

Build Week Bonus

W W W . Z E R O D A Y K N I G H T S . C O M

INDICE

03 **Bonus 1**

08 **Bonus 2**

20 **Bonus 3**

Bonus 1

CASO ANYRUN

Analisi

L'analisi effettuata sulla macchina virtuale Windows 10 Professional, configurata con build 19045 a 64-bit, ha evidenziato la presenza di anomalie significative riconducibili a un'infezione malware. Il file sospetto analizzato, denominato Jvczfhe.exe, è stato individuato come un possibile Dropper. Questo tipo di malware è progettato per introdurre altri componenti malevoli nel sistema compromesso, fungendo da veicolo per payload secondari, che potrebbero includere ransomware, spyware o strumenti di controllo remoto.

Durante il comportamento osservato in ambiente sandbox, il file Jvczfhe.exe ha effettuato modifiche al registro di sistema, in particolare sulle impostazioni di sicurezza di Internet Explorer e sulle politiche di connessione proxy. Inoltre, il malware ha eseguito comandi tramite il processore di comando di Windows (cmd.exe) e utilizzato timeout.exe per ritardare l'esecuzione di alcune operazioni, una tecnica comunemente impiegata per sfuggire ai meccanismi di rilevamento automatizzati. Un altro comportamento anomalo è stato il rilascio di ulteriori eseguibili dannosi, tra cui Muadnrd.exe, che ha replicato molte delle attività nocive.

Un aspetto interessante del comportamento del malware è stato il suo tentativo di connettersi a porte non usuali tramite InstallUtil.exe. L'uso di strumenti legittimi del sistema operativo come InstallUtil rappresenta una strategia nota come "Living off the Land" (LotL), progettata per sfruttare risorse già presenti sul sistema vittima, riducendo così il rischio di rilevamento. Inoltre, sono stati disabilitati i log di traccia, suggerendo l'intento di rendere più difficile l'analisi forense successiva all'attacco.

Cosa è?

Un dropper è un malware che introduce altri malware in un sistema, scaricandoli o estraendoli per esecuzione, spesso evitando i controlli di sicurezza.

Fonti non certificate

L'origine del malware analizzato sembra risalire a un repository GitHub sospetto. Questo evidenzia come il download di software da fonti non verificate rappresenti un rischio elevato, soprattutto in contesti dove non vengono utilizzate protezioni adeguate come soluzioni antivirus avanzate o sistemi di monitoraggio della rete. È plausibile che l'utente abbia inconsapevolmente scaricato ed eseguito il file dannoso, magari attratto da una descrizione legittima o da un'apparente utilità del programma.

MITRE ATT&CK Matrix

La MITRE ATT&CK Matrix è un framework che classifica e descrive le tecniche usate dagli attaccanti informatici durante le varie fasi di un attacco, come l'accesso iniziale, l'esecuzione di codice malevolo e la persistenza. È progettata per aiutare i professionisti della sicurezza a comprendere e anticipare il comportamento degli attaccanti, mappando le attività osservate alle tecniche documentate, e supporta lo sviluppo di strategie di rilevamento e difesa più efficaci.

Subtechniques ▾

[T1059](#)

"Command and Scripting Interpreter"

Permissions required:

Data sources: Process: Process Creation, Module: Module Load.

Subtechniques ▾

[T1036](#)

"Masquerading"

"Query Registry"

Permissions required: User, Administrator, SYSTEM

Data sources: Process: OS API Execution, Process: Process Creation, Command: Command Execution, Windows Registry

"Non-Standard Port"

Windows Command Shell ▾

- Starts CMD.EXE for commands execution (2)

7492 Jvczfhe.exe (1)

7824 Muadnrd.exe (1)

- Uses TIMEOUT.EXE to delay execution (2)

7520 cmd.exe (1)

7876 cmd.exe (1)

Rename System Utilities ▾

- Process drops legitimate windows executable (1)

6596 firefox.exe (1)

- Reads Microsoft Office registry keys (1)

6596 firefox.exe (1)

- Reads security settings of Internet Explorer (1)

7492 Jvczfhe.exe (1)

7824 Muadnrd.exe (1)

- Checks Windows Trust Settings (2)

7492 Jvczfhe.exe (1)

7824 Muadnrd.exe (1)

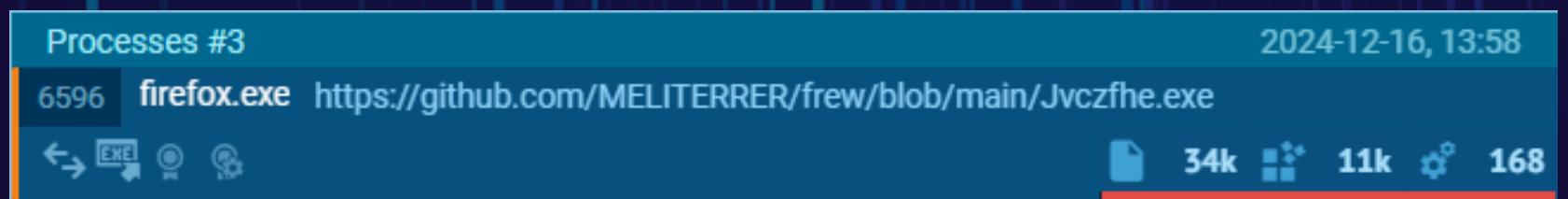
- Connects to unusual port (1)

5152 InstallUtil.exe (1)

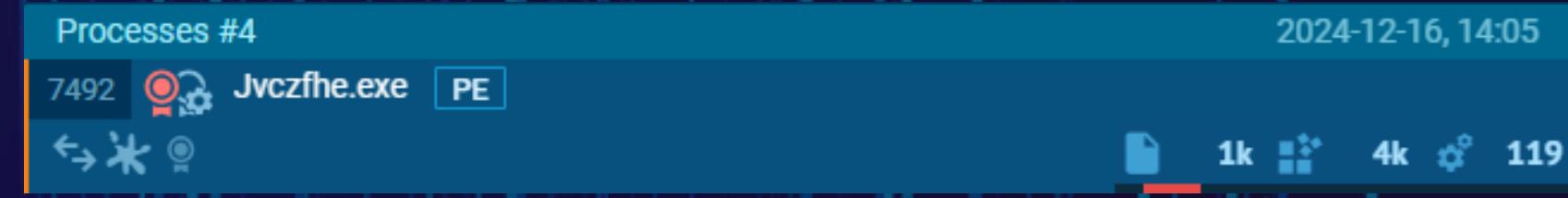
DETTAGLI EVENTI



Il processo descritto riguarda l'esecuzione di "Muadnrd.exe" dalla cartella "Downloads" dell'utente "admin", con lo stesso file come processo padre. Sebbene l'esecuzione di file eseguibili possa essere legittima, la presenza di incidenti legati al registro e ai dump suggerisce attività dannose. Il programma sta leggendo informazioni dal registro (nome del computer, GUID della macchina) e rilevando un protettore .NET Reactor, comportamenti tipici di malware o software che cerca di eludere il rilevamento. Pertanto, il processo è probabilmente associato a un programma dannoso.



