

Report Bonus 2: Isolare un Host Compromesso Usando la 5-Tupla

A cura di Iris Canole, Federico Giannini, Daniele Castello, Luca Pani, Rosario Papa, Yari Olmi, Alessandro Salerno

Obiettivi

In questo laboratorio, esaminerai i log raccolti durante lo sfruttamento di una vulnerabilità documentata per determinare gli host e il file compromessi.

- PARTE 1: Esaminare gli Alert in Sguil.
- PARTE 2: Passare a Wireshark (Pivoting).
- PARTE 3: Passare a Kibana (Pivoting).

Dopo l'attacco, gli utenti non hanno più accesso al file chiamato `confidential.txt`. Ora esamineremo i log per determinare come il file è stato compromesso.

Parte 1: Esaminare gli Alert in Sguil

1. Abbiamo avviato la VM Security Onion ed effettuato il login.

Credenziali:

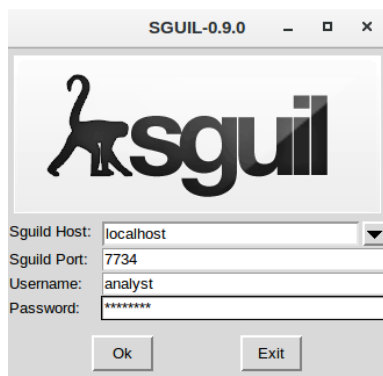
- UTENTE: `analyst`
- PASSWORD: `cyberops`



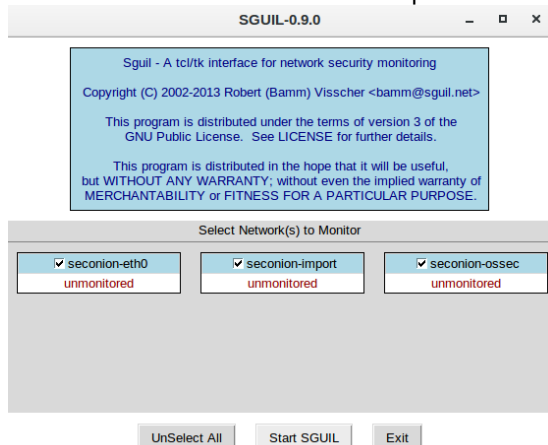
2. Abbiamo aperto Sguil e poi effettuato il login.

Credenziali:

- USERNAME: analyst
- PASSWORD: cyberops



Abbiamo cliccato su Select All per selezionare le interfacce e poi su Start SGUIL.



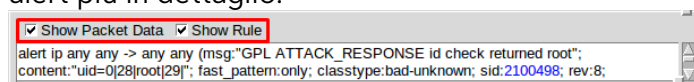
3. Abbiamo esaminato gli eventi elencati nella colonna Event Message.

Uno di questi messaggi è GPL ATTACK_RESPONSE id check returned root, ed indica che potrebbe essere stato ottenuto l'accesso root durante un attacco.

L'host 209.165.200.235 ha restituito l'accesso root a 209.165.201.17.

ST	CNT	Sensor	Alert ID	DateTime	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	17	seconion...	5.234	2019-07-19 18:53:12	172.16.4.205	49249	185.243.115.84	80	6	ET POLICY Data POST to a...
RT	114	seconion...	5.251	2019-07-19 18:57:23	172.16.4.205	49255	31.7.62.214	443	6	ET POLICY HTTP traffic on ...
RT	2	seconion...	5.365	2020-02-21 00:53:55	172.17.8.174	62362	172.17.8.8	53	17	ET POLICY DNS Update Fro...
RT	13	seconion...	5.366	2020-02-21 00:55:07	49.51.172.56	80	172.17.8.174	49731	6	ET CURRENT_EVENTS Lik...
RT	13	seconion...	5.379	2020-02-21 00:55:07	49.51.172.56	80	172.17.8.174	49731	6	ET CURRENT_EVENTS Win...
RT	13	seconion...	5.392	2020-02-21 00:55:07	49.51.172.56	80	172.17.8.174	49731	6	ET POLICY PE EXE or DLL ...
RT	4	seconion...	5.406	2020-02-21 01:11:48	91.211.88.122	443	172.17.8.174	49760	6	ET TROJAN ABUSE.CH SS...
RT	1	seconion...	5.1	2020-06-11 03:41:20	209.165.200.235	6200	209.165.201.17	45415	6	GPL ATTACK_RESPONSE I...

