S10-L5
Splunk

Emanuele Benedetti | 14 febbraio 2025

Consegna

Obiettivo

Lo scopo di questo esercizio è analizzare il seguente file di log con Splunk e

1. Notare e documentare evidenze di anomalie o attacchi
2. Preparare un report tecnico dettagliato per spiegare le remediation (non voglio tipo "installare MFA" ma voglio dei passi per implementare una soluzione, a livello pratico)
3. Creare una conclusione per i manager che indica brevemente le anomalie/attacchi rilevati e delle remediation per coprire le situazioni (descritte dal file log)

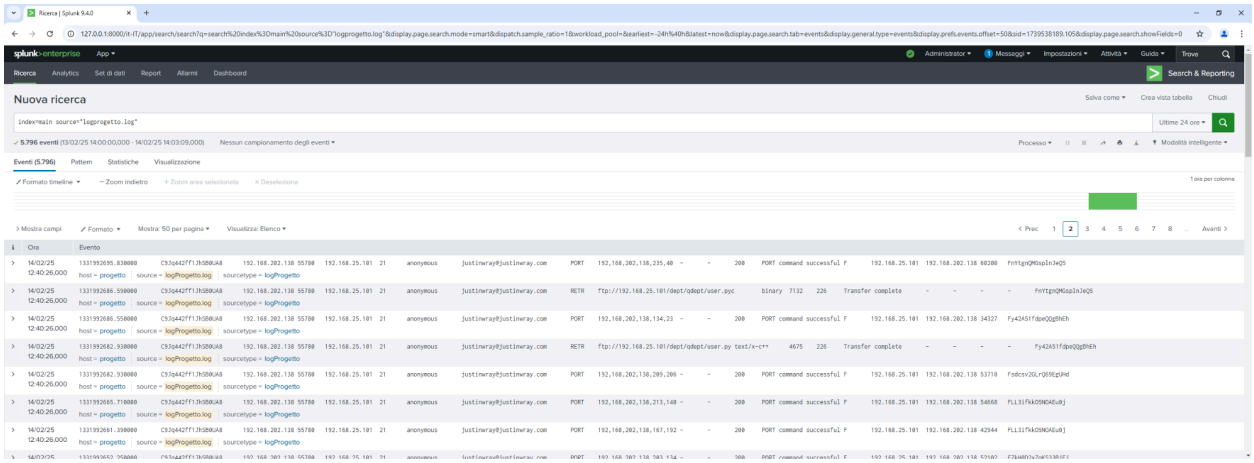
Svolgimento

Ho iniziato il laboratorio accedendo al server Splunk installato su macchina virtuale Windows 10 tramite la URL 127.0.0.1:8000. Dopo aver inserito le credenziali di accesso dell'account enterprise, ho aggiunto il file log da analizzare andando su *Aggiungi dati* > *Carica*.

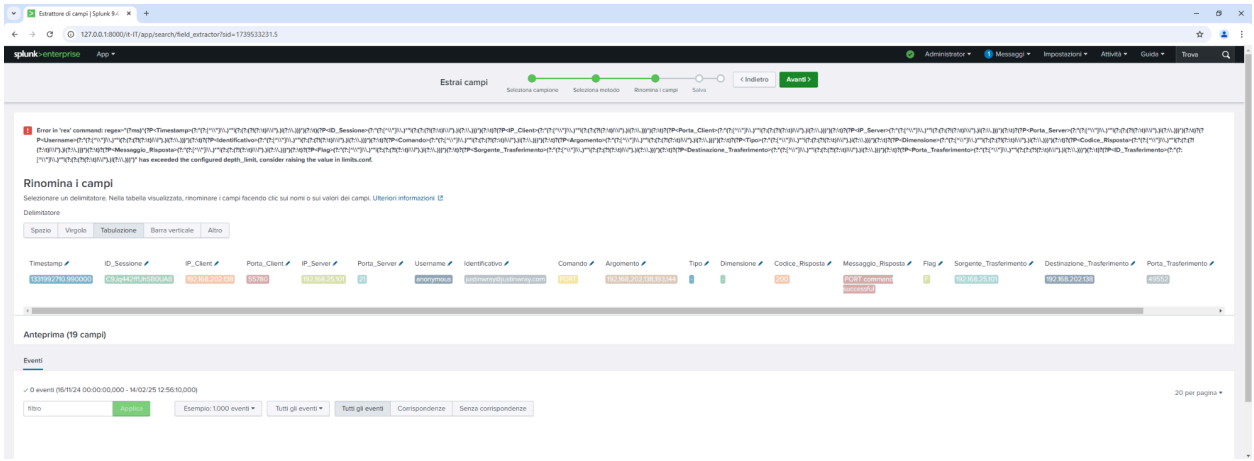
Dopo aver importato, prima di eseguire un'analisi approfondita ho letto il file per cercare di capire quali dati fossero stati raccolti.

