Interpretare Dati HTTP e DNS per Isolare l'Attore della Minaccia

Data: 24 Febbraio 2026
Laboratorio: Security Onion - Analisi Forense di Rete

Introduzione

Questo documento presenta l'analisi forense di un attacco informatico rilevato tramite Security Onion. L'esercizio si articola in due parti principali: l'identificazione di un attacco SQL injection tramite traffico HTTP e l'analisi di un'esfiltrazione dati tramite query DNS manipolate. L'obiettivo è isolare l'attore della minaccia, comprendere le tecniche utilizzate e documentare l'intero processo investigativo.

Parte 1: Analisi dell'Attacco SQL Injection via HTTP

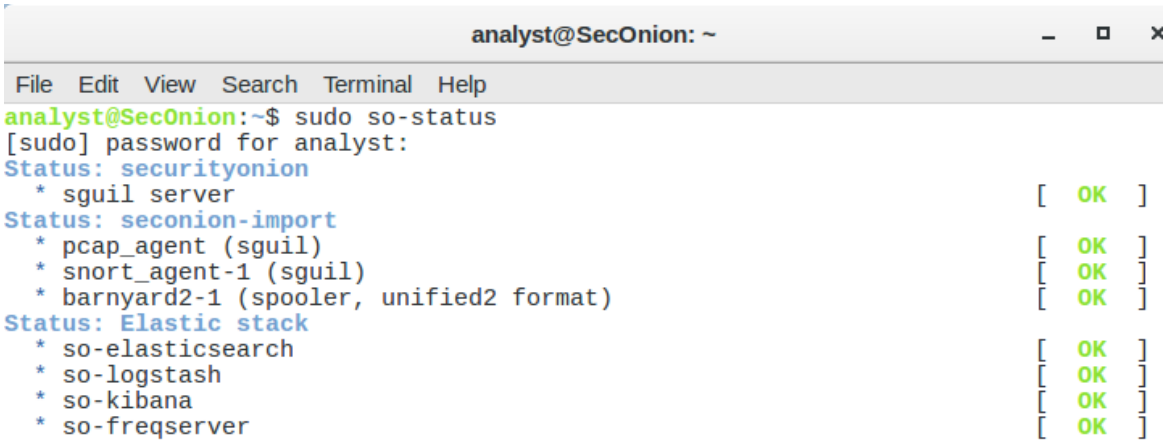
Passo 1: Configurazione dell'Ambiente di Analisi

L'analisi è stata condotta sulla macchina virtuale **Security Onion**, un sistema specializzato per il monitoraggio della sicurezza di rete che integra strumenti come Zeek (Bro), Suricata e l'interfaccia Kibana per l'analisi dei log.

Verifica dello Stato dei Servizi

Dopo aver effettuato il login con le credenziali **analyst/cyberops**, è stato verificato lo stato di tutti i servizi tramite il comando:

