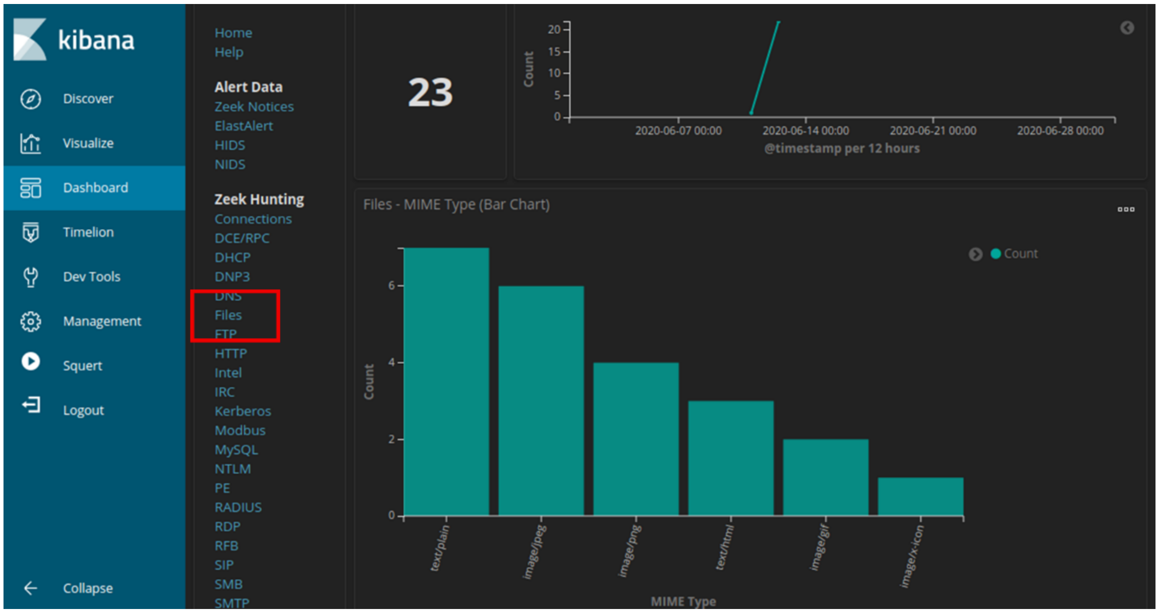
h. Review the transcript for the transactions between the attacker and the target. If desired, you can download the pcap and review the traffic using Wireshark.

What are the user credentials to access the FTP site?

analyst cyberops

i. Now that you have verified that the attacker has used FTP to copy the content of the file confidential.txt and then deleted it from the target. So what is the content of the file? Remember one of the services listed in the pie chart is ftp_data.

j. Navigate to the top of the dashboard. Select **Files** under the Zeek Hunting heading in the left panel, as shown in the figure. This will allow you to review the types of the files that were logged.

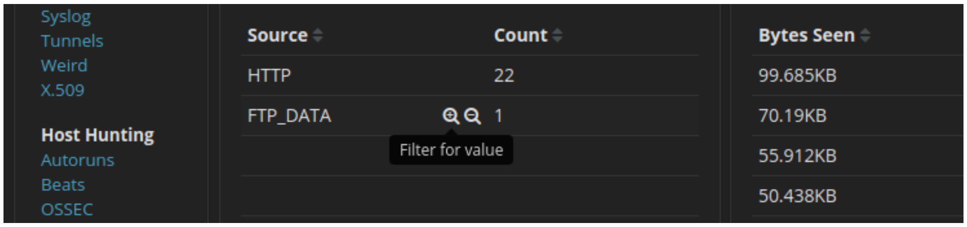


What are the different types of files? Look at the MIME Type section of the screen.

Scroll to the **Files - Source** heading. What are the file sources listed?

http and ftp data

k. Filter for **FTP_DATA** by hovering over the empty space next to the Count for FTP_DATA and click **+**.



Source	Count	Bytes Seen
HTTP	22	99.685KB
FTP_DATA	1	70.19KB
		55.912KB
		50.438KB

l. Scroll down to review the filtered results.

What is the MIME type, source and destination IP address associated with the transfer of the FTP data? When did this transfer occur?

plain/text 2020-06-11 at 00:00

m. In the File logs, expand the entry associated with FTP data. Click the link associated with alert **_id**.

What is the text content of the file that was transferred using FTP?

info about last security breach

With all the information has gathered so far, what is your recommendation for stopping further unauthorized access?

change password, remove user myroot disable ftp