Implementieren eines IDS/IPS in einem beispielhaften Firmennetzwerk

Kai Kauffmann

Abgabedatum: 11.03.2025 Prüfungsjahr: 2025

Inhaltsverzeichnis

[Inhaltsverzeichnis 2](#_Toc1324167662)

[1. Projektbeschreibung 2](#_Toc1827851766)

[2. Konzeption & Planung 3](#_Toc1075668116)

[2.2 Zeitplan / Aufwand 4](#_Toc460328943)

[3. Realisierungsphase 5](#_Toc926594870)

[4. Testphase 12](#_Toc1061625083)

[6. Fazit 14](#_Toc340492985)

[Anhang (Anlagen) 17](#_Toc245300786)

[Anlage 3 - Übersicht Schnittstellen, Clients, Netzwerke 18](#_Toc1766236572)

[Anlage 5 – pfSense “Alerts” wegen asymmetrischem Routing 20](#_Toc1859661047)

[Regelkategorien mit aktiven SIDmgmt. 26](#_Toc616274444)

[SIDmgmt mit aktiven “enable-” und “drop List” 29](#_Toc33209694)

[Anlage 8 – Testergebnisse auf das LAN-Netz IPS 29](#_Toc1159354981)

[29](#_Toc817903309)

[Anlage 8 – Testergebnisse auf das OPT1-Netz IDS 31](#_Toc827406453)

[Anlage 9 – Beispiel Snort icmp Rules 32](#_Toc59970874)

[(und wieviel der Regeln standardmäßig deaktiviert sind) 32](#_Toc402425077)

[Abkürzungsverzeichnis 33](#_Toc605112988)

[Quellenverzeichnis 34](#_Toc1347649272)

1. Projektbeschreibung

1.1 Ziel des Projekts

Das Ziel meines Projektes ist der Aufbau einer exemplarischen Testumgebung (beispielhaftes Firmennetz) und die Implementierung eines IDS/IPS, das das Netzwerk überwacht, protokolliert, Angriffe erkennt und darauf reagiert. Dabei sollen typische Angriffsversuche wie Portscans oder DoS-Angriffe zuverlässig erkannt werden, um eine aktive Reaktion des Netzwerks zu ermöglichen. Durch die Integration in eine virtuelle Umgebung kann die Lösung flexibel getestet und auf andere Unternehmensnetze übertragen werden.

1.2 Projektumfeld & Rahmenbedingungen

Für die Durchführung des Projekts stehen zwei Laptops zur Verfügung, wobei einer von Cloudcommand bereitgestellt wurde und ein weiterer privater Laptop genutzt wird. Beide Laptops verfügen nicht über Netzwerk-Interfaces mit Netmap-Unterstützung, was bei der Performance und Funktionsweise des IDS/IPS berücksichtigt werden muss. Um genügend Arbeitsspeicher für die Virtualisierung mit Hyper-V zu haben, werden beide Geräte gleichzeitig eingesetzt. Die pfSense-Firewall wird an das private Heimnetz angebunden und bildet die Grundlage für eine virtuelle Netzwerkstruktur.

Diese Struktur umfasst:

Zwei LAN-Netze für jeweils einen Windows-10-Clienten

Ein weiteres LAN-Netz für Kali Linux als Test- und Angriffstool

Alle Systeme laufen auf virtuellen Maschinen unter Hyper-V. Dadurch kann die Lösung flexibel in unterschiedlichen Umgebungen eingesetzt und um zusätzliche Hosts oder Netze erweitert werden.

1.3 Eingesetzte Werkzeuge

**Hyper-V:** Virtualisierungsplattform zur Erstellung und Verwaltung der virtuellen Maschinen.

**Windows Server:** Betriebssystem für verschiedene Serverdienste innerhalb der Testumgebung, z. B. als Active Directory Domain Services (ADDS) oder Fileserver.

