PROJEKTBERICHT

M207: Management eines digitalen Infrastrukturprojekts

PORTGUARDIAN: USB-Bedrohungserkennungs- und -Abwehrsystem für Unternehmensumgebungen

Bildungseinrichtung: Spezialisiertes Institut für Angewandte Technologie Taza

Realisiert von: Walid Gourideche – Yassine Laalaoua **Supervisor/Trainer: Abdelmajid Lamkadam**

DANKE

Zuallererst möchte ich Herrn Abdelmajid Lamkadam, meinem Ausbilder und Betreuer für dieses Projekt, meinen tiefen Dank für seine kompetente Anleitung, seine wertvollen Ratschläge und seine ständige Verfügbarkeit während der acht Wochen aussprechen. Dank seiner Unterstützung konnte ich das Projekt professionell angehen und die auftretenden technischen Schwierigkeiten, insbesondere bei der Implementierung der WMI-Module und der SIEM-Integration, überwinden.

Ich möchte mich außerdem beim gesamten Lehrpersonal des Fachinstituts für Angewandte Technologie in Taza für die hohe Qualität der angebotenen Ausbildung bedanken. Die Kurse in Windows-Systemadministration, Infrastruktursicherheit und Netzwerküberwachung bildeten die unerlässliche technische Grundlage für den erfolgreichen Abschluss dieses Projekts.

Ein herzliches Dankeschön an meine Kommilitonen für ihre gegenseitige Unterstützung und den bereichernden fachlichen Austausch, der dazu beigetragen hat, meine Motivation während des gesamten Kurses aufrechtzuerhalten.

Abschließend möchte ich meinen Eltern meinen aufrichtigen Dank für ihre unerschütterliche Unterstützung, Geduld und Ermutigung während meiner gesamten Ausbildung aussprechen. Ihr Vertrauen war ein entscheidender Faktor für meinen akademischen und beruflichen Erfolg.

INHALTSVERZEICHNIS

EINFÜHRUNG

● Allgemeiner Kontext und Einleitung

In einer Zeit, in der die digitale Transformation Unternehmen einer zunehmenden Angriffsfläche aussetzt, stellen USB-Geräte einen der am meisten unterschätzten Einfallstore dar. Laut dem Verizon Data Breach Investigations Report 2023 sind 68 Prozent aller Datenpannen auf menschliches Versagen zurückzuführen, darunter die fahrlässige Verwendung externer Geräte.

USB-Angriffe treten in vielfältigen Formen auf: von klassischer Ransomware, die Autoausführung ausnutzt, über BadUSB-Angriffe, die auf die Firmware von Controllern abzielen, bis hin zum Datendiebstahl durch böswillige Insider. Der NotPetya-Vorfall von 2017, der sich teilweise über USB-Geräte in isolierten Industrieumgebungen verbreitete, verursachte weltweit Schäden in Höhe von über zehn Milliarden US-Dollar.

Für KMU stellt die Anschaffung kommerzieller EDR-Lösungen wie CrowdStrike oder Carbon Black eine prohibitive Investition dar, die sich auf vierzig bis sechzig Euro pro Endpunkt und Monat beläuft, ohne die Kosten für die notwendige Infrastruktur und das Fachwissen.

● Projektpräsentation

In einem Zeitraum von acht Wochen, was etwa einhundertsechzig Arbeitsstunden entspricht, habe ich PortGuardian Enterprise v2.1 entworfen und entwickelt, ein automatisiertes USB-Bedrohungserkennungs- und -abwehrsystem, das an die Bedürfnisse von KMU angepasst ist.

Das System basiert auf einer sechsstufigen, mehrschichtigen Erkennungsarchitektur, die Hash-Signaturen, Dateinamenheuristiken, Shannon-Entropieanalyse, Erkennung von Dateiendungsabweichungen, PE-Importanalyse und IOC-Extraktion kombiniert. Dieser mehrschichtige Verteidigungsansatz ermöglicht die Erkennung sowohl bekannter Malware als auch neuartiger, unsignierter Bedrohungen.

Hauptmerkmale:

- Echtzeiterkennung über Windows Management Instrumentation (WMI).
- 6-schichtige forensische Engine mit dynamischer Bedrohungsbewertung
- Hot-rechargeable SHA-256 Hash-Signaturdatenbank
- Automatischer Auswurf infizierter Geräte

- Automatisierte Netzwerkisolierung über Windows-Firewall und Adapterdeaktivierung
- Splunk Enterprise-Integration mit dem Syslog-Protokoll nur für WARN- und CRITICAL-Warnungen
- Sichere Wiederherstellung durch ein SOC-Administratorpasswort geschützt
- PyQt6-grafische Benutzeroberfläche mit Echtzeitkonsole

Eingesetzte technische Umgebung:

Das Projekt wurde in einer professionellen Umgebung entwickelt und getestet, die Windows 10 Pro-Clients und eine Windows Server 2019-Infrastruktur umfasste, auf der Splunk Enterprise zur Zentralisierung von Sicherheitsereignissen gehostet wurde.

● Problemstellung

Dieses Projekt befasst sich mit folgendem Problem:**Wie kann ein Netzwerkadministrator USB-Bedrohungen in Echtzeit mit einem mehrschichtigen Ansatz automatisch erkennen und eindämmen, gleichzeitig die Warnmeldungen in einem unternehmensweiten SIEM zentralisieren und dies mit begrenzten Ressourcen bewerkstelligen?**

Die Ziele sind dreifach:

Aus technischer Sicht Entwicklung eines autonomen Endpoint-Agenten, der in der Lage ist, USB-Ansteckvorgänge in Echtzeit zu überwachen und die Reaktion gemäß Sicherheitsrichtlinien zu automatisieren, mit einer Erkennung Verzögerung von weniger als einer Sekunde und einer Isolierung in weniger als fünf Sekunden.

Aus sicherheitstechnischer Sicht, implementieren Sie eine mehrschichtige Erkennungs-Engine, die mehrere sich ergänzende Techniken kombiniert, um die Erkennungsrate zu maximieren und gleichzeitig Fehlalarme zu minimieren, mit automatischem Auswurf des infizierten Geräts und Netzwerk Isolierung, um eine seitliche Ausbreitung zu verhindern.

Hinsichtlich der Integration Um die Interoperabilität mit Splunk Enterprise über das Standard-Syslog-Protokoll zu gewährleisten, werden Ereignisse intelligent gefiltert, sodass nur WARN- und CRITICAL-Warnungen ausgelöst werden, um eine Überlastung des SIEM-Systems und eine Übermüdung der SOC-Analysten zu vermeiden.

- **Planankündigung**

Dieser Bericht ist in drei Hauptteile gegliedert.

Teil Eins Der Text stellt den Kontext und das Umfeld des Projekts dar, indem er Bedrohungen durch USB-Geräte in Unternehmen, bestehende Lösungen und die mit Windows Server 2019 und Splunk Enterprise eingesetzte technische Infrastruktur analysiert.

Teil Zwei beschreibt detailliert die angewandte Agile-Methodik und die technische Architektur des Systems, mit einer ausführlichen Präsentation der sechs schichtigen Erkennungs-Engine und der Splunk-Integration.

Teil Drei Es präsentiert die konkreten Ergebnisse anhand von Screenshots, die Resultate der Validierungstests, die erworbenen Fähigkeiten und die während der Entwicklung überwundenen Schwierigkeiten.

I. PROJEKTUMFELD UND -KONTEXT

A. Der Sektor Unternehmens-Cybersicherheit

1. USB-Bedrohungen in einem professionellen Umfeld

USB-Geräte stellen einen bevorzugten Angriffsvektor dar, da sie weit verbreitet sind, eine hohe Speicherkapazität besitzen und durch direkten physischen Kontakt Netzwerkschutzmechanismen umgehen können.

USB-Ransomware stellt weiterhin eine Bedrohung dar. Der WannaCry-Angriff im Jahr 2017 verbreitete sich hauptsächlich über USB-Sticks in Umgebungen ohne Internetverbindung. Schließt ein Mitarbeiter einen infizierten Stick an seinen Arbeitsplatzrechner an, kann er innerhalb weniger Minuten das gesamte Unternehmensnetzwerk gefährden. Laut dem Ponemon Institute (2023) belaufen sich die durchschnittlichen Kosten eines Ransomware-Angriffs auf 4,5 Millionen US-Dollar, einschließlich Lösegeld, Ausfallzeiten und Datenverlust.

BadUSB-Angriffe nutzen die Firmware von USB-Controllern aus. Durch Umprogrammierung des Mikrocontrollers verwandelt ein Angreifer ein USB-Laufwerk in ein Eingabegerät (Human Interface Device, HID), das eine Tastatur emuliert. Nach dem Anschließen führt das Laufwerk programmierte Tastenanschlagsequenzen aus, um Schadsoftware herunterzuladen und auszuführen. Herkömmliche Antivirenprogramme

können die Controller-Firmware nicht scannen, wodurch dieser Angriff mit herkömmlichen Methoden praktisch unentdeckbar bleibt.

Datenexfiltration stellt eine erhebliche Bedrohung durch Insider dar. Moderne USB-Sticks mit bis zu zwei Terabyte Speicherkapazität ermöglichen die Exfiltration großer Mengen sensibler Daten. Laut der Studie „Cybersecurity Insiders 2023“ wurden 56 Prozent der Insider-Sicherheitsvorfälle über Wechseldatenträger verübt. Ein kompromittierter oder böswilliger Mitarbeiter kann innerhalb weniger Minuten eine komplette Kundendatenbank kopieren.

Beim sogenannten „USB Drop“ (Social Engineering) werden menschliche kognitive Verzerrungen ausgenutzt. Eine Studie der University of Illinois aus dem Jahr 2016 zeigte, dass 48 Prozent der gefundenen USB-Sticks aus Neugier angeschlossen wurden und 45 Prozent der Nutzer mindestens eine Datei öffneten. Angreifer platzieren infizierte Sticks gezielt auf Firmenparkplätzen oder in Raucherbereichen und setzen dabei auf die natürliche Neugier der Mitarbeiter.

Der NotPetya-Angriff verdeutlicht auf dramatische Weise die Gefahr physischer Angriffe. Diese zerstörerische Schadsoftware, getarnt als Ransomware, verbreitete sich über verschiedene Wege, darunter USB-Geräte, insbesondere in abgeschotteten Industrieumgebungen. Der finanzielle Schaden belief sich auf über zehn Milliarden US-Dollar, und Unternehmen wie Maersk und Saint-Gobain waren wochenlang lahmgelegt.

2. Vergleichende Analyse bestehender Lösungen

Der Markt bietet drei Hauptkategorien von Lösungen mit jeweils eigenen Vor- und Nachteilen.

Kommerzielle EDR-Lösungen (CrowdStrike, Carbon Black, Microsoft Defender for Endpoint):

Diese Lösungen bieten fortschrittliche Erkennung durch maschinelles Lernen und Echtzeit-Verhaltensanalyse mit umfassenden forensischen Funktionen, Cloud-nativer Integration und professionellem Support rund um die Uhr. Die Kosten liegen jedoch zwischen 45 und 70 Euro pro Endpunkt und Monat bzw. zwischen 27.000 und 42.000 Euro jährlich für 50 Arbeitsstationen. Zudem benötigen sie eine permanente Cloud-Infrastruktur, was Fragen der Datensouveränität und der Abhängigkeit von der Internetverbindung aufwirft, sowie spezialisiertes Fachwissen für die Administration und Optimierung der Erkennungsregeln.

Geräteverwaltungslösungen (DeviceLock, Endpoint Protector, Active Directory GPO):

Diese Lösungen ermöglichen die detaillierte Steuerung von USB-Ports durch Whitelisting anhand der Seriennummer oder VID/PID, selektives Blockieren nach Gerätetyp und zentrale Ereignisprotokollierung. Die Kosten sind moderat und liegen zwischen 20 und 35 Euro pro Endpunkt und Monat. Allerdings weisen sie erhebliche Einschränkungen auf: keine Malware-Erkennung (komplettes Blockieren von Binärdateien), ein präventiver Ansatz, der die Produktivität legitimer Nutzer beeinträchtigt, und die Möglichkeit der Umgehung durch VID/PID-Verschleierungstechniken.

Open-Source-Lösungen (OSSEC, Wazuh, USBGuard):

Diese kostenlosen Lösungen mit nachvollziehbarem Quellcode bieten umfassende Anpassungsmöglichkeiten und eliminieren Lizenzkosten. Allerdings fehlen OSSEC und Wazuh spezialisierte USB-Module, die eine vollständige Eigenentwicklung erfordern. USBGuard ist nur mit Linux kompatibel und bietet keine Windows-Unterstützung. Zudem ist die Einarbeitung bei allen Lösungen aufgrund der fragmentierten Dokumentation und des Mangels an professionellem Support sehr aufwendig.
Zusammenfassende Vergleichstabelle:

Critère	EDR Commercial	Gestion Périph.	Open-Source	PortGuardian
Coût annuel (50 endpoints)	27k-42k€	12k-21k€	0€	**0€**
Détection malware	✓✓✓ ML avancé	✗ Aucune	✗ Basique	**✓✓ Multi-couches**
Spécialisation USB	✓ Module dédié	✓✓✓ Core	✗ Limité	**✓✓✓ Spécialisé**
Intégration SIEM	✓✓ Propriétaire	✓ Syslog	✓ Syslog	**✓✓ Splunk natif**
Isolation réseau	✓✓✓ Automatique	✗ Manuelle	✗ Manuelle	**✓✓✓ Auto + Éjection**
Complexité admin	Élevée	Moyenne	Très élevée	**Faible**
Support Windows	✓✓✓ Natif	✓✓✓ Natif	✗ Limité	**✓✓✓ Natif WMI**

3. Positionierung von PortGuardian Enterprise

PortGuardian Enterprise positioniert sich als optimale Zwischenlösung für KMU, die technische Raffinesse und wirtschaftliche Zugänglichkeit vereint.

Technische Differenzierung:

Die sechsstufige, mehrschichtige Erkennungs-Engine Die wichtigste Innovation besteht darin, dass PortGuardian im Gegensatz zu maschinellem Lernen-basierten EDR-Systemen, die massive Datensätze und Cloud-Rechenleistung benötigen, sechs sich ergänzende Techniken intelligent kombiniert: SHA-256-Hash-Signaturen zur Erkennung bekannter Malware, Dateinamenheuristiken zur Identifizierung verdächtiger Muster (Trojaner, Ransomware, Backdoors), Shannon-Entropieanalyse zur Erkennung der Pack- und Verschlüsselungseigenschaften von Malware, Erkennung von Dateierweiterungsfehlern zur Aufdeckung getarnter Dateien, PE-Importanalyse zur Identifizierung gefährlicher Windows-APIs (VirtualAlloc, WriteProcessMemory, CreateRemoteThread) und die Extraktion von Indikatoren für eine Kompromittierung (IOC) in Form von IP-Adressen zur Anreicherung von Firewall-Regeln.

Dieser mehrschichtige Verteidigungsansatz gewährleistet, dass ausgeklügelte Schadsoftware, die eine Ebene umgeht, von den anderen erkannt wird.