4. Abbiamo selezionato le caselle di controllo Show Packet Data e Show Rule per visualizzare ogni alert più in dettaglio.



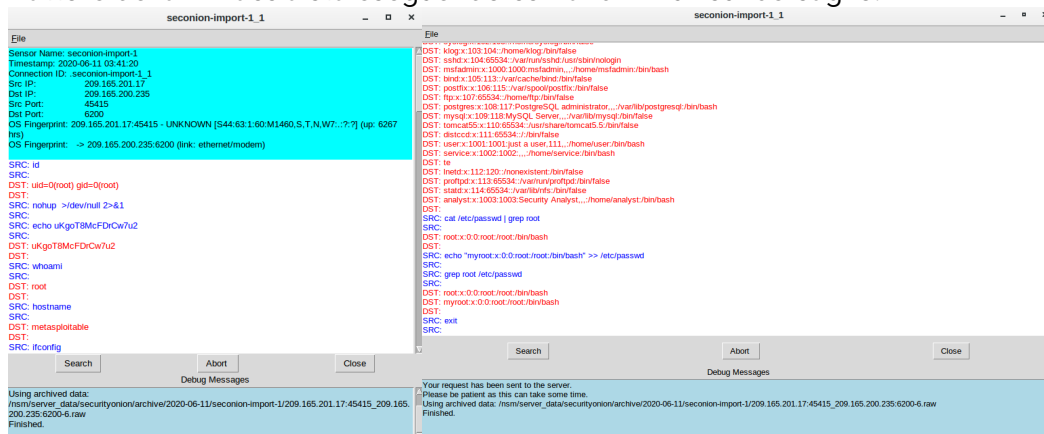
5. Abbiamo cliccato con il pulsante destro sull'ID dell'alert 5.1 e selezionato Transcript.

RT	1	seconion-...	5.1	2020-06-11 03:41:20
RT	351	seconion-...	Event History	8:09:28
RT	23	seconion-...	Transcript	8:09:28
RT	7	seconion-...	Transcript (force new)	8:10:04
RT	7	seconion-...	Wireshark	8:10:04
RT	2	seconion-...	Wireshark (force new)	8:14:41
RT	1	seconion-...	NetworkMiner	8:18:41
			NetworkMiner (force new)	
			Bro	
			Bro (force new)	

6. Abbiamo esaminato le trascrizioni per l'alert.

Mostrano le transazioni tra l'attore della minaccia (SRC) e il bersaglio (DST) durante l'attacco.

L'attore della minaccia sta eseguendo comandi Linux sul bersaglio.



DOMANDA: Che tipo di transazioni si sono verificate tra il client e il server in questo attacco?

In questo attacco, l'attaccante ha compromesso il server, ha confermato di essere root, ha rubato gli hash delle password e ha creato un account amministratore nascosto **myroot** per garantirsi l'accesso futuro.

Cronologia delle transazioni

Reconnaissance

- **Comando:** `id`
Risposta: `uid=0(root) gid=0(root)`
- **Comando:** `whoami`
Risposta: `root`

L'attaccante verifica i suoi privilegi.

La risposta conferma che ha accesso root sulla macchina.

- **Comando:** `hostname`

Risposta: metasploitable

L'attaccante identifica il nome della macchina.

—

- **Comando:** ifconfig

L'attaccante controlla la configurazione di rete.

Information Gathering

- **Comando:** cat /etc/passwd

L'attaccante legge il file che contiene la lista degli utenti del sistema

—

- **Comando:** cat /etc/shadow

L'attaccante legge il file che contiene gli hash delle password degli utenti.

Con questi hash, l'attaccante può tentare di decifrare le password offline.

Creating a Backdoor User

- **Comando:** echo "myroot:x:0:0:root:/root:/bin/bash" >> /etc/passwd
- **Comando:** echo "myroot::14747:0:99999:7:::" >> /etc/shadow

L'attaccante ha modificato il sistema per assicurarsi di poter rientrare anche se la vulnerabilità originale venisse corretta.