**pfSense:** Open-Source-Firewall zur Netzwerkverwaltung und als Basis für die IDS/IPS-Integration.

**Suricata:** Open-Source-IDS/IPS zur Erkennung und Prävention von Angriffen im Netzwerk.

**Kali Linux:** Spezialisierte Linux-Distribution mit Sicherheitstools zur Durchführung von Angriffssimulationen.

**WinSCP:** Dateiübertragungsprogramm für den sicheren Transfer von Konfigurationsdateien zwischen Hosts.

**PuTTY:** SSH-Client zur Verwaltung und Konfiguration von Netzwerkkomponenten und Servern.

**Nmap/Zenmap:** Nmap und GUI für Nmap zur Durchführung und Visualisierung von Netzwerkscans.

2. Konzeption & Planung

2.1 Grober Projektablauf

Zu Beginn des Projekts wird die Virtualisierungsumgebung eingerichtet. Dazu gehörten die Installation und Konfiguration von Hyper-V auf den bereitgestellten Geräten. Die Planung sieht vor, dass die pfSense-Firewall, die Windows-10-Clients sowie der Windows-Server auf dem privaten Laptop bereitgestellt werden. Der Cloudcommand-Laptop hingegen wird für die Bereitstellung der Kali-Linux-VM genutzt.

(Der Cloudcommand-Laptop dient als Zugangspunkt für die WebGUI und Konsole von pfSense, dazu muss die grundlegende Einrichtunge von pfsense über die Konsole abgeschlossen sein und der Zugang zur WebGUI, über den Windows-10-Clienten vom LAN-Netz aus auch für den Cloudcommand-Laptop eingerichtet werden. Mit PuTTY kann dann vom Cloudcommand-Laptop die pfsense Konsole erreicht werden.)

Nachdem die VMs bereitgestellt sind, erfolgt die Konfiguration des Netzwerks und der Firewall. Es werden virtuelle Netzwerke für die Windows-Clients und das Kali-Linux-System angelegt. Die grundlegenden Netzwerkeinstellungen für pfSense werden vorgenommen, Firewall-Regeln sowie NAT-Regeln konfiguriert und die gesamte Netzwerkstruktur so abgesichert.

Nach dieser grundlegenden Einrichtung wird die Netzwerkkonnektivität getestet. Dabei wird überprüft, ob die Windows-10-Clients wie geplant über das WAN-Netzwerk Internetzugang erhalten, aber voneinander isoliert sind. Gleichzeitig wird getestet, ob das Kali-Netzwerk Zugriff auf beide Windows-Netzwerke hat, jedoch vom WAN-Netzwerk und dem Internet isoliert bleibt. Diese Konnektivitätsprüfung stellt sicher, dass die Netzwerksegmentierung korrekt umgesetzt wurde, bevor weitere Sicherheitsmaßnahmen implementiert werden.

Als nächstes wird Suricata als IDS/IPS in die pfSense-Firewall integriert. Hierbei wird Suricata installiert und anschließend auf den einzelnen Netzwerkschnittstellen (NICs) konfiguriert. Dies stellt den entscheidenden Schritt bei der Implementierung von IDS und IPS dar, da Suricata an den jeweiligen Schnittstellen aktiviert und mit den passenden Konfigurationen und Rulesets versehen wird. Dabei wird sichergestellt, dass der eingehende und ausgehende Datenverkehr entsprechend analysiert und Angriffsversuche erkannt werden.

Im Anschluss beginnt die Testphase, in der die Funktionsweise des IDS/IPS überprüft wird. Dazu werden NMAP-Scans durchgeführt, Angriffssimulationen mit Kali Linux getestet und die Firewall- sowie Suricata-Logs analysiert, um die Erkennung zu evaluieren.

Am Ende des Projekts werden der gesamte Ablauf der Umsetzung sowie die erzielten Ergebnisse dokumentiert. Die finale Dokumentation fasst die gesamten durchgeführten Maßnahmen, die gewonnenen Testergebnisse zusammen und bietet ein ein umfangreiches Fazit zum implemtieren und Betreiben eines IDS/ IPS.

