# Windows Server 2022 con IDS/IPS pfSense, Splunk e Bot Telegram

**Data:** 22/09/2025

**analista:** gianluca agostini

---

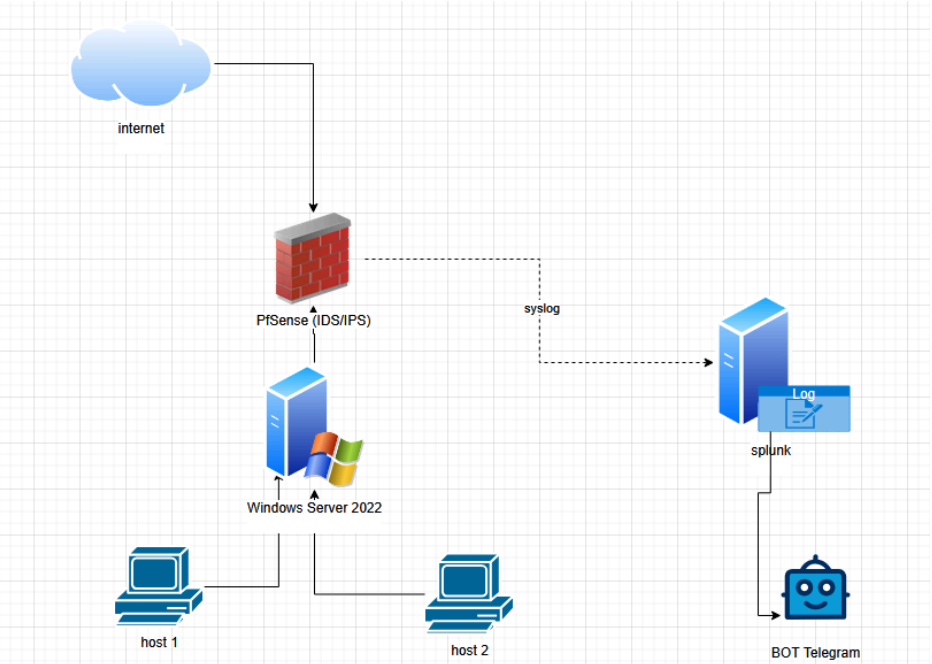
# 1. Introduzione

L’obiettivo di questo lavoro è stato quello di unire un esercizio didattico di gestione dei gruppi in Windows Server 2022 con la progettazione di una piccola architettura di sicurezza basata su tecnologie enterprise. In particolare, oltre alla creazione di gruppi di utenti e all’assegnazione dei relativi permessi, è stata realizzata un’integrazione con:

- pfSense configurato come IDS/IPS per il monitoraggio e la protezione del traffico di rete.
- Splunk per la raccolta, l’indicizzazione e l’analisi centralizzata dei log.
- Bot Telegram collegato a Splunk per l’invio in tempo reale di notifiche sugli eventi critici (es. alert di Suricata con priorità 1).

Questa architettura, pur in scala ridotta e a scopo dimostrativo, mostra come la gestione dei gruppi in un dominio Windows possa essere inserita in un contesto più ampio di sicurezza informatica, dove il controllo degli accessi si combina con strumenti di monitoraggio, logging e automazione degli allarmi.

# 2. Topologia della rete




La topologia implementata prevede un'architettura a livelli volta a garantire sicurezza e controllo.

- **Perimetro**: il traffico proveniente da Internet è filtrato da un **firewall con funzionalità IDS/IPS integrate (pfSense con Suricata)**. Questo componente rappresenta la prima linea di difesa, gestendo il controllo degli accessi e l'ispezione del traffico in tempo reale.

- **Zona interna**: dietro il firewall è collocato un **server Windows/Linux** che gestisce i servizi interni e funge da punto di snodo per le comunicazioni verso gli host. Gli **endpoint interni** risultano segmentati per garantire isolamento logico e limitare l'impatto in caso di compromissione.

- **Logging e analisi**: tutti gli eventi rilevati vengono inviati a una piattaforma di **log management centralizzata (Splunk)**, che consente correlazione, filtraggio e visualizzazione dei dati.

- **Alerting**: a valle della correlazione, gli eventi di sicurezza critici generano notifiche automatiche verso un **canale dedicato (Telegram Bot)**, garantendo tempi di reazione rapidi.

Questa architettura permette di integrare controllo perimetrale, analisi centralizzata e meccanismi di risposta automatizzata, migliorando il livello complessivo di sicurezza della rete.