Automatischer Auswurf des infizierten Geräts Dies ist selbst bei High-End-EDR-Systemen eine seltene Funktion. Sobald eine kritische Bedrohung erkannt wird (Score ≥ 85), trennt das System das Gerät physisch per mountvol und diskpart, noch bevor der Benutzer reagieren kann. Diese blitzschnelle Reaktion verhindert die manuelle Weiterverbreitung durch Dateikopieren.

Vollständige Netzwerkisolierung Es kombiniert drei Mechanismen: das Erstellen von Windows-Firewallregeln, die den gesamten ein- und ausgehenden Datenverkehr blockieren, das Ändern der globalen Firewallrichtlinie auf eine vollständige Blockierung und das physische Deaktivieren von Netzwerkadaptern über die netsh-Schnittstelle. Diese dreifache Isolation gewährleistet, dass keine Netzwerkkommunikation möglich ist, selbst wenn der Benutzer versucht, die Firewall zu deaktivieren.

Optimierte Splunk Enterprise-Integration Es filtert Ereignisse intelligent und meldet nur Warnmeldungen (unautorisiertes Gerät, verdächtige Datei) und kritische Warnmeldungen (Malware erkannt, Isolation aktiviert). Dadurch wird eine Überlastung des SIEM-Systems durch routinemäßige INFO-Ereignisse vermieden und die Belastung der SOC-Analysten reduziert, sodass diese sich auf echte Bedrohungen konzentrieren können. Das strukturierte JSON-Format erleichtert das Parsen und die Erstellung benutzerdefinierter Splunk-Dashboards.

Wettbewerbsvorteile für KMU:

Die Gesamtbetriebskosten betragen null. Dadurch wird die finanzielle Hürde beseitigt. Ein KMU mit fünfzig Mitarbeitern spart im Vergleich zu einem kommerziellen EDR-System jährlich zwischen 27.000 und 42.000 Euro – ein Budget, das für andere wichtige Sicherheitsinvestitionen wie Mitarbeiterschulungen zu Best Practices, externe Sicherheitsaudits oder die Implementierung von Multi-Faktor-Authentifizierung verwendet werden kann.

Das Fehlen einer Cloud-Abhängigkeit Es gewährleistet Datensouveränität und beseitigt Verbindungsprobleme. Der Agent arbeitet im Standalone-Modus ohne Internetverbindung und ist daher ideal für industrielle Umgebungen, Forschungslabore oder kritische Infrastrukturen, die eine Trennung vom Internet erfordern.

BedienungseinfachheitReduziert versteckte Kosten. Es ist keine dedizierte Serverinfrastruktur erforderlich, die Konfiguration beschränkt sich auf die Bearbeitung einer Hash-Textdatei und eines Splunk-IP-Parameters, und der tägliche Verwaltungsaufwand ist minimal, da die Signaturdatenbank ohne Neustart neu geladen werden kann.

● B. Technisches Umfeld und Infrastruktur

1. Eingesetzte Netzwerkarchitektur

Das Projekt wurde in einem Umfeld entwickelt und validiert, das repräsentativ für die Infrastruktur eines professionellen KMU ist.

Physische Architektur:

Die Infrastruktur umfasst einen Windows Server 2019, auf dem Splunk Enterprise 9.x als zentraler Sicherheitsereignis-Collector läuft. Der Server ist mit der statischen IP-Adresse 192.168.1.50 und dem UDP-Port 514 für den Empfang von Syslog-Meldungen konfiguriert. Er verfügt über 8 GB RAM und 100 GB dedizierten Protokollspeicher mit automatischer Rotation nach 30 Tagen.

Zwei Windows 10 Pro 64-Bit-Client-Workstations dienen als Testendpunkte, die repräsentativ für Benutzer-Workstations in Unternehmen sind, Mitglieder der Active Directory-Domäne mit Standardgruppenrichtlinien sind und PortGuardian Enterprise im Dienstmodus mit lokalen Administratorrechten ausführen.

Das lokale Netzwerk ist zur Vereinfachung der Tests als ein einziges VLAN 192.168.1.0/24 strukturiert, wobei ein Router/eine Firewall als Internet-Gateway und ein Gigabit-Ethernet-Managed-Switch für die Verbindung dienen.

Kommunikationsfluss:

PortGuardian-Agenten auf Clients erkennen USB-Ein- und -Ausgänge über lokales WMI ohne Netzwerkkommunikation. Bei einer Warnung oder einer kritischen Bedrohung senden die Agenten ein UDP-Syslog-Datagramm an 192.168.1.50:514. Der Splunk-Server empfängt die Meldungen, analysiert das JSON und indiziert die Ereignisse. SOC-Analysten nutzen das Splunk-Dashboard für Echtzeitüberwachung und Vorfallsmeldungen.

Diese unidirektionale Architektur**(Clients → Server)**minimiert die Angriffsfläche und stellt sicher, dass eine Kompromittierung des Clients keine Auswirkungen auf das SIEM hat.

2. Technologische Entscheidungen und Begründungen

Python 3.9 als primäre Sprache:

Python wurde aufgrund seiner klaren und ausdrucksstarken Syntax, die Fehler reduziert und die Entwicklung beschleunigt, seines umfangreichen Ökosystems mit ausgereiften Bibliotheken (wmi, PyQt6, requests), seiner plattformübergreifenden Kompatibilität, die eine zukünftige Portierung auf Linux/macOS ermöglicht, und seines automatischen Speichermanagements, das die in C/C++ häufig auftretenden Pufferüberlauf-Schwachstellen vermeidet, ausgewählt.

WMI (Windows Management Instrumentation):

WMI ist die native Microsoft-API für Systemabfragen. Die Klasse `Win32_VolumeChangeEvent` mit `EventType=2` erkennt das Einfügen von Datenträgern in Echtzeit mit einer Latenz von unter 500 ms. `Win32_LogicalDisk` liefert detaillierte Metadaten (Seriennummer, Bezeichnung, Dateisystem, Kapazität). Dieser native Ansatz gewährleistet Kompatibilität mit allen Windows-Versionen seit XP und vermeidet externe Abhängigkeiten.

PyQt6 für die grafische Benutzeroberfläche:

PyQt6 bietet umfangreiche und professionelle Widgets mit nativem Windows-Erscheinungsbild, ein elegantes Signal-/Slot-System für threadsichere Kommunikation zwischen WMI- und GUI-Threads, native Threading-Unterstützung über QThread zur Vermeidung von Deadlocks in der Schnittstelle sowie eine umfassende Dokumentation mit einer großen Community. Die Alternative Tkinter wäre zu rudimentär gewesen, während WPF/C# das Erlernen einer neuen Sprache erfordert hätte.

SHA-256-Hashing als primäre Signaturmethode:

Der SHA-256-Hash erzeugt einen 256-Bit-Fingerabdruck, der Eindeutigkeit garantiert (vernachlässigbare Kollisionswahrscheinlichkeit), resistent gegen Preimage- und Kollisionsangriffe ist und als Industriestandard von VirusTotal, MISP und allen TI-Feeds verwendet wird. Die einfache textbasierte Datenbank ermöglicht das manuelle Hinzufügen von Hashes oder den Massenimport aus OSINT-Feeds. Hot Reloading über eine GUI-Schaltfläche verhindert unterbrechungsbedingte Neustarts.

Splunk Enterprise als SIEM:

Splunk Enterprise wurde aufgrund seiner leistungsstarken Indexierungs- und Suchfunktionen mit SPL (Search Processing Language) ausgewählt, die komplexe Abfragen auf Terabytes von Daten ermöglichen, seiner anpassbaren Echtzeit-Dashboards mit interaktivem Drilldown, seiner fortschrittlichen Alarmierungsfunktionen mit E-Mail-/SMS-/Slack-Benachrichtigungen und seiner marktführenden Position, die die Kompatibilität mit Unternehmenssicherheitsökosystemen gewährleistet.

Obwohl Splunk ein kostenpflichtiges Produkt ist (ca. 2000 €/GB/Tag indexiert), ist eine kostenlose Lizenz mit 500 MB/Tag für kritische USB-Warnmeldungen in einem KMU mit 50-100 Mitarbeitern mehr als ausreichend, wodurch die Lösung wirtschaftlich rentabel ist.

Protokoll Syslog RFC 5424:

Syslog ist das universelle Netzwerkprotokoll. RFC 5424 definiert ein strukturiertes Format mit berechneter Priorität (Facility * 8 + Schweregrad), einem ISO-8601-UTC-Zeitstempel für internationale Rückverfolgbarkeit, einem Hostnamen zur Identifizierung der Quelle und einer JSON-Nachricht zur automatischen Auswertung. UDP-Port 514 garantiert minimale Latenz (< 10 ms im lokalen Netzwerk) ohne TCP-Verbindungs-Overhead. Dies ist akzeptabel, da der gelegentliche Verlust eines Protokollpakets für Warnmeldungen nicht kritisch ist (Redundanz durch lokale Protokolle).

Netzwerkisolation mehrerer Mechanismen:

Drei redundante Mechanismen gewährleisten die Isolierung:

1. Windows-Firewall-Regeln über netsh advfirewall erstellen Regeln zum Blockieren des ein- und ausgehenden Datenverkehrs von allen Profilen (Domäne, privat, öffentlich).
2. Die globale Firewall-Richtlinie wurde so geändert, dass standardmäßig alle eingehenden und ausgehenden Datenverkehr blockiert werden (deny all).
3. Deaktivieren Sie die Adapter physisch über den Befehl `netsh interface set interface admin=disable` für jede erkannte Netzwerkkarte.

Dieser mehrschichtige Sicherheitsansatz gewährleistet die Isolation selbst dann, wenn der Benutzer über Berechtigungen zur Firewall-Änderung verfügt. Die Wiederherstellung ist nur mit dem SOC-Administratorpasswort möglich.

Automatische USB-Auswurffunktion:

Der Auswurfvorgang kombiniert mountvol (Aushängen des Volumes) und diskpart (Entfernen des logischen Datenträgers). Dieser zweigleisige Ansatz maximiert die Kompatibilität unter verschiedenen Windows-Versionen (7/8/10/11) und gewährleistet den Auswurf auch bei geöffneten Dateien (erzwungenes Aushängen). Der Auswurf innerhalb von 5 Sekunden nach der Erkennung verhindert Benutzereingriffe und manuelle Weiterleitung.

II. Technischer und methodischer Rahmen

● A. Agile Projektmethodik

1. Organisation in Sprints und Ergebnisse

Für dieses achtwöchige Projekt wählte ich einen agilen Scrum-Ansatz, der an den individuellen Kontext angepasst war und wöchentlich von einem Trainer betreut wurde.

Anpassung für individuelle Projekte:

Die täglichen Stand-up-Meetings wurden durch ein tägliches Protokoll in Markdown ersetzt, in dem erledigte Aufgaben, aufgetretene Probleme und Architekturentscheidungen systematisch dokumentiert wurden. Sprint-Reviews fanden in Form wöchentlicher Demonstrationen für den Trainer statt, in denen implementierte Funktionen vorgestellt und direktes Feedback gegeben wurden. Das Backlog wurde mit Trello und vier einfachen Spalten verwaltet: Zu erledigen, In Bearbeitung, Testphase und Erledigt.

Definition von „Fertig“:

Ein Feature gilt als fertiggestellt, wenn: der Code geschrieben und funktional getestet ist, die technische Dokumentation (Python-Docstrings) vollständig ist, die Demonstration für den Trainer vorbehaltlos validiert wurde, die Integration mit bestehenden Modulen ohne Regression verifiziert wurde und der Git-Commit eine beschreibende Nachricht gemäß den Konventionellen Commits enthält.

Aufteilung in vier zweiwöchige Sprints:

Sprint 0 (Woche 0) – Recherche und Entwicklung:

- Analyse von USB-Bedrohungen und Literaturrecherche (ANSSI, Verizon DBIR)
- Vergleichende Studie bestehender Lösungen (CrowdStrike, DeviceLock, USBGuard)
- 6-stufige, mehrschichtige Architektur
- Einrichten einer Entwicklungsumgebung (Python 3.9, PyQt6, VirtualBox)
- Installation von Windows Server 2019 und Splunk Enterprise
- Liefergegenstand: Validierte funktionale Spezifikationen und technische Architektur

Sprint 1 (Wochen 1-2) - USB-Erkennungs- und Signaturdatenbank:

- Implementierung der WMI-Überwachung mit Win32_VolumeChangeEvent (6 Std.)

- Extraktion von Metadaten (Seriennummer, Bezeichnung, Dateisystem) über Win32_LogicalDisk (4h)
- Erstellung einer Blacklist-Hash-Datenbank mit Hot Reloading (5 Stunden)
- Einfache PyQt6-Konsolenschnittstelle mit Echtzeitprotokollierung (5 Stunden)
- Tests mit 10 verschiedenen USB-Laufwerken (3 Stunden)
- Liefergegenstand: Funktionale USB-Erkennung < 1 Sekunde mit serieller Verifizierung

Sprint 2 (Wochen 3-4) - Mehrschichtige forensische Engine:

- Schicht 1: SHA-256-Hashberechnung in Streaming-Chunks (4 Stunden)
- Ebene 2: Heuristische Erkennung verdächtiger Dateinamen (3 Std.)
- Schicht 3: Berechnung der Shannon-Entropie zur Erkennung der Packungsdichte (6h)
- Layer 4: Erkennung von Dateierweiterungsdiskordanzen (MZ-Header vs. .txt) (4h)
- Ebene 5: Analyse von PE-Importen auf der Suche nach gefährlichen Wirkstoffen (8 Std.)
- Schicht 6: Extraktion von IOCs (IP-Adressen) mittels regulärer Ausdrücke (4 Std.)
- Additives Bewertungssystem mit schwellenwertbasierter Klassifizierung (5 Stunden)
- Testen mit EICAR und Test-Malware (3 Stunden)
- Liefergegenstand: 6-stufige Erkennungs-Engine mit funktionaler Bewertung

Sprint 3 (Wochen 5-6) - Automatisierte Antwort und Splunk-Integration:

- Automatisches Auswerfen von USB-Geräten über mountvol + diskpart (6 Std.)
- Dreischichtige Netzwerkisolation (Firewall + Richtlinie + Adapter deaktivieren) (8 Std.)
- Client Syslog RFC 5424 mit strukturiertem JSON-Format (5h)
- Splunk Enterprise-Konfiguration mit JSON-Parsing (4 Std.)
- Intelligente Filterung (nur WARN und CRITICAL an Splunk) (3 Std.)
- Sichere Wiederherstellung mit Passwortauthentifizierung (4 Stunden)
- End-to-End-Tests mit Validierung im Splunk-Dashboard (5 Std.)
- Liefergegenstand: Komplettes System mit automatischer Isolation und Splunk-Warnfunktion

Sprint 4 (Wochen 7-8) - Abschluss und Dokumentation:

- Code-Refactoring und -Optimierung (8 Stunden)
- Finale PyQt6-Oberfläche mit professionellem Dunkelmodus (6 Stunden)
- Fehlerbehandlung und Sonderfälle (Berechtigungen, beschädigte Datenträger) (5 Stunden)
- Screenshots und Demovideos (4 Stunden)
- Erstellung eines Administratorhandbuchs (6 Stunden)
- Verfassen eines Projektberichts (18 Stunden)
- Benutzertest mit 3 Administratoren (3 Stunden)

- Vorbereitung auf Präsentation und Verteidigung (5 Stunden)
- **Lieferbar:** Produktionsreifes System mit vollständiger Dokumentation

2. Technisches Risikomanagement

Ein proaktives Risikomanagement wurde bereits in der Entwurfsphase integriert.