sudo so-status

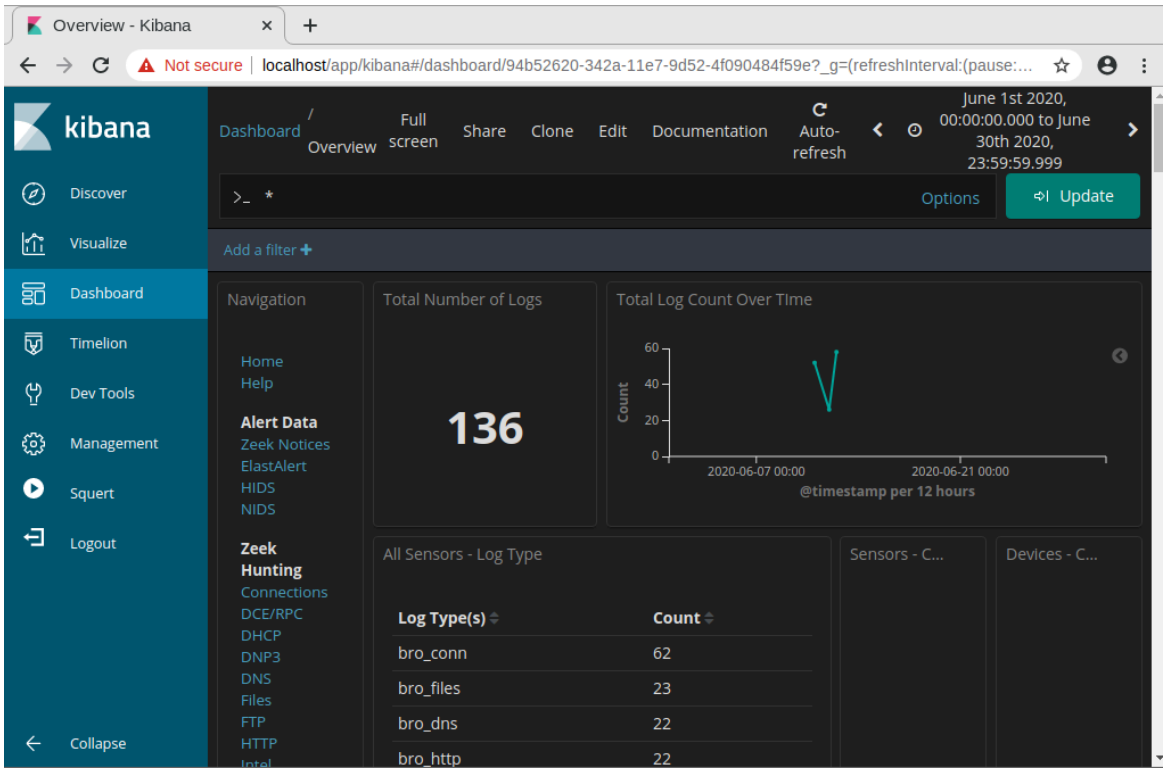


(Verifica dello stato dei servizi Security Onion - tutti i componenti risultano OK)

Come mostrato, tutti i servizi critici risultano operativi (status: **OK**).

Configurazione dell'Intervallo Temporale in Kibana

È stato necessario modificare l'intervallo temporale predefinito di Kibana (ultime 24 ore) per concentrarsi sul periodo dell'attacco: **giugno 2020**. Dall'interfaccia Kibana, è stata selezionata l'opzione **Absolute** sotto **Time Range** e configurato l'intervallo dal **1 giugno 2020** al **30 giugno 2020**.



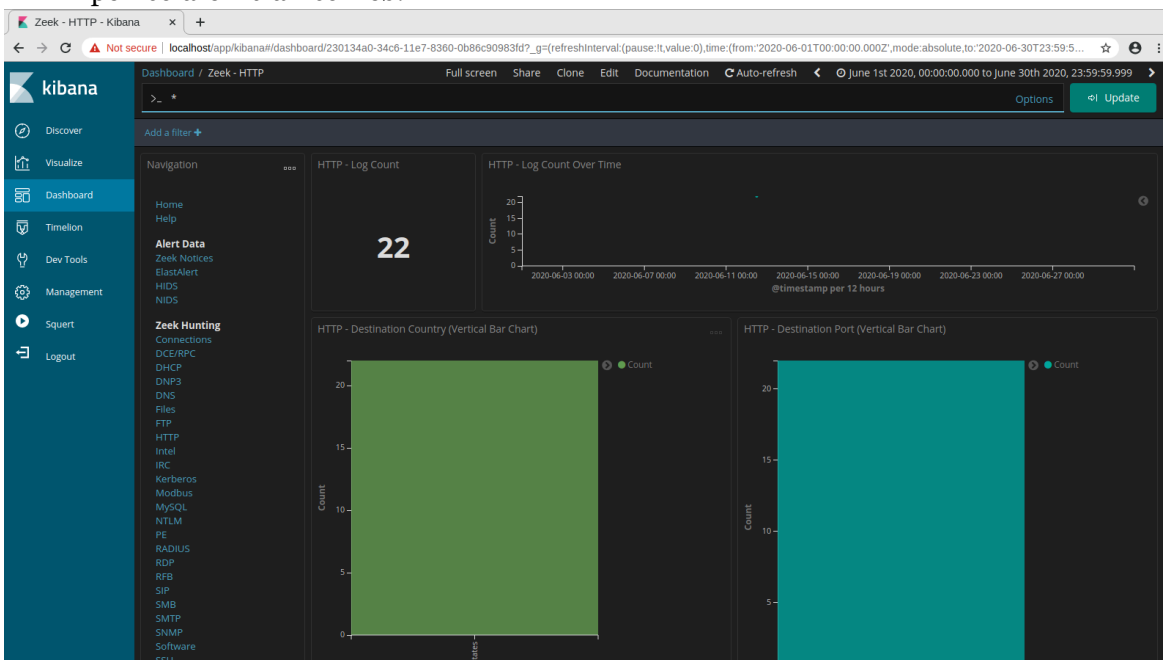
(Configurazione dell'intervallo temporale in Kibana per giugno 2020)

Lo screenshot mostra il dashboard Overview di Kibana con l'intervallo temporale corretto. Il dashboard evidenzia **136 log totali** per il periodo analizzato, con un picco di attività visibile nel grafico temporale.

Passo 2: Filtraggio per Traffico HTTP

Applicazione del Filtro HTTP

Dal dashboard principale, navigando nella sezione **Zeek Hunting**, è stato applicato il filtro **HTTP** per isolare il traffico web.

