Security Information and Event Management System

Proiect TPI

UNSTB

ETTI

2024– 2025

III RST

Andronie Vasile Laurentiu

Maravela Viorel

Anul

1.Introducere

Security Information and Event Management (prescurtat SIEM), reprezinta un sistem de aparare a companiilor impotriva atacurilor cibernetice.

Un astfel de manager combina functionalitatea de gestiune a informatiilor de securitate precum si a evenimentelor, oferind astfel capacitatea companii sa se apere atat in mod reactiv cat si proactiv.

Printre altele un SIEM ofera transparenta traficului prin retea, alerteaza asupra posibilelor amenintari, raspunde in mod activ la evenimente suspicioase sau malitioase.

2. Analiza Pietei

Cel mai cunoscut si utilizat SIEM este SPLUNK, care apartine de Cisco.

Printre altele Splunk ofera:

- Analiza si vizualizarea datelor de tip log
- Functionalitati avansate de cautare, alertate si creare de dashboarduri
- Monitorizare de performanta, securitate si conformitate

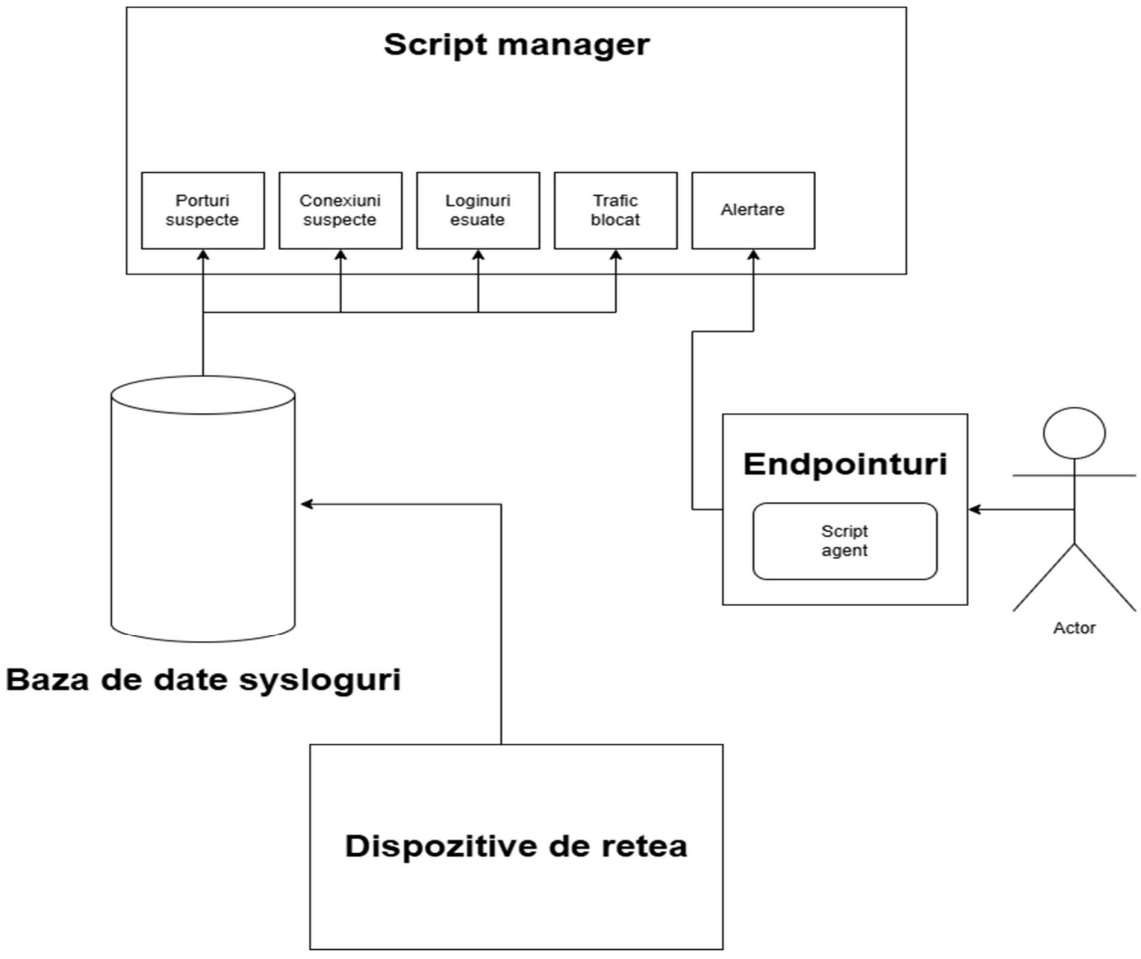
O alta unealta disponibila de piata este Wazuh, care este mai orientat pe dispositive endpoint.