# 3. Analisi Tecnica

## 1.Windows Server 2022 e Host

### 1.Preparazione

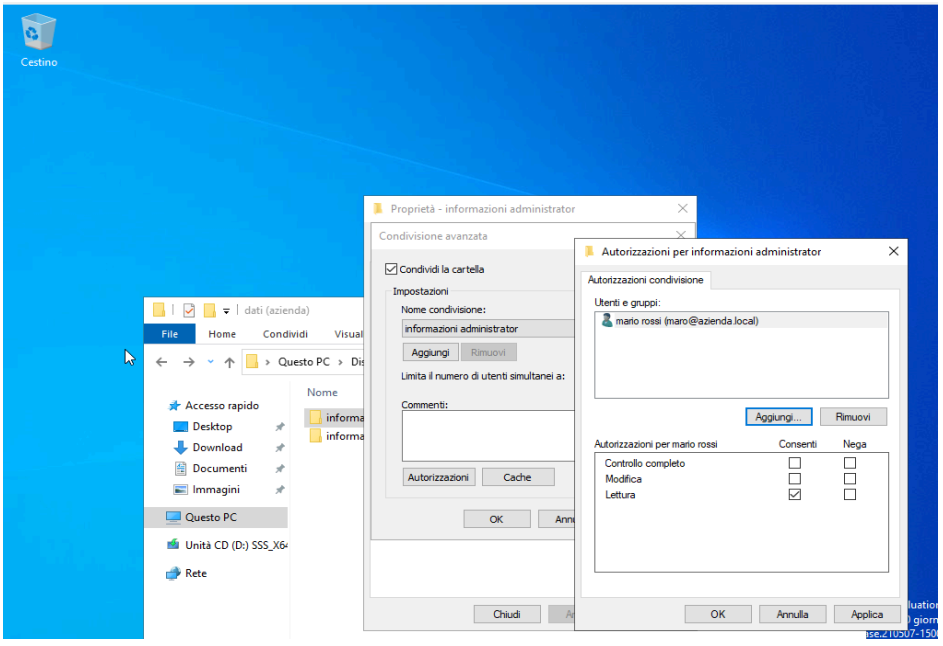
- Installazione Windows Server 2022.
- Configurato IP statico `192.168.61.10` con DNS che punta su se stesso.
- Rinomina della macchina in **server** tramite Server Manager.

### 2.Creazione Dominio e AD DS

- Installato ruolo **Active Directory Domain Services**.
- Creata nuova foresta con dominio `azienda.local`.
- Configurate le **OU**:
    - **OU Administrator** → utente *Mario Rossi* (privilegi Domain Admin).
    - **OU Utenti Generali** → utenti *Chiara Zucchetti* e *Paolo Foglio* (permessi standard).
- Creato gruppo associato agli utenti di **Utenti Generali** per policy comuni.

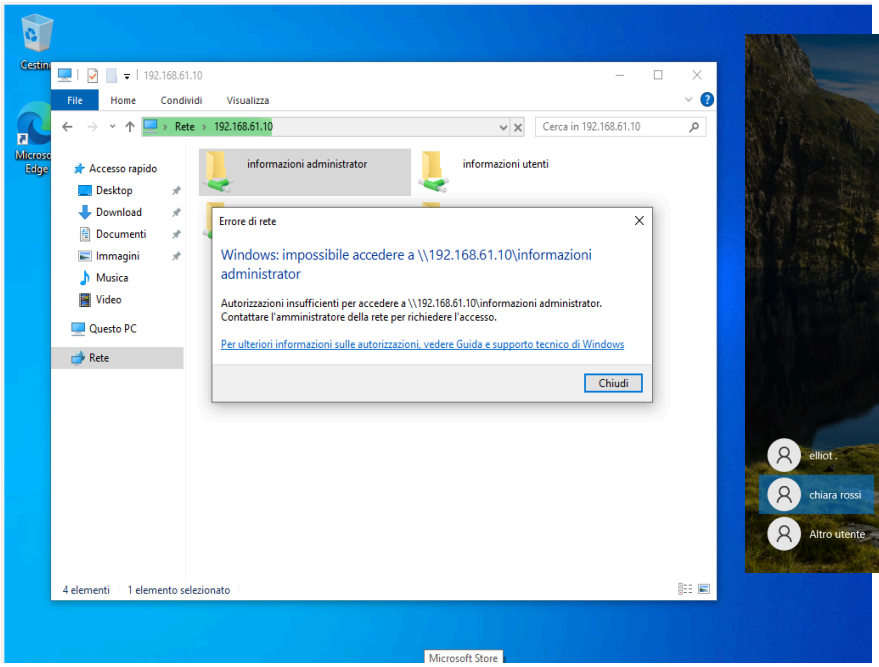
### 3.Assegnazione Permessi

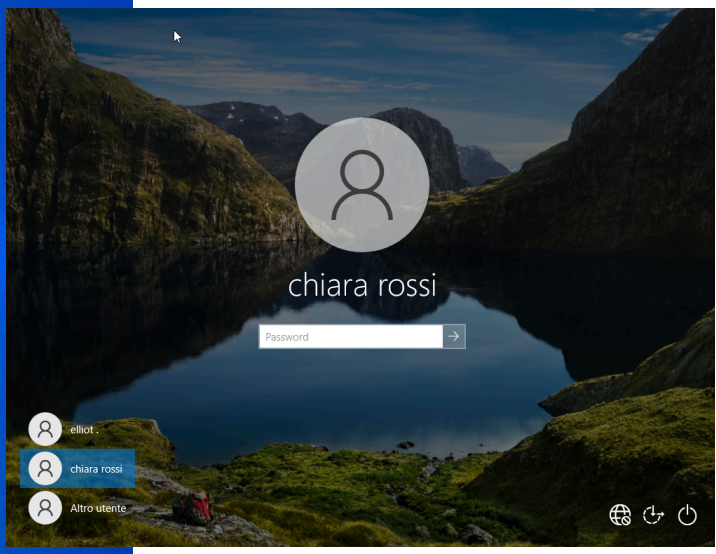
- Mario Rossi: accesso completo a tutte le cartelle e privilegi amministrativi.
- Utenti Generali: accesso solo alle proprie cartelle personali, blocco sulle cartelle non autorizzate.