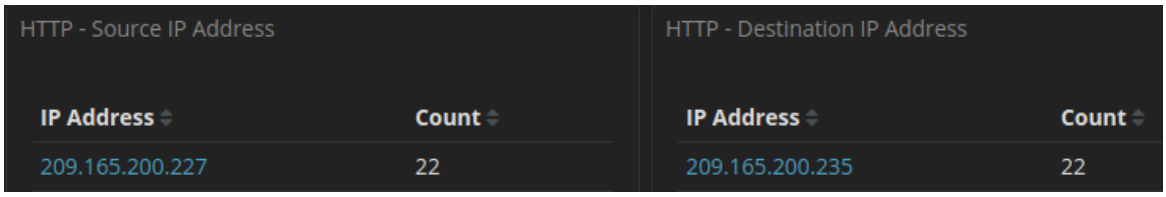


(*Dashboard Zeek Hunting con filtro HTTP applicato - 22 eventi HTTP rilevati)*

Sono stati rilevati **22 eventi HTTP** durante il periodo analizzato.

Identificazione degli Indirizzi IP

L'analisi dei log HTTP ha permesso di identificare i principali attori della comunicazione:



(Tabella degli indirizzi IP sorgente e destinazione per il traffico HTTP)

Dallo screenshot emergono i seguenti dati:

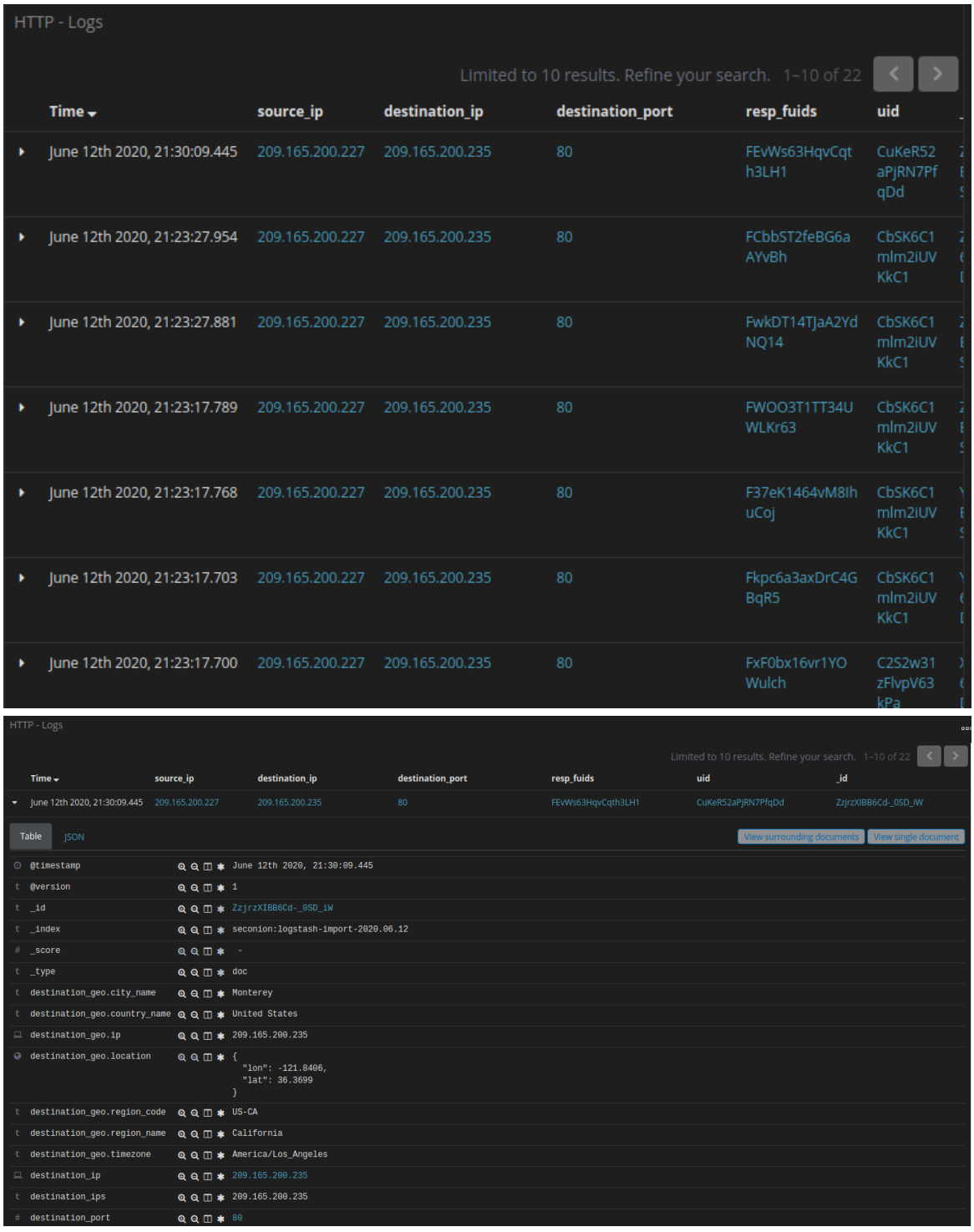
Parametro	**Valore**
IP Sorgente (Source)	209.165.200.227
IP Destinazione (Destination)	209.165.200.235
Porta Destinazione	80 (HTTP)
Numero di Connessioni	22