In generale appare evidente che il file di log contiene registrazioni di attività FTP,

con l'esecuzione di vari comandi come *PASV*, *PORT*, *RETR*, *APPE*, *STORE*, *DEL*.

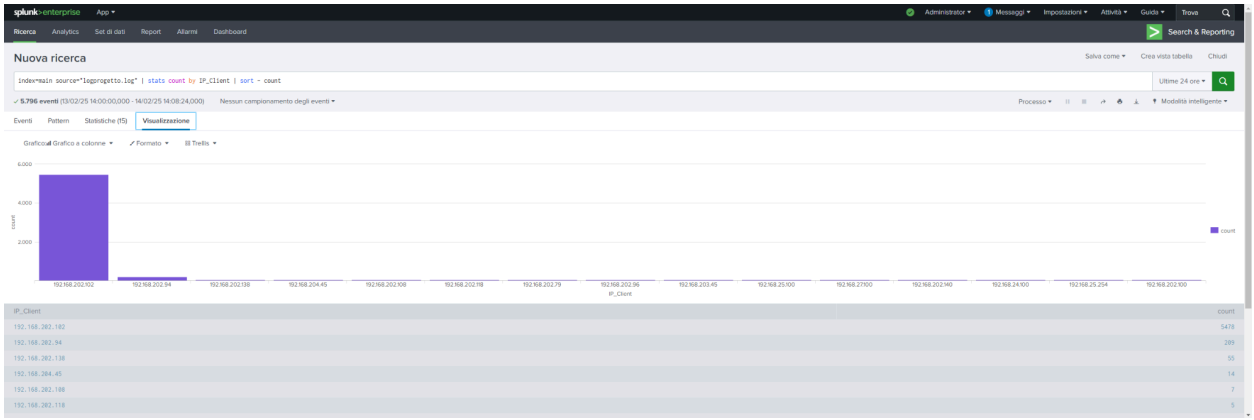


Per poter analizzare meglio il file tuttavia ho estratto ulteriori campi che Splunk non è riuscito ad estrarre automaticamente. Ho selezionato un evento di esempio ed ho rinominato i campi di interesse come mostrato nello screenshot che segue

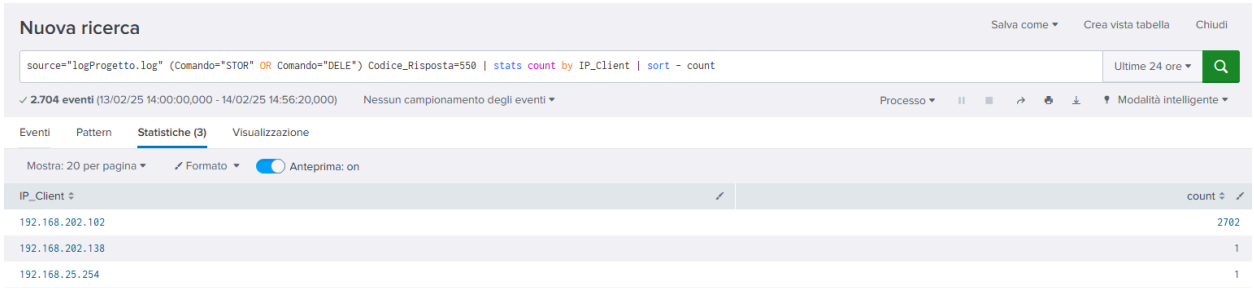


Ho quindi avviato la mia analisi usando la query *index=main source="logProgetto.log" | stats count by IP_Client | sort - count* per capire quali fossero gli indirizzi IP con maggiore traffico e potenzialmente pericolosi.

Il risultato della ricerca con la query inserita, mostra che su 5796 eventi registrati, 5478 provengono dallo stesso indirizzi IP (192.168.202.102)



Ho tentato poi di filtrare le entry sulla base dei codici di risposta FTP 550 che indica accesso negato, con la query *source="logProgetto.log" (Comando="STOR" OR Comando="DELE") Risposta=550 | stats count by IP_Client | sort - count*



Il risultato mostra che 2702 eventi sono generati dallo stesso IP precedentemente individuato.

Ho quindi approfondito questa ricerca con *source="logProgetto.log" (Comando="STOR" OR Comando="DELE") Risposta=550 | table _time IP_Client IP_Server Username Comando Argomento* per individuare tentativi di scrittura o eliminazione falliti, che potrebbero indicare tentativi di compromissione.