## 4.Verifica

- Aggiunto client Windows 10 Pro al dominio (`192.168.61.20`, DNS 192.168.61.10).
- Accesso utente *Chiara Zucchetti*: cambio password obbligatorio al primo login, riuscito.
- Test di permessi: confermata impossibilità di accedere a cartelle non autorizzate, corretto funzionamento delle policy.




## 5.Errori riscontrati e risolti

- Problema DNS → client non comunicava col dominio; risolto correggendo rete e DNS.
- Errore database AD → mancava installazione corretta; completato con PowerShell.

## 2.pfSense

### Preparazione

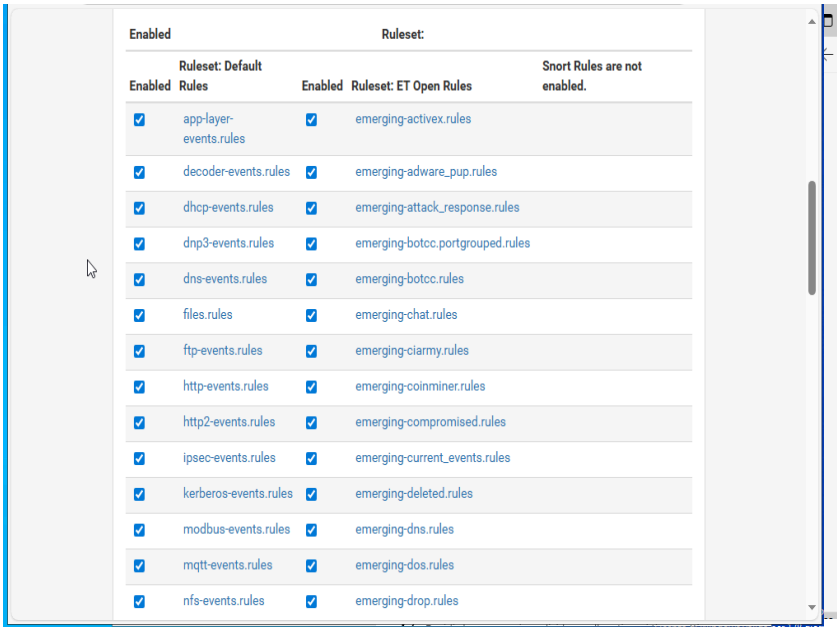
- **Installazione VM pfSense in VirtualBox.**
- **Configurate due interfacce:**
    - **WAN:** collegata a NAT per l'uscita verso Internet.
    - **LAN:** collegata a rete interna `192.168.61.1/24`.

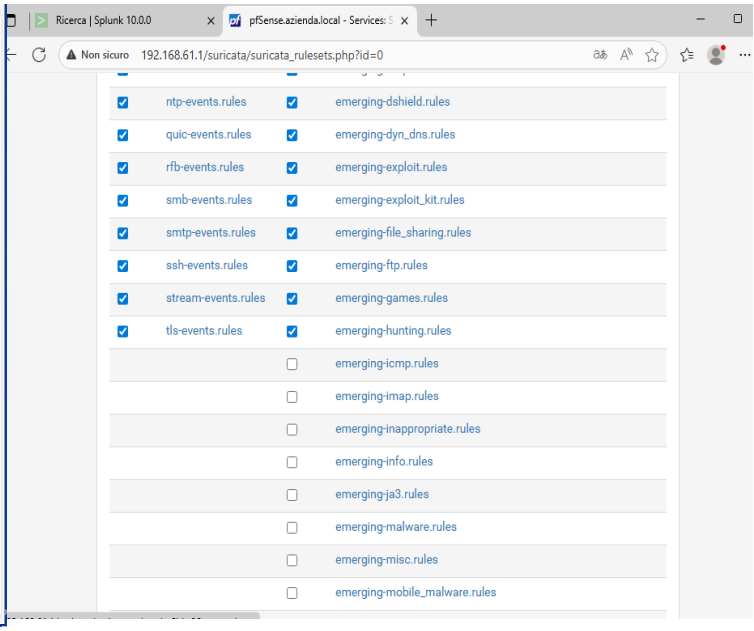
### Configurazione iniziale

- **Assegnato IP `192.168.61.1` all'interfaccia LAN.**
- **Host Windows Server configurato come client della LAN (`192.168.61.10`).**
- **DNS per i client puntati su win server(`192.168.61.10`).**