Risposta alle domande:

- **Qual è l'indirizzo IP sorgente?** 209.165.200.227
- **Qual è l'indirizzo IP destinazione?** 209.165.200.235
- **Qual è il numero di porta destinazione?** 80

Analisi dei Log HTTP Dettagliati

Scorrendo verso il basso fino alla sezione **HTTP - Logs**, è possibile esaminare i singoli eventi HTTP. Espandendo il primo risultato si ottengono informazioni dettagliate sulla richiesta.



(Dettagli del primo evento HTTP - log espanso con timestamp e metadati)

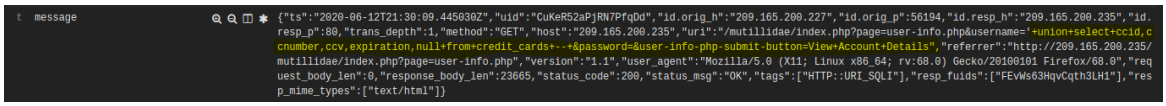
Lo screenshot mostra i dettagli del primo log HTTP:

Risposta alle domande:

- **Qual è il timestamp del primo risultato?** June 12th 2020, 21:30:09.445 (2020-06-12T21:30:09.445Z)

- **Qual è il tipo di evento?** `doc` (documento di log Zeek)
- **Cosa è incluso nel campo message?**

Il campo `message` contiene una richiesta HTTP GET completa.



(Campo message con dettagli della richiesta HTTP GET e payload SQL injection)

Il campo `uri` nella richiesta HTTP contiene:

/mutillidae/index.php?page=user-info.php&username='+union+select+ccid,ccnumber,ccv,expiration,null+from+credit_cards+--+&password=

Qual è il significato di queste informazioni?

Questa stringa rappresenta un chiaro tentativo di **SQL Injection**. Analizzando la query string:

- **Target:** `/mutillidae/index.php` - applicazione web vulnerabile
- **Parametri manipolati:** `username` e `password`
- **Payload SQL:** `'+union+select+ccid,ccnumber,ccv,expiration,null+from+credit\_cards+--+`

La tecnica utilizzata è un **UNION-based SQL Injection**, dove:

- L'apice singolo (') chiude la query SQL originale
- `union select` unisce il risultato della query legittima con una query malevola
- `ccid`, `ccnumber`, `ccv`, `expiration` sono i campi della tabella `credit_cards` che l'attaccante vuole estrarre
- `--` commenta il resto della query originale per evitare errori di sintassi

Questo attacco sfrutta la mancata sanitizzazione degli input utente per accedere direttamente al database sottostante ed esfiltrare **dati sensibili di carte di credito**.

Passo 3: Analisi della Trascrizione PCAP con capME!

Accesso all'Interfaccia capME!

Per visualizzare il contenuto completo delle richieste e risposte HTTP, è stata utilizzata l'interfaccia **capME!**, che fornisce una trascrizione PCAP con evidenziazione delle richieste (SRC - blu) e delle risposte (DST - rosso).

(Interfaccia capME! con trascrizione delle richieste HTTP contenenti username)

Utilizzando la funzione di ricerca (Ctrl+F) per la parola chiave **username**, sono state identificate numerose occorrenze nella trascrizione.

Scoperta dei Dati Esfiltrati

Scorrendo la trascrizione, emerge che il server ha risposto alle richieste SQL injection restituendo **dati sensibili in chiaro**:

Figure 8: Risposta del server (DST) contenente credenziali esfiltrate: username, password e signature

Risposta alla domanda: Cosa vedi più avanti nella trascrizione riguardo ai nomi utente?