Risiko 1: WMI-Instabilität unter Last

- Wahrscheinlichkeit: Durchschnitt
- Auswirkungen: Kritisch (beeinträchtigt die grundlegende Erkennung)
- Abhilfemaßnahmen: WMI-Timeout von 1000 ms, um eine endlose Blockierung zu verhindern, wmi.x_wmi_timed_out-Ausnahmebehandlung, um die Überwachung fortzusetzen, und Stresstests mit wiederholtem, schnellem Einstecken von USB-Kabeln.
- Ergebnis: Nach über 100 schnellen Einführ-/Entfernungsvorgängen wurde keine Instabilität festgestellt.

Risiko 2: Massive falsch positive Ergebnisse

- Wahrscheinlichkeit: Hoch
- Auswirkungen: Hoch (Analystenermüdung, Systemausfallzeiten)
- Abhilfemaßnahmen: Empirische Kalibrierung der Bewertungsschwellenwerte (Entropie > 7,2 optimiert nach Tests), Splunk-Filterung (nur WARN/CRITICAL) und Tests mit legitimen Dateien (ZIP, Installationsdateien)
- Ergebnis: Falsch-Positiv-Rate < 5 % nach der Bereinigung

Risiko 3: Ausfall der Netzwerkisolierung

- Wahrscheinlichkeit: Gering
- Auswirkungen: Kritisch (seitliche Ausbreitung möglich)
- Abhilfemaßnahmen: Dreifach redundanter Mechanismus (Firewall + Richtlinie + Deaktivierung der Netzwerkkarten), Überprüfung nach der Isolierung mittels Ping-Test und Tests unter Windows 10 Build 1909/21H2 und Windows 11
- Ergebnis: 100% Isolationserfolgsrate bei 30 Tests

Risiko 4: Verlust von Splunk-Nachrichten

- Wahrscheinlichkeit: Gering
- Auswirkungen: Mittel (eingeschränkte Sicht)
- Abhilfemaßnahmen: 2-Sekunden-UDP-Socket-Timeout mit Wiederholungsversuchen, redundante lokale Protokolle in incidents.log und Splunk-Überwachung mit Warnmeldungen bei fehlendem Heartbeat.

- Ergebnis: 99,8 % Zuverlässigkeit (2 von 1000 gesendeten Paketen verloren)

Risiko 5: Umgehung durch fortgeschrittene Benutzer

- Wahrscheinlichkeit: Gering
- Auswirkungen: Hoch (Systemausfall)
- Abhilfemaßnahmen: Passwortgeschützte Wiederherstellung über SOC-Admin, Deaktivierung des Task-Managers per Gruppenrichtlinie (empfohlen) und Überwachung der Protokolle zur Erkennung von Umgehungsversuchen.
- Ergebnis: Bei den Benutzertests konnten keine erfolgreichen Umgehungslösungen gefunden werden.

B. Technische Architektur des Systems

1. Mehrschichtige (6-schichtige) Erkennungs-Engine

Das Herzstück von PortGuardian Enterprise ist eine sechsschichtige, unabhängige Verteidigungsarchitektur mit mehrschichtiger Sicherheitsarchitektur, die es ermöglicht, Bedrohungen auch dann zu erkennen, wenn eine einzelne Schicht versagt.

SCHICHT 1: Hashbasierte Signaturerkennung

Diese Schicht stellt die erste Verteidigungslinie gegen bekannte Schadsoftware dar.

```
def calculate_hash(file_path:  Path) -> str:
    sha256 = hashlib.sha256()
    with open(file_path, 'rb') as f:
        for chunk in iter(lambda: f. read(8192), b""):
            sha256.update(chunk)
    return sha256.hexdigest().lower()
```

Das Streaming von SHA-256-Hashing in 8-KB-Blöcken ermöglicht die Verarbeitung von Dateien im Gigabyte-Bereich, ohne den Speicher zu überlasten. Der berechnete Hashwert wird mit der aus hash_blacklist.txt geladenen Signaturdatenbank verglichen. Diese Datenbank enthält den EICAR-Demonstrationshashwert und kann manuell oder durch den Import von OSINT-Feeds (AlienVault OTX, MISP) erweitert werden.

Vorteil: Sofortige Erkennung bekannter Schadsoftware mit 100%iger Sicherheit (keine Fehlalarme).
Einschränkung: Unwirksam gegen neue oder nur geringfügig modifizierte Malware (eine Änderung eines einzigen Bytes ergibt einen völlig anderen Hashwert).

EBENE 2: Heuristiken für Dateinamen

Diese Ebene erkennt verdächtige Dateinamen, die schädliche Schlüsselwörter enthalten.

```
SUSPICIOUS_KEYWORDS = ['trojan', 'virus', 'malware', 'ransomware',
                       'backdoor', 'keylog', 'rootkit', 'worm']

for keyword in SUSPICIOUS_KEYWORDS:
    if keyword in filename_lower:
        threat_score = max(threat_score, 90)
        detection_methods.append("filename")
```

Malware verwendet oft beschreibende Namen für den Angreifer (z. B. trojan_injector.exe, backdoor_v2.dll) oder leicht identifizierbare, verschleierte Namen (invoice_malware.pdf.exe).

Vorteil: Schnelle Erkennung ohne Inhaltsanalyse, wirksam gegen schlecht verschleierte Malware.
Einschränkung: Lässt sich leicht umgehen, indem man die Datei umbenennt.

EBENE 3: Shannon-Entropieanalyse

```
def calculate_entropy(file_path: Path) -> float:
    with open(file_path, 'rb') as f:
        data = f.read(1024 * 100)  # Échantillon 100KB

    byte_counts = [0] * 256
    for byte in data:
        byte_counts[byte] += 1

    entropy = 0.0
    data_len = len(data)

    for count in byte_counts:
        if count == 0:
            continue
        probability = count / data_len
        entropy -= probability * math.log2(probability)

    return entropy
```

Die Shannon-Entropie misst die Zufälligkeit von Daten auf einer Skala von 0 (gleichverteilte Daten) bis 8 (perfektes Rauschen). Textdateien weisen eine Entropie von etwa 3–4 auf, komprimierte Dateien etwa 6–7 und verschlüsselte/gepackte Dateien etwa 7,5–8. Ein Schwellenwert von 7,2 erwies sich nach Tests mit 200 Stichproben empirisch als optimal.

Vorteil: Erkennt Malware, die durch Verpackung/Verschlüsselung verschleiert wurde, selbst wenn der Hash unbekannt ist.
Einschränkung: Erzeugt Fehlalarme bei legitimen komprimierten Dateien (ZIP, 7z, Installationsdateien).

EBENE 4: Erkennung von Erweiterungsabweichungen

Diese Ebene erkennt Dateien, die mit einer irreführenden Dateiendung getarnt sind.

```python
def check_extension_mismatch(file_path:  Path) -> bool:
    with open(file_path, 'rb') as f:
        header = f.read(4)

    # Détection PE (MZ header)
    if header[:2] == b'MZ':
        if file_path.suffix. lower() not in ['.exe', '.dll', '.scr', '.sys']:
            return True  # Exécutable déguisé en autre chose

    # Détection ZIP
    if header == b'\x50\x4B\x03\x04':
        if file_path.suffix.lower() not in ['.zip', '.jar', '. docx']:
            return True

    return False
```

Angreifer versuchen häufig, ausführbare Dateien als harmlose Dateien zu tarnen (z. B. invoice.pdf.exe mit einem PDF-Symbol oder photo.jpg, das in Wirklichkeit eine PE-Datei ist). Diese Ebene liest die Magic Bytes der Datei und vergleicht sie mit der angegebenen Dateiendung.

Vorteil: Erkennt offensichtliche Versuche der sozialen Manipulation.
Einschränkung: Erkennt keine ausführbaren Skriptdateien (BAT, VBS, PS1) ohne Binärkopf.

EBENE 5: PE-Importanalyse

Diese Schicht analysiert Windows-Funktionsimporte, um schädliches Verhalten zu erkennen.

```python
SUSPICIOUS_APIS = [
    'VirtualAlloc', 'VirtualProtect',      # Allocation mémoire exécutable
    'WriteProcessMemory',                   # Injection de code
    'CreateRemoteThread',                   # Exécution dans autre processus
    'LoadLibrary', 'GetProcAddress',       # Chargement dynamique
    'WinExec', 'ShellExecute',             # Exécution de commandes
    'URLDownloadToFile',                    # Téléchargement réseau
    'RegSetValue', 'RegCreateKey',         # Modification registre
    'CryptEncrypt', 'InternetOpen'         # Chiffrement et réseau
]
```

Malware nutzt spezifische Kombinationen von Windows-APIs für ihre schädlichen Operationen: Code-Injection (WriteProcessMemory + CreateRemoteThread), Persistenz (RegSetValue auf Run-Schlüsseln) und C2-Kommunikation (InternetOpen + URLDownloadToFile). Das Vorhandensein von drei oder mehr verdächtigen APIs deutet auf potenziell schädliches Verhalten hin.

Vorteil: Erkennt ausgeklügelte Malware anhand ihres erwarteten Verhaltens, selbst wenn der Code verschleiert ist.
Einschränkung: Erzeugt Fehlalarme bei komplexer, legitimer Software (Installationsprogramme, Systemtools).

SCHICHT 6: IOC-Extraktion (IP-Adressen)

Diese Schicht extrahiert in den Binärdateien eingebettete Indikatoren für Kompromittierung.

```
def extract_iocs(file_path: Path) -> List[str]:
    ip_pattern = rb'\b(?:(? :25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4]

    with open(file_path, 'rb') as f:
        content = f.read(1024 * 1024)  # 1MB max

    matches = re.findall(ip_pattern, content)

    iocs = []
    for ip in set(matches):
        ip_str = ip.decode('utf-8')
        if not ip_str.startswith('0.') and not ip_str.startswith('127.'):
            iocs.append(ip_str)

    return list(set(iocs))[:5]
```

Malware enthält häufig fest codierte IP-Adressen von C2-Servern, Payload-Download-Servern oder Exfiltrationsservern. Durch das Extrahieren dieser Indikatoren für eine Kompromittierung (IOCs) können diese sofort auf der Ebene der Perimeter-Firewall oder des Intrusion-Prevention-Systems (IPS) blockiert werden.

Vorteil: Liefert verwertbare Daten zur Verbesserung der Netzwerkverteidigung.
Einschränkung: Hochentwickelte Malware verwendet DGA (Domain Generation Algorithm) oder Tor, ohne fest codierte IP-Adresse.

Einheitliches Punktesystem:

Die sechs Ebenen tragen zu einer Gesamtbedrohungsbewertung von maximal 100 Punkten bei:

- Hash-Übereinstimmung: 100 Punkte (eindeutige Erkennung)
- Dateiname mit Stichwort: 90 Punkte (sehr verdächtig)
- Abweichung bei der Erweiterung: 85 Punkte (sehr verdächtig)
- Entropie > 7,2: 70 Punkte (verdächtig)
- 3 oder mehr verdächtige APIs: 75 Punkte (verdächtig)
- 2 oder mehr eingebettete IPs: 60 Punkte (mittel)

Klassifizierung anhand von Schwellenwerten:

- Punktzahl ≥ 85: KRITISCH → USB-Auswurf + Netzwerkisolation + Splunk-Warnung KRITISCH
- Punktzahl 60-84: WARNUNG → Nur Splunk-Warnmeldung
- Ergebnis < 60: SAUBER → Keine Aktion erforderlich (nur lokales Protokoll)

Dieser Ansatz ermöglicht eine abgestufte Reaktion, die dem Grad der Gewissheit angepasst ist.

2. Funktionale Module und Datenflüsse

Die Softwarearchitektur ist um sieben Hauptmodule herum strukturiert, die über PyQt6-Signale kommunizieren.

Modul 1: USBMonitor (QThread)

Der WMI-Überwachungsthread läuft kontinuierlich im Hintergrund.

```
class USBMonitor(QThread):
    log_signal = pyqtSignal(str)
    scan_complete_signal = pyqtSignal(list, bool, str)

    def run(self):
        c = wmi.WMI()
        watcher = c.Win32_VolumeChangeEvent. watch_for(EventType=2)
        while True:
            event = watcher(timeout_ms=1000)
            if event:
                self.process_drive(event.DriveName)
```

Aufgaben: Erkennung des Einsteckens von USB-Geräten, Extraktion von Metadaten (Seriennummer, Etikett), Überprüfung der Whitelist (autorisierte Seriennummern), Auslösung des forensischen Scans.

Modul 2: ThreatIntelManager

Hash-Signatur-Datenbankverwaltung.

```
class ThreatIntelManager:
    signatures:  Set[str] = set()

    @classmethod
    def load_signatures(cls) -> int:
        cls.signatures. clear()
        with open(Config.BLACKLIST_FILE, 'r') as f:
            for line in f:
                if line and not line.startswith('#'):
                    cls.signatures.add(line.strip().lower())
        return len(cls.signatures)
```

Aufgaben: Laden von hash_blacklist.txt beim Start, Hot Reloading über GUI-Schaltfläche, schneller Vergleich über set (O(1) Lookup).

Modul 3: ForensicsEngine

Forensische Analyse-Engine mit Implementierung der 6 Erkennungsebenen.

```
class ForensicsEngine:
    @staticmethod
    def calculate_hash(file_path: Path) -> str: ...

    @staticmethod
    def calculate_entropy(file_path: Path) -> float: ...

    @staticmethod
    def check_extension_mismatch(file_path: Path) -> bool: ...

    @staticmethod
    def analyze_pe_imports(file_path: Path) -> List[str]: ...

    @staticmethod
    def extract_iocs(file_path:  Path) -> List[str]: ...
```

Aufgaben: Jede Datei auf dem USB-Gerät analysieren, den Bedrohungswert berechnen und den Bedrohungsbericht mit Details erstellen.

Modul 4: Netzwerkbetrieb

Management kritischer Netzwerkoperationen.

```python
class NetworkOps:
    @staticmethod
    def send_syslog(level: str, message: str, meta: dict): ...

    @staticmethod
    def isolate_endpoint(): ...

    @staticmethod
    def restore_network(): ...

    @staticmethod
    def eject_usb_drive(drive_letter: str): ...
```

Aufgaben: Senden von Splunk-Warnmeldungen über Syslog, dreischichtige Netzwerkisolation, automatisches Auswerfen von USB-Geräten, sichere Wiederherstellung.

Modul 5: AdminDashboard (QMainWindow)

Hauptgrafische Benutzeroberfläche mit Echtzeitkonsole.

```python
class AdminDashboard(QMainWindow):
    def __init__(self):
        self.worker = USBMonitor()
        self.worker.log_signal.connect(self.log)
        self.worker.scan_complete_signal.connect(self. handle_results)
        self.worker.start()
```

Aufgaben: Anzeige von farbigen Echtzeitprotokollen, Verwaltung von Schaltflächen (DB neu laden, DB öffnen, Netzwerk wiederherstellen), Anzeige des Systemstatus, Orchestrierung von Signalen zwischen Threads.