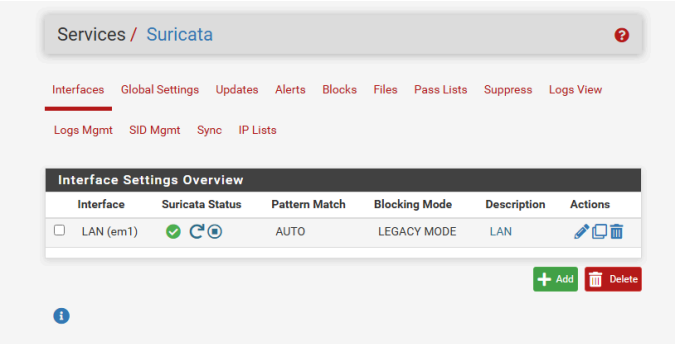
### IDS/IPS (Suricata)

- Installato pacchetto Suricata su pfSense,aggiornato dal bottone update e avviato le regole per pfsense scelgiedno il fomrato basiche

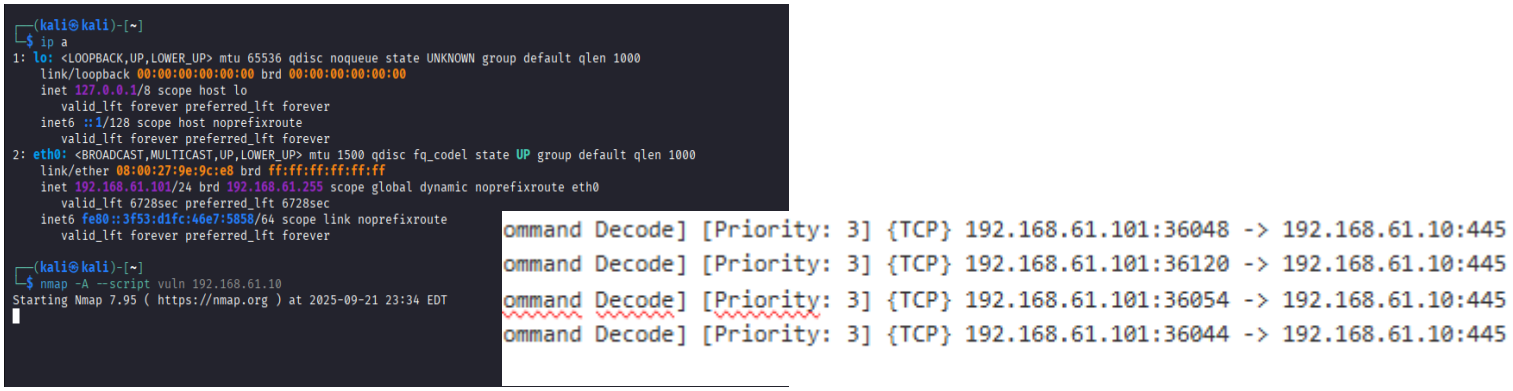


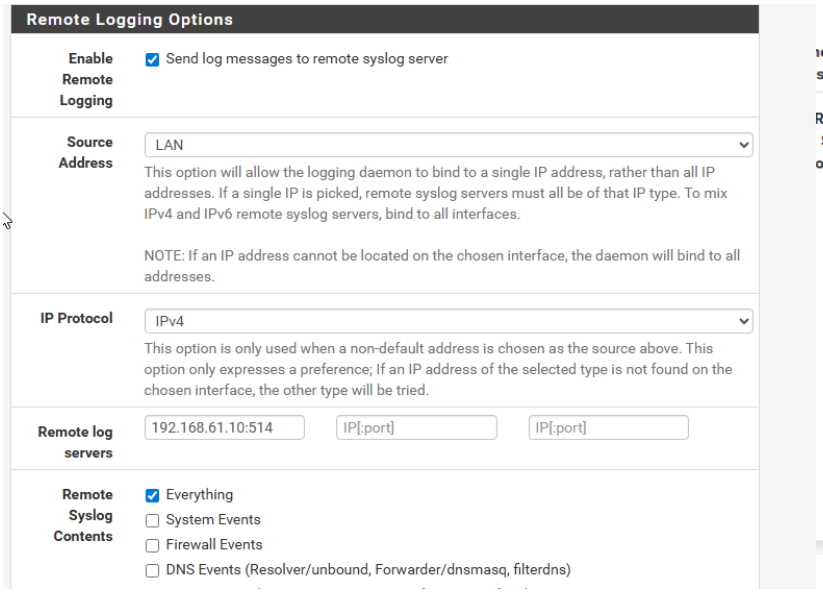
- **Configurata interfaccia LAN in modalità Legacy (IDS).**