La Figura 8 mostra chiaramente che il server ha restituito **username, password e signature** di diversi utenti. Alcuni esempi identificati:

Username	Password	Signature
4444111122223333	--	--
7746536337776330	--	--
8242325748474749	--	--

7725653200487633	--	--
1234567812345678	--	--

Table 2: Esempi di dati sensibili esfiltrati (numeri di carte di credito)

Dalla trascrizione completa emergono anche credenziali utente con il seguente formato:

- Username: formato numerico o alfanumerico
- Password: varie (numeri come 745, 722, 461, 230, 627)
- Signature: date nel formato YYYY-MM-DD (es. 2012-03-01, 2015-04-01, 2016-03-01, 2017-06-01, 2018-11-01)

Fornisci alcuni esempi di nome utente, password e firma che sono stati esfiltrati:

Basandosi sulla Figura 8 e sulla trascrizione:

- **Utente 1:** Username: `4444111122223333`, Password: `745`, Signature: `2012-03-01`
- **Utente 2:** Username: `7746536337776330`, Password: `722`, Signature: `2015-04-01`
- **Utente 3:** Username: `8242325748474749`, Password: `461`, Signature: `2016-03-01`
- **Utente 4:** Username: `7725653200487633`, Password: `230`, Signature: `2017-06-01`
- **Utente 5:** Username: `1234567812345678`, Password: `627`, Signature: `2018-11-01`

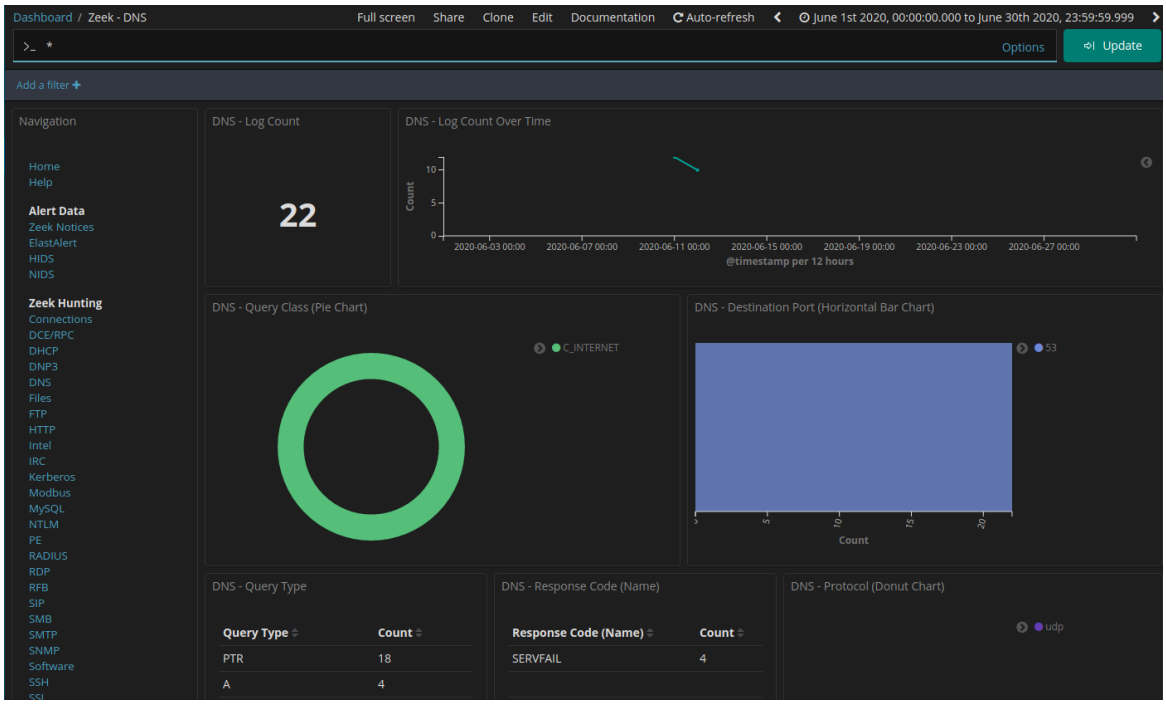
Questi dati rappresentano una **grave violazione della sicurezza**, poiché l'attaccante ha ottenuto accesso a informazioni riservate che possono essere utilizzate per frodi, accessi non autorizzati e ulteriori attacchi.

Parte 2: Analisi dell'Esfiltrazione DNS

Passo 1: Filtraggio per Traffico DNS

Applicazione del Filtro DNS

Tornando alla dashboard principale di Kibana, è stato rimosso il filtro HTTP e applicato il filtro **DNS** nella sezione Zeek Hunting, mantenendo l'intervallo temporale di giugno 2020.



(Dashboard DNS con metriche di query, tipi di record e porte di destinazione)

Lo screenshot mostra il dashboard DNS con **22 log DNS totali**.

