

BW3- Esercizio Bonus 1

Indagine su Exploit SQLi ed Esfiltrazione Dati via DNS

Autore: Cybereagles

Sintesi

Il presente report documenta l'attività di **digital forensics** e **incident response** focalizzata sull'analisi di **log di traffico di rete**. L'obiettivo primario è stato l'individuazione e la validazione di una doppia compromissione di sistema: lo sfruttamento di una vulnerabilità di **SQL Injection** veicolata tramite protocollo **HTTP** e la successiva **esfiltrazione occulta di dati sensibili** (PII) perpetrata attraverso tecniche di **DNS Tunneling**. L'analisi ha confermato il successo di entrambi i vettori d'attacco, portando all'esposizione di informazioni finanziarie e documenti confidenziali aziendali al di fuori del perimetro di sicurezza.

Scopo del test e analisi dello scenario

Scenario e Obiettivi

L'attività investigativa è stata condotta all'interno dell'ambiente operativo virtualizzato **Security Onion**, impiegato per l'analisi profonda dei pacchetti e l'ispezione dei log di sicurezza. L'obiettivo è rintracciare la catena d'attacco, ricostruire i payload malevoli e dimostrare l'avvenuta **violazione dei dati (Data Breach)**. Gli attori di rete individuati durante le indagini sono strutturati come segue:

- **Attacker:** Esterno (IP 209.165.200.227) utilizzato per la fase offensiva di **SQL Injection**.
- **Target:** Server Web aziendale compromesso (IP 209.165.200.235) con focus sulla vulnerabilità dell'applicativo web esposto sulla porta **TCP/80**.
- **Attacker (DNS Tunneling):** Workstation interna compromessa (IP 192.168.0.11) utilizzata come origine dell'esfiltrazione.
- **Target (DNS Tunneling):** Server DNS controllato dall'attore della minaccia (IP 209.165.200.235) impiegato per la ricezione occulta dei dati rubati.

Strumenti

- **Kibana:** Piattaforma di data visualization utilizzata per esplorare, filtrare e analizzare i log degli eventi di sicurezza generati dai sensori di rete (es. Zeek) in modo interattivo e granulare.
 - **capME!:** Strumento di ispezione integrato per la trascrizione completa e l'analisi dettagliata delle interazioni client-server estratte direttamente dai file di cattura **pcap**.
 - **xxd:** Utility da riga di comando nativa in ambiente Linux impiegata per la manipolazione di hexdump e per la decodifica rapida in formato testuale ASCII dei payload malevoli codificati.
-

Svolgimento

Fase 1: Isolamento del traffico web sospetto (SQL Injection)

L'indagine ha preso avvio nell'interfaccia di **Kibana** rimodulando il parametro temporale assoluto (*Absolute Time Range*) per coprire gli eventi registrati nel mese di giugno 2020. Per circoscrivere i log pertinenti al potenziale attacco, è stato applicato un filtro mirato sul traffico *HTTP* all'interno del modulo **Zeek Hunting**. L'analisi dei widget grafici ha permesso di identificare immediatamente il traffico anomalo generato dall'indirizzo IP sorgente verso l'indirizzo IP destinazione, scambiato esclusivamente in chiaro sulla porta *TCP/80*.