Successivamente vediamo l'esecuzione di Mozilla Firefox che apre un URL sospetto. Sebbene l'uso di Firefox per navigare sia legittimo, il comportamento osservato, come il "drop" o la sovrascrittura di contenuti eseguibili e la lettura delle chiavi del registro di Microsoft Office, suggerisce potenziale attività dannosa. Il fatto che venga scaricato un eseguibile legittimo potrebbe mascherare contenuti malevoli. È necessaria un'ulteriore analisi per determinare l'intento del processo.



L'evento descritto riguarda "Jvczfhe.exe", eseguito da "firefox.exe" e scaricato da GitHub nella cartella "Downloads" dell'utente. Sebbene il download di file possa essere legittimo, le azioni del programma, come la disabilitazione dei log, la lettura dei GUID del sistema e il controllo delle impostazioni proxy, indicano un comportamento dannoso. Tali tecniche sono tipiche di malware che cerca di eludere il rilevamento, raccogliere informazioni sul sistema e comunicare con server di comando e controllo. Il crash del programma durante l'esecuzione rafforza il sospetto di attività malevole.



Infine, questo processo riguarda "InstallUtil.exe", uno strumento legittimo per gestire componenti .NET. Tuttavia, la connessione a una porta insolita, la lettura di valori ambientali e il rilevamento di un protettore .NET Reactor suggeriscono un possibile uso malevolo. Questi comportamenti sono tipici di malware che tenta di comunicare con server di comando e controllo o proteggersi dall'analisi.

Remediation

Per affrontare situazioni simili e prevenire futuri attacchi, risulta cruciale adottare alcune strategie. In primo luogo, è essenziale educare gli utenti sui rischi associati al download di contenuti da fonti non affidabili e sull'importanza di verificare l'autenticità di un file tramite firme digitali o hash. Inoltre, la configurazione del sistema dovrebbe essere rafforzata per bloccare l'esecuzione di file non attendibili e per registrare tutte le modifiche al sistema, incrementando così la capacità di risposta alle minacce. La prevenzione passa anche attraverso strumenti di sicurezza robusti, come gli EDR, che analizzano il comportamento dei processi in tempo reale.

Una volta compromesso il sistema, la priorità è l'isolamento immediato della macchina per evitare ulteriori propagazioni. Segue una scansione completa del sistema per identificare e rimuovere tutte le componenti nocive. Qualora il danno fosse esteso, il ripristino da un backup sicuro rappresenta l'unica opzione percorribile per garantire l'integrità delle operazioni. Infine, è necessario un monitoraggio continuo del sistema per rilevare eventuali attività residue o tentativi di riconnessione da parte degli attaccanti.

La natura di questo attacco sottolinea un punto critico: i Dropper non solo rappresentano una minaccia diretta, ma sono anche il primo passo di un'infezione più complessa. Attraverso le loro azioni, possono compromettere la sicurezza del sistema, consentire il furto di informazioni sensibili o destabilizzare il funzionamento delle risorse aziendali. La lettura e la modifica delle chiavi di registro legate al GUID del sistema e alle impostazioni proxy suggerisce che il malware potrebbe tentare di reindirizzare il traffico o raccogliere dati specifici relativi alla configurazione della rete.

Bonus 2

SQL INJECTION E DNS EXFILTRATION

DI COSA SI TRATTA

In questo secondo bonus approfondiremo un caso studio che riguarda l'analisi di due distinti tipi di attacco informatico: una **SQL Injection** e una **DNS Exfiltration**. Questo studio ci permetterà di esaminare le tecniche utilizzate dagli attaccanti per compromettere la sicurezza dei sistemi, evidenziando le caratteristiche e le implicazioni specifiche di ciascun attacco.

Sql injection

La **SQL Injection** è un attacco che sfrutta vulnerabilità nei sistemi web che interagiscono con database SQL, inserendo comandi SQL malevoli all'interno di input forniti dall'utente, come campi di login o URL. Questo avviene quando le applicazioni non validano o sanitizzano correttamente i dati in ingresso, permettendo agli attaccanti di manipolare query SQL esistenti. Gli obiettivi possono includere l'accesso non autorizzato ai dati, la modifica o l'eliminazione di informazioni, o l'ottenimento di controllo sul database sottostante.

DNS Exfiltration

La **DNS Exfiltration** è una tecnica di attacco in cui gli attaccanti utilizzano il protocollo DNS (Domain Name System) per trasferire in modo furtivo dati sensibili da un sistema compromesso a un server controllato da loro. Questo metodo sfrutta il fatto che il traffico DNS è spesso considerato legittimo e non viene monitorato rigorosamente. Gli attaccanti codificano i dati rubati in richieste DNS, inviandoli sotto forma di sottodomini all'interno di query apparentemente normali. Il server DNS malevolo decodifica poi queste informazioni, completando l'esfiltrazione senza destare sospetti.