2.2 Zeitplan / Aufwand

10 Std. Aufbau der Virtualisierungsumgebung

3 Std. Abweichung der Planung: Lösung der Routing-probleme (siehe 3.2.2)

11 Std. Implementierung IDS/IPS

8 Std. Testen IDS/IPS

8 Std. Dokumentation

2 Std. Finale Überprüfung & Korrekturen Dokumentation

Gesamt: 40 Stunden

3. Realisierungsphase

3.1 Einrichtung Hyper-V und VMs

Auf dem Cloudcommand-Laptop ist Hyper-V bereits vorinstalliert und kann direkt genutzt werden. Auf dem privaten Laptop hingegen läuft Windows 11 Home, weshalb Hyper-V zunächst über PowerShell aktiviert werden muss. Dazu wird PowerShell als Administrator geöffnet und der Code vom Screenshot aus Anlage1 eingegeben und ausgeführt. Jetzt erstellt PowerShell eine Batchdatei und führt diese aus, wodurch die features für Hyper-V geladen, installiert und aktiviert werden. Nach der Aktivierung wird Hyper-V auf beiden Geräten so konfiguriert, dass eine stabile Virtualisierungsumgebung entsteht.

Für das Projekt wurde die Version Kali Linux 2024.4 vorkonfiguriert und von der offiziellen Kali-Website (kali.org) für Hyper-V heruntergeladen. Für die übrigen virtuellen Maschinen standen bereits ISO-Dateien aus der bisherigen Weiterbildung zur Verfügung, welche zur Erstellung der Windows- und pfSense-VMs genutzt wurden. Sobald die Virtualisierung eingerichtet ist, werden die benötigten virtuellen Maschinen erstellt. Die pfSense-VM sowie die beiden Windows-10-VMs für die Client-Systeme werden auf dem privaten Laptop bereitgestellt. Die Kali-Linux-VM für die Durchführung von Tests und Angriffssimulationen wird auf dem Cloudcommand-Laptop installiert.

Nach der Erstellung der VMs werden die erforderlichen virtuellen Switches in Hyper-V eingerichtet. Der WAN-Switch wird als externer Switch konfiguriert, um die Kommunikation mit dem Heimnetzwerk zu ermöglichen. Die Kali-VM, die auf dem Cloudcommand-Laptop ausgeführt wird, benötigt ebenfalls einen externen Switch, der jedoch als LAN-Schnittstelle fungiert, da Kali nur über diese Verbindung die pfSense-Schnittstelle erreichen kann. Die Windows-10-Clients und der Windows-Server werden mit dedizierten privaten Switches ausgestattet, um eine vollständige Isolation dieser Netzwerke sicherzustellen. Die pfSense-Firewall wird so konfiguriert, dass über Hyper-V vier virtuelle Switches hinzugefügt werden. Zwei dieser Switches sind externe Switches, wobei einer für die WAN-Schnittstelle und der andere für die Verbindung mit der Kali-VM vorgesehen ist. Dadurch kann Kali, das auf dem Cloudcommand-Laptop läuft, mit der pfSense-Schnittstelle kommunizieren. Zusätzlich werden zwei private Switches konfiguriert: Einer dient dem LAN-Netzwerk mit dem IPS, der andere stellt die Verbindung zum separaten LAN-Netzwerk mit dem IDS sicher.

Diese strukturierte Vorkonfiguration gewährleistet eine gezielte Trennung der Netzwerke, sodass jedes System gemäß seiner vorgesehenen Rolle isoliert oder verbunden werden kann.