In particolare la risposta mostra che il client 192.168.202.102 ha tentato più volte di scrivere e cancellare file in diverse directory sul server 192.168.21.101, il che potrebbe indicare un comportamento malevolo, come un tentativo di stabilire una backdoor.

Nuova ricerca

Salva come | Crea vista tabella | Chiudi

```
source="logProgetto.log" (Comando="STOR" OR Comando="DELE") Codice_Risposta=550 | table _time IP_Client IP_Server Username Comando Argomento
```

Ultime 24 ore

✓ **2.704 eventi** (13/02/25 15:00:00,000 - 14/02/25 15:03:46,000) Nessun campionamento degli eventi

Processo | Modalità intelligente

Eventi | Pattern | **Statistiche (2.704)** | Visualizzazione

Mostra: 20 per pagina | Formato | Anteprima: on

< Prec 1 2 3 4 5 6 7 8 ... Avanti >

_time	IP_Client	IP_Server	Username	Comando	Argomento
2025-02-14 12:40:26	192.168.202.102	192.168.22.101	ftp	STOR	ftp://192.168.22.101/dept/env/lib/python2.7/site-packages/Flask_SQLAlchemy-0.15-py2.7.egg-info/.ftpdBUL2Da
2025-02-14 12:40:26	192.168.202.102	192.168.22.101	ftp	DELE	ftp://192.168.22.101/dept/env/lib/python2.7/site-packages/Flask_SQLAlchemy-0.15-py2.7.egg-info/.ftpdBUL2Da
2025-02-14 12:40:26	192.168.202.102	192.168.22.101	ftp	STOR	ftp://192.168.22.101/dept/env/lib/python2.7/site-packages/Flask_Login-0.1-py2.7.egg-info/.ftpdBUL2Da
2025-02-14 12:40:26	192.168.202.102	192.168.22.101	ftp	DELE	ftp://192.168.22.101/dept/env/lib/python2.7/site-packages/Flask_Login-0.1-py2.7.egg-info/.ftpdBUL2Da
2025-02-14 12:40:26	192.168.202.102	192.168.22.101	ftp	STOR	ftp://192.168.22.101/dept/env/lib/python2.7/site-packages/Flask_Celery-2.4.1-py2.7.egg-info/.ftpdBUL2Da
2025-02-14 12:40:26	192.168.202.102	192.168.22.101	ftp	DELE	ftp://192.168.22.101/dept/env/lib/python2.7/site-packages/Flask_Celery-2.4.1-py2.7.egg-info/.ftpdBUL2Da
2025-02-14 12:40:26	192.168.202.102	192.168.22.101	ftp	STOR	ftp://192.168.22.101/dept/env/lib/python2.7/site-packages/Flask-0.8-py2.7.egg-info/.ftpdBUL2Da
2025-02-14 12:40:26	192.168.202.102	192.168.22.101	ftp	DELE	ftp://192.168.22.101/dept/env/lib/python2.7/site-packages/Flask-0.8-py2.7.egg-info/.ftpdBUL2Da
2025-02-14 12:40:26	192.168.202.102	192.168.22.101	ftp	STOR	ftp://192.168.22.101/dept/qdept/static/css/blueprint/.ftpdBUL2Da
2025-02-14 12:40:26	192.168.202.102	192.168.22.101	ftp	DELE	ftp://192.168.22.101/dept/qdept/static/css/blueprint/.ftpdBUL2Da
2025-02-14 12:40:26	192.168.202.102	192.168.22.101	ftp	STOR	ftp://192.168.22.101/dept/env/lib/python2.7/site-packages/.ftpdBUL2Da
2025-02-14 12:40:26	192.168.202.102	192.168.22.101	ftp	DELE	ftp://192.168.22.101/dept/env/lib/python2.7/site-packages/.ftpdBUL2Da
2025-02-14 12:40:26	192.168.202.102	192.168.22.101	ftp	STOR	ftp://192.168.22.101/dept/env/lib/python2.7/distutils/.ftpdBUL2Da
2025-02-14 12:40:26	192.168.202.102	192.168.22.101	ftp	DELE	ftp://192.168.22.101/dept/env/lib/python2.7/distutils/.ftpdBUL2Da
2025-02-14 12:40:26	192.168.202.102	192.168.22.101	ftp	STOR	ftp://192.168.22.101/dept/qdept/static/images/.ftpdBUL2Da
2025-02-14 12:40:26	192.168.202.102	192.168.22.101	ftp	DELE	ftp://192.168.22.101/dept/qdept/static/images/.ftpdBUL2Da