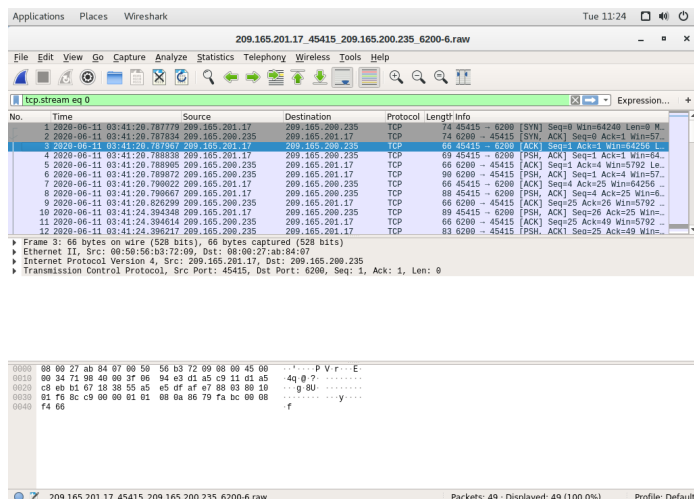
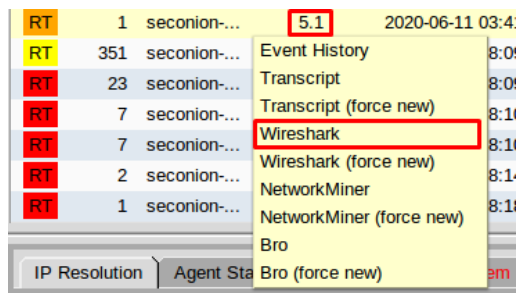
Verifica della persistenza

- **Comando:** grep root /etc/passwd
- **Risposta:** gli ha mostrato sia l'utente root originale sia il nuovo utente myroot.

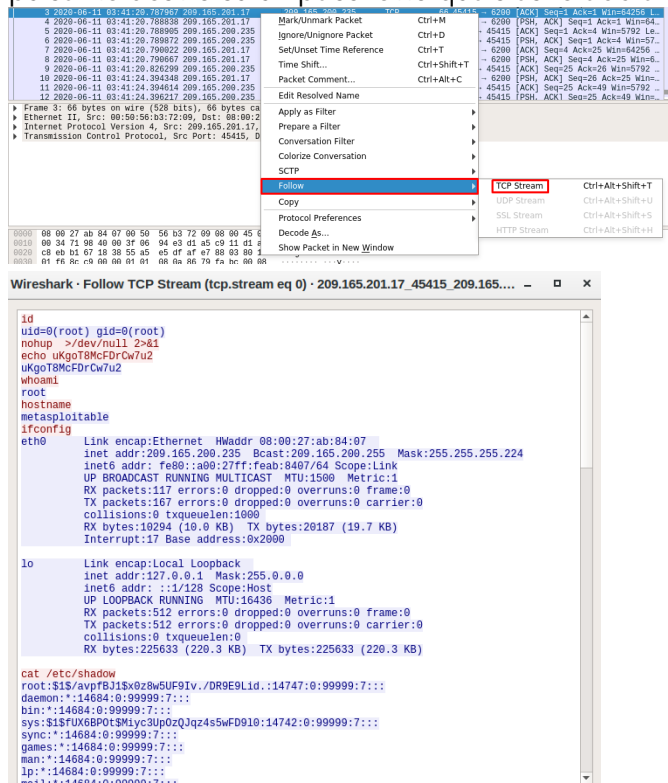
L'attaccante ha confermato che la modifica al file è andata a buon fine.

Parte 2: Passare a Wireshark

1. Abbiamo selezionato l'alert che ci ha fornito la trascrizione nel passo precedente. Abbiamo cliccato con il pulsante destro sull'ID dell'alert 5.1 e selezionato **Wireshark**.



- Per visualizzare tutti i pacchetti assemblati in una conversazione TCP, Abbiamo cliccato con il pulsante destro su un pacchetto qualsiasi e abbiamo selezionato Follow > TCP Stream.



DOMANDA: Cosa hai osservato? Cosa indicano i colori del testo rosso e blu?

RISPOSTA: Si è aperta una finestra che mostra l'interazione tra l'attaccante e il server Target. In **rosso** vediamo ciò che scrive l'attaccante, quindi i comandi, mentre in **blu** è ciò che il server nella console risponde.