Sql Injection

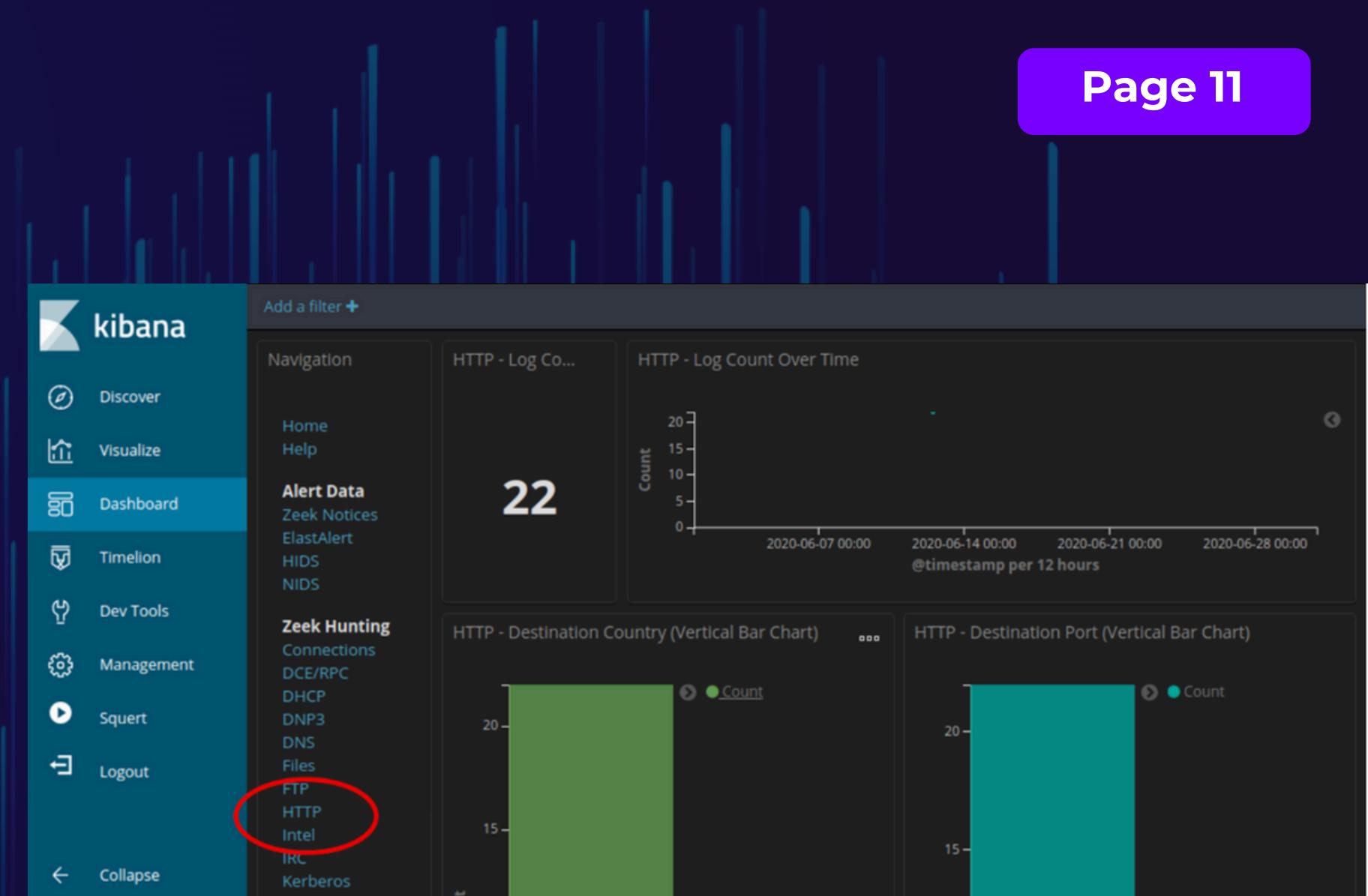
ANALISI

Kibana

Per analizzare in modo approfondito il traffico di rete, utilizzerò **Kibana**, una potente piattaforma di visualizzazione e analisi dei dati. Kibana è uno strumento open source che consente di esplorare, filtrare e rappresentare graficamente i dati raccolti da Elasticsearch, spesso utilizzato per monitorare log di sistema, traffico di rete e altri tipi di dati strutturati o semi-strutturati. Grazie alla sua interfaccia intuitiva, Kibana permette di individuare facilmente pattern, anomalie e dettagli rilevanti all'interno di grandi quantità di informazioni.

In questo contesto, Kibana sarà utilizzato per monitorare tutto il traffico di rete in entrata e in uscita, restringendo l'analisi a un preciso intervallo temporale, ovvero il mese di **giugno 2020**.

Inoltre, lo strumento consente di applicare filtri mirati per isolare specifici protocolli. Dato che stiamo analizzando un attacco di SQL Injection, concentreremo la nostra attenzione sui pacchetti **HTTP**, che rappresentano il veicolo principale attraverso cui questo tipo di attacco viene generalmente eseguito. Kibana ci aiuterà a identificare eventuali richieste sospette o anomalie all'interno di queste comunicazioni.



Analisi Pacchetto

Procederemo ora ad analizzare una specifica richiesta effettuata il 12 giugno alle 21:30:09, con IP source 209.165.200.227 e IP destination 209.165.200.235. Per approfondire questa richiesta, utilizzerò lo strumento **capME!**, che facilita l'analisi dettagliata del traffico di rete.

capME! è uno strumento integrato in piattaforme di analisi forense di rete, progettato per decodificare e visualizzare le comunicazioni in modo leggibile. Consente agli analisti di ricostruire richieste HTTP, messaggi DNS e altre interazioni di rete, partendo dai pacchetti acquisiti. Grazie alle sue capacità di filtraggio e interpretazione, **capME!** rende più semplice individuare comportamenti sospetti o attività malevole, mostrando il contenuto delle richieste e delle risposte.

In questo caso, **capME!** sarà utilizzato per esaminare nel dettaglio la richiesta **GET** effettuata dall'attaccante. Lo strumento ci permetterà di filtrare i dati rilevanti e interpretare la natura e gli obiettivi dell'azione malevola all'interno del flusso di traffico analizzato.

Per visualizzare nel dettaglio la richiesta in esame, sarà sufficiente aprire **l'ID univoco** associato alla richiesta in una **nuova scheda**. Questo permetterà di accedere direttamente alle informazioni specifiche della comunicazione, consentendo un'analisi approfondita del contenuto e del contesto della richiesta effettuata.

The screenshot shows the Kibana interface with a sidebar on the left containing links: Discover, Visualize, Dashboard (which is selected), Timelion, Dev Tools, Management, Squert, and Logout. The main area is titled "HTTP - Logs" and displays a table of search results. The columns are: Time, source_ip, destination_ip, destination_port, resp_fuids, uid, and _id. One row is highlighted with a red box around the _id column value "ZzjrzXIBB6Cd-_0SD_IW". Below the table, there are two tabs: "Table" (selected) and "JSON". Under the JSON tab, there is a list of field names with their corresponding values and search filters. One entry in this list is circled in red and shows the value "ZzjrzXIBB6Cd-_0SD_IW" for the field "t _id".

Time	source_ip	destination_ip	destination_port	resp_fuids	uid	_id
June 12th 2020, 21:30:09.445	209.165.200.227	209.165.200.235	80	FEvWs63HqvCqt h3LH1	CuKeR52aP JRNPfqDd	ZzjrzXIBB6Cd-_0SD_IW

Table JSON

- @timestamp June 12th 2020, 21:30:09.445
- @version 1
- t _id ZzjrzXIBB6Cd-_0SD_IW
- t _index seconion:logstash-import-2020.06.12
- # _score -
- t _type doc
- t destination_geo.city_name Monterey
- t destination_geo.country_name United States
- t destination_geo.ip 209.165.200.235
- t destination_geo.location {"lon": -121.8406, "lat": 36.3699}
- t destination_geo.region_code US-CA