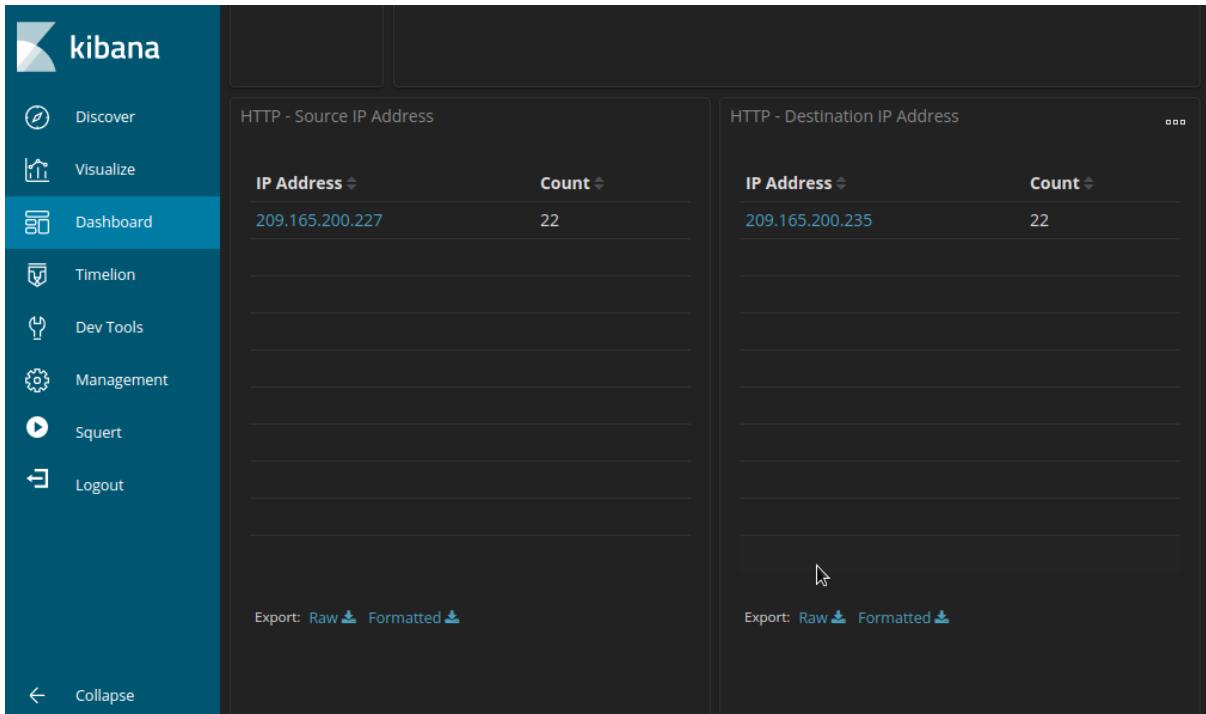


Figura 1 Visualizzazione in Kibana degli indirizzi IP sorgente e destinazione relativi al traffico HTTP anomalo isolato tramite filtro Zeek Hunting.

Fase 2: Ispezione del payload malevolo e decodifica PCAP

Si è proceduto all'ispezione di dettaglio del primo log *HTTP* registrato nei sistemi (timestamp: June 12th 2020, 21:30:09.445). L'espansione del record e l'analisi del campo *message* hanno rivelato l'iniezione, via URI, di un payload tipico degli attacchi **SQL Injection**, chiaramente evidenziato dalla presenza della sintassi `+union+select+...` inserita dopo il campo *username*. Tramite l'estrapolazione dell'identificativo *alert_id*, la sessione è stata investigata nello strumento **capME!** per l'analisi della trascrizione *pcap*. Esaminando la risposta del server web in chiaro in seguito alla richiesta *GET* verso la pagina */mutillidae/index.php*, è stata verificata l'avvenuta esfiltrazione di informazioni sensibili (PII): i campi originariamente destinati alle credenziali mostravano a schermo numeri di carte di credito (es. 4444111122223333) e codici CVV (es. 745) forzati dal database.

209.165.200.227:56194_209.165.200.235:80-6-1312669804.pcap

```
Log entry:  
{"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_dept_h": 1, "method": "GET", "host": "209.165.200.235", "uri": "/mutillidae/index.php?page=user-info.php&username=%union+select+ccid,ccnumber,ccv,expiration,null+from+credit_cards+-+&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": ["FEVWs63HqvCqth3LH1"], "resp_mime_types": ["text/html"]}  
  
Sensor Name: seconion-import  
Timestamp: 2020-06-12 21:30:09  
Connection ID: CLI  
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 [S44:64:1:60:M1460,S,T,N,W?::?:?] (up: 2829 hrs)  
OS Fingerprint: -> 209.165.200.235:80 (link: ethernet/modem)  
SRC: GET /mutillidae/index.php?page=user-info.php&username=%27+union+select+ccid%2Cccnumber%2Cccv%2Cexpiration%2Cnull+from+credit_cards+-+&password=&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/mutillidae/index.php?page=user-info.php  
SRC: Connection: keep-alive  
SRC: Cookie: PHPSESSID=9fd8860958f924a43cd529dc4120d1cb  
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/5.2.4-2ubuntu5.10  
DST: Expires: Thu, 19 Nov 1981 08:52:00 GMT  
DST: Logged-In-User:  
DST: Cache-Control: public  
  
DST: <b>User</b>name=</b>4444111122223333<br>  
DST:  
DST: 17  
DST: <b>Password=</b>745<br>  
DST:  
DST: 22  
DST: <b>Signature=</b>2012-03-01<br><p>  
DST:  
DST: 24  
DST: <b>User</b>name=</b>7746536337776330<br>  
DST:  
DST: 17  
DST: <b>Password=</b>722<br>  
DST:  
DST: 22  
DST: <b>Signature=</b>2015-04-01<br><p>  
---
```

Figura 2 Interfaccia di capME! che mostra la trascrizione della sessione HTTP e il dump a schermo dei dati sensibili delle carte di credito esfiltrate in chiaro.