Nach der Erstellung und Zuordnung der Switches erfolgt die grundlegende Konfiguration der Systeme. Die Windows-Clients werden lediglich grundlegend eingerichtet, um eine Basisfunktionalität sicherzustellen. Zusätzlich wird das Tool Zenmap heruntergeladen und installiert, um die Netzwerkkonnektivität zu überprüfen. Die Kali-Linux-VM wurde in der Version 2024.4 heruntergeladen und bereits mit den relevanten Sicherheitstools vorkonfiguriert.

Ein besonderer Fokus liegt auf der effizienten Ressourcennutzung, da die Virtualisierungsumgebung auf zwei Laptops verteilt wird, um ausreichend Arbeitsspeicher und Rechenleistung zur Verfügung zu haben.

3.2 Netzwerk- und Firewallkonfiguration (pfSense)

3.2.1 Netzwerk- und Firewallkonfiguration (Konsole)

Die Netzwerkkonfiguration in pfSense umfasst die Einrichtung der Netzwerkschnittstellen sowie die statische IP-Adressvergabe für die verbundenen Systeme. Nach dem ersten Start der pfSense-VM erscheint das Konfigurationsmenü in der Konsole, dass die Zuweisung der Netzwerkschnittstellen veranlasst. Später auch unter Menüpunkt 1 zu finden, werden hier die Schnittstellen den Netz-Bezeichnungen zugewiesen (bspw. hn1 zu LAN). Eine korrekte Zuweisung der Schnittstellen ist wichtig, da jede Schnittstelle einem bestimmten Switch in Hyper-V zugewiesen ist. Eine fehlerhafte Zuweisung und eine Kombination unterschiedlicher Switch-Typen kann zu Verbindungsproblemen führen.

Dann folgt Menüpunkt 2, die IP-Konfiguration der einzelnen Schnittstellen. Zunächst wird die WAN-Schnittstelle (hn0) konfiguriert. Im Menü wird die Option zur Verwendung von DHCP ausgewählt, wodurch die Schnittstelle automatisch eine IP-Adresse aus dem Heimnetz 192.168.178.73/24, zugewiesen bekommt. Dabei wird im Konfigurationsdialog explizit nach der gewünschten Schnittstelle und dem Verbindungsmodus gefragt – hier ist DHCP mit JA zu bestätigen, alle Anfragen zu IPv6 werden jetzt und beim Einrichten der LAN-Schnittstellen verneint bzw. durch das Bestätigen ohne Eingabe deaktiviert.

Anschließend erfolgt die manuelle Konfiguration der internen LAN-Netzwerkeschnittstellen, die LAN (hn1) Schnittstelle erhält die statische IP-Adresse 192.168.50.1/24. Im Konfigurationsdialog wird dazu die Anfrage zu DHCP mit NEIN beantwortet, dann die IP-Adresse eingeben und die Subnetzmaske (255.255.255.0) mit Eingabe der 24 bestätigt. Wenn die IP-Konfiguration für die LAN-Schnittstelle (hn1) abgeschlossen ist, erscheint der Hinweis das die WebGUI von pfSense über die LAN-Adresse als URL im Browser vom LAN-Netz aus erreichbar ist. Danach wird auch das zweite interne LAN (hn2) und das dritte interne LAN (hn3) mit den gleichen Schritten eingerichtet. Hierbei wird die pfSense-Schnittstelle hn2 auf die statische IP-Adresse 192.168.60.1/24 konfiguriert und dementsprechend erhält die Schnittstelle (hn3) die statische IP-Adresse 192.168.80.1/24. Zu beachten ist das bei der IP-Konfiguration der internen Schnittstellen kein Upstream Gateway angegeben wird, da pfSense sonst diese Schnittstellen als WAN-Schnittstellen einstuft.