Analisi Richiesta

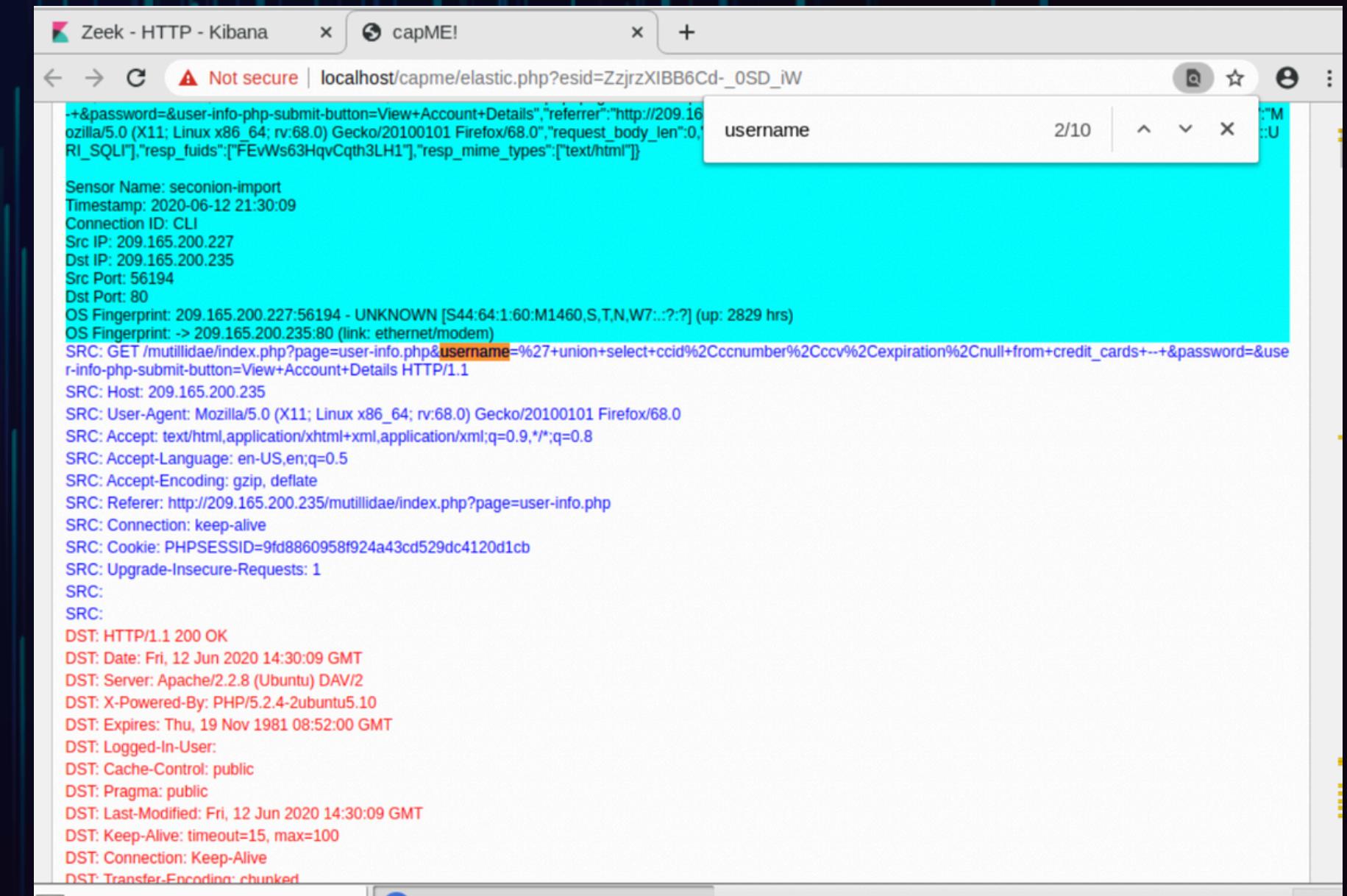
Utilizzando capME!, procederemo ad analizzare la richiesta GET effettuata dall'attaccante. Applicando un filtro sul contenuto basato sulla parola chiave "username", siamo in grado di identificare la specifica query SQL inviata dall'attaccante e di osservare le risposte generate dal server.

La query SQL malevola individuata è la seguente:

**username='+union+select+ccid,ccnumber,ccv,expiration,n
ull+from+credit_cards+--+&password=.**

Questa richiesta sfrutta una vulnerabilità di SQL Injection per unire i risultati di una query legittima con quelli di una query malevola (**UNION SELECT**).

L'attaccante mira ad accedere a **informazioni sensibili** archiviate nel database, come identificativi di carte di credito, numeri di carta, codici CVV e date di scadenza, estraendole dalla tabella denominata credit_cards. Le risposte ricevute dal server confermano che la vulnerabilità è stata sfruttata con successo, evidenziando una grave compromissione della sicurezza del sistema.



Dati Recuperati

Dall'analisi completa dei dati tramite capME!, siamo riusciti a ricostruire una tabella che rappresenta i dati sensibili esfiltrati attraverso l'attacco di SQL Injection. La tabella include i campi **Username**, **Password** e **Signature**.

Questi dati dimostrano il successo dell'attacco, che ha portato al furto di informazioni presumibilmente sensibili e utili all'attaccante. La vulnerabilità sfruttata evidenzia l'importanza di implementare misure di sicurezza adeguate per proteggere i sistemi da SQL Injection.

Username	Password	Signature
444411122223333	745	2012-03-01
774653633776330	722	2015-04-01
8242325748474749	461	2016-03-01
7725653200487633	230	2017-06-01
1234567812345678627	627	2018-11-01

Remediation E Mitigation

La **mitigazione e la remediation** di un attacco di SQL Injection richiedono un approccio strutturato che combini misure preventive, rilevamento e azioni correttive per eliminare la vulnerabilità e proteggere il sistema da attacchi futuri.

Per mitigare il rischio, è essenziale adottare una rigorosa validazione degli input forniti dagli utenti. Tutti i dati in ingresso devono essere verificati e sanitizzati per assicurarsi che non contengano caratteri o sequenze che possano essere interpretati come comandi SQL. Questo processo può essere potenziato utilizzando query parametrizzate o dichiarazioni preparate, che separano il codice SQL dai dati forniti dall'utente, prevenendo l'iniezione di comandi malevoli.

Un'altra misura fondamentale è l'implementazione di principi di **least privilege** nella configurazione del database. Gli account utilizzati dalle applicazioni per accedere al database dovrebbero avere solo i permessi strettamente necessari, limitando così l'impatto di un eventuale attacco. Ad esempio, un'applicazione che esegue semplici operazioni di lettura non dovrebbe mai avere permessi di modifica o cancellazione dei dati.

Per quanto riguarda la remediation, una volta rilevato un attacco di SQL Injection, è cruciale identificare e correggere immediatamente la vulnerabilità che lo ha reso possibile. Questo include l'analisi del codice sorgente dell'applicazione per individuare le **query SQL dinamiche non sicure** e modificarle per utilizzare metodi più sicuri, come l'ORM (Object-Relational Mapping) o le query parametrizzate.

Parallelamente, è necessario analizzare i log del sistema per identificare le azioni compiute dall'attaccante e determinare se sono stati compromessi dati sensibili. **Nel caso di compromissione**, deve essere notificata la violazione secondo le normative applicabili, come il **GDPR**, e prese le misure necessarie per limitare ulteriori danni, come il reset delle credenziali degli utenti.

Infine, l'implementazione di strumenti di **monitoraggio e rilevamento**, come Web Application Firewall (WAF), può prevenire ulteriori tentativi di attacco, bloccando richieste sospette prima che raggiungano il database. È fondamentale adottare un approccio proattivo alla sicurezza, mantenendo sempre aggiornato il software e formando gli sviluppatori sulle migliori pratiche per la scrittura di codice sicuro.