Acesta ofera:

- Detectie de amenintari si monitorizare integritate fisiere
- Management al vulnerabilitatilor si raspuns la incidente
- Capacitate de functionare impreuna cu alte platforme

3.Proiectare

Schema bloc (prototip)



1. **Script manager**

Este un script care va rula pe un server. Va indeplini majoritatea functionalitatilor ale SIEM-ului de alertare si/sau prevenire a amenintarilor precum, inchidere de porturi, alertate de conexiuni suspecte, alertare si/sau blocare incercari de brute-force, alertare si/sau prevenire port scan.

2. **Script agent**

Este un script care va rula pe dispozitive de tip endpoint. Acesta va trimite alerte scriptului manager care va anunta utilizatorii retele de posibile amenintari. In general, dar nu limitat la, va indeplini 3 functii:

verificare loginuri esuate local, verificare loginuri remote, verificare posibila infectie malware. Acest script va avea si capacitatea proactiva de a termina (kill) procesele suspecte. Este compatibil cu Linux (Debian) si Windows.

3. **Baza de date sysloguri**

Dispozitivele de retea pot trimite evenimente ce se petrec la nivelul lor

prin protocolul syslog. Baza de date va pastra aceste jurnale de loguri astfel incat managerul sa le poata prelucra riguros, analizand si evenimente anterioare pentru corelatie intre aceste loguri si o mai buna luare de decizie.

4.Implementare

Pentru implementare am ales limbajul GO. Motivele pentru aceasta alegere au fost disponibilitatea librariilor de interes in dezvoltare unui SIEM si securitatea oferita de un limbaj compilat (fata de Python). Serverul de baza de date este MySQL community.

Proiectul contine doua foldere principale : cmd si pkg. Folderul cmd contine toate scripturile ce au o functie main, functie ce e apelata atunci cand rulezi un fisier .go . Folderul pkg contine diferite metode ce ajuta la implementare functionalitatilor principale ale SIEM-ului cat si doua structure de date: Alerta si MesajSyslog.

In general, scriptul manager.go reprezinta centrul sistemului si implementeaza urmatoarele functii: deschide un server HTTP pe care asculta POST-uri venite de la dispozitve endpoint. Acesta primeste alerte sub forma strcturii de date construite de noi si le afiseaza in consola; el de asemenea deschide o subrutina go numita RuleazaDetectii(). In aceasta subrutina se apeleaza periodic trei functii care folosesc informatii din baza de date pentru analiza si luare de decizii. Fiecare din cele 3 functii reactioneaza astfel in cazul unei amenintari: afiseaza in consola o alerta si adauga un IP in spatele caruia se afla atacatorul intr-un blacklist.

Pentru blacklist, exista scriptul blacklist_server.go care deschide un alt server HTTP unde dispozitivele de retea (routere si firewalluri) pot citi si bloca trafic catre respoectivele IP-uri.

Pe langa asta scriptul syslog_listener.go deschide pe masina care gazduieste managerul portul UDP 514, port dedicate comunicatiilor de tip syslog. Pentru fiecare syslog primit, acest script il parseaza si il transforma in urmatoare strctura: prival, timestamp, hostname, program, message.

Pentru simplitate si generealizare se presupune ca dispozitivele de retea au setat formatul pentru mesajele syslog de tipul RFC3164. Odata parsate

aceste mesaje sun salvate in baza de date care are aceleasi coloane precum campurile strcturii descries anterior.

Serverul HTTP din scriptul manager care primeste alerte de la endpointuri, va asculta pentru alerte trimise de scriptul agent.go. Acest script ruleaza pe dispozitive de tip endpoint care au sistemul de operare Linux sau Windows. Functiile principale ale agentului sunt de a verifica incercari de log-in in sistem local esuat, incercari de loguri remote (prin reverse shell de exemplu), sau modificari ale fisierelor de sistem care sunt cel mai probabil cauzate de malware *.

Daca una din conditii este indeplinita agentul trimite un obiect JSON catre manager, managerul il decodeaza sub forma structurii de alerta si afiseaza alerta. Metoda TrimiteAlerta() este implementata pentru asigurarea comunicarii intre agent si manager. Pentru implementare unei reactii proactive, agentul se ocupa de asemenea de terminarea proceselor in cazul unui login remote.

In final, pentru o analiza manuala (daca este necesara) scriptul baza_de_date.go poate fi apelat din consola cu un set de flag-uri pentru a afisa toate liniile in baza de date ce contin un anumit cuvant cheie, ales arbitrar.

*Aceste functionalitati functioneaza in realitate doar pe Linux cum trebuie deoarece pe windows nu pot citi sau scrie in anumite fisiere nici daca esti administratorul sistemului