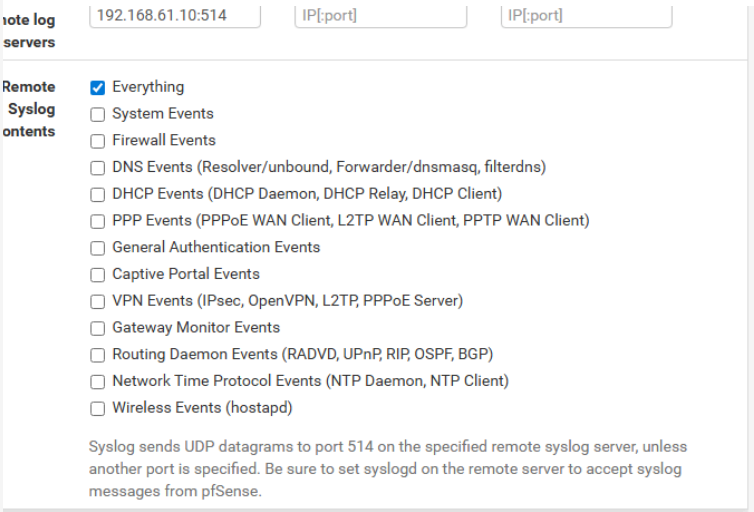



- **Test: lanciato scansione con nmap e nei log precedenti prima che configuravo di mandare tutto a splunk, i log di pfsense rivelano una prima ricerca delle vulnerabilità.**




- **dopo Attivati i log dettagliati per splunk tramite porta 514 UDP.**



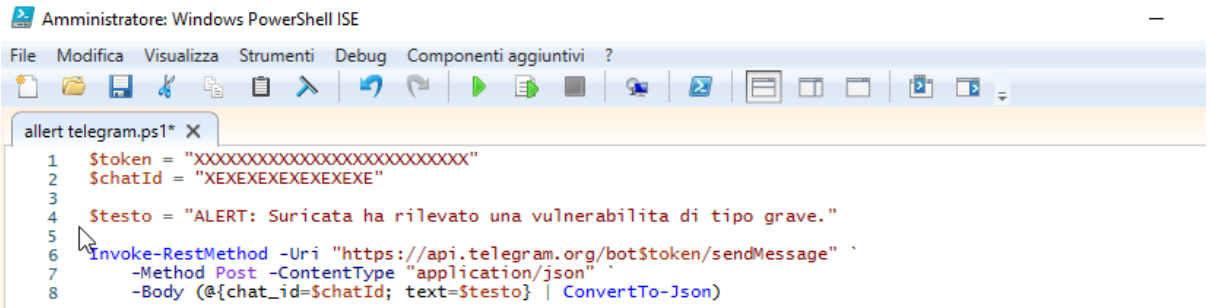
## 3..Bot Telegram

### Preparazione

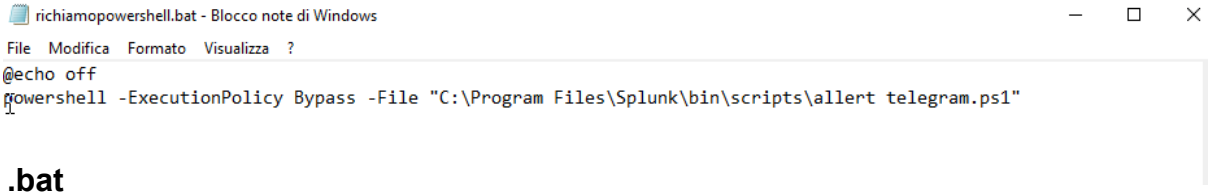
- **Creato bot dedicato tramite BotFather su Telegram.**
- **Ricevuto token di autenticazione per l'API.**
- **Recuperato chat_id tramite chiamata alle API (`getUpdates`).**

### Configurazione script

- **ho chiesto a chat gpt di farmi questi due script perchè non eccello in powershell e .bat(soprattutto), ho preferito richiede a un'intelligenza artificiale essendo che da un lato sono state create anche per semplificare quello non semplificabile all'epoca.**

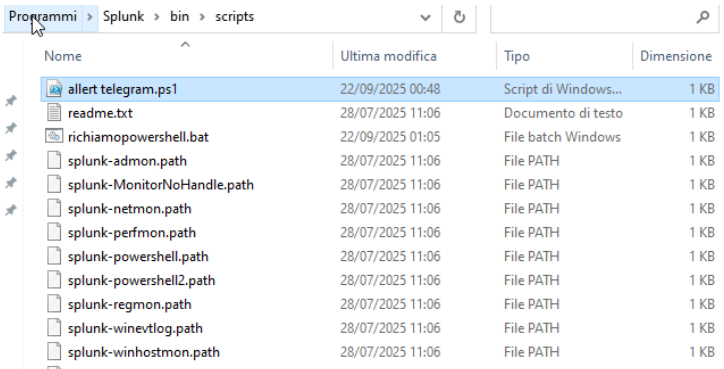



**power shell**




**.bat**

- **Salvati entrambi gli script nella cartella `C:\Program Files\Splunk\bin\scripts\`.**