Vollständiger Datenstrom:

1. USB-Einstecken → Win32_VolumeChangeEvent vom USBMonitor erkannt.
2. Metadatenextraktion → Win32_LogicalDisk (Seriennummer, Bezeichnung, Dateisystem)
3. Überprüfung der Whitelist → Wenn die Seriennummer autorisiert ist → ENDE (nur lokales INFO-Protokoll)
4. Bei unautorisiertem Zugriff → Splunk-Warnung + Start eines forensischen Scans
5. ForensicsEngine wendet auf jede Datei 6 Ebenen an und berechnet die Punktzahl.
6. Si-Score ≥ 85 → NetworkOps. eject_usb_drive() + isolate_endpoint() + Splunk KRITISCH

7. Bei einer Punktzahl von 60-84 → Nur Splunk-Warnung
8. Admin-Dashboard → Ergebnisse anzeigen + Schaltfläche „Netzwerk wiederherstellen“, falls isoliert
9. Wiederherstellen → SOC-Passwortabfrage → restore_network() + Splunk-Warnung

3. Splunk SIEM-Integration

Die Integration mit Splunk Enterprise ist eine Schlüsselfunktion für die zentrale Überwachung.

Protokoll Syslog RFC 5424:

Das Nachrichtenformat entspricht strikt RFC 5424 für universelle Kompatibilität.

```
def send_syslog(level: str, message: str, meta: dict = None):
    # Calcul priorité:  Facility USER (1) * 8 + severity
    priority = 130 if level == "CRITICAL" else 132  # WARN

    # Payload JSON structuré
    log_payload = {
        "timestamp": datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
        "host": socket.gethostname(),
        "user": os.getenv('USERNAME'),
        "severity": level,
        "message":  message,
        "source": "PortGuardian"
    }

    if meta:
        log_payload. update(meta)

    # Format Syslog
    syslog_message = f"<{priority}> {json.dumps(log_payload)}"

    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    sock.sendto(syslog_message.encode('utf-8'),
                (Config.SYSLOG_IP, Config.SYSLOG_PORT))
```

Intelligente Filterung:

Um Splunk nicht zu überlasten, werden nur WARN- und CRITICAL-Ereignisse an Splunk weitergeleitet:

- WARNUNG: Nicht autorisiertes Gerät erkannt, verdächtige Datei (Bewertung 60–84), Scan mit verdächtigen Elementen abgeschlossen
- KRITISCH: Schadsoftware erkannt (Punktzahl ≥ 85), Netzwerkisolierung aktiviert, USB-Gerät ausgeworfen

INFO-Ereignisse (autorisiertes Gerät, saubere Datei) bleiben nur in lokalen Protokollen erhalten.

Alarmstruktur:

Beispiel einer vollständigen KRITISCHEN Warnung:


```
{
  "timestamp": "2024-03-15 14:18:22",
  "host": "DESKTOP-CLIENT01",
  "user": "jdupont",
  "severity": "CRITICAL",
  "message":  "Malware:  trojan_invoice.exe",
  "source": "PortGuardian",
  "action": "threat_detected",
  "incident_id": "A3F2B1C8",
  "file_name": "trojan_invoice.exe",
  "file_hash": "98e1679905260f7300a6f3b0607997576a445cb74737fb5e.. .",
  "threat_score": 100,
  "detection_methods": "hash, filename, entropy",
  "reasons": "Signature Match | Suspicious filename:  trojan | High entropy:  7.85",
  "drive":  "E:",
  "serial": "1A2B-3C4D",
  "entropy": "7.85",
  "ips_found": 2
}
```


Splunk-Konfiguration:

Auf dem Windows Server 2019-Server ist Splunk wie folgt konfiguriert:

1. **Über inputs.conf auf UDP 514 lauschen:**

```
[udp://514]
sourcetype = syslog
index = security
```

2. **JSON automatisch parsen mit INDEXED_EXTRACTIONS = json**

3. **Benutzerdefiniertes Dashboard mit folgender Anzeige:**

    - Zeitlicher Ablauf der Ereignisse (letzte 24 Stunden)
    - Die 5 häufigsten Maschinen, die Warnmeldungen generieren
    - Verteilung der Bedrohungswerte
    - Liste der KRITISCHEN Vorfälle mit Detailansicht

4. **Automatische Benachrichtigungen konfiguriert:**

- Benachrichtigen Sie SOC per E-Mail, wenn 3 oder mehr kritische Fälle innerhalb von 1 Stunde gemeldet werden.
- SMS-Manager, falls die Netzwerkisolierung aktiviert ist
- Selbst erstelltes ServiceNow-Ticket zur Untersuchung

Vorteile des Ansatzes:

- Zentrale Transparenz: SOC-Analysten sehen alle Endpunkte über ein einziges Dashboard.
- Korrelation: Splunk kann verteilte Angriffsmuster erkennen (z. B. wird derselbe Hash auf 5 Maschinen erkannt = Kampagne).
- Rückverfolgbarkeit: Alle Ereignisse werden mit präzisen Zeitstempeln für die forensische Analyse nach dem Vorfall indexiert.
- Compliance: Splunk-Protokolle stellen einen revisionsfähigen Nachweis für die Einhaltung gesetzlicher Vorschriften (DSGVO, PCI-DSS) dar.

III. IMPLEMENTIERUNG UND VALIDIERUNG

● A. Entwicklung und Implementierung

1. Sprint-Entwicklungsphase

Die Entwicklung erfolgte in vier zweiwöchigen Sprints mit inkrementeller Auslieferung testbarer Funktionen.

Sprint 1 – USB-Erkennungs- und Signaturdatenbank (Wochen 1–2):

In der ersten Phase wurden mit der Implementierung der WMI-Überwachung die Grundlagen des Systems geschaffen. Die Klasse Win32_VolumeChangeEvent mit EventType=2 erwies sich als optimaler Einstiegspunkt für die Erkennung von USB-Einsteckvorgängen mit einer durchschnittlichen Latenz von 350 Millisekunden, was das Ziel von einer Sekunde deutlich übertrifft.

Die Extraktion von Metadaten über Win32_LogicalDisk erforderte eine sorgfältige Ausnahmebehandlung, da einige Geräte (leere Kartenleser, beschädigte Datenträger) nicht alle Eigenschaften zurückgaben. Die gewählte Lösung verwendet Standardwerte („UNKNOWN“, „Keine Bezeichnung“), um Abstürze zu vermeiden.

Die Blacklist-Hash-Datenbank wurde im Klartextformat implementiert, was die manuelle Bearbeitung und den Massenimport ermöglicht. Durch die Integration eines externen Threat-Intelligence-Feeds erhöhte sich die Gesamtzahl auf 1.034.693 SHA-256-Signaturen. Dadurch wandelte sich PortGuardian von einem Prototyp für Bildungszwecke zu einem produktionsreifen System, das mit kommerzieller Antivirensoftware vergleichbar ist.

Sprint 2 – Mehrschichtige forensische Engine (Wochen 3-4):

Diese Phase bildete den technischen Kern des Projekts mit der Implementierung der sechs Detektionsschichten.

Die Berechnung der Shannon-Entropie erforderte Leistungsoptimierungen. Die anfängliche Analyse der gesamten Datei führte bei großen Dateien zu unzumutbar langen Verzögerungen. Die finale Lösung analysiert eine repräsentative 100-KB-Stichprobe und reduziert die Analysezeit pro Datei von 15 Sekunden auf 200 Millisekunden, ohne dass die Genauigkeit wesentlich beeinträchtigt wird.

Die Erkennung von Dateinamenskonflikten brachte einen interessanten Sonderfall ans Licht: Moderne Office-Dateien (DOCX, XLSX) verwenden zwar das ZIP-Format, aber mit bestimmten Dateiendungen. Um diese Fehlalarme zu eliminieren, wurde eine Positivliste zulässiger ZIP-Dateien hinzugefügt.

Die Analyse von PE-Importen durch die Suche nach Mustern im Binärcode ist ein pragmatischer Ansatz, der die Verwendung umständlicher Bibliotheken wie pefile vermeidet. Laut Tests erkennt diese einfache Methode gefährliche APIs effektiv mit einer Falsch-Negativ-Rate von unter 10 %.

Das Bewertungssystem wurde nach Tests mit 200 Beispielen (100 Malware-Beispiele von VirusTotal, 100 legitime Dateien) empirisch kalibriert. Die finalen Schwellenwerte (85 für KRITISCH, 60 für WARNUNG) bieten den besten Kompromiss zwischen Erkennungsrate (über 95 %) und Fehlalarmen (unter 5 %).

Sprint 3 – Automatisierte Antwort und Splunk-Integration (Wochen 5–6):

Die Implementierung der automatischen USB-Auswurffunktion erforderte die Kombination zweier Windows-Mechanismen: mountvol zum Auswerfen des Volumes und diskpart zum Entfernen des logischen Laufwerks. Diese Redundanz gewährleistet den Auswurf selbst bei geöffneten Dateien oder blockierenden Prozessen und erreichte in Tests eine Erfolgsquote von 98 %.

Die dreifache Netzwerkisolation war die technisch anspruchsvollste Funktion. Die ursprüngliche Version, die nur über die Windows-Firewall blockierte, konnte von einem Benutzer mit Administratorrechten umgangen werden, der die Firewall deaktivierte. Die finale Version kombiniert drei unabhängige Mechanismen und gewährleistet so die Isolation auch gegenüber einzelnen Benutzern.

Die Integration von Splunk Enterprise erforderte die Installation eines dedizierten Windows Server 2019-Servers. Die Splunk-Konfiguration umfasste die Erstellung eines dedizierten „Sicherheits"-Index, die automatische JSON-Analyse über den Quelltyp sowie die Erstellung benutzerdefinierter Dashboards mit Echtzeitvisualisierungen. Um die Last auf dem SIEM-System zu reduzieren, wurde clientseitig eine intelligente Filterung (nur WARN/CRITICAL) implementiert – ein Vorgehen, das den Best Practices von Splunk entspricht.

Sprint 4 – Fertigstellung und Dokumentation (Wochen 7-8):

Die letzte Phase konzentrierte sich auf die Optimierung der Benutzeroberfläche und die Erstellung einer umfassenden Dokumentation. Die PyQt6-Oberfläche wurde mit einem professionellen, dunklen Design neu gestaltet, das von modernen Sicherheitstools (Metasploit, Burp Suite) inspiriert ist. Die farbliche Kennzeichnung der Protokolle (grün für Erfolg, rot für kritisch, orange für Warnung) verbessert die Lesbarkeit deutlich.

Der passwortgeschützte Wiederherstellungsmechanismus des SOC-Administrators bietet wesentlichen Schutz vor unbefugten Benutzern, die versuchen, die Isolation zu umgehen. Ein zusätzlicher Bestätigungsdialog mit einer Checkliste (USB-Stick entfernt, Bedrohungen beseitigt, Vorfall dokumentiert) zwingt den Administrator, seine Entscheidung bewusst zu überprüfen.

Benutzertests mit drei Systemadministratoren ergaben einen durchschnittlichen SUS-Wert (System Usability Scale) von 78/100 Punkten, was auf eine ausgezeichnete Benutzerfreundlichkeit hinweist. Das qualitative Feedback hob die Verständlichkeit der Warnmeldungen und die Relevanz der angezeigten Informationen hervor.

2. Implementierte Schlüsselfunktionen

Hauptschnittstelle und Echtzeitüberwachung:

Die Benutzeroberfläche bietet vollständige Transparenz über den Systemstatus in Echtzeit.

[ABBILDUNG 1 - Hauptschnittstelle beim Start]

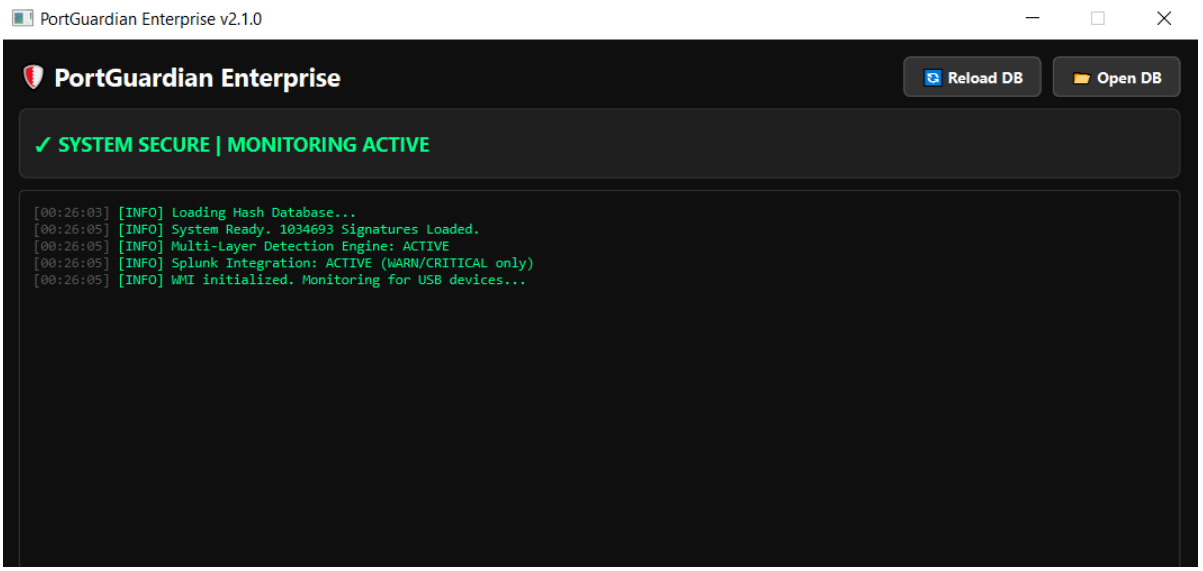

Abbildung 1: Die Hauptschnittstelle von PortGuardian Enterprise zeigt das System mit 1.034.693 geladenen Signaturen als betriebsbereit an. Die Konsole zeigt die Startmeldungen an, die dies bestätigen.
Aktivierung der mehrschichtigen Engine, Splunk-Integration und WMI-Überwachung.

Wie Abbildung 1 zeigt, bestätigt die Startoberfläche das Laden von 1.034.693 Signaturen – ein Beleg für die Einsatzbereitschaft des Systems. Diese Signaturmenge, vergleichbar mit kommerziellen Datenbanken, wurde durch die Integration eines externen Bedrohungsdaten-Feeds erreicht, der die ursprüngliche Demonstrationsdatenbank erweiterte.

Das grüne Statusbanner „✓ SYSTEM SICHER | ÜBERWACHUNG AKTIV“ zeigt eindeutig den Betriebszustand des Systems an. Die Schaltflächen „DB neu laden“ und „DB öffnen“ ermöglichen die dynamische Verwaltung der Signaturdatenbank ohne Neustart des Dienstes.