Nach Abschluss der Schnittstellenkonfiguration in der pfSense-Konsole wird die Implementierung auf den Client-Systemen vorgenommen. Für den Windows-10-Client im LAN werden die Netzwerkadaptereinstellungen unter Eigenschaften auf "statische IP" gestellt und folgende Werte eingetragen:

IP-Adresse: 192.168.50.50

Subnetzmaske: 255.255.255.0

Standardgateway: 192.168.50.1

Analog dazu wird im zweiten internen LAN (OPT1) für den weiteren Windows-10-Client die Einstellung vorgenommen aber mit folgendem Werten:

IP-Adresse: 192.168.60.60

Subnetzmaske: 255.255.255.0

Standardgateway: 192.168.60.1

Für die Kali-VM, die im isolierten Netzwerk betrieben wird, erfolgt die Konfiguration ebenfalls im statischen Modus. Die Netzwerkeinstellungen wird über die Kommandozeile konfiguriert. Dazu muss wie bei Anlage2 mit Root-Rechten und bspw. dem Nanoprogramm die Interfaces-Datei (/etc/Network/interfaces) bearbeitet werden:

IP-Adresse: 192.168.80.80

Subnetzmaske: 255.255.255.0

Standardgateway: 192.168.80.1

Übersicht zu den IP-Adressen, Schnittstellen, Netzen, Clients in Anlage 3

Diese Netzwerkkonfiguration sorgt für eine gezielte Trennung der einzelnen Netzwerksegmente.

3.2.2 pfSense WebGUI-Konfiguration und pfSense-Regeln

Nachdem die grundlegende Schnittstellenkonfiguration in der pfSense-Konsole abgeschlossen ist, erfolgt die weitere Einrichtung über die WebGUI. Um auf diese zugreifen zu können, wird im Webbrowser des Windows-10-Clients im LAN-Netzwerk die URL „https://192.168.50.1“ eingegeben. Beim ersten Zugriff erfolgt die Anmeldung mit den Standardzugangsdaten (Benutzername: „admin“ und Passwort: „pfsense“). Aus Sicherheitsgründen wird unmittelbar nach der Anmeldung das Standardpasswort geändert.

Im ersten Schritt der WebGUI-Konfiguration werden, die zuvor in der Konsole eingerichteten Netzwerkschnittstellen überprüft. Unter dem Menüpunkt Interfaces → Assignments wird sichergestellt, dass jeder physischen Netzwerkkarte (hn0, hn1, hn2, hn3) die korrekte Bezeichnung (WAN, LAN, OPT1, OPT2) zugewiesen wurde. Falls nötig, können hier noch Anpassungen vorgenommen werden.

Anschließend erfolgt die detaillierte Konfiguration jeder Schnittstelle über den Menüpunkt Schnittstelle:

WAN-Schnittstelle (hn0):

Die WAN-Schnittstelle bleibt auf DHCP eingestellt, um eine automatische IP-Vergabe aus dem Heimnetz zu gewährleisten. Es wird die Option „Block private networks“ deaktiviert, um Konflikte mit dem privaten Heimnetz zu vermeiden.

LAN-Schnittstelle (hn1):

Diese Schnittstelle wird mit einer statischen IPv4-Adresse konfiguriert (192.168.50.1/24). Der DHCP-Server wird hier deaktiviert, da die Windows-Clients statisch konfiguriert werden.

OPT1-Schnittstelle (hn2):

Die OPT1-Schnittstelle erhält ebenfalls eine statische IPv4-Adresse (192.168.60.1/24). Auch hier erfolgt die manuelle Konfiguration der angeschlossenen Windows-Clients.

OPT2-Schnittstelle (hn3):

Das isolierte Netz mit der Kali-VM erhält die statische IPv4-Adresse (192.168.80.1/24). DHCP wird hier nicht verwendet, da auch Kali statisch konfiguriert ist.

Nach Abschluss der Schnittstellenkonfiguration wird der Zugang zur WebGUI über die WAN-Schnittstelle vom Cloudcommand-Laptop ermöglicht. Dazu wird unter „Firewall“ → „Rules“ auf dem WAN-Interface eine Regel erstellt, die es dem Laptop (192.168.178.36) vom Heimnetz aus erlaubt, via HTTPS (Port 443) auf die WAN-IP der pfSense zuzugreifen.