Passo 2: Identificazione delle Query Anomale

Analisi dei Client e Server DNS

Scorrendo il dashboard, la sezione **DNS - Client** e **DNS - Server** rivela gli attori coinvolti:

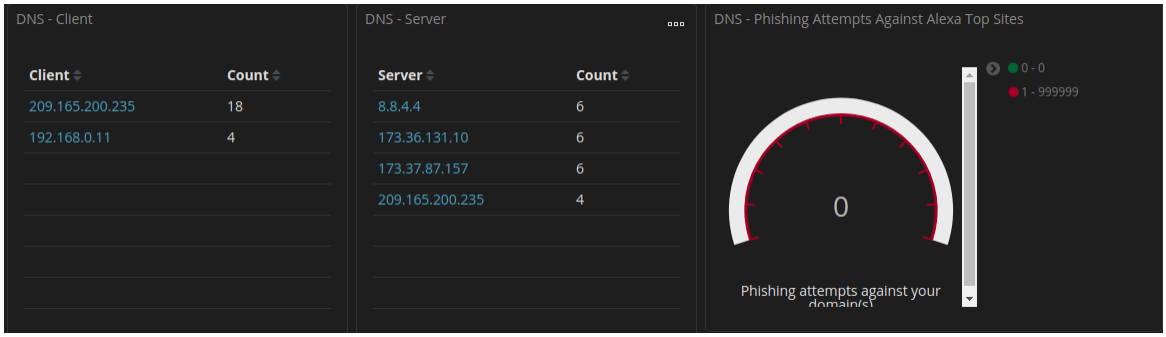


Tabelle DNS Client e DNS Server con conteggi delle query

Dalla Figura 10 si identificano:

DNS Client	Count
209.165.200.235	18
192.168.0.11	4

(Client DNS attivi)

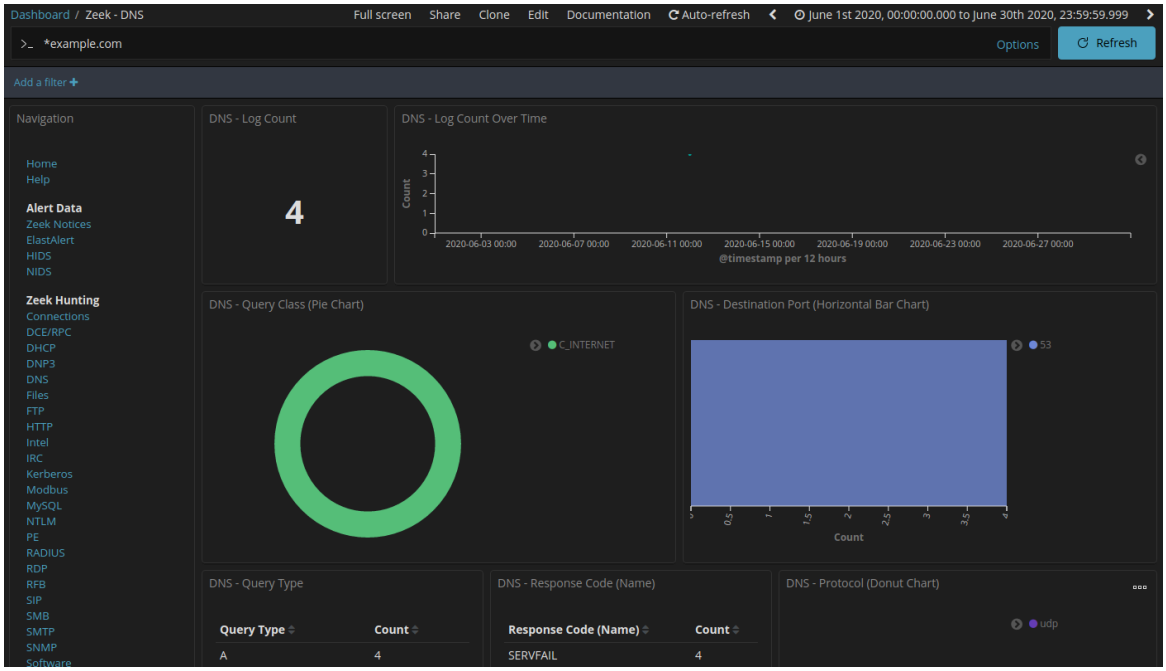
DNS Server	Count
8.8.4.4	6
173.36.131.10	6
173.37.87.157	6
209.165.200.235	4

(Server DNS utilizzati)

Passo 3: Applicazione del Filtro example.com

Isolamento delle Query Sospette

Applicando il filtro `*example.com` nella barra di ricerca, il numero di log si riduce drasticamente a **4 eventi**.



(Dashboard DNS filtrato per example.com)

Lo screenshot mostra che tutte e 4 le query verso `example.com` hanno ricevuto risposta **SERVFAIL**, indicando che il dominio non è stato risolto correttamente. Questo è coerente con l'utilizzo del DNS per esfiltrazione dati piuttosto che per risoluzione legittima di nomi.