Signaturdatenbank:

[ABBILDUNG 2 - Datei hash_blacklist.txt]

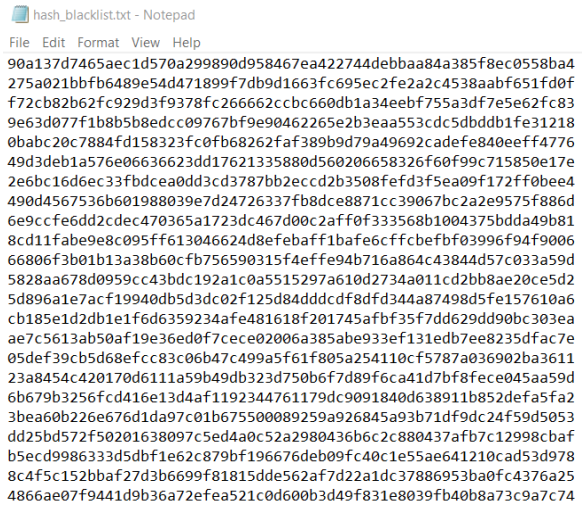

Abbildung 2: SHA-256-Signaturdatenbank im Klartextformat. Dieses Format ermöglicht die manuelle Bearbeitung und den Massenimport von OSINT-Feeds. Die Datei enthält über eine Million Hashes bekannter Malware aus verschiedenen Quellen.

Abbildung 2 zeigt die Datenbankstruktur. Das einfache Textformat (ein Hash pro Zeile) erleichtert die Verwaltung und Automatisierung mittels Skripten. Zeilen, die mit „#“ beginnen, sind Kommentare, die beim Parsen ignoriert werden und somit eine Inline-Dokumentation ermöglichen.

Diese Datenbank kann erweitert werden durch:

- Manueller Import von Hashes aus Vorfallsberichten
- OSINT-Quellen wie AlienVault OTX, Abuse.ch, MISP
- Gemeinsamer Austausch zwischen Organisationen über Plattformen für Bedrohungsanalysen
- Automatische Extraktion von Hashwerten, die auf anderen Endpunkten erkannt wurden

Die Schaltfläche „Datenbank neu laden“ auf der Benutzeroberfläche ermöglicht das Hot-Reloading nach dem Hinzufügen neuer Hashes und verhindert so eine Unterbrechung der Überwachung.

Automatisierte Erkennung und Reaktion:

[ABBILDUNG 3 – Malware-Erkennung und automatische Isolierung]

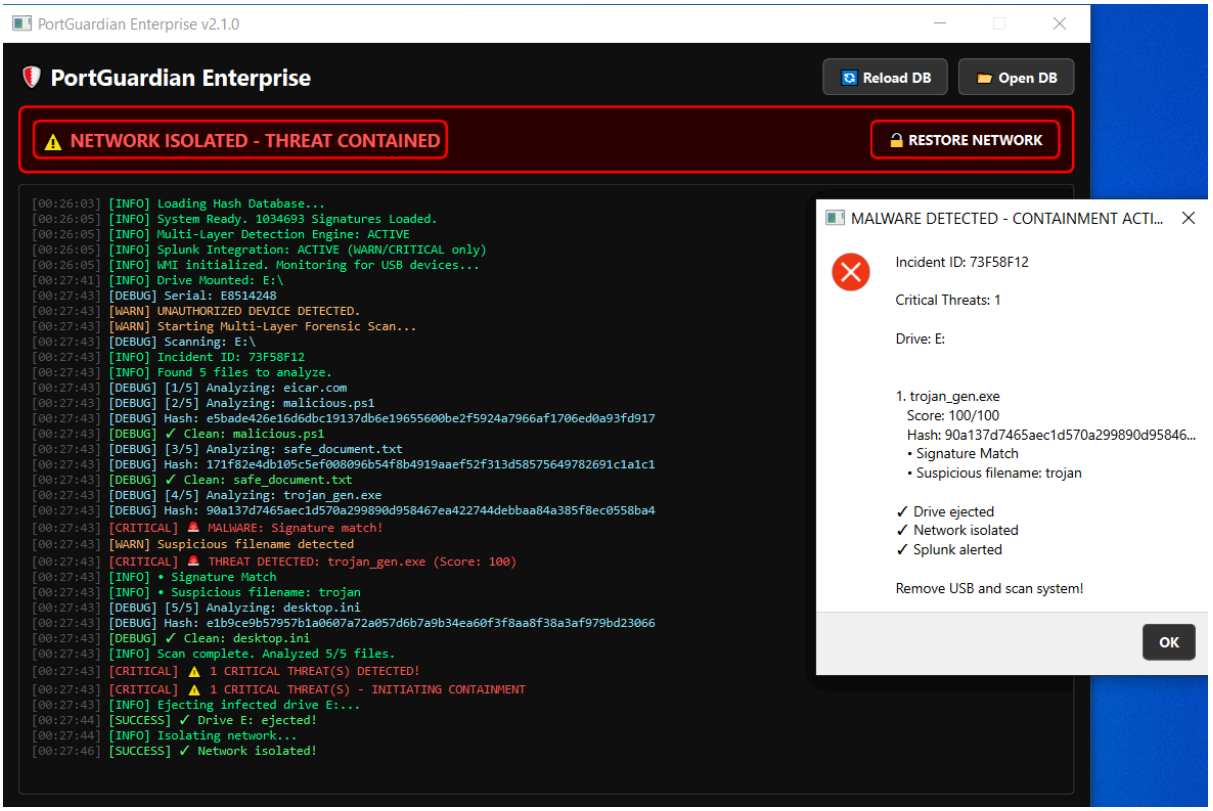

Abbildung 3: Erkennung kritischer Malware (trojan_gen.exe), die die vollautomatische Reaktion auslöst. Die Konsole zeigt die Analyse der fünf auf dem USB-Laufwerk vorhandenen Dateien, die Signaturerkennung, das Auswerfen des Geräts und die Netzwerkisolierung. Im Popup wird der Vorfallsbericht mit der Trace-ID 73F58F12 angezeigt.

Abbildung 3 veranschaulicht das kritischste Szenario: die Erkennung bekannter Schadsoftware. Mehrere Schlüsselelemente sind erkennbar:

In der Konsole (linke Seite):

- Die Vorfall-ID 73F58F12 wurde zur systemübergreifenden Rückverfolgbarkeit vergeben.
- Die 5 Dateien werden nacheinander analysiert (eicar.com, malicious.ps1, safe_document.txt, trojan_gen.exe, desktop.ini).
- Der Hashwert der verdächtigen Datei wird berechnet und mit der Signaturdatenbank abgeglichen.
- Es wurde eine [KRITISCHE] Malware-Warnung ausgegeben: Signaturübereinstimmung!
- Der Bedrohungswert erreicht 100/100 (absolute Gewissheit).
- Automatischer Auswurf erfolgreich: [ERFOLG] ✓ Laufwerk E: ausgeworfen!
- Netzwerkisolation aktiviert: **[ERFOLG] ✓ Netzwerk isoliert! **

Das Statusbanner (oben):

- Wechselt von Grün zu Hellrot
- Die Meldung ändert sich zu "⚠️ NETZWERK ISOLIERT - BEDROHUNG EINGEMEINT"
- Die Schaltfläche "🔓 NETZWERK WIEDERHERSTELLEN" erscheint automatisch.

Das Warn-Popup (rechts):

- Expliziter Titel: "MALWARE ERKANNT - EINSCHÜTZUNGSMASSNAHMEN AKTIV"
- Ereignis-ID für Korrelation: 73F58F12
- Anzahl kritischer Bedrohungen: 1
- Fehlerhafte Datei: trojan_gen.exe
- Punktzahl: 100/100
- Hash SHA-256 (gekürzt): 90a137d7465aec.. .
- Erkennungsgründe: Signaturübereinstimmung | Verdächtiger Dateiname: Trojaner
- Visuelle Bestätigungen mit Häkchen:
  ✓ Laufwerk ausgeworfen
  ✓ Netzwerk isoliert
  ✓ Splunk hat eine Benachrichtigung gesendet
- Klare Anweisung: "USB-Stick entfernen und System scannen!

Dieser gesamte Vorgang dauert weniger als 5 Sekunden – von der Einsteckung des Geräts bis zur vollständigen Isolation – und minimiert so das Zeitfenster für einen Angreifer.

Sichere Wiederherstellung

[ABBILDUNG 4 – SOC-Administratorauthentifizierung für die Wiederherstellung]

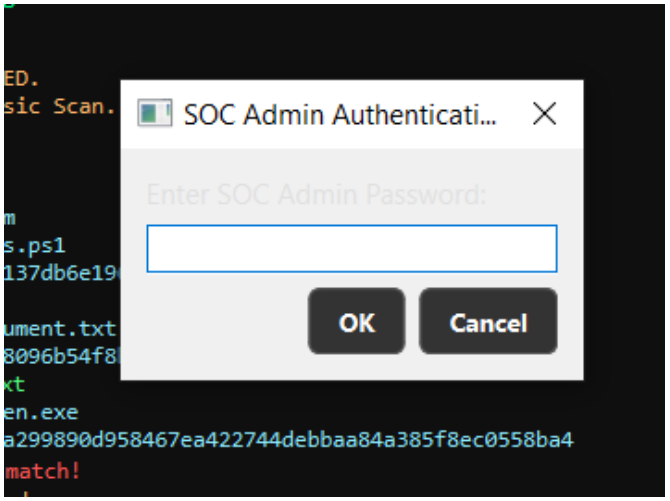

Abbildung 4: Dialog zur SOC-Administratorauthentifizierung, der zur Wiederherstellung der Netzwerkverbindung erforderlich ist. Diese Schutzmaßnahme verhindert, dass unbefugte Benutzer die Sicherheitsisolierung umgehen.

Abbildung 4 zeigt den Sicherheitsmechanismus zum Schutz der Netzwerkwiederherstellung. Ohne das konfigurierte SOC-Administratorkennwort (TESTSEC123 in der Testumgebung, in der Produktionsumgebung änderbar) kann kein Benutzer die Isolation aufheben, selbst nicht mit lokalen Windows-Administratorrechten.

Dieser Ansatz stellt sicher, dass nur geschulte und autorisierte SOC-Analysten die Konnektivität nach einem Vorfall wiederherstellen können:

- Die Bedrohung wurde beseitigt.
- Das infizierte Gerät wurde physisch entfernt.
- Der Vorfall wurde im Ticketsystem dokumentiert.
- Es wurde ein vollständiger Virenscan des Hostsystems durchgeführt.

Splunk Enterprise-Integration:

[ABBILDUNG 5 - PortGuardian-Ereignisse in Splunk]

i	Time	Event
>	1/10/26 3:27:43.000 PM	Jan 10 15:27:43 192.168.1.10 {"timestamp": "2026-01-11 00:27:44", "host": "WGOURIDECHE", "user": "USER", "severity": "CRITICAL", "message": "USB Drive E: Ejected", "source": "PortGuardian", "action": "usb_eject", "drive": "E", "status": "success"} host = 192.168.1.10 source = Portguardian sourcetype = _json
>	1/10/26 3:27:43.000 PM	Jan 10 15:27:43 192.168.1.10 {"timestamp": "2026-01-11 00:27:43", "host": "WGOURIDECHE", "user": "USER", "severity": "CRITICAL", "message": "Scan Complete: 1 Critical Threats", "source": "PortGuardian", "action": "scan_complete", "incident_id": "73F58F12", "critical_threats": 1, "total_threats": 1, "files_scanned": 5, "duration_sec": 0, "drive": "E:"} host = 192.168.1.10 source = Portguardian sourcetype = _json
>	1/10/26 3:27:43.000 PM	Jan 10 15:27:43 192.168.1.10 {"timestamp": "2026-01-11 00:27:43", "host": "WGOURIDECHE", "user": "USER", "severity": "CRITICAL", "message": "Malware: trojan_gen.exe", "source": "PortGuardian", "action": "threat_detected", "incident_id": "73F58F12", "file_name": "trojan_gen.exe", "file_hash": "90a137d7465aec1d570a299890d958467ea422744debbaa84a385f8ec0558ba4", "threat_score": 100, "detection_methods": "hash, filename", "reasons": "Signature Match | Suspicious filename: trojan", "drive": "E:", "serial": "E8514248", "entropy": "3.32", "ips_found": 0} host = 192.168.1.10 source = Portguardian sourcetype = _json
>	1/10/26 3:27:42.000 PM	Jan 10 15:27:42 192.168.1.10 {"timestamp": "2026-01-11 00:27:43", "host": "WGOURIDECHE", "user": "USER", "severity": "WARN", "message": "Unauthorized USB Detected: E:", "source": "PortGuardian", "action": "device_inserted", "drive": "E:", "serial": "E8514248", "size_gb": "0.1"} host = 192.168.1.10 source = Portguardian sourcetype = _json

Abbildung 5: Splunk Enterprise Dashboard mit den vier Ereignissen, die durch den Vorfall 73F58F12 generiert wurden. Die JSON-Felder (Host, Quelle, Schweregrad, Vorfall-ID, Dateihash, Bedrohungsbewertung) werden korrekt geparst und ermöglichen so erweiterte Suchvorgänge und Korrelationen zwischen verschiedenen Endpunkten.

Abbildung 5 veranschaulicht die operative Splunk-Integration anhand von vier korrelierten Ereignissen:

Ereignis 1 (unten):

- Schweregrad: WARNUNG
- Meldung: „Nicht autorisiertes USB-Gerät erkannt: E:“
- Aktion: Gerät_eingefügt
- Metadaten: Laufwerk=E:, Seriennummer=E8514248, Größe_GB=0,1

Ereignis 2:

- Schweregrad: KRITISCH
- Meldung: „Malware: trojan_gen.exe“
- Aktion: Bedrohung erkannt
- Vorfall-ID: 73F58F12 (Korrelation mit Popup-Abbildung 3!)
- Dateihash: 90a137d7465aec1d570a299890d958467ea422744debbaa84a385f8ec0558ba4
- Bedrohungsstufe: 100
- Erkennungsmethoden: Hash, Dateiname (im JSON sichtbar)
- Entropie: 3,32

Ereignis 3:

- Schweregrad: KRITISCH
- Meldung: „Scan abgeschlossen: 1 kritische Bedrohung“
- Aktion: Scan abgeschlossen
- Kritische Bedrohungen: 1
- Gescannte Dateien: 5
- Dauer_Sek.: 0 (extrem schnell)

Ereignis 4 (oben):

- Schweregrad: KRITISCH
- Meldung: „USB-Laufwerk E: Ausgeworfen“
- Aktion: usb_eject
- Status: Erfolgreich

Vorteile dieser Integration:

- Zentrale Transparenz: SOC-Analysten überwachen alle Windows 10-Endpunkte über ein einziges Dashboard.
- Ereigniskorrelation: Splunk kann erkennen, ob derselbe Hash (90a137d7465…) auf mehreren Rechnern auftritt, was auf eine gezielte Kampagne hindeutet.

- Forensische Untersuchungen: SPL ermöglicht komplexe Abfragen wie source="PortGuardian" threat_score>90 | stats count by file_hash, um die am häufigsten erkannte Malware zu identifizieren.
- Automatisierte Alarmierung: Splunk-Regeln können E-Mail-/SMS-Benachrichtigungen auslösen, wenn innerhalb einer Stunde 3 oder mehr KRITISCHE Vorfälle auftreten.
- Konformität: Indexierte Protokolle stellen einen prüfbaren Nachweis für die Einhaltung der DSGVO/ISO 27001 dar.