Quello che emerge da questo screenshot è che

- l'utente ftp sta cercando di caricare (*STOR*) e cancellare (*DELE*) file nascosti (*.ftpdBUL2Da*) in percorsi specifici che contengono librerie Python. Il file *.ftpdBUL2Da* potrebbe essere un tentativo di nascondere un payload malevolo all'interno di librerie Python
- Le operazioni di *STOR* e *DELE* vengono bloccate (*Codice_Risposta=550*). Infatti il messaggio *550 Operation not permitted* indica che il server ha impedito la scrittura e la cancellazione di questi file

Dato la mole di traffico generato e la tipologia di traffico, il client 192.168.202.102 con utente ftp è da considerarsi altamente sospetto e probabilmente compromesso.

splunk>enterprise | App | Administrator | 1 Messaggi | Impostazioni | Attività | Guida | Trova

Ricerca | Analytics | Set di dati | Report | Allarmi | Dashboard | Search & Reporting

Nuova ricerca

Salva come | Crea vista tabella | Chiudi

```
source="logProgetto.log" Argomento="*.ftpdBUL2Da" NOT Codice_Risposta=550
```

Ultime 24 ore

✓ **0 eventi** (13/02/25 15:00:00,000 - 14/02/25 15:17:05,000) Nessun campionamento degli eventi

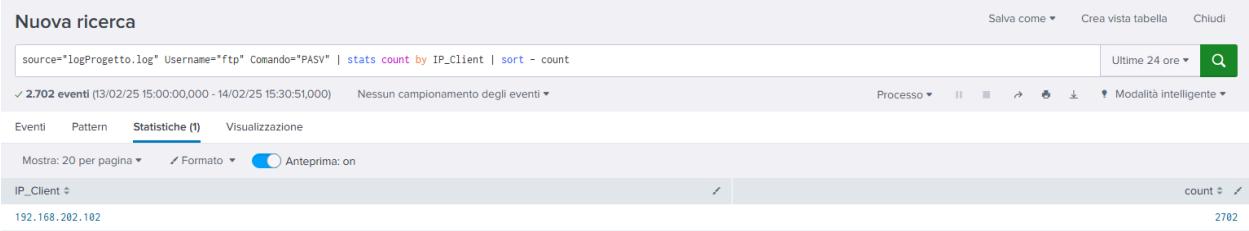
Processo | Modalità intelligente

Eventi (0) | Pattern | Statistiche | Visualizzazione

Nessun risultato trovato. Provare ad ampliare l'intervallo temporale

Lo screenshot sopra mostra l'utilizzo della query *source="logProgetto.log" Argument="*.ftpdBUL2Da*" NOT Codice_Risposta=550* per verificare che l'host non sia ancora riuscito a caricare il file sul server.

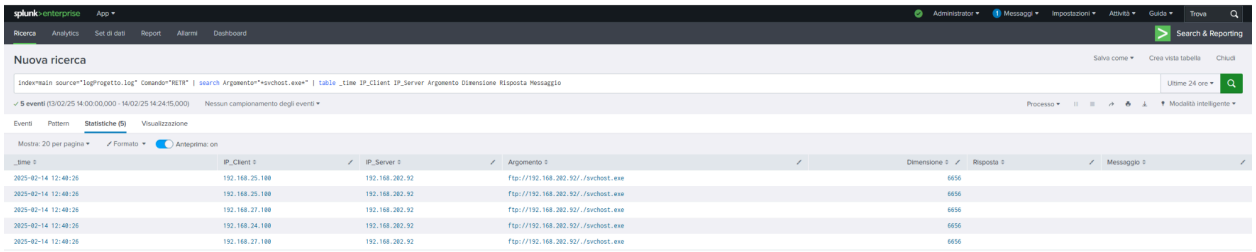
Ho considerato sospetto anche il ricorrente utilizzo (2702 eventi) del comando FTP PASV.



PASV (passive mode) è il comando utilizzato nel protocollo FTP per stabilire la modalità di trasferimento passiva tra client e server. In modalità passiva, è il server ad aprire una porta casuale e comunicare al client quale porta usare per il trasferimento dei dati.

Cercando le tipologie di attacco ho infatti visto che questa tecnica viene utilizzata per attacchi FTP tunneling in cui viene sfruttata la modalità passiva per aggirare firewall e creare canali di comunicazione nascosti.

Un altro indicatore o anomalia che ho trovato è il download del file *svchost.exe* che corrisponde ad un processo comune su Windows ma viene spesso utilizzato dai malware per mascherarsi. Lo screen mostra la query utilizzata per filtrare i client che hanno avuto accesso al file e potrebbe indicare una compromissione del server 192.168.202.92



Ho infine notato che è attivo l'utente *anonymous*, un utente speciale che permette di accedere al server senza autenticazione. Viene spesso utilizzato per download pubblici ma potrebbe essere un rischio per la sicurezza.