# 4.Splunk

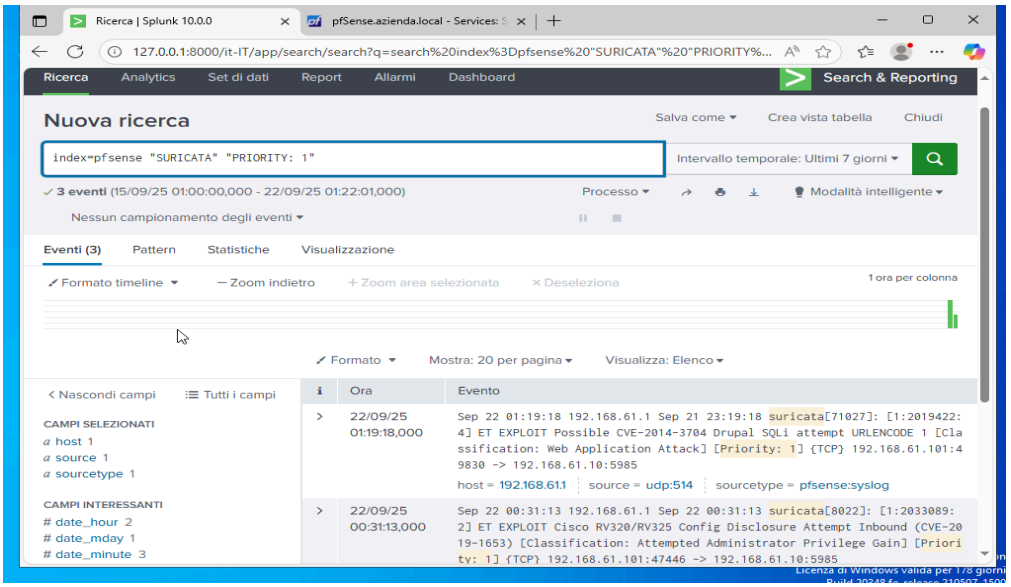
## Preparazione

- **Installato Splunk Enterprise su Windows Server 2022 (`192.168.61.10`).**

- **Definito nuovo Data Input: tipo *UDP*, porta 514, sourcetype `pfsense:syslog`, per far venire i log da pfsense a splunk tramite systemlog**

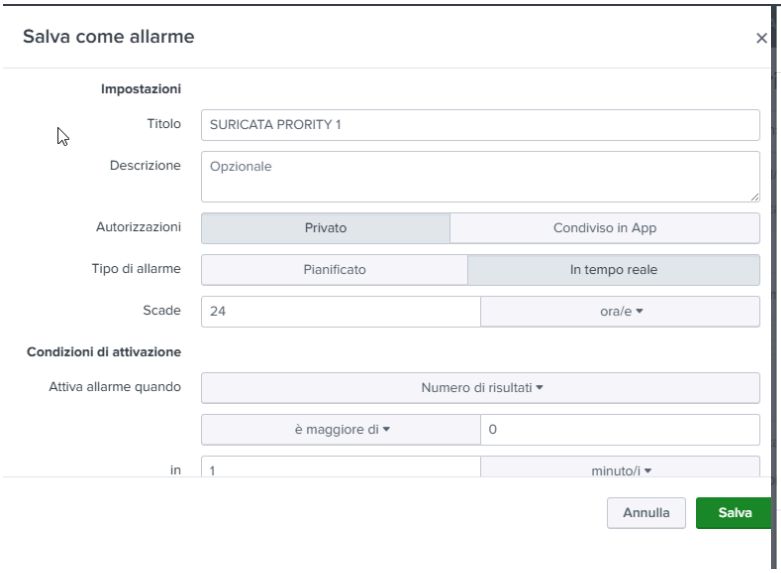

## Configurazione

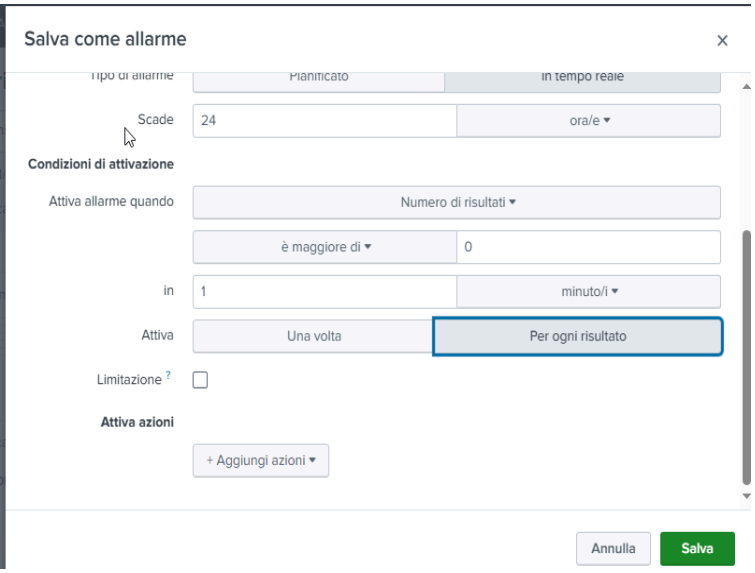
- **Creata query di ricerca per eventi critici:**




- **`cosi` rileva solo alert ad alta priorità(così da non avere tantissimi falsi positivi)**

- **Salvata la query come Allarme programmato ogni 1 minuto.**
- **Condizione → attiva se *Numero risultati > 0*.**