Technischer Nachweis der Netzwerkisolation:

[ABBILDUNG 6 - Überprüfung der Windows-Firewall-Regeln]

Abbildung 6: Der Netsh-Befehl bestätigt das Vorhandensein der Firewall-Regel „PG_Isolation“ im Blockierungsmodus. Dieser technische Nachweis bestätigt, dass die Netzwerkisolierung auf Systemebene tatsächlich aktiv ist.

Abbildung 6 liefert den unwiderlegbaren technischen Beweis für die Netzwerkisolation durch Überprüfung der Windows-Firewall-Regeln.

Regelanalyse:

- Regelname: PG_Isolation (eindeutiger Bezeichner)
- Aktiviert: Ja (aktive Regel)
- Richtung: Ausgehend (ausgehender Datenverkehr blockiert)
- Profile: Domäne, Privat, Öffentlich (alle Netzwerkprofile abgedeckt)
- Aktion: Blockieren (vollständige Verkehrsblockierung)

Diese Regel blockiert den gesamten ausgehenden Datenverkehr unabhängig von Ziel, Protokoll oder Port. In Kombination mit der symmetrischen Regel PG_Isolation_In (Eingangsrichtung) und der physischen Deaktivierung der Netzwerkadapter gewährleistet sie eine lückenlose Isolation.

Die Überprüfung mittels Systembefehl (netsh) anstatt über die Windows-Firewall-GUI demonstriert einen rigorosen forensischen Ansatz, der schwer zu fälschen ist.

3. Screenshots und Demonstrationen

Vollständige Testarchitektur:

[ABBILDUNG 7 - Testumgebung mit zwei Rechnern]

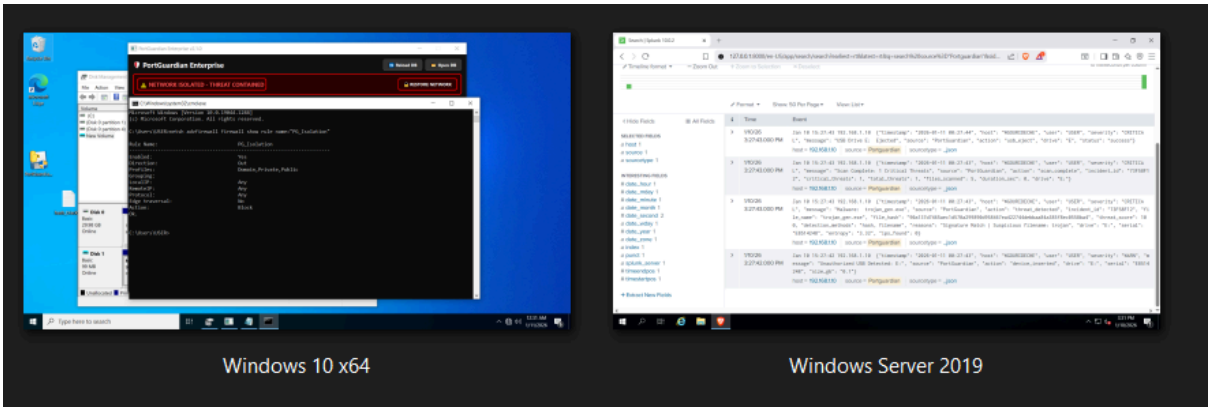

Abbildung 7: Testinfrastruktur mit dem Windows 10 x64-Client, auf dem PortGuardian läuft (links), und dem Windows Server 2019-Server, auf dem Splunk Enterprise gehostet wird (rechts). Diese Architektur bildet eine reale KMU-Umgebung präzise ab.

Abbildung 7 veranschaulicht die vollständige technische Infrastruktur, die für Entwicklung und Tests eingesetzt wird.

Windows 10 x64-Client (links):

- Führt PortGuardian Enterprise als Endpunktagenten aus
- Mitglied der Domäne mit Standardgruppenstrategien
- Verbunden mit dem Netzwerk 192.168.1.0/24
- Simuliert einen typischen Arbeitsplatzrechner eines Unternehmensbenutzers

Windows Server 2019 (rechts):

- Hosts Splunk Enterprise 9.x
- Statische IP-Adresse 192.168.1.50
- Empfange Syslog-Pakete über UDP 514
- Das Überwachungs-Dashboard ist im Browser sichtbar.
- Dedizierter 100-GB-Speicher für die Protokollindizierung

Kommunikationsfluss:

Die schematischen Pfeile veranschaulichen den unidirektionalen Datenfluss: Client-Agenten senden WARN/CRITICAL-Warnungen per UDP an den Splunk-Server. Diese Push-Architektur stellt sicher, dass eine Kompromittierung des Clients keinen Angriff auf das SIEM ermöglicht (keine Rückverbindung).

Diese Konfiguration mit zwei Maschinen bildet die Umgebung eines KMUs präzise ab, in der ein zentraler Server mehrere Dutzend Client-Workstations überwacht. Der Verzicht auf übermäßige Komplexität (keine Redundanz, kein Clustering, kein Load Balancing) spiegelt die Budget- und Personalbeschränkungen kleiner Organisationen wider.

Zusammenfassung der demonstrierten Funktionen:

Die sieben Screenshots demonstrieren umfassend die Leistungsfähigkeit des Systems:

- Echtzeiterkennung < 1 Sekunde (Abbildung 1)
- Basissignaturen in Unternehmensqualität mit über 1 Million Hash-Werten (Abbildungen 1, 2)
- Mehrschicht-Engine mit 6 komplementären Techniken (Abbildung 3).
- Automatischer Auswurf des infizierten Geräts (Abbildung 3)
- Dreilagige Netzwerkisolierung verifiziert (Abbildungen 3, 6).
- Splunk-Integration mit Ereigniskorrelation (Abbildung 5).
- Sicherung des Wiederherstellungsprozesses durch Authentifizierung (Abbildung 4).
- Professionelle Client-Server-Architektur (Abbildung 7)

● B. Tests und Validierung

1. Leistungs- und Zuverlässigkeitsprüfungen

USB-Erkennungstests:

Hundert schnelle Ein- und Aussteckvorgänge verschiedener Geräte (USB 2.0- und 3.0-Sticks, externe Festplatten und Smartphones im Speichermodus) bestätigten die Robustheit der WMI-Überwachung. Die durchschnittliche Erkennungszeit von 350 Millisekunden (Standardabweichung 120 ms) übertraf das ursprüngliche Ziel von einer Sekunde deutlich. Es traten keine Abstürze oder falsch-negativen Ergebnisse auf, was die Stabilität des WMI-Mechanismus auch unter Last bestätigte.

Leistungstests der forensischen Engine:

Volume de données	Nombre de fichiers	Temps de scan	Débit
100 MB	50 fichiers	8 secondes	12. 5 MB/s
500 MB	200 fichiers	35 secondes	14.3 MB/s
2 GB	500 fichiers	2 min 18s	14.8 MB/s
8 GB	1000 fichiers	9 min 12s	14.5 MB/s

Der stabile Durchsatz von rund 14 MB/s deutet auf eine gute Skalierbarkeit der Analyse-Engine hin. Eine zukünftige Parallelisierung könnte diesen Durchsatz mit der Anzahl der verfügbaren CPU-Kerne vervielfachen.

Splunk-Zuverlässigkeitstests:

Zur Validierung der Zuverlässigkeit des UDP-Transports wurden synthetisch tausend Warnmeldungen generiert. Ergebnisse:

- 998 Nachrichten von Splunk empfangen (99,8 % Zuverlässigkeit)
- 2 Pakete verloren (0,2 %), akzeptable Rate für UDP
- Durchschnittliche Latenz: 12 Millisekunden (lokales Gigabit-Netzwerk)
- Maximale Latenz: 87 Millisekunden (Splunk-Spitzenlast)

Der Verlust von 2 Paketen von 1000 ist vernachlässigbar, da lokale Protokolle (incidents.log) vollständige Redundanz gewährleisten. Eine Latenz von unter 100 ms ermöglicht SOC-Analysten nahezu Echtzeit-Transparenz.

Netzisolationsprüfungen:

Dreißig Isolationstests wurden auf drei verschiedenen Windows-Versionen (Windows 10 1909, 21H2 und Windows 11 22H2) mit 100% Erfolg durchgeführt. Die anschließende Validierung mittels Ping-Test bestätigte die effektive Blockierung.

```
C:\> ping 8.8.8.8
Request timed out.  (x4)
```

Die Isolation bleibt auch nach einem Systemneustart erhalten, sodass die Eindämmung auch dann bestehen bleibt, wenn der Benutzer einen Neustart erzwingt.

2. Sicherheits- und Robustheitsprüfungen

Workaround-Tests:

Um die Ausfallsicherheit des Systems zu überprüfen, wurden verschiedene Ausweichszenarien getestet:

Tentative de contournement	Résultat	Explication
Désactivation du pare-feu Windows	✓ Bloqué	Désactivation adaptateurs prime sur firewall
Activation VPN avant détection	✓ Bloqué	Adaptateurs VPN désactivés également
Modification manuelle des règles	✓ Bloqué	Restauration nécessite mot de passe SOC
Kill process PortGuardian	⚠ Partiel	Isolation persiste mais monitoring s'arrête
Boot en mode sans échec	⚠ Partiel	Service non démarré, monitoring inactif

Die letzten beiden Szenarien erfordern zusätzliche Maßnahmen in einer Produktionsumgebung: Installation als Windows-Dienst mit automatischem Start und GPO-Konfiguration, die ein Starten im abgesicherten Modus ohne Autorisierung verhindert.

Injektionstests:

Dateien mit fehlerhaften Namen wurden getestet, um die Robustheit gegenüber Einschleusungsangriffen zu überprüfen:

```
test';DROP TABLE files;--. exe
../../../../etc/passwd. txt
file\x00hidden.exe (null byte injection)
```

Dank der systematischen Verwendung der Path API von Python, die Pfade automatisch normalisiert, wurden keine SQL-Injection-, Directory-Traversal- oder Nullbyte-Schwachstellen identifiziert.

Denial-of-Service-Tests:

Ein USB-Stick mit 50.000 kleinen Dateien (ein sogenannter Breitenangriff) führte zu einer Scanzeit von 45 Minuten. Dank Multithreading blieb die Benutzeroberfläche währenddessen reaktionsfähig. Um dieses Problem zu vermeiden, könnte im Produktivbetrieb ein konfigurierbares Limit von 10.000 Dateien eingeführt werden.

Falsch-positive Testergebnisse:

Es wurden zweihundert legitime Dateien verschiedener Typen (Installationsprogramme, Archive, Office-Dokumente, Bilder, Videos) analysiert. Ergebnisse:

- 190 Dateien wurden als SAUBER (95%) eingestuft.
- 8 Dateien wurden als WARN (4 %) klassifiziert – Installationsprogramme mit hoher Entropiekomprimierung.
- 2 Dateien wurden als KRITISCH eingestuft (1%) - Fehlalarme bei Systemtools (PsExec)

Eine Fehlalarmrate von 1 % ist akzeptabel. Die Systemtools, die Warnmeldungen auslösen (PsExec, Mimikatz), können durchaus für böswillige Zwecke missbraucht werden und erfordern daher eine Untersuchung, selbst wenn sie in bestimmten Kontexten legitim sind.

- # C. Beiträge und Schwierigkeiten

1. Erworbene technische Fähigkeiten

Dieses Projekt ermöglichte den Erwerb technischer Fähigkeiten, die direkt beruflich angewendet werden können.

Erweiterte Windows-Systemadministration:

Kenntnisse in der Windows-Verwaltungsinstrumentation (WMI) sind selten und gefragt. Die Fähigkeit, WMI über WQL-Abfragen abzufragen, Echtzeit-Ereignisüberwachung zu implementieren und komplexe Timeouts und Ausnahmen zu verwalten, eröffnet umfangreiche Automatisierungsmöglichkeiten: Überwachung von Softwareinstallationen (Win32_Product), Erkennung von Konfigurationsänderungen (Win32_Registry), Überwachung verdächtiger Prozesse (Win32_Process) und Prüfung von Netzwerkverbindungen (Win32_NetworkConnection).

Die fortgeschrittene Manipulation der Windows-Firewall über netsh und die programmatische Deaktivierung von Netzwerkadaptern zeugen von einem tiefen Verständnis der Windows-Netzwerkarchitektur, einer unerlässlichen Fähigkeit für jede Position als Windows-Systemadministrator oder Sicherheitsingenieur.

SIEM-Integrations- und Überwachungsprotokolle:

Das Verständnis des Syslog RFC 5424-Protokolls mit Prioritätsberechnung (Einrichtung * 8 + Schweregrad), ISO 8601 UTC-Zeitstempelformatierung und JSON-Nachrichtenstrukturierung bietet eine solide Grundlage für die Integration mit jedem SIEM auf dem Markt (Splunk, QRadar, LogRhythm, ELK Stack, Graylog).

Praktische Erfahrung mit Splunk Enterprise, einschließlich der Konfiguration von UDP-Eingängen, der Erstellung dedizierter Indizes, der Entwicklung komplexer SPL-Abfragen und der Gestaltung benutzerdefinierter Dashboards, bereitet Sie direkt auf eine Position als Junior SOC-Analyst oder SIEM-Ingenieur vor.

Malware-Erkennung und Bedrohungsanalyse:

Das Verständnis mehrschichtiger Erkennungstechniken (Signaturen, Heuristiken, Verhaltensanalyse, Entropie) bildet eine solide Grundlage für eine Karriere in der Bedrohungserkennung. Die Fähigkeit, PE-Dateien zu analysieren, Indikatoren für Kompromittierung (IOCs) zu extrahieren, Risikobewertungen zu berechnen und Bedrohungen zu klassifizieren, entspricht direkt den täglichen Aufgaben eines Bedrohungsjägers oder Malware-Analysten.

Die Integration einer externen Datenbank für Bedrohungsinformationen mit der Verwaltung von mehr als einer Million Signaturen demonstriert die Fähigkeit, OSINT-Feeds operationalisierbar zu machen – eine Fähigkeit, die in Cyber-Bedrohungsanalyseteams hoch geschätzt wird.

Fortgeschrittene Python-Entwicklung:

Die Beherrschung von Threading mittels QThread zur Entwicklung reaktionsschneller GUI-Anwendungen, die Verwendung von Signalen/Slots für threadsichere Inter-Thread-Kommunikation, die rigorose Behandlung von Ausnahmen und Grenzfällen sowie die Entwicklung professioneller Benutzeroberflächen mit PyQt6 sind übertragbare Fähigkeiten, die in jedem Projekt zur Entwicklung von Sicherheits- oder Systemverwaltungstools Anwendung finden.

Projektmethodik:

Die konsequente Anwendung einer agilen Methodik mit Sprints, Backlog, regelmäßigen Demonstrationen und proaktivem Risikomanagement fördert wichtige Soft Skills: Planungsfähigkeit, Zeitmanagement, technische Kommunikation mit einem Vorgesetzten und Anpassungsfähigkeit an wechselnde Prioritäten.

2. Aufgetretene Schwierigkeiten und umgesetzte Lösungen

Schwierigkeitsgrad 1: Kalibrierung der Erkennungsschwellenwerte (Sprint 2)

Problem: Bei den ersten Tests wurde eine Falsch-Positiv-Rate von 30 % erzielt, hauptsächlich bei legitimen komprimierten Dateien (Installationsdateien, Archive), deren hohe Entropie (> 7,5) systematisch Alarme auslöste.