Die Firewall-Regeln selbst werden sorgfältig und mit Fokus auf die Sicherheitsanforderungen erstellt und angeordnet. pfSense arbeitet als „Stateful Firewall“ nach dem „Default-Deny-Prinzip“, das bedeutet alles, was nicht explizit erlaubt ist, wird automatisch blockiert. Firewall-Regeln in pfSense werden nach dem Prinzip der Top-Down-Verarbeitung abgearbeitet. Jede Schnittstelle hat eine eigene Regelstruktur, die von oben nach unten gelesen wird. Sobald eine Regel auf einen bestimmten Datenverkehr zutrifft, wird sie angewendet und alle darunterliegenden Regeln werden ignoriert.

Aus diesem Grund werden explizite Allow-Regeln immer oben platziert, um gewünschten Traffic zuzulassen, gefolgt von restriktiveren Block-Regeln, die jeglichen nicht explizit erlaubten Verkehr verhindern.

Dementsprechend werden auf den internen Schnittstellen folgende Regeln konfiguriert:

LAN-Netzwerk (192.168.50.0/24):

Zunächst wird eine Allow-Regel definiert, die ausgehenden Traffic ins WAN erlaubt, damit die Windows-Clients Internetzugang erhalten. Eine weitere Allow-Regel gestattet den Datenverkehr von LAN zu OPT2, sodass Kommunikation mit der Kali-VM möglich ist. Anschließend folgt eine Block-Regel, die jeglichen Datenverkehr von LAN zu OPT1 unterbindet, um die Isolation zwischen diesen beiden Netzwerken sicherzustellen.

OPT1-Netzwerk (192.168.60.0/24):

Analog zum LAN-Netzwerk wird eine Allow-Regel erstellt, die den Internetzugang für die Windows-Clients im OPT1-Netz erlaubt. Zusätzlich wird eine Allow-Regel definiert, die den Datenverkehr von OPT1 zu OPT2 ermöglicht. Danach folgt eine Block-Regel, die die Kommunikation zwischen OPT1 und LAN verhindert, um sicherzustellen, dass beide Netzwerke voneinander getrennt bleiben.

OPT2-Netzwerk (192.168.80.0/24):

Da dieses Netz für Sicherheits- und Angriffssimulationen vorgesehen ist, werden gezielte Allow-Regeln gesetzt, die den Zugriff von OPT2 auf die Netzwerke LAN und OPT1 erlauben. Jeglicher Datenverkehr in Richtung WAN und Internet wird jedoch durch eine Block-Regel vollständig unterbunden, sodass die Kali-VM ausschließlich mit den internen Netzwerken interagieren kann. Optional kann zusätzlich ein Logging für geblockte Verbindungen aktiviert werden, um unerlaubte Verbindungsversuche nachzuvollziehen.

Zur Unterstützung dieser Regelkonfiguration wird zusätzlich die automatische Outbound-NAT-Regelgenerierung überprüft („Firewall“ → „NAT“ → „Outbound“ im „Automatisch“-Modus). Dabei wird überprüft, ob der Modus auf „Automatisch“ gesetzt ist und keine NAT-Regel auf der Schnittstelle OPT2 existiert. Dadurch wird sichergestellt, dass der Datenverkehr aus den Netzwerken LAN und OPT1 zum WAN übersetzt wird, während der gesamte Datenverkehr aus OPT2 durch die Block-Regel definitiv am Internet- und WAN-Zugang gehindert wird. Abschließend werden DNS- und NTP-Dienste („System“ → „General Setup“) konfiguriert, um sicherzustellen, dass alle Systeme über korrekte Zeit- und Namensauflösung verfügen.