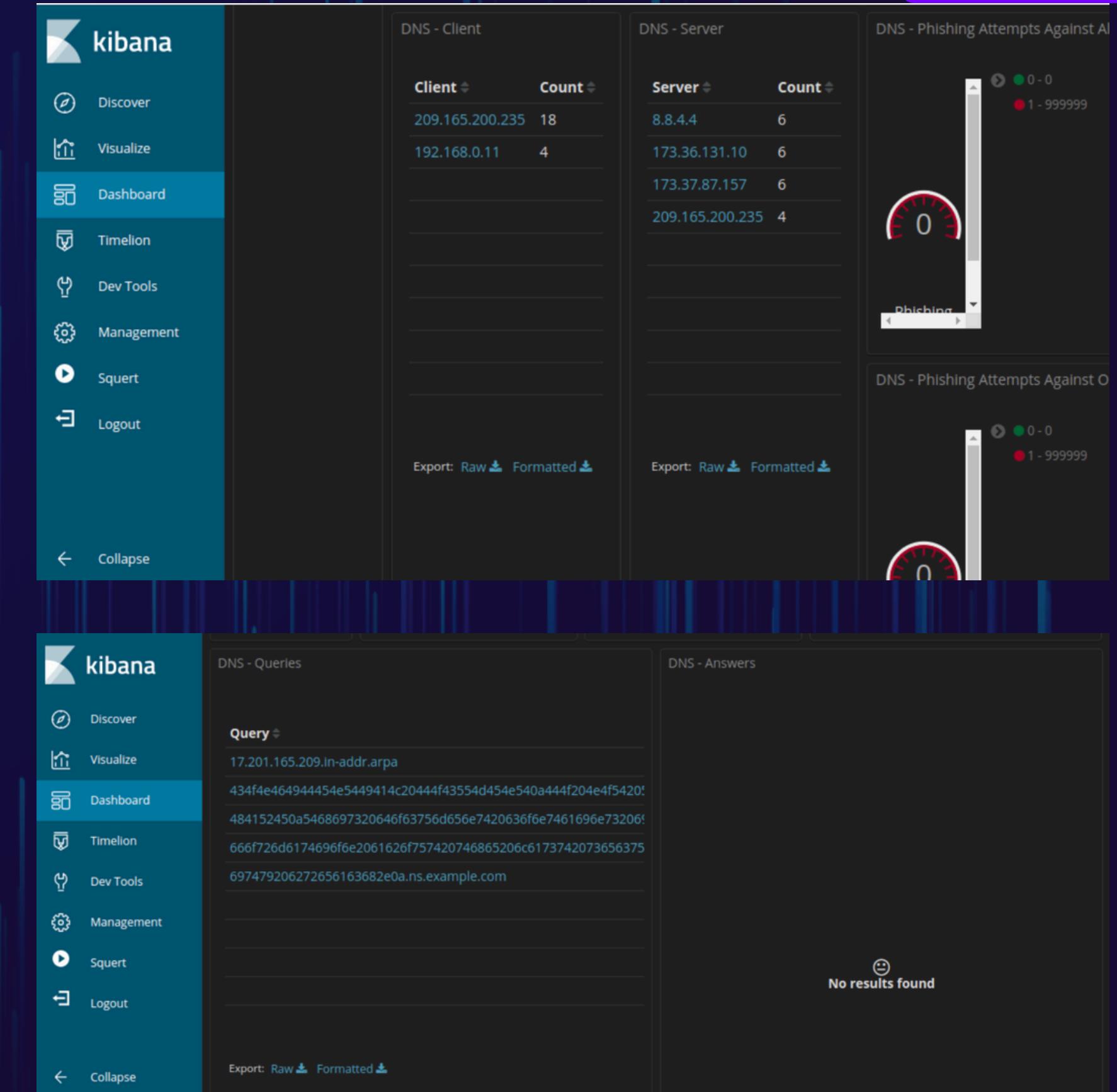
DNS Exfiltration

ANALISI

Kibana

La seconda parte del laboratorio si è concentrata sull'analisi di richieste DNS anomale, impiegate per esfiltrare dati sensibili. Le richieste analizzate erano caratterizzate da **sottodomini con lunghezze inconsuete**, contenenti stringhe di caratteri codificati in formato **esadecimale**. Tali anomalie suggerivano l'utilizzo del DNS come canale per trasferire informazioni sensibili in modo nascosto, evitando i tradizionali sistemi di rilevamento.

L'analisi è stata condotta su un log di traffico DNS risalente a giugno 2020, con un focus particolare su richieste indirizzate al dominio **example.com**. Tra queste, sono state individuate quelle che presentavano sottodomini sospetti, dotati di lunghezze e strutture decisamente atipiche per una normale comunicazione DNS. I dati relativi agli indirizzi IP coinvolti hanno rivelato che le richieste venivano originariamente inviate da un client con indirizzo IP **192.168.0.11**, mentre il server DNS che rispondeva alle richieste aveva l'indirizzo IP **209.165.200.235**.



Dati Esfiltrati

Un passo fondamentale per completare questa analisi è determinare quali dati sono stati **effettivamente esfiltrati**. Per fare ciò, bisogna estrarre e decodificare le stringhe esadecimali contenute nei sottodomini sospetti.

Dopo aver analizzato il traffico DNS, è possibile identificare le query sospette che contengono sottodomini insolitamente lunghi, indirizzati al dominio ns.example.com. Le stringhe che compaiono nei sottodomini, formate da numeri e lettere esadecimali (0-9, a-f), sono un chiaro segnale che i dati trasmessi potrebbero essere codificati in questo formato, anziché essere semplicemente nomi di sottodomini legittimi.

Una volta individuati questi sottodomini sospetti, è possibile procedere con l'esportazione dei log contenenti queste query. Per farlo, si può cliccare sul link "**Export: Raw**" per scaricare i dettagli delle query in un file esterno. Il file CSV generato verrà salvato nella cartella **/home/analyst/Downloads**, permettendo di analizzare ulteriormente i dati e identificare esattamente quali informazioni sono state esfiltrate dal sistema.

The screenshot shows a terminal window titled "analyst@SecOnion: ~/Downloads". The window displays the command "xxd -r -p "DNS - Queries.csv" > secret.txt" being run, followed by the output of the command, which is a file named "secret.txt". The content of "secret.txt" is shown in the terminal window, revealing sensitive information:

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

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.
analyst@SecOnion:~/Downloads$
```

*Il comando **xxd -r -p** permette di **decodificare** una stringa esadecimale e convertirla in un formato leggibile. Nel caso specifico, il comando è stato utilizzato per prendere il contenuto del file CSV, che presumibilmente conteneva dati esadecimali, e decodificarlo nel formato originale. Il flag **-r** indica a xxd di eseguire l'operazione di "**reverse**" (cioè di decodifica), mentre **-p** specifica che i dati devono essere trattati come un flusso di esadecimale puro, senza alcuna formattazione aggiuntiva.

In pratica, grazie a questo comando, i dati esfiltrati, precedentemente codificati in esadecimale, vengono convertiti in un formato leggibile, che può poi essere visualizzato nel file **secret.txt** con il comando **cat secret.txt**. Questo permette di vedere chiaramente il contenuto nascosto nel traffico DNS e di capire quali dati sono stati esfiltrati.

Remediation E Mitigation

Mitigazione dell'attacco

La prima fase è fermare l'attacco in corso. Ciò implica il **blocco immediato** delle comunicazioni DNS sospette. Un modo per farlo è monitorare il traffico DNS in tempo reale e individuare richieste verso domini sospetti o sottodomini anomali. Se identificati, è possibile **bloccare il traffico DNS** che punta a indirizzi IP o domini noti per essere associati a server di comando e controllo (C&C) o a esfiltrazione di dati. Un altro approccio utile è la configurazione di firewall e sistemi di sicurezza per limitare le query DNS a un insieme di server DNS autorizzati, riducendo la possibilità che attaccanti sfruttino DNS non controllati. Inoltre, l'uso di strumenti di rilevamento delle anomalie nel traffico DNS, come **intrusion detection systems (IDS)** che analizzano i modelli di traffico, può aiutare a individuare attività sospette, come le lunghe stringhe esadecimali nei sottodomini.

Remediation dell'incidente

Una volta che l'attacco è stato fermato, è fondamentale intraprendere una serie di azioni per rimuovere le vulnerabilità e prevenire futuri attacchi. La prima cosa da fare è esaminare i sistemi compromessi e determinare l'ampiezza dell'esfiltrazione. Questo implica l'analisi di log e traffico di rete per comprendere i dati sottratti e i metodi utilizzati dagli aggressori per compromettere i sistemi. Successivamente, occorre identificare e rimuovere eventuali malware o strumenti utilizzati dagli attaccanti per generare richieste DNS anomale. È anche importante verificare la sicurezza dei server DNS interni, implementando tecniche di hardening come la configurazione di DNSSEC (Domain Name System Security Extensions), che aiuta a proteggere l'integrità delle risposte DNS e prevenire manipolazioni.

Un altro passo cruciale è rivedere le **configurazioni dei firewall** e dei sistemi di sicurezza per garantire che le comunicazioni DNS siano ben controllate. È fondamentale applicare una segmentazione più rigida della rete e implementare un monitoraggio continuo del traffico DNS per individuare tempestivamente attività sospette.