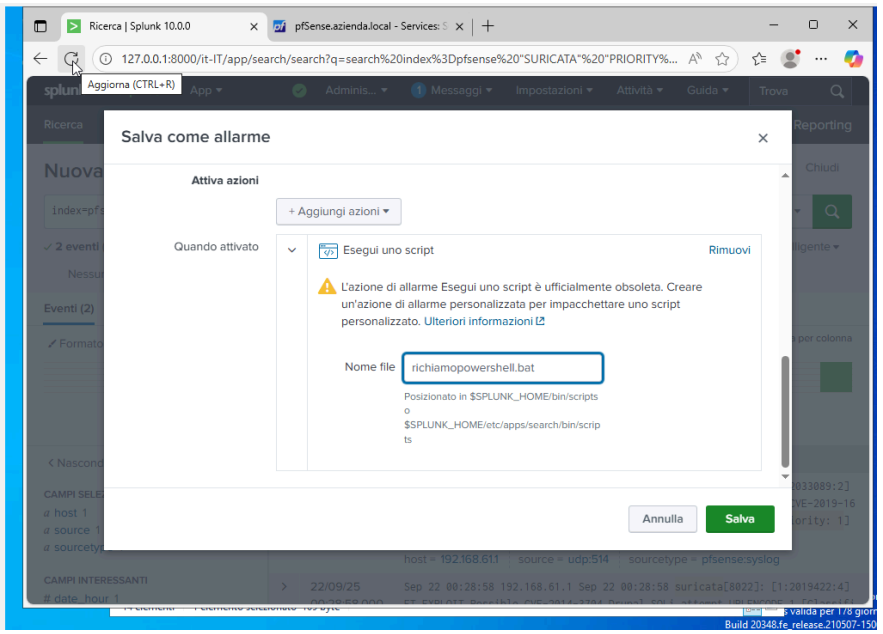


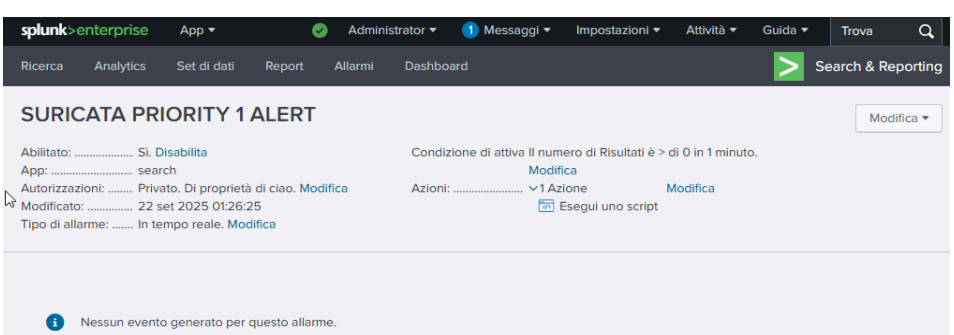




## Allarmi e Azioni

- **Azione associata,Esegui script esterno.**

- **Script collegato: file `.bat` che richiama PowerShell per invio notifiche su Telegram.**



## 5..Verifica pratica con scansione Nmap

**1. Generazione traffico sospetto**

- **Dal client interno è stata lanciata una scansione aggressiva con:**
- **Sono stati utilizzati anche script di vulnerabilità (`--script vuln`) per aumentare il numero di eventi rilevabili.**

**2. Rilevazione con Suricata**

- **Suricata, configurato in modalità IDS su pfSense, ha identificato le attività di scanning come potenziali minacce.**

- **Sono stati generati diversi alert con classificazione Priority 1 (eventi critici).**

**3. Invio log a Splunk**

- **pfSense ha inviato i log Suricata a Splunk tramite syslog UDP 514.**

- **La query `index=pfsense "Suricata" "Priority: 1"` ha mostrato in tempo reale gli eventi generati.**

**4. Attivazione allarme Splunk**

- **L'allarme pianificato in Splunk ha rilevato che il numero di risultati > 0.**