Passo 4: Determinazione dei Dati Esfiltrati

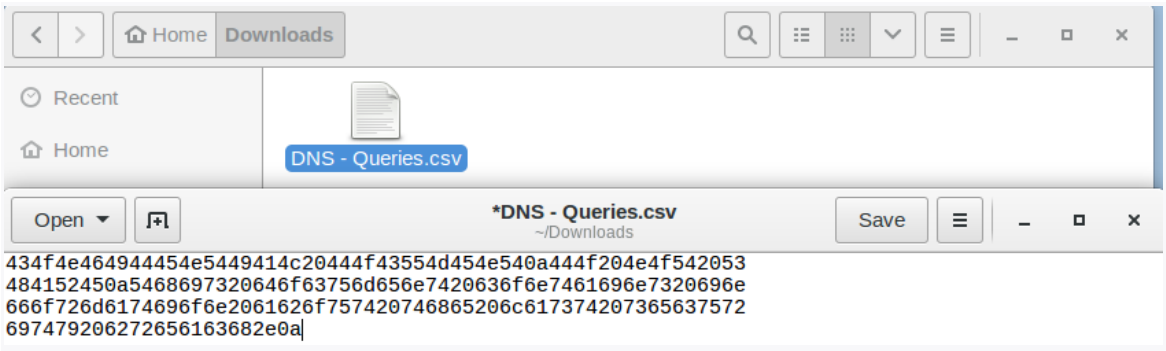
Esportazione delle Query DNS

Non disponendo di screenshot specifici del download delle query DNS, il procedimento descritto nelle istruzioni prevede:

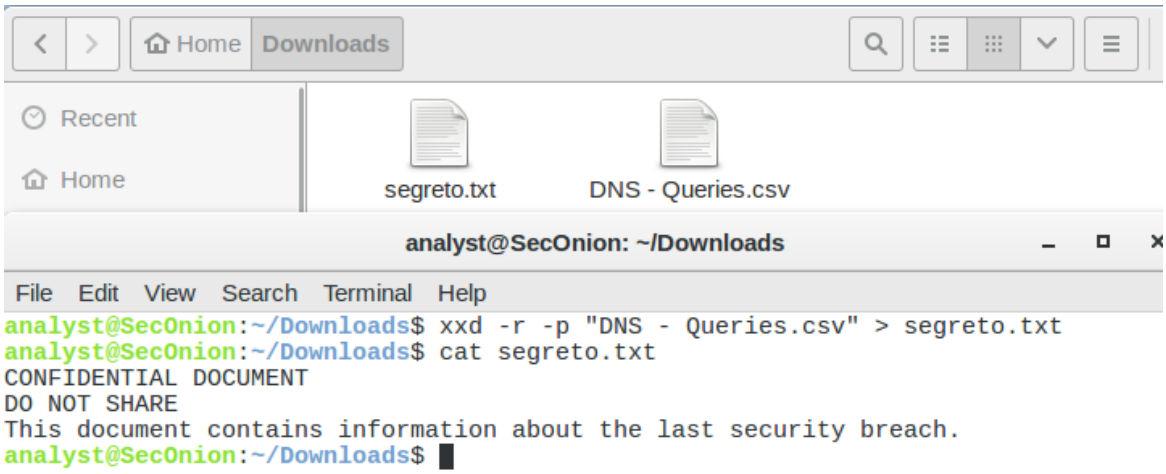
1. Scorrere verso il basso per identificare le 4 voci di log uniche per le query DNS a `*.ns.example.com`
2. Notare come alcune query abbiano sottodomini insolitamente lunghi composti da caratteri esadecimali (0-9, a-f)
3. Fare clic sul link di download **Export: Raw** per scaricare le query su file esterno (CSV)
4. Salvare il file nella cartella `/home/analyst/Downloads`

Decodifica dei Sottodomini Esadecimali

Il file CSV scaricato contiene righe del tipo:



Queste stringhe esadecimali sono state decodificate utilizzando il comando: *xxd -r -p "DNS - Queries.csv" > secret.txt*, il risultato della decodifica rivela il seguente testo:



Risposta alle Domande DNS

I sottodomini delle query DNS erano sottodomini? Se no, qual è il testo?

No, i cosiddetti "sottodomini" non erano veri sottodomini DNS ma **frammenti di un file di testo codificato in formato esadecimale**. Decodificando le stringhe esadecimali si ottiene il messaggio:

Cosa implica questo risultato riguardo a queste particolari richieste DNS? Qual è il significato più ampio?

Questo risultato dimostra che le query DNS sono state utilizzate come **canale nascosto per l'esfiltrazione di dati sensibili**. La tecnica è nota come **DNS Tunneling** o **DNS Data Exfiltration**.