Proposta di remediation tecniche

1. **Blocco dell'attività sospetta del client 192.168.202.102**

Obiettivo: Prevenire ulteriori tentativi di caricamento/eliminazione di file dannosi e indagare sul dispositivo compromesso.

Isolamento del client

- Rimuovere immediatamente il client 192.168.202.102 dalla rete interna per prevenire ulteriori interazioni con i server FTP.
- Eseguire una scansione completa del sistema con strumenti antivirus
- Verificare la presenza di malware, rootkit o script automatizzati che potrebbero essere stati utilizzati per generare i comandi FTP sospetti.

Analisi

- Esaminare i log del sistema operativo del client (Event Viewer su Windows o /var/log/ su Linux) per identificare processi anomali o connessioni sospette.
- Controllare eventuali script o programmi in esecuzione che potrebbero aver generato i comandi STOR, DELE e PASV
- Verificare se il client è stato compromesso da un attacco di tipo "brute force" o da un exploit.

Blocco del traffico FTP dal client

- Configurare il firewall della rete per bloccare tutto il traffico FTP proveniente dall'IP 192.168.202.102 verso i server FTP.
- Implementare regole di filtraggio basate su indirizzi IP e porte FTP (tipicamente 21 per il controllo e 20 per i dati).

Monitoraggio continuo del traffico

- Implementare un IDS/IPS (Intrusion Detection/Prevention System) come Snort o Suricata per monitorare il traffico FTP e rilevare comportamenti anomali in tempo reale.

2. **Mitigazione del server FTP compromesso (192.168.202.92)**

Obiettivo: Rimuovere il malware e prevenire ulteriori download di svchost.exe.

Isolamento del server

- Rimuovere il server FTP 192.168.202.92 dalla rete interna per evitare ulteriori download di svchost.exe
- Eseguire una scansione completa del filesystem del server con strumenti antivirus avanzati per individuare e rimuovere il file svchost.exe

Rimozione del file dannoso

- Individuare il percorso esatto del file svchost.exe sul server FTP
- Eliminare il file e verificare che non ci siano copie nascoste o varianti del malware.

Implementazione di autenticazione sicura

- Abilitare l'autenticazione basata su utenti registrati con password complesse.
- Considerare l'uso di certificati SSL/TLS per cifrare le comunicazioni FTP (FTPS).

3. **Disattivazione accesso anonimo**

E' necessario modificare la configurazione dei server FTP per disabilitare l'accesso anonymous.

- Modificare il file di configurazione con *sudo nano /etc/vsftpd.conf*
- Impostare *anonymous_enable=NO*
- Riavviare il servizio con *sudo systemctl restart vsftpd*

Relazione e conclusione per i manager

Anomali ed attacchi rilevati

1. **Attività sospetta del client 192.168.202.102**
    - Il client ha eseguito un elevato numero di tentativi di caricamento (STOR) e eliminazione (DELE) di file potenzialmente dannosi su uno o più server FTP, fallendo a causa di permessi insufficienti (codice 550).
    - L'uso frequente del comando PASV suggerisce un tentativo di bypassare il firewall

2. **Server FTP 192.168.202.92 compromesso**
    - Il server distribuisce il file svchost.exe, notoriamente associato a malware, ai client che lo scaricano
    - L'accesso anonimo è abilitato, aumentando il rischio di accessi non autorizzati.

Remediation Implementate

1. **Per il client 192.168.202.102**
    - Isolamento dalla rete e scansione antivirus per identificare eventuali malware o script automatizzati
    - Blocco del traffico FTP dal client tramite firewall per prevenire ulteriori tentativi di attacco

2. **Per il server FTP 192.168.202.92 compromesso**
    - Rimozione del file dannoso (svchost.exe) e disattivazione dell'accesso anonimo
    - Migrazione da FTP a SFTP per garantire comunicazioni cifrate e autenticazione sicura

3. **Miglioramenti generali**
    - Implementazione di monitoraggio avanzato (IDS/IPS) per rilevare comportamenti anomali in tempo reale.
    - Limitazione dell'accesso ai server FTP solo agli utenti fidati e revisione delle autorizzazioni utente.

È consigliabile condurre un'analisi forense approfondita per identificare la causa principale dell'incidente e valutare l'adozione di soluzioni di trasferimento file più sicure, come SFTP o HTTPS, per sostituire completamente il protocollo FTP.