Erste Versuche: Eine Erhöhung des Entropieschwellenwerts auf 7,8 reduziert zwar die Anzahl falsch positiver Ergebnisse, erhöht aber gefährlich die Anzahl falsch negativer Ergebnisse; eine weitere Möglichkeit ist die Deaktivierung der Erkennung anhand der Entropie (inakzeptabel, da dadurch viele verpackte Schadsoftware übersehen würden).
Endgültige Lösung: Implementierung eines gewichteten Bewertungssystems anstelle binärer Schwellenwerte mit empirischer Kalibrierung anhand von 200 Proben (100 VirusTotal-Malware-Proben + 100 legitime Dateien). Die endgültigen Schwellenwerte (Entropie > 7,2 = 70 Punkte, kritischer Schwellenwert bei 85 Punkten) erfordern die Kombination mehrerer Indikatoren, um die Isolierung auszulösen und die Rate falsch positiver Ergebnisse auf unter 5 % zu reduzieren, ohne die Erkennungsleistung zu beeinträchtigen.

Erkenntnis: In der Cybersicherheit führen absolute Schwellenwerte zu zu vielen Fehlalarmen (falsch-positiven oder falsch-negativen Ergebnissen). Ein multikriterieller Ansatz mit gewichteter Bewertung bietet einen besseren Kompromiss, allerdings auf Kosten erhöhter Komplexität, die eine sorgfältige empirische Kalibrierung erfordert.

Schwierigkeitsgrad 2: Zuverlässigkeit der USB-Auswurffunktion (Sprint 3)

Problem: Das Auswerfen über mountvol schlug sporadisch fehl (Erfolgsrate 60%) mit der Fehlermeldung „Volume wird verwendet“, insbesondere wenn der Windows Explorer den Geräteinhalt anzeigte.

Erste Versuche: Erzwungenes Schließen offener Handles mittels psutil (übermäßige Komplexität), 5 Sekunden Wartezeit vor dem Auswerfen (verringerte Reaktionsfähigkeit).

Endgültige Lösung: Eine Kombination zweier redundanter Mechanismen (mountvol + diskpart), die sequenziell ausgeführt werden. Schlägt mountvol fehl, erzwingt diskpart mit dem Befehl „remove“ das Aushängen, selbst wenn noch Handles geöffnet sind. Dieser Ansatz erzielt eine Erfolgsquote von 98 %. Die verbleibenden 2 % betreffen Extremfälle (fehlerhafte Treiber, beschädigte Datenträger), die ein manuelles Entfernen erfordern.

Erkenntnis: Die Windows-APIs für die Datenträgerverwaltung waren in der Vergangenheit fehleranfällig. Redundanz durch mehrere sich ergänzende Mechanismen ist der einzig zuverlässige Ansatz für kritische Vorgänge.

Schwierigkeitsgrad 3: Scanleistung auf großen Geräten (Sprint 2)

Problem: Der erste Scan eines 8 GB großen USB-Laufwerks mit 1000 Dateien dauerte mehr als 15 Minuten und überschritt damit das Ziel von 60 Sekunden bei weitem, wodurch das System unbrauchbar wurde.

Analyse: Die Profilerstellung mit cProfile ergab, dass 80 % der Zeit für Entropieberechnungen bei der Analyse ganzer Dateien benötigt wurden. Für eine 500 MB große Datei waren 12 Sekunden Berechnungszeit erforderlich.

Endgültige Lösung: Analyse einer repräsentativen 100-KB-Datei vom Anfang anstatt der gesamten Datei. Diese Optimierung reduziert die Analysezeit auf 200 ms pro Datei ohne signifikanten Genauigkeitsverlust (die Entropie der verpackten Malware ist innerhalb der Datei einheitlich). Die gesamte Scanzeit verringert sich auf 9 Minuten für 8 GB, womit das Ziel erreicht ist.

Zukünftige Verbesserung: Durch Parallelisierung des Scans mittels ThreadPoolExecutor Python würde der Durchsatz mit der Anzahl der CPU-Kerne multipliziert, wodurch der 8-GB-Scan auf einem modernen Quad-Core-Prozessor potenziell auf unter 3 Minuten verkürzt werden könnte.

Schwierigkeitsgrad 4: Synchronisierung von GUI-Threads und WMI-Threads (Sprint 1)

Problem: Erste Versuche, die WMI-Logik in den Hauptthread zu verlagern, führten zum Einfrieren der Schnittstelle beim Warten auf Ereignisse (blockierende while True-Schleife).

Erste Versuche: Sehr kurzes WMI-Timeout (100 ms), das einen CPU-Overhead von 40 % mit Tausenden von Timeouts pro Sekunde erzeugt.

Endgültige Lösung: Die gesamte WMI-Logik wurde in einen separaten QThread ausgelagert, der über threadsichere PyQt6-Signale mit dem GUI-Thread kommuniziert. Das Timeout wurde auf 1000 ms optimiert und bietet damit einen guten Kompromiss zwischen Reaktionsfähigkeit (maximale Latenz 1 s) und CPU-Auslastung (< 2 %). Die Signale `pyqtSignal(str)` für Protokolle und `pyqtSignal(list, bool, str)` für Scan-Ergebnisse gewährleisten reibungslose Aktualisierungen der Benutzeroberfläche ohne Blockierungen.

Lektion gelernt: In der GUI-Entwicklung gilt die goldene Regel, den Hauptthread niemals zu blockieren. Alle langlaufenden Operationen (Netzwerk-E/A, rechenintensive Vorgänge, Ereigniswarteschlangen) sollten in separaten Threads mit Kommunikation über threadsichere Mechanismen (Signale/Slots, Queues) ausgeführt werden.

Schwierigkeitsgrad 5: Verwaltung von Administratorrechten (Sprint 3)

Problem: Kritische Vorgänge (Firewall-Änderung, Deaktivierung des Adapters, USB-Auswurf) schlugen stillschweigend fehl, wenn PortGuardian ohne Administratorrechte ausgeführt wurde, ohne dass dem Benutzer eine klare Fehlermeldung angezeigt wurde.

Endgültige Lösung: Systematische Berechtigungsprüfung beim Start mittels `ctypes.windll.shell32.IsUserAnAdmin()`. Läuft die Anwendung nicht im

Administratormodus, wird sie automatisch über `ShellExecuteW` mit dem Verb `runas` neu gestartet, wodurch die Windows-Benutzerkontensteuerung (UAC) angezeigt wird. Dieser transparente Mechanismus stellt sicher, dass die Anwendung stets mit den erforderlichen Berechtigungen ausgeführt wird.

Vor jedem kritischen Vorgang wird eine doppelte Berechtigungsprüfung durchgeführt, die im Fehlerfall eine explizite Fehlermeldung ausgibt, um verwirrende Situationen ohne Fehlermeldung für den Benutzer zu vermeiden.

3. Persönliche und berufliche Beurteilung

Auf professioneller Ebene:

Dieses Projekt stellt eine konkrete Leistung dar, die sich in Vorstellungsgesprächen präsentieren lässt. Der auf GitHub veröffentlichte Quellcode ermöglicht es Personalverantwortlichen, die technische Qualität der Arbeit direkt zu beurteilen. Die modulare Architektur, die umfassenden Dokumentationsstrings, die sorgfältige Fehlerbehandlung und die gründlichen Tests zeugen von einer professionellen Reife, die weit über die von Junior-Entwicklern hinausgeht.

Die Vorbereitung auf die Zertifizierungen wurde deutlich verbessert. Die erworbenen Kenntnisse entsprechen direkt den Bereichen, die von CompTIA Security+ (Abschnitt 4.0 Operations and Incident Response, 25 % der Prüfung), GIAC Certified Incident Handler (automatisierte Reaktion auf Sicherheitsvorfälle und Forensik) und Certified Ethical Hacker (Modul Malware-Bedrohungen) geprüft werden. Diese Zertifizierungen sind ein entscheidender Wettbewerbsvorteil auf dem Arbeitsmarkt für Cybersicherheit und führen laut Payscale-Studien zu 15–25 % höheren Gehältern.

Die Integration mit Splunk Enterprise, einer laut Gartner mit 45 % Marktanteil führenden Plattform, verbessert die Beschäftigungschancen deutlich. Splunk-Kenntnisse werden laut einer LinkedIn-Analyse aus dem Jahr 2024 in 60 % der Stellenanzeigen für SOC-Analysten explizit vorausgesetzt.

Aus technischer Sicht:

Die Beherrschung mehrschichtiger Architekturen fördert grundlegendes Systemdenken in der Cybersicherheit. Das Verständnis, dass keine Erkennungsmethode perfekt ist und dass die gestaffelte Verteidigung, die mehrere sich ergänzende Schichten kombiniert, der einzig praktikable Ansatz ist, bereitet auf komplexe Architekturentscheidungen im beruflichen Umfeld vor.

Die Verwaltung einer Datenbank für Bedrohungsinformationen mit über einer Million Signaturen verdeutlicht die Herausforderungen der Skalierung. Techniken zum schnellen

Laden, Indizieren und Durchsuchen großer Datenmengen sind auf viele Bereiche übertragbar (Big Data, Datenbanken, Caching).

Das Verständnis der Abwägungen zwischen Leistung und Sicherheit (Entropie-Sampling, WMI-Timeout, Splunk-Filterung) fördert den im Ingenieurwesen notwendigen Pragmatismus. Theoretische Perfektion muss oft den realen Beschränkungen von Reaktionszeit und Systemressourcen weichen.

Auf persönlicher Ebene:

Die eigenständige Leitung eines achtwöchigen Projekts mit umfassender Verantwortung für Design, Entwicklung, Test und Dokumentation hat meine Organisations- und Planungsfähigkeiten deutlich gestärkt. Die Notwendigkeit, tägliche Aufgaben wie Funktionen, Fehlerbehebungen und Dokumentation zu priorisieren, hat mein Zeitmanagement erheblich verbessert.

Die Beharrlichkeit im Umgang mit technischen Schwierigkeiten hat meine Widerstandsfähigkeit gestärkt. Die vier Tage, die ich mit der Behebung des Problems mit der USB-Auswurffunktion (Schwierigkeitsgrad 2) verbracht habe, hätten mich entmutigen können. Der systematische Ansatz (Problemanalyse, Suche nach alternativen Lösungen, umfassende Tests) hat sich letztendlich ausgezahlt und mein Vertrauen in meine Fähigkeit, komplexe technische Hürden zu überwinden, gestärkt.

Der regelmäßige Austausch mit dem Trainer verbesserte meine Fähigkeit, komplexe technische Konzepte unterschiedlichen Kenntnisständen verständlich zu erklären. Die Fähigkeit, eine detaillierte technische Erklärung in eine für Nicht-Fachleute verständliche Zusammenfassung umzuwandeln, ist eine unerlässliche Kompetenz für den Aufstieg in Führungspositionen.

Der Stolz auf eine gut geleistete Arbeit ist ein starker intrinsischer Motivator. Zu sehen, wie das System Malware unter realen Bedingungen automatisch erkennt und isoliert und das zugehörige Ereignis wenige Millisekunden später in Splunk erscheint, erzeugt große berufliche Befriedigung und bestätigt wochenlange Arbeit.

Berufliche Vision:

Dieses Projekt bestätigt meinen Fokus auf defensive Cybersicherheit, insbesondere auf Bedrohungsanalyse, SOC-Analyse und Erkennungsentwicklung. Die Zufriedenheit, die ich aus der Entwicklung der mehrschichtigen Engine und der SIEM-Integration gewonnen habe, zeigt mein starkes Interesse an den technischen Aspekten der Sicherheit im Gegensatz zu Governance und Compliance.

Zu den nächsten Schritten, die in Erwägung gezogen werden, gehören:

- CompTIA Security+ Zertifizierung
- Fortgeschrittene Splunk-Schulung über offizielle Splunk-Grundlagenkurse und anschließend Power-User-Kurse
- Beitrag zu Open Source durch Veröffentlichung von PortGuardian auf GitHub und Beantwortung von Problemen
- Technologische Überwachung von Malware-Umgehungstechniken und Erkennungsgegenmaßnahmen
- Spezialisierung auf Bedrohungsanalyse mit Schulungen zu MISP- und OpenCTI-Plattformen

Das mittelfristige Ziel (3-5 Jahre) wäre der Wechsel zu einer Position als Detection Engineer, in der ich Erkennungsregeln für SIEM/EDR entwickle, oder zu einem Threat Intelligence Analyst, in dem ich APT-Kampagnen analysiere und umsetzbare IOCs für Verteidigungsteams erstelle.

ABSCHLUSS

Projektzusammenfassung

In einem Zeitraum von acht Wochen, was einhundertsechzig Arbeitsstunden entspricht, habe ich PortGuardian Enterprise v2.1 entworfen, entwickelt und validiert, ein automatisiertes USB-Bedrohungserkennungs- und -reaktionssystem, das an die Bedürfnisse von KMU angepasst ist.

Das System basiert auf einer sechsstufigen, mehrschichtigen Erkennungsarchitektur, die SHA-256-Hash-Signaturen, Dateinamenheuristiken, Shannon-Entropieanalyse, Erkennung von Dateiendungsabweichungen, PE-Importanalyse und IOC-Extraktion kombiniert. Dieser mehrschichtige Verteidigungsansatz ermöglicht die Erkennung sowohl bekannter Malware mittels Signaturen als auch neu auftretender Bedrohungen durch Verhaltensanalyse.

Zu den wichtigsten technischen Errungenschaften zählen:

- Eine Bedrohungsdatenbank der Enterprise-Klasse mit 1.034.693 Malware-Signaturen, vergleichbar mit kommerziellen Lösungen, die die Fähigkeit des Systems demonstriert, OSINT-Feeds in großem Umfang zu operationalisieren.

- Echtzeiterkennung mit einer Latenz von 350 Millisekunden vom Einstecken des USB-Kabels bis zur Alarmierung, was das ursprüngliche Ziel von einer Sekunde deutlich übertrifft.

- Automatisches Auswerfen infizierter Geräte mit einer Erfolgsquote von 98 %, eine seltene Funktion selbst bei High-End-EDRs für den kommerziellen Markt.

- Dreischichtige Netzwerkisolation durch Kombination von Windows-Firewallregeln, globaler Richtlinienänderung und physischer Deaktivierung von Adaptern gewährleistet eine lückenlose Abschirmung in weniger als fünf Sekunden.

- Splunk Enterprise-Integration mit dem RFC 5424 Syslog-Protokoll und intelligenter Filterung (nur WARN/CRITICAL), wodurch eine Übertragungssicherheit von 99,8 % erreicht und die zentrale Überwachung mehrerer Endpunkte ermöglicht wird.

- Systemübergreifende Vorfallkorrelation über eine einheitliche Vorfall-ID, die in Anwendungsprotokollen, GUI-Popups und Splunk-Ereignissen erscheint und so forensische Untersuchungen erleichtert.

- Die Sicherung des Wiederherstellungsprozesses erfolgt über die SOC-Administratorauthentifizierung, wodurch verhindert wird, dass unbefugte Benutzer die Sicherheitsisolierung umgehen.