Nach einer gründlichen Prüfung aller Einstellungen wurden abschließende Konnektivitätstests mittels Ping- und Traceroute-Befehlen durchgeführt, um sicherzustellen, dass alle Regeln und Schnittstellen korrekt arbeiten (dazu wurde die Windows-Firewall der Windows-Clients deaktiviert).

Dabei wurde jedoch ein asymmetrisches Routing festgestellt. Dies äußerte sich insbesondere in inkonsistenten Traceroute-Ergebnissen zwischen den Netzsegmenten, ungleichmäßiger Antwortzeit bei Ping-Anfragen und unerwartetem Routing über alternative Pfade und damit verbundene Alerts der Firewall (eingehender und ausgehender Traffic nahm unterschiedliche Pfade, somit konnte in der pfSense, der "State" für diesen Traffic nicht in der Tabelle abgeglichen werden und löste einen Alarm aus siehe Anlage 5).

Nachdem die Problemstellen erkannt wurden, sind alle Firewall-Regeln, mehrfach auch in anderer Kombination neu angelegt, die NAT-Regeln sowie alle Konfigurationen der pfSense erneut geprüft worden, um das symmetrische Routing sicherzustellen. Trotzdem blieb das Problem bestehen.

Um das Problem weiter einzugrenzen, wurde die gesamte Netzwerkkonfiguration aller einzelnen Systeme erneut überprüft und systematisch rekonstruiert. Nach jedem Konfigurationsschritt wurden die Tests wiederholt, jedoch blieb das Problem des asymmetrischen Routings weiterhin bestehen.

Das Routing-Problem trat zwischen den LAN-/ OPT1-Netz und dem OPT2 Netz auf. Beim Traceroute-Scan mit Zenmap ergab sich eine inkonsistente Anzahl von Hops zwischen den Netzsegmenten.

Vom LAN- und OPT1-Netz zum OPT2-Netz wurden vom Windows-Client aus zwei Hops registriert, während von der Kali-VM im OPT2-Netz nur ein Hop festgestellt wurde. Zudem trat vereinzelt das Problem auf, dass das Ziel nicht erreicht wurde und die Fehlermeldung "Host down" erschien. Dies deutet darauf hin, dass die voreingestellten Versuche im Traceroute-Befehl möglicherweise nicht ausreichten, um den zielführenden Netzwerkpfad zu ermitteln.

Im nächsten Schritt wurde das Routing zwischen LAN-Netz und OPT1-Netz getestet, mit dem Ergebnis das hier symmetrisches Routing besteht und die Anzahl der Hops korrekt ist. Deshalb folgte der Entschluss den Windows-Client mit der IP 192.168.50.50 aus dem LAN-Netz mit der Kali-VM mit der IP 192.168.80.80 aus dem OPT2-Netz samt der IP-Nummern zu tauschen, um zu sehen, ob Kali von diesem Netz beim Traceroute-Scan immer noch nur einen Hops registriert.

Auch hier blieb das Problem bestehen, dies zeigt jedoch das es nicht an der Konfiguration von pfSense liegt und nicht an der Netzwerk-Konfiguration, weil sonst der Windows-Client bei den Scans aus dem OPT2-Netz auch nur ein Hops und Kali aus dem LAN-Netz zwei Hops hätte anzeigen müssen und grundsätzlich die Alerts auf der pfSense hätten verschwinden müssen.

Als nächstes wurde die Kali VM neu aufgesetzt und konfiguriert, doch das Problem blieb bestehen. Aus diesem Grund wurde das bestehende Setup, bei dem die Kali-VM ausschließlich auf dem Cloudcommand-Laptop über Hyper-V betrieben wurde, überarbeitet. Nun laufen alle virtuellen Maschinen zentral auf dem privaten Laptop, während der Cloudcommand-Laptop weiterhin für den Zugriff auf die WebGUI und die pfSense-Konsole genutzt wird. Dies stellt sicher, dass später bei der Implementierung von Suricata genügend Arbeitsspeicher für das Laden und Verarbeiten der umfangreichen Regelsets (die je nach Konfiguration über 30.000 Einträge umfassen können) bei dem System, auf dem der Webbrowser läuft, zur Verfügung steht.