Infine, l'**educazione e la formazione** continua degli utenti e dei team di sicurezza sono cruciali. Gli utenti devono essere sensibilizzati sui rischi di attacchi di esfiltrazione tramite DNS e sugli indicatori di compromissione (IoC) che potrebbero suggerire un attacco in corso. Gli amministratori di sistema e gli analisti di sicurezza dovrebbero essere formati su come identificare e rispondere rapidamente a minacce legate al traffico DNS anomalo.

In sintesi, la mitigazione e la remediation di un attacco DNS exfiltering richiedono un intervento tempestivo e coordinato che prevede l'analisi dei log, il blocco delle comunicazioni sospette, la rimozione di malware e vulnerabilità, il rafforzamento delle difese DNS e la sensibilizzazione degli utenti. Implementare questi passaggi aiuta a ridurre i rischi di futuri attacchi e a rafforzare la postura di sicurezza complessiva.

Bonus 3

**ISOLATE COMPROMISED
HOST USING 5-TUPLE**

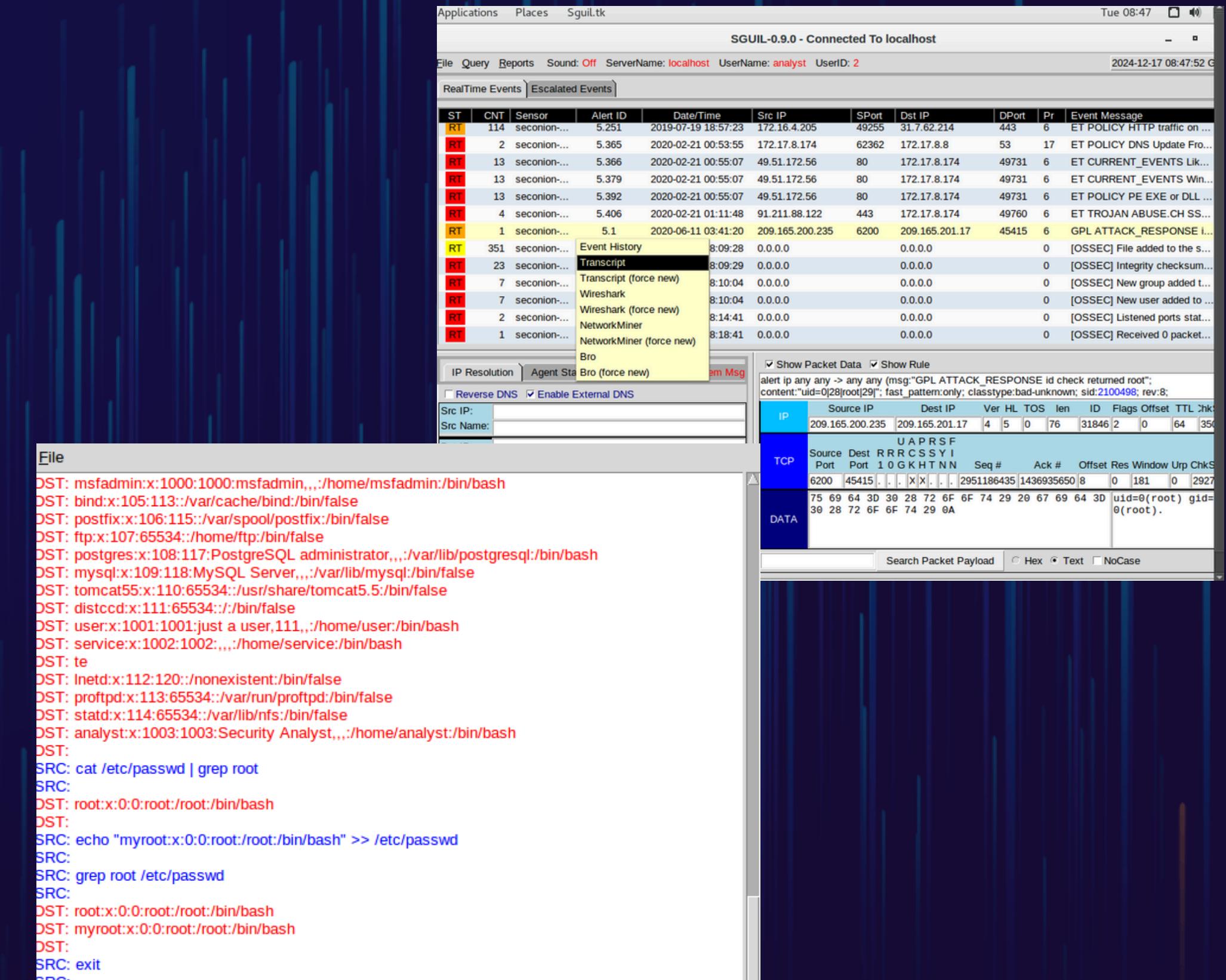
Durante l'analisi di un attacco informatico utilizzando Security Onion, abbiamo esaminato una serie di eventi e attività che hanno rivelato come un attaccante sia riuscito a compromettere un server, acquisire privilegi di root e sottrarre informazioni sensibili. Il processo si è articolato in diverse fasi, sfruttando strumenti come Sguil, Wireshark e Kibana per investigare a fondo l'incidente e comprendere la portata dell'attacco.

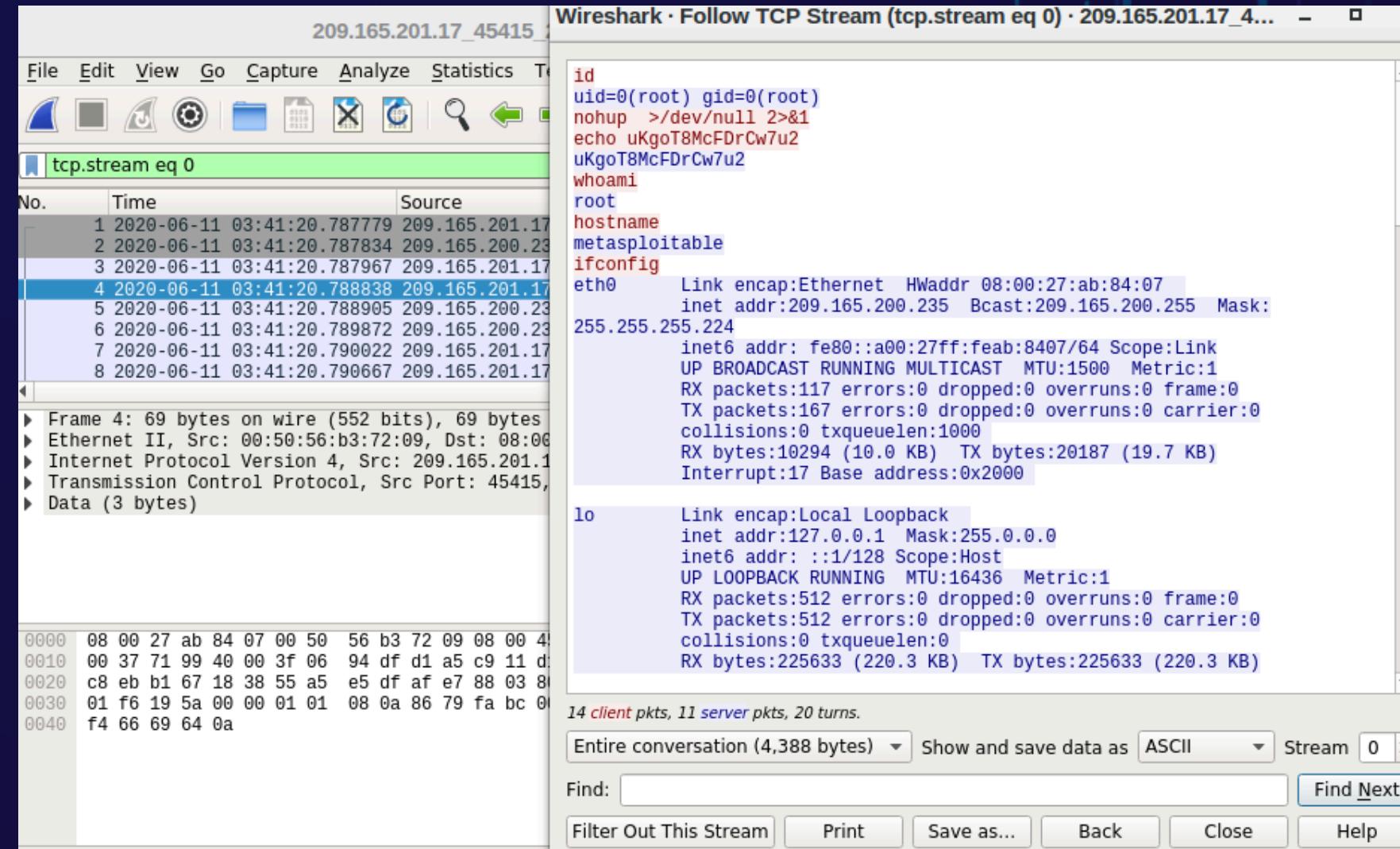
Inizialmente, attraverso Sguil, è stato individuato un allarme particolarmente rilevante, denominato "GPL ATTACK_RESPONSE id check returned root". Questo allarme indicava che l'host con indirizzo IP 209.165.200.235 aveva restituito l'accesso root a un altro host, identificato con l'IP 209.165.201.17.

Approfondendo i dettagli dell'allarme e analizzando i transcript delle comunicazioni, è emerso chiaramente come l'attaccante fosse riuscito a eseguire comandi Linux sul server compromesso.

Tra le operazioni svolte, l'attaccante ha navigato nel file system, copiato il file critico shadow e manipolato file di sistema fondamentali come /etc/shadow e /etc/passwd.

Questi interventi indicano una compromissione completa del sistema, con l'obiettivo di ottenere un controllo persistente.





L'indagine è poi proseguita con l'utilizzo di Wireshark, che ha consentito di analizzare in dettaglio il traffico di rete associato all'attacco. Seguendo lo stream TCP relativo alla connessione tra l'attaccante e il target, è stato possibile confermare che il server compromesso aveva come hostname "metasploitable" e utilizzava l'indirizzo IP 209.165.200.235.

Durante questa fase, è stato osservato che l'attaccante ha eseguito il comando `whoami`, confermando di aver ottenuto privilegi di root sulla macchina. Inoltre, l'analisi del traffico ha rivelato che l'attaccante stava accedendo a dati sensibili, come informazioni sugli account utente.

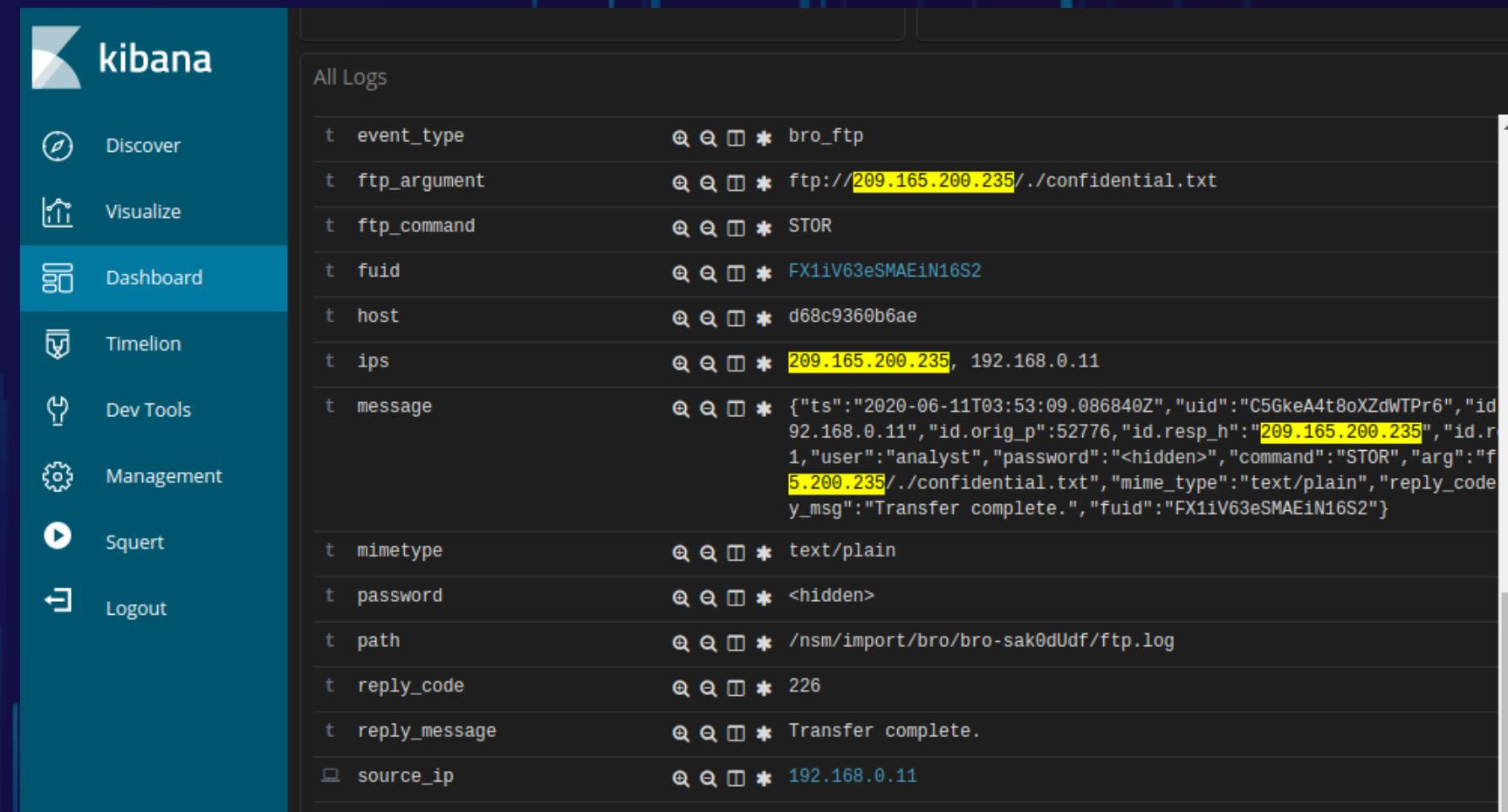
Un ulteriore approfondimento è stato condotto utilizzando Kibana, per analizzare in dettaglio i log relativi alle attività dell'attaccante.

Concentrandoci sul traffico FTP, è stato possibile identificare una serie di operazioni legate all'esfiltrazione di dati.

In particolare, è emerso che l'attaccante aveva scaricato un file denominato confidential.txt dal server compromesso, utilizzando l'indirizzo `ftp://209.165.200.235./confidential.txt`.