DOMANDA: L'attaccante esegue il comando whoami sul bersaglio. Cosa rivela questo sul ruolo dell'attaccante sul computer bersaglio?

RISPOSTA: Questo comando rivela all'attaccante che ha accesso root sul Target.

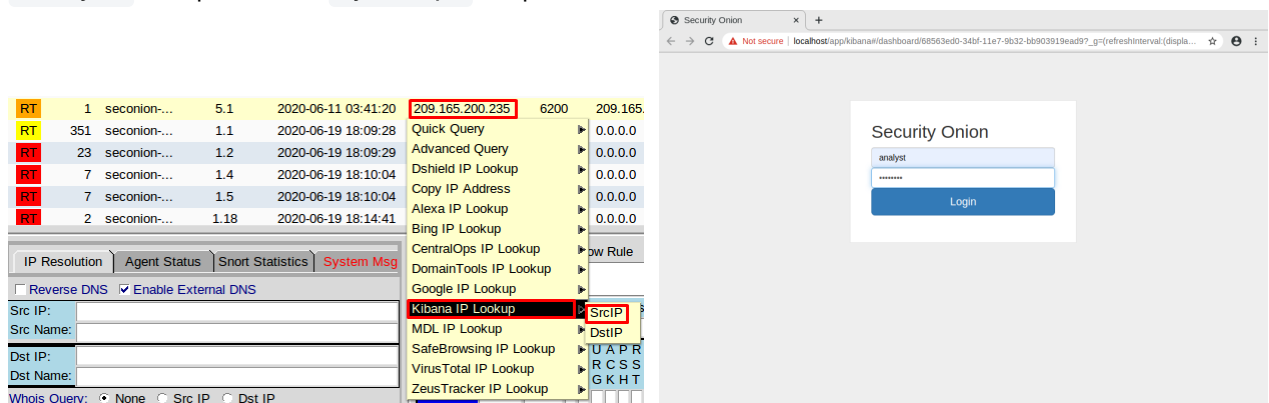
DOMANDA: Scorri il flusso TCP. Che tipo di dati ha letto l'attore della minaccia?

RISPOSTA: L'attaccante ha letto dati come, la configurazione di rete, la lista degli utenti del sistema e il file che contiene gli hash delle password degli utenti

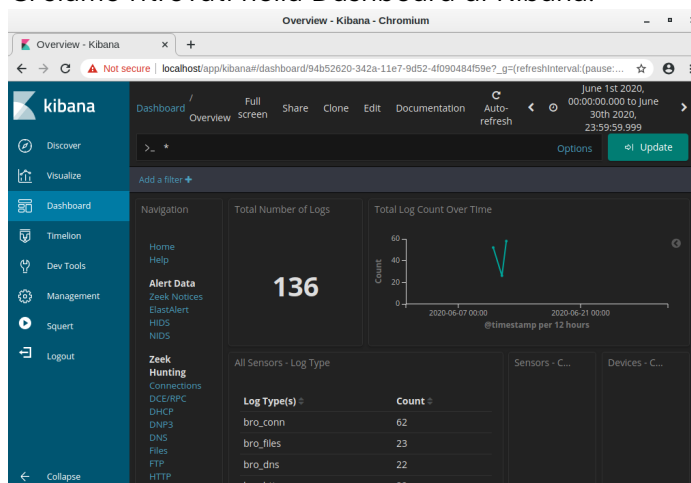
3. Siamo usciti dalla finestra del flusso TCP. E infine abbiamo chiuso Wireshark quando abbiamo finito di esaminare le informazioni fornite.

Parte 3: Passare a Kibana

1. Su Sguil abbiamo fatto click con il pulsante destro sull'indirizzo IP di origine per l'ID dell>alert 5.1 e abbiamo selezionato Kibana IP Lookup > SrcIP. Abbiamo poi inserito il nome utente `analyst` e la password `cyberops` in quanto richiesto da Kibana.



2. Ci siamo ritrovati nella Dashboard di Kibana.



E abbiamo cambiato l'intervallo di tempo in modo tale che l'11 giugno fosse incluso nell'intervallo.