- **Condizione soddisfatta → esecuzione dello script batch configurato.**

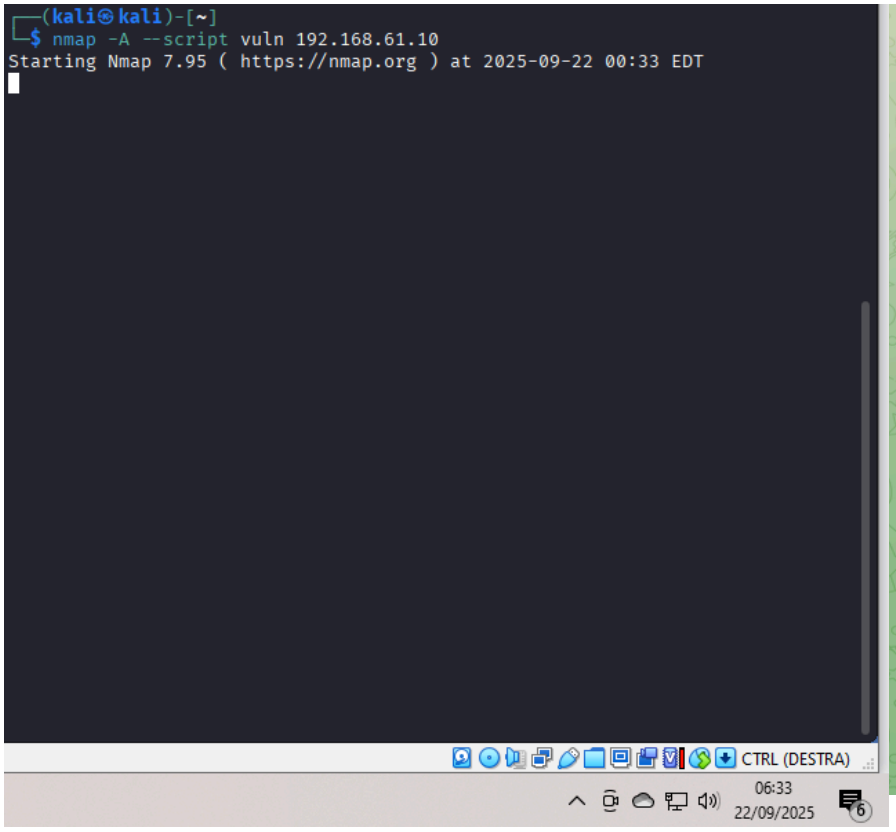
**5. Esecuzione script Telegram**

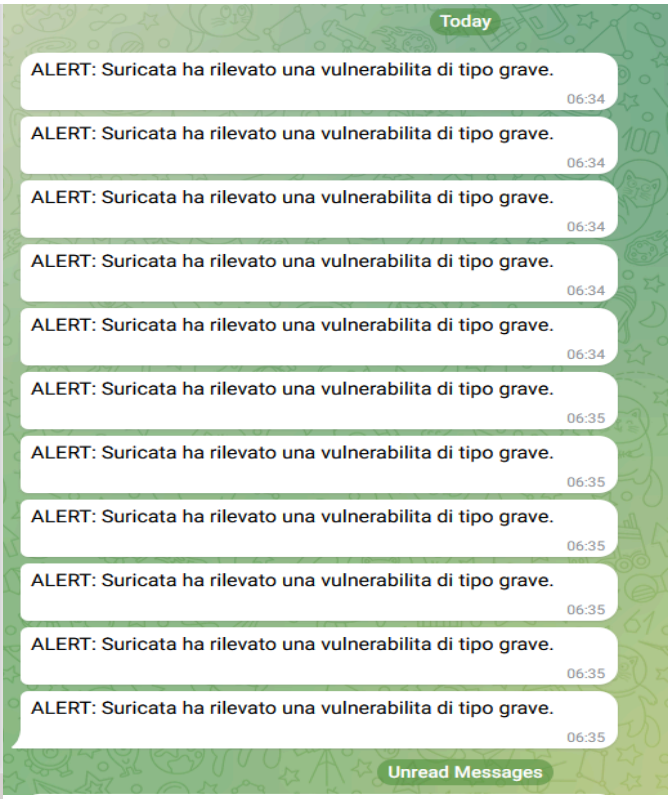
- **Lo script batch ha richiamato `alert_telegram.ps1`.**

- **Lo script PowerShell ha inviato una richiesta alle API Telegram con token e chat_id del bot.**

**6. Ricezione notifica**

- **Il bot Telegram ha inviato il messaggio al canale/chat configurata.**
- **Durante i test sono state ricevute x notifiche critiche, una per ciascun evento Priority 1 rilevato da Suricata**




**06:33 ho avviato la prima scansione e alle 06:34/35 mi aveva gia trovato vulnerabilita con priority uno considerate tipologie di vulnerabilità gravi.**

## 6.Miglioramenti e Ottimizzazioni

**purtroppo come anche detto sopra per mancanza di tempo potevo rendere molto più sicuro esempio con segmentation,chiusura di porte che non servono, regole firewall più restrittive, ip sin line con giuste regole, alerting avanzato con magari ip e cosa stava cercando di fare l'attaccante ecc**

## 7.CONCLUSIONI

**Il progetto ha dimostrato con successo la possibilità di integrare diversi strumenti in un'unica pipeline di sicurezza:**

- **Windows Server 2022 per la gestione dei gruppi e dei permessi.**
- **pfSense con Suricata per il monitoraggio IDS e la protezione perimetrale.**
- **Splunk come piattaforma di raccolta, analisi e correlazione dei log.**
- **Bot Telegram come sistema di alerting in tempo reale.**

**La simulazione di attacchi tramite scansioni nmap ha confermato il corretto funzionamento del flusso: gli eventi sono stati rilevati da Suricata, inviati a Splunk, trasformati in allarmi e notificati su Telegram.**

**Nonostante la natura dimostrativa, il lavoro rappresenta un proof of concept valido che unisce amministrazione di dominio e sicurezza di rete. Le migliorie proposte (segmentazione, regole firewall più restrittive, Suricata in modalità IPS, alerting avanzato) mostrano come questa architettura possa evolversi in una soluzione più solida e vicina agli standard di un SOC aziendale.**

**Il risultato finale è una base concreta su cui costruire un'infrastruttura capace non solo di rilevare, ma anche di prevenire e mitigare attacchi informatici, garantendo maggiore protezione e tempi di risposta più rapidi.**