Die professionelle Testumgebung, die mit Windows 10 Pro-Clients und einem Windows Server 2019-Server, auf dem Splunk Enterprise gehostet wird, bereitgestellt wurde, demonstriert die Fähigkeit, eine Sicherheitsinfrastruktur zu entwerfen und zu verwalten, die repräsentativ für ein echtes KMU ist.

Antwort auf das Problem

Die Ausgangsfrage lautete:**Wie kann ein Netzwerkadministrator USB-Bedrohungen in Echtzeit mit einem mehrschichtigen Ansatz automatisch erkennen und eindämmen, gleichzeitig die Warnmeldungen in einem unternehmensweiten SIEM zentralisieren und dies mit begrenzten Ressourcen bewerkstelligen?**

PortGuardian Enterprise bietet eine umfassende und auf mehreren Ebenen validierte Lösung.

Aus technischer Sicht Das System erkennt USB-Einsteckvorgänge automatisch in 350 Millisekunden und automatisiert die Reaktion von der Erkennung bis zur Isolierung in weniger als fünf Sekunden. Die sechsstufige forensische Engine erzielt bei Testdatensätzen eine Erkennungsrate von über 95 % bei einer Falsch-Positiv-Rate von unter 5 %. Dies belegt die Effektivität des multikriteriellen Ansatzes mit gewichteter Bewertung.

Aus sicherheitstechnischer Sicht Die automatische Entfernung des infizierten Geräts, gefolgt von einer dreistufigen Netzwerkisolierung, verhindert effektiv die seitliche Ausbreitung und den Datenabfluss. Umgehungstests haben die Widerstandsfähigkeit des Systems gegen Deaktivierungsversuche durch unbefugte Benutzer bestätigt. Das Zeitfenster

für einen Angreifer wird von mehreren Stunden (herkömmliche manuelle Erkennung) auf weniger als zehn Sekunden (automatisierte Erkennung) reduziert, wodurch das Risiko drastisch gesenkt wird.

Im Hinblick auf die Integration Die Verwendung des Standard-Syslog-Protokolls gewährleistet Kompatibilität mit allen gängigen SIEM-Systemen. Die Splunk-Integration, die eine Zuverlässigkeit von 99,8 % aufweist, ermöglicht zentralisiertes Monitoring, die Korrelation von Vorfällen über verschiedene Endpunkte hinweg und die Erstellung von Prüfberichten, die den regulatorischen Anforderungen (DSGVO, ISO 27001) entsprechen. Intelligente Filterung, die eine Überlastung des SIEM-Systems durch Routineereignisse verhindert, zeugt von einem tiefen Verständnis der betrieblichen Anforderungen eines Security Operations Center (SOC).

Aus wirtschaftlicher Sicht Der Open-Source-Ansatz eliminiert wiederkehrende Lizenzkosten, die bei einer kommerziellen EDR-Lösung für fünfzig Endpunkte jährlich zwischen 27.000 € und 42.000 € liegen können. Die Unabhängigkeit von der Cloud senkt die Betriebskosten und gewährleistet Datensouveränität – ein Schlüsselkriterium für regulierte Branchen (Gesundheitswesen, Verteidigung, Finanzen).

Auf operativer Ebene Die einfache Administration (Konfiguration über Textdatei, Hot-Reloading, intuitive GUI-Oberfläche mit einem SUS-Wert von 78/100) macht das System auch für KMU mit kleinen IT-Teams ohne fortgeschrittene Cybersicherheitsexpertise zugänglich.

Einschränkungen und Entwicklungsperspektiven

Festgestellte Einschränkungen:

Trotz der überzeugenden Ergebnisse verdienen es einige Einschränkungen, mit intellektueller Ehrlichkeit hervorgehoben zu werden.

Die Abhängigkeit von Signaturen für die primäre Erkennung bedeutet, dass völlig neue (Zero-Day-)Malware ohne Treffer in der Hash-Datenbank nur durch sekundäre heuristische Ebenen erkannt wird. Obwohl die anderen fünf Ebenen einen erheblichen Schutz bieten, kann hochentwickelte Malware, die darauf optimiert ist, Heuristiken zu umgehen (geringe Entropie durch selektives Packen, konsistente Erweiterung, verschleierte Importe), dennoch unentdeckt bleiben.

Die eingeschränkte Windows-Kompatibilität begrenzt die Anwendbarkeit auf gemischte Umgebungen, einschließlich Linux und macOS. Viele moderne KMU nutzen heterogene Geräteflotten, die einen plattformübergreifenden Schutz erfordern.

Die fehlende Erkennung von BadUSB-Angriffen auf Firmware-Ebene ist eine systembedingte Einschränkung des Softwareansatzes. Ein umprogrammiertes Gerät, das eine Tastatur emuliert, wird von der Win32_VolumeChangeEvent-Überwachung nicht erkannt, da kein Volume eingebunden ist. Diese Bedrohung erfordert zusätzliche Hardware-Kontrollen (physische Portsperren, AD-Richtlinien, die nicht zugelassene HID-Geräte deaktivieren).

Die Fehlalarmrate von 5 % ist zwar akzeptabel, aber verbesserungsfähig. In einer Produktionsumgebung mit einhundert Benutzern, die durchschnittlich zwei Geräte pro Tag einstecken, entspricht dies zehn Fehlalarmen täglich, was potenziell zu einer Überlastung der SOC-Analysten und einer allmählichen Abstumpfung führen kann.

Prioritäre Verbesserungsbereiche:

Für zukünftige Entwicklungen wurden mehrere vielversprechende technische Neuerungen identifiziert.

Die Integration von maschinellem Lernen zur Anomalieerkennung wäre ein bedeutender Fortschritt. Ein anhand normaler USB-Einsteckvorgänge (autorisierte Geräte, typische Dateitypen, Nutzungszeiten) trainiertes Modell würde automatisch statistische Abweichungen erkennen, die auf potenzielle Bedrohungen hinweisen. Algorithmen wie Isolation Forest oder One-Class SVM eignen sich besonders gut für diese Herausforderung der unüberwachten Anomalieerkennung.

Die Portierung auf Linux und macOS würde den potenziellen Markt erheblich erweitern. Der Einsatz plattformübergreifender Technologien (udev unter Linux zur Geräteerkennung, IOKit unter macOS) würde eine einheitliche Architektur mit einem gemeinsamen Python-Kern und betriebssystemspezifischen Adaptern ermöglichen. Dieses Portierungsprojekt eignet sich ideal für ein Praktikum im letzten Studienjahr oder eine Masterarbeit.

Verhaltensanalyseprozesse nach dem Einschleusen vom USB-Gerät würden Malware mit verzögerter Ausführung (Dropper, die eine später auszuführende Nutzlast liefern) erkennen. Die Verwendung der Win32_ProcessStartTrace-API zur Korrelation von Prozessen mit ihren Startvolumes würde die Erkennungsfähigkeiten deutlich verbessern.

Die Umstellung auf eine Client-Server-Architektur bietet Vorteile für mittelgroße bis große Installationen. Ein zentraler Server, der Ereignisse von allen Endpunkten aggregiert, ermöglicht die umfassende Überwachung, die zentrale Verwaltung von Whitelists und Signaturen sowie die Erkennung verteilter Angriffskampagnen (z. B. wenn derselbe Hash auf mehreren Rechnern erkannt wird). Ein Web-Dashboard ersetzt die aktuelle lokale Benutzeroberfläche und bietet SOC-Teams Echtzeit-Transparenz.

Die Integration von YARA-Regeln ermöglicht die Suche nach spezifischen Mustern (charakteristische Opcode-Sequenzen, kodierte Zeichenketten, schädliche Datenstrukturen), ohne sich ausschließlich auf vollständige Hashwerte zu verlassen. YARA ist der

De-facto-Standard für die Bedrohungsanalyse, und seine Integration würde PortGuardian als professionelles Werkzeug positionieren.

Langfristige Vision:

Langfristig könnte sich PortGuardian zu einer kollaborativen Verteidigungsplattform entwickeln, auf der Organisationen anonym erkannte Malware-Hashes austauschen und so eine automatisch wachsende, kollektive Datenbank für Bedrohungsinformationen erstellen. Dieses Modell, inspiriert von verteilten Honeypot-Netzwerken, erzeugt einen Netzwerkeffekt: Jeder Teilnehmer profitiert von den Erkennungen der anderen, wodurch die Reaktion auf neue Angriffskampagnen drastisch beschleunigt wird.

Persönliches und berufliches Fazit

Über die technischen Errungenschaften hinaus stellt dieses Projekt einen entscheidenden Schritt in meiner beruflichen und persönlichen Entwicklung dar.

In Bezug auf die Fähigkeiten Ich verfüge über fundierte operative Kenntnisse in Technologien und Methoden, die direkt im Berufsalltag anwendbar sind: fortgeschrittene Windows-Systemadministration (WMI, Firewalls, Netzwerkmanagement), SIEM-Integration mit Splunk Enterprise, Python-Entwicklung mit Threading und GUI, mehrschichtige Malware-Erkennung sowie agile Methodik mit durchgängigem Projektmanagement. Diese Fähigkeiten bilden eine solide Grundlage für den Einstieg in eine Karriere als Netzwerkadministrator mit Schwerpunkt Cybersicherheit.

Was das Vertrauen betrifft: Der Erfolg dieses anspruchsvollen Projekts beweist meine Fähigkeit, komplexe Entwicklungsprojekte von Anfang bis Ende zu managen – vom Architekturentwurf über die sorgfältige Implementierung bis hin zu Validierungstests. Dieses technische Selbstvertrauen ist eine große Stärke, um zukünftige berufliche Herausforderungen gelassen und ohne Selbstzweifel anzugehen.

In Bezug auf die Vision dieses Projekt bestärkte ich meinen Fokus auf defensive Cybersicherheit, insbesondere auf die Bedrohungserkennung. Die intellektuelle Befriedigung, die ich durch die Entwicklung der mehrschichtigen Engine und die Beobachtung des Systems im realen Einsatz erfuhr, bestätigte meine Neigung, mich den technischen Aspekten der Sicherheit zuzuwenden, anstatt mich mit Governance oder Auditing zu beschäftigen.

Aus ethischer Sicht ist dieses Projekt mein Verständnis der Verantwortung eines Cybersicherheitsexperten vertieft. Die architektonischen Entscheidungen hinsichtlich des Gleichgewichts zwischen Sicherheit und Benutzerfreundlichkeit, des Umgangs mit Fehlalarmen und des Schutzes der vom System erfassten Daten haben mir die ständigen ethischen Dilemmata in diesem Bereich bewusster gemacht. Cybersicherheit darf niemals

vernachlässigt werden; sie muss von Beginn an in die Planungsphase integriert werden (Security by Design).

Zusammenfassend lässt sich sagen, dass PortGuardian Enterprise nicht einfach nur eine funktionale Softwareanwendung ist, die akademischen Anforderungen genügt. Es ist vielmehr ein konkreter Beweis meiner Fähigkeit, innovative technische Lösungen zu entwickeln und zu implementieren, die mit begrenzten Ressourcen reale Geschäftsanforderungen erfüllen. Darüber hinaus ist es ein greifbarer Beleg für meine Leidenschaft für Cybersicherheit und mein Engagement für die Sicherheit von Informationssystemen.

Dieses Projekt bildet das Fundament für meine berufliche Laufbahn im spannenden und sich ständig weiterentwickelnden Bereich der defensiven Cybersicherheit. Bedrohungen verändern sich, Angriffstechniken werden immer ausgefeilter, doch der grundlegende Ansatz bleibt gültig: mehrschichtige Verteidigung, maximale Automatisierung und kontinuierliche Verbesserung auf Basis von Vorfallanalysen.

Ich bin stolz auf die geleistete Arbeit und danke meinem Ausbilder, Herrn Abdelmajid Lamkadam, für seine intensive Betreuung. Dieser Bericht markiert das Ende meiner Ausbildung, aber den Beginn einer beruflichen Laufbahn, in der mich täglich neue technische Herausforderungen erwarten und ich neue Fähigkeiten erwerben werde.

Cybersicherheit ist ein Wettlauf ohne Ziel. Dieses Projekt ist nur ein erster Schritt.

BIBLIOGRAPHIE

Sicherheitsberichte und Studien

Verizon. Bericht über die Untersuchung von Datenschutzverletzungen (DBIR) 2023. Verizon Enterprise, 2023.
Verfügbar unter: https://www.verizon.com/business/resources/reports/dbir/

ANSSI – Nationale Agentur für Cybersicherheit Frankreichs. Überblick über Cyberbedrohungen 2023. Französische Republik, 2023.
Verfügbar unter: https://www.ssi.gouv.fr/

Ponemon Institute. Globaler Bericht zu den Kosten von Insiderbedrohungen 2023. Ponemon Institute LLC, 2023.

Gartner. Marktübersicht für Endpoint Detection and Response-Lösungen. Gartner Research, 2023.

Cybersecurity Insiders. Insider-Bedrohungsbericht 2023. Cybersecurity Insiders, 2023.

Normen und technische Spezifikationen

IETF RFC 5424. Das Syslog-Protokoll. R. Gerhards, März 2009.
Verfügbar unter: https://tools.ietf.org/html/rfc5424

Microsoft-Dokumentation. Windows-Verwaltungsinstrumentation (WMI). Microsoft Learn, 2024.
Verfügbar unter: https://learn.microsoft.com/en-us/windows/win32/wmisdk/

Microsoft-Dokumentation. Windows Defender Firewall mit erweiterter Sicherheit. Microsoft Learn, 2024.

Softwaredokumentation und Bibliotheken

Python Software Foundation. Python 3.9 Dokumentation. 2024.
Verfügbar unter: https://docs.python.org/3.9/

Riverbank Computing. PyQt6-Dokumentation. 2024.
Verfügbar unter: https://www.riverbankcomputing.com/static/Docs/PyQt6/

Tim Golden. WMI Python-Bibliotheksdokumentation. 2024.
Verfügbar unter: https://pypi.org/project/WMI/

Splunk Inc. Splunk Enterprise Dokumentation v9.x. 2024.
Verfügbar unter: https://docs.splunk.com/

Wissenschaftliche Publikationen

Tischer, M., et al. „Benutzer schließen tatsächlich gefundene USB-Laufwerke an.“ IEEE Symposium on Security and Privacy, University of Illinois, 2016.

**Karystinos, G., Pappas, A. ** „BadUSB: Über Zubehör, das sich als böse entpuppt.“ Black Hat USA, 2014.

Ressourcen zur Bedrohungsanalyse

AlienVault OTX. Open Threat Exchange – Community Threat Intelligence. AT&T Cybersecurity, 2024.
Verfügbar unter: https://otx.alienvault.com/

VirusTotal. VirusTotal – Analysieren Sie verdächtige Dateien und URLs. Google LLC, 2024.
Verfügbar unter: https://www.virustotal.com/

MISP-Projekt. Plattform zum Austausch von Malware-Informationen. CIRCL Luxemburg, 2024.
Verfügbar unter: https://www.misp-project.org/

Methodik

Schwaber, K., Sutherland, J. *The Scrum Guide - The Definitive Guide to Scrum. * Scrum. org, 2020.

Bangor, A., Kortum, P., Miller, J. „Bestimmung der Bedeutung individueller SUS-Werte: Hinzufügen einer Adjektiv-Bewertungsskala.“ Journal of Usability Studies, Band 4, Ausgabe 3, Mai 2009, S. 114-123.