Fase 3: Investigazione sull'esfiltrazione occulta via DNS

La seconda fase dell'analisi si è concentrata sulla ricerca di un'eventuale esfiltrazione basata su **DNS Tunneling**. Ritornando alla dashboard di **Kibana**, è stato selezionato il modulo relativo al traffico *DNS* per identificare anomalie. È stato individuato un volume sospetto di richieste verso il dominio specifico *example.com*. Applicando un filtro su tale FQDN, è stata isolata la comunicazione diretta tra il client interno compromesso (192.168.0.11) e il server ricevitore (209.165.200.235). L'esame approfondito della tabella *DNS - Queries* ha portato alla luce la presenza di **falsi sottodomini** strutturati come lunghe stringhe codificate in formato esadecimale e dirette al nameserver *ns.example.com*.

Figura 3 Tabella delle query DNS in Kibana evidenziante i falsi sottodomini anomali in formato esadecimale associati al dominio example.com.

Fase 4: Decodifica del payload esfiltrato

Al fine di estrarre il reale contenuto mascherato nei pacchetti di rete, le query grezze sono state esportate localmente in formato CSV e accuratamente ripulite dai metadati per isolare unicamente il payload esadecimale all'interno del file *DNS Queries.csv*. Utilizzando la shell integrata nella macchina virtuale e l'utilità **xxd**, si è proceduto alla conversione del flusso dati in testo ASCII. È stato lanciato il comando `$ xxd -r -p "DNS Queries.csv" > secret.txt`, reindirizzando l'output finale in un file di testo. La lettura del file *secret.txt* ha rivelato la stringa "CONFIDENTIAL DOCUMENT DO NOT SHARE This document contains information about the last security breach", confermando in maniera inequivocabile l'uso illecito del protocollo DNS come canale di esfiltrazione (Data Exfiltration).

```

analyst@SecOnion:~$ xxd -r -p "DNS - Queries.csv" > secret.txt
xxd: DNS - Queries.csv: No such file or directory
analyst@SecOnion:~$ cd Downloads/
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$ sudo cat secret.txt
[sudo] password for analyst:
CONFIDENTIAL DOCUMENT
DO NOT SHARE
This document contains information about the last security breach.
analyst@SecOnion:~/Downloads$ 

```

Figura 4 Output del terminale Linux che mostra la decodifica delle stringhe esadecimale tramite xxd e il contenuto in chiaro del documento confidenziale esfiltrato.

Conclusioni

L'attività di indagine ha confermato l'avvenuta compromissione dell'infrastruttura aziendale attraverso due distinti vettori di attacco. È stato dimostrato lo sfruttamento critico di vulnerabilità applicative web mediante tecniche di **SQL Injection**, con conseguente esposizione pubblica di dati finanziari (Carte di Credito). Successivamente, è stata provata l'esecuzione di uno script malevolo in grado di effettuare tattiche evasive di **Data Exfiltration**, aggirando completamente i controlli perimetrali tradizionali mediante la creazione di un **DNS Tunneling**. L'impatto sulla riservatezza e sull'integrità del capitale informativo risulta severo.

Si raccomandano le seguenti azioni di mitigazione:

- **Sanitizzazione degli Input (Prepared Statements):** Implementare urgentemente controlli di validazione e sanificazione rigorosi sulle variabili passate via URL nell'applicazione web, adottando query parametrizzate per separare il codice logico dai dati inseriti dall'utente e disinnescare ulteriori attacchi **SQLi**.
- **Web Application Firewall (WAF):** Installare e configurare un **WAF** a protezione dei servizi esposti su porta *TCP/80* per intercettare pattern malevoli noti (es. firme associate a union select) e bloccare le richieste anomale prima che raggiungano il web server.
- **Controllo e Monitoraggio del Traffico DNS:** Disabilitare le risoluzioni DNS libere verso l'esterno per le macchine non autorizzate e implementare un sistema di **DNS Filtering/IPS**. Questo dovrà essere configurato per monitorare e bloccare interrogazioni caratterizzate da elevata entropia e lunghezza anomala dei sottodomini, tipiche del **DNS Tunneling (C2)**.