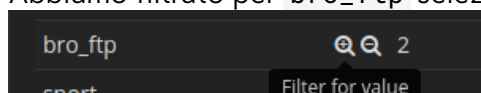
June 1st 2020,
00:00:00.000 to June
30th 2020,
23:59:59.999

3. Nella schermata è presente un elenco di diverso tipo di dati.

All Sensors - Log Type

Log Type(s) ▾	Count ▾
bro_conn	62
bro_files	23
bro_dns	22
bro_http	22
bro_ssh	4
bro_ftp	2
snort	1

4. Abbiamo filtrato per `bro_ftp` selezionando il `+` nel Filter for value.



5. Siamo andati fino in fondo alla Dashboard e abbiamo trovato `All Logs`, le voci elencate erano quelle nello Screenshot sotto:

All Logs

1-2 of 2 < >

Time ▾	source_ip	source_port	destination_ip	destination_port	_id
▸ June 11th 2020, 03:53:09.086	192.168.0.11	52776	209.165.200.235	21	LDjqzXIB B6Cd-0 SbfgO
▸ June 11th 2020, 03:53:09.086	192.168.0.11	52776	209.165.200.235	21	LTjqzXIB B6Cd-0 SbfgO

1-2 of 2 < >

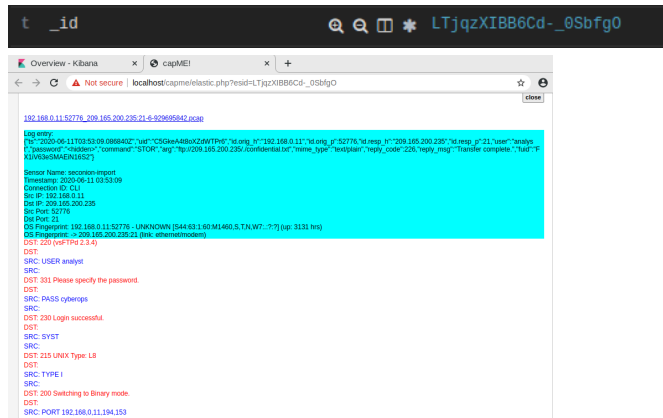
DOMANDA: Quali sono gli indirizzi IP e i numeri di porta di origine e destinazione per il traffico FTP?

RISPOSTA: L'indirizzo IP e la porta di origine è `192.168.0.11` e `52776`, mentre quello di destinazione è `209.165.200.235` e `21`.

6. Espandendo la seconda voce di Log, nel campo `ftp_argument` abbiamo trovato la voce `ftp://209.165.200.235/./confidential.txt`

t ftp_argument 🔍 209.165.200.235/./confidential.txt

7. Abbiamo aperto poi il campo `_id` schiacciando sul link e abbiamo trovato le transazioni tra l'attaccante e il server Target.



DOMANDA: Quali sono le credenziali utente per accedere al sito FTP?

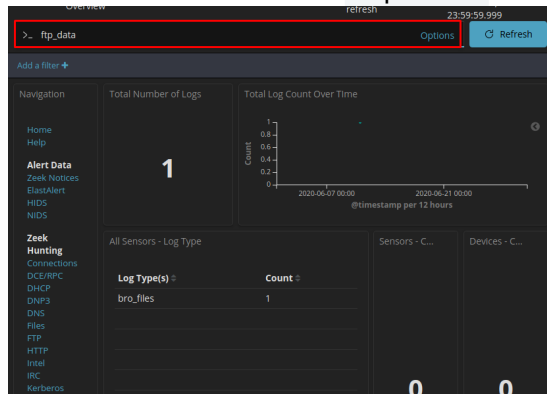
RISPOSTA: Le credenziali utente per accedere al sito FTP sono:

- **USER:** analyst
- **PASS:** cyberops

8. Ora che sappiamo che l'attaccante ha usato FTP per copiare il contenuto del file `confidential.txt` e poi cancellarlo dal bersaglio.

DOMANDA: Qual è il contenuto del file? Ricorda che uno dei servizi elencati nel grafico a torta è `ftp_data`.

Abbiamo messo nel filtro `ftp_data` e il risultato è stato 1.