Implicazioni specifiche:

- Le query DNS non avevano scopo di risoluzione legittima
- I dati sensibili sono stati frammentati e codificati in esadecimale
- Ogni frammento è stato inserito come "sottodominio" nelle query verso ns.example.com
- Il DNS server autoritativo per example.com (controllato dall'attaccante) riceve e ricostruisce i dati

Significato più ampio:

- Il DNS è raramente bloccato da firewall e proxy, rendendolo un canale ideale per esfiltrazione
- Questa tecnica può bypassare molti sistemi di sicurezza tradizionali
- La lunghezza e la complessità dei nomi di dominio possono rivelare attività malevole
- È necessario monitorare pattern anomali nelle query DNS (lunghezza, entropia, frequenza)

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

Cosa ha creato le query:

Molto probabilmente un **malware o uno script di esfiltrazione** installato sul sistema compromesso (209.165.200.235). Il processo tipico include:

1. Lettura del file confidenziale dal filesystem
2. Conversione del contenuto in formato esadecimale
3. Frammentazione in blocchi di dimensioni compatibili con i limiti DNS (max 63 caratteri per label, 253 totali)
4. Generazione automatica di query DNS verso ns.example.com includendo i blocchi come "sottodomini"
5. Invio sequenziale delle query al DNS resolver

Perché è stato scelto il DNS:

Il DNS è stato scelto per diverse ragioni strategiche:

1. **Universalmente permesso:** il traffico DNS (porta 53 UDP/TCP) è quasi sempre consentito attraverso firewall aziendali

2. **Basso sospetto:** le query DNS sono considerate "traffico normale" e raramente vengono ispezionate in profondità
3. **Nessuna connessione diretta:** l'attaccante non deve stabilire connessioni in uscita verso server C&C esterni
4. **Difficile da rilevare:** senza analisi approfondita di lunghezza e contenuto dei domini, l'attività appare legittima
5. **Resilienza:** anche se alcuni DNS server bloccano il dominio, l'attaccante può cambiare dominio rapidamente
6. **Quantità di dati:** nonostante i limiti di dimensione, è possibile esfiltrare file di dimensioni considerevoli con query multiple

---

Conclusioni

Remediation

Immediate (0-24 ore)

1. Isolare il server compromesso 209.165.200.235 dalla rete
2. Modificare tutte le password degli utenti compromessi
3. Bloccare le carte di credito esfiltrate
4. Implementare regole firewall per bloccare comunicazioni con example.com
5. Avviare analisi forense completa del sistema compromesso

Breve termine (1-7 giorni)

1. Implementare Web Application Firewall (WAF) con protezione SQL injection
2. Applicare parametrized queries e prepared statements nell'applicazione
3. Configurare monitoraggio DNS avanzato per rilevare query anomale
4. Implementare anomaly detection per query DNS con alta entropia
5. Rivedere e rafforzare la configurazione del database
6. Applicare principio del minimo privilegio per l'utente del database web

Medio-lungo termine (1-3 mesi)

1. Condurre security code review dell'applicazione Mutillidae
2. Implementare SIEM con correlazione eventi HTTP e DNS
3. Configurare alerting per pattern di SQL injection
4. Implementare DNS Security Extensions (DNSSEC)
5. Considerare implementazione di DNS firewall
6. Formare il team di sviluppo su secure coding practices
7. Stabilire programma di vulnerability assessment periodico

8. Implementare Data Loss Prevention (DLP) per monitorare esfiltrazione dati

Indicatori di Compromissione (IoC)

Tipo	Valore
IP Attaccante	209.165.200.227
IP Vittima	209.165.200.235
Dominio malevolo	ns.example.com
Dominio malevolo	example.com
Pattern URL	/mutillidae/index.php?page=user-info.php&username=
SQL Pattern	'+union+select+ccid,ccnumber,ccv,expiration
Tabella target	credit_cards
Timestamp iniziale	2020-06-12T21:30:09.445Z

(Indicatori di compromissione identificati)

Conclusione

L'analisi ha dimostrato come un attaccante abbia combinato due tecniche sofisticate per compromettere un sistema e esfiltrare dati sensibili:

1. **SQL Injection** per ottenere accesso non autorizzato al database
2. **DNS Tunneling** per esfiltrare documenti confidenziali in modo nascosto

La capacità di Security Onion di correlare eventi HTTP e DNS è stata fondamentale per identificare l'intera catena dell'attacco. Questo caso sottolinea l'importanza di:

- Validazione rigorosa degli input utente
- Monitoraggio multi-layer del traffico di rete
- Analisi comportamentale del traffico DNS
- Correlazione di eventi apparentemente non correlati
- Defense-in-depth approach alla sicurezza

L'implementazione delle raccomandazioni proposte ridurrà significativamente il rischio di attacchi simili in futuro.