```
192.168.0.11:52776 209.165.200.235:21-6-565165033.pcap

Log entry:
{"ts": "2020-06-11T03:53:09.086840Z", "uid": "C5GkeA4t8oXZdWTPr6", "id.orig_h": "192.168.0.11", "id.orig_p": 52776, "id.resp_h": "209.165.200.235", "id.resp_p": 21, "user": "analyst", "password": "<hidden>", "command": "STOR", "arg": "ftp://209.165.200.235./confidential.txt", "mime_type": "text/plain", "reply_code": 226, "reply_msg": "Transfer complete.", "fuid": "FX1iV63eSMAEiN16S2"}  
  
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: 52776  
Dst Port: 21  
OS Fingerprint: 192.168.0.11:52776 - UNKNOWN [S44:63:1:60:M1460,S,T,N,W7...?:?] (up: 3131 hrs)  
OS Fingerprint: -> 209.165.200.235:21 (link: ethernet/modem)  
DST: 220 (vsFTPd 2.3.4)  
DST:  
SRC: USER analyst  
SRC:  
DST: 331 Please specify the password.  
DST:  
SRC: PASS cyberops  
SRC:  
DST: 230 Login successful.  
DST:  
SRC: SYST  
SRC:  
DST: 215 UNIX Type: L8  
DST:  
SRC: TYPE I  
SRC:  
DST: 200 Switching to Binary mode.  
DST:  
SRC: PORT 192.168.0.11,194,153  
SRC:
```



All Logs	
t event_type	bro_ftp
t ftp_argument	ftp://209.165.200.235./confidential.txt
t ftp_command	STOR
t fuid	FX1iV63eSMAEiN16S2
t host	d68c9360b6ae
t ips	209.165.200.235, 192.168.0.11
t message	{"ts": "2020-06-11T03:53:09.086840Z", "uid": "C5GkeA4t8oXZdWTPr6", "id.orig_h": "192.168.0.11", "id.orig_p": 52776, "id.resp_h": "209.165.200.235", "id.resp_p": 21, "user": "analyst", "password": "<hidden>", "command": "STOR", "arg": "ftp://209.165.200.235./confidential.txt", "mime_type": "text/plain", "reply_code": 226, "reply_msg": "Transfer complete.", "fuid": "FX1iV63eSMAEiN16S2"}
t mimetype	text/plain
t password	<hidden>
t path	/nsm/import/bro/bro-sak0dUdf/ftp.log
t reply_code	226
t reply_message	Transfer complete.
source_ip	192.168.0.11

Le credenziali utilizzate per accedere al server FTP erano `analyst` come username e `cyberops` come password. Questo dimostra che l'attaccante ha sfruttato credenziali valide per ottenere l'accesso, il che solleva interrogativi sulla sicurezza e sull'implementazione delle misure di protezione delle credenziali nella rete.

192.168.0.11:49817_209.165.200.235:20-6-1644278317.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": ["C2Jv8MWV6Xg4Ibb51"], "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": "f7f54acee0342f6161f8e63a10824ee11b330725"}
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 0.40 seconds: 0.08 0.18 0.00 0.15 0.00

[192.168.0.11:49817_209.165.200.235:20-6-1644278317.pcap](#)

Analizzando il contenuto del file esfiltrato, è stato scoperto che esso conteneva informazioni sensibili riguardanti una violazione di sicurezza precedente.

Il trasferimento del file è avvenuto l'11 giugno 2020 alle 3:53, utilizzando un protocollo FTP non sicuro. Inoltre, i log hanno rivelato che, dopo aver copiato il file, l'attaccante lo ha cancellato dal server, probabilmente per nascondere le tracce del furto.

Considerazioni

L'intera analisi ha evidenziato diverse falle di sicurezza nella rete e nei sistemi coinvolti. L'utilizzo di credenziali statiche e facilmente accessibili rappresenta un grave rischio, soprattutto se associate a protocolli di comunicazione non sicuri come l'FTP. Per mitigare futuri attacchi, è indispensabile adottare misure correttive immediate, come il cambio delle password per gli account compromessi, l'implementazione di protocolli sicuri come SFTP e una gestione rigorosa dei privilegi di accesso.

Inoltre, un monitoraggio costante delle attività di rete e dei file di sistema è fondamentale per identificare tempestivamente eventuali comportamenti sospetti. In conclusione, questo attacco mette in evidenza la necessità di un approccio proattivo alla sicurezza informatica. Solo attraverso una combinazione di misure preventive, strumenti di monitoraggio avanzati e revisione periodica delle configurazioni di sistema è possibile proteggere efficacemente le risorse critiche da minacce simili.