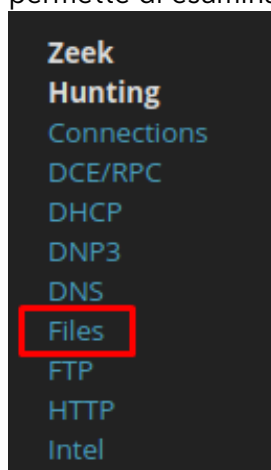
RISPOSTA: Il contenuto del file è stato il seguente:

```
192.168.0.11:49817 -> 209.165.200.235:20-6-2105080620.pcap

Log entry
[TS: 2020-06-11T03:53:09.088773Z; "uid": "FXJIV3eSMAEN1852"; "tx_hosts": ["192.168.0.11"]; "rx_hosts": ["209.165.200.235"]; "conn_uids": ["C2v8MWV6Xg4bb51"]; "source": "FTP_DATA"; "depth": 0; "analyzers": ["SHA1", "MD5"]; "name_type": "text/plain"; "duration": 0.0; "is_orig": false; "seen_bytes": 102; "missing_bytes": 0; "overflow_bytes": 0; "timedout": false; "msg5": "e7bc20c2069566365379c91294d530b"; "sha1": "f7f54ace094261018e63a10824ee11b530725"]
Sensor Name: seconion-import
Timestamp: 2020-06-11 03:53:09
Connection ID: CLI
Src IP: 192.168.0.11
Dst IP: 209.165.200.235
Src Port: 49817
Dst Port: 20
OS Fingerprint: 209.165.200.235:20 - Linux 2.6 (newer: 1) (up: 1 hrs)
OS Fingerprint -> 192.168.0.11:49817 (distance 0, link: ethernet/modem)
SRC: CONFIDENTIAL DOCUMENT
SRC: DO NOT SHARE
SRC: This document contains information about the last security breach.
SRC:

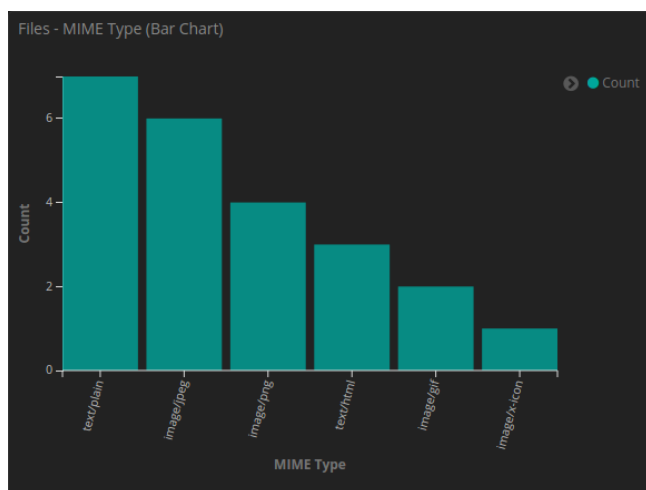
DEBUG: Using archived data: /usr/share/data/seconionion/archive/2020-06-11/seconion-import/192.168.0.11:49817_209.165.200.235:20-6-raw
QUERY: SELECT sid FROM sensor WHERE hostname=seconion-import AND agent_type=pcap LIMIT 1
CAPME: Processed transcript in 0.26 seconds: 0.08 0.10 0.00 0.08 0.00
192.168.0.11:49817 -> 209.165.200.235:20-6-2105080620.pcap
```


9. Abbiamo selezionato Files sotto l'intestazione Zeek Hunting nel pannello di sinistra. Questo permette di esaminare i tipi di file che sono stati registrati.



DOMANDA: Quali sono i diversi tipi di file? Guarda la sezione MIME Type dello schermo.

RISPOSTA: Sono presenti diversi tipi di file tra cui text/plain, image/jpeg, etc.



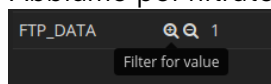
DOMANDA: Scorri fino all'intestazione Files - Source. Quali sono le sorgenti dei file elencate?

RISPOSTA: Le sorgenti dei file elencate sono HTTP e FTP_DATA.

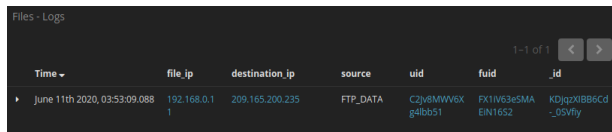
A screenshot of the 'Files - Source' table. The table has two columns: 'Source' and 'Count'. The data is as follows:

Source	Count
HTTP	22
FTP_DATA	1

10. Abbiamo poi filtrato per FTP_DATA cliccando sul + di Filter for value.



11. Abbiamo esaminato i risultati filtrati in fondo alla pagina.



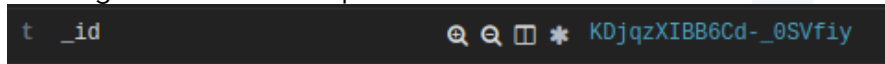
Time	file_ip	destination_ip	source	uid	fuid	_id
June 11th 2020, 03:53:09.088	192.168.0.1	209.165.200.235	FTP_DATA	C2jv8MWV6Xg4lbb51	FX1iV63eSMAEIN16S2	KDjqzXIBB6Cd-_0SVfiy

DOMANDA: Qual è il tipo MIME, l'indirizzo IP di origine e di destinazione associato al trasferimento dei dati FTP? Quando si è verificato questo trasferimento?

RISPOSTA:

- Il MIME type è `FTP_DATA`
- L'indirizzo IP di origine e di destinazione sono `192.168.0.11` e `209.165.200.235`
- Il trasferimento si è verificato l'11th June 2020 alle ore 03:53:09

12. Nei Log dei file abbiamo poi cliccato sul link associato a `_id`.



DOMANDA: Qual è il contenuto testuale del file trasferito tramite FTP?

RISPOSTA: Il contenuto del file è riportato nello Screenshot seguente:

[192.168.0.11:49817_209.165.200.235:20-6-1922484408.pcap](#)

```
Log entry:
{"ts":"2020-06-11T03:53:09.088773Z","fuid":"FX1iV63eSMAEIN16S2","tx_hosts":["192.168.0.11"],"rx_hosts":["209.165.200.235"],"conn_uids":["C2jv8MWV6Xg4lbb51"],"source":"FTP_DATA","depth":0,"analyzers":{"SHA1":"MD5"},"mime_type":"text/plain","duration":0.0,"is_orig":false,"seen_bytes":102,"missing_bytes":0,"overflow_bytes":0,"timedout":false,"md5":"e7bc9c20bfd5666365379c91294d536b","sha1":"7f54acee0342f6161f8e63a10824ee11b330725"}
```

```
Sensor Name: seconion-import
Timestamp: 2020-06-11 03:53:09
Connection ID: CLI
Src IP: 192.168.0.11
Dst IP: 209.165.200.235
Src Port: 49817
Dst Port: 20
OS Fingerprint: 209.165.200.235:20 - Linux 2.6 (newer, 1) (up: 1 hrs)
OS Fingerprint -> 192.168.0.11:49817 (distance 0, link: ethernet/modem)
```

```
SRC: CONFIDENTIAL DOCUMENT
SRC: DO NOT SHARE
SRC: This document contains information about the last security breach.
SRC:
```

```
DEBUG: Using archived data: /nsm/server_data/securityonion/archive/2020-06-11/seconion-import/192.168.0.11:49817_209.165.200.235:20-6.raw
QUERY: SELECT sid FROM sensor WHERE hostname='seconion-import' AND agent_type='pcap' LIMIT 1
CAPME: Processed transcript in 0.24 seconds: 0.08 0.09 0.00 0.00 0.07 0.00
```

[192.168.0.11:49817_209.165.200.235:20-6-1922484408.pcap](#)

DOMANDA: Con tutte le informazioni raccolte finora, qual è la tua raccomandazione per fermare ulteriori accessi non autorizzati?

RISPOSTA: È fondamentale scollegare dalla rete il sistema `192.168.0.11` per consentire l'indagine forense. È necessario effettuare un audit delle regole FTP restringendo l'invio di dati verso l'esterno e verificare l'eventuale presenza di attività di esfiltrazione analoghe. Applicare subito una regola di blocco sul firewall per l'IP `209.165.200.235`.

Conclusioni

Grazie a strumenti come Sguil, Wireshark e Kibana, siamo riusciti a ricostruire completamente l'evento di sicurezza.

L'analisi ha confermato che l'attaccante è riuscito a diventare `root` su una macchina non sicura, utilizzandola poi per trasferire all'esterno il documento `confidential.txt` mediante una connessione FTP.