Jetzt bildet die gesamte Konfiguration eine stabile Basis für die nahtlose Integration und den reibungslosen

3.3 Implementierung und Konfiguration von Suricata (IDS/IPS)

Die Implementierung erfolgt auf der pfSense-Firewall, wobei Suricata auf den relevanten Netzwerkschnittstellen konfiguriert wird, um eine umfassende Überwachung und Schutz der Infrastruktur zu gewährleisten. Zunächst wird die Software über das pfSense-Paketmanagement installiert. Danach erfolgt die Konfiguration der Netzwerkschnittstellen, um den Datenverkehr gezielt zu überwachen. Anschließend werden die Rulesets eingerichtet, die festlegen, welche Arten von Netzwerkverkehr als Bedrohung erkannt werden. Diese Rulesets können über die Funktionen des SIDmgmt bei bedarf noch individuell bearbeitet und angepasst werden.

Die einzelnen Steps werden nachfolgend nochmal detaillierter dargestellt.

Suricata bietet eine Vielzahl an Konfigurationsoptionen für die effektive Erkennung und Verhinderung von Bedrohungen, die für die einzelnen Schnittstellen festgelegt werden können. Unter den Einstellungen der Schnittstelle werden als erstes die Logging-Settings festgelegt, wie bspw. das Suricata Alarme von der Schnittstelle an System-Log der Firewall sendet, ob HTTP-Logs erstellt werden sowie TLS-Logs. Als nächstes folgen EVE Output Settings, hier besteht die Möglichkeite eine spezielle Datei, die EVE JSON Datei zu erstellen und individuell das logging der Datei festzulegen (Anlage 7). Diese Datei kann auch für Externe SIEM-Systeme genutzt werden. Die nächste Kategorie ist "Alert an Block Settings". Unter diesen Einstellungen wird der Modus eingestellt, mit dem Suricata auf der Schnittstelle arbeitet. In unserem Fall auf der LAN-Schnittstelle im IPS-Legacy-Modus. Der IPS Inline-Modus wird vom System nicht unterstützt, da dies Netmap erfordert. Der Unterschied von beiden ist folgender:

Im IPS-Legacy-Modus wird der Datenverkehr vom Prinzip kopiert und von Suricata analysiert, sodass es die Pakete nur passiv inspiziert und bei verdächtigen Aktivitäten Warnungen ausgibt. Im Gegensatz dazu ist der IPS-Inline-Modus direkt im Netzwerk integriert: Suricata analysiert und modifiziert den Datenverkehr in Echtzeit, wodurch schädliche Pakete unmittelbar blockiert werden können. Deshalb können im Legacy-Modus geringe Teile des Traffic “durchkommen”, bevor dieser blockiert wird.

Bei den Einstellung Alert und Block Settings ist es wichtig den haken bei Block On DROP zu setzen, weil später die Funktion der Rules Policy genutzt werden sollen. Außerdem kann unter IP Pass List bspw. IP-Adressen oder ganze Netze geladen werden, die vorab unter dem Reiter Pass List angelegt wurden, diese werden von Suricata nicht geblockt.

Dann ist noch der Haken beim Promiscuous Mode zu setzen, so kann Suricata den gesamten Traffic der Schnittstelle "einsehen".

Für unseren Case müssen wir noch die Einstellungen unter "Networks Suricata Should Inspect and Protect" ändern. Im normalen Fall können diese auf Standard verbleiben, da wir jedoch von Kali aus, die anderen Netze, zum Beispiel Scannen wollen und Suricata reagieren soll, müssen wir hier Änderungen vornehmen, denn Suricata stuft alle Internen Netze (LAN, OPT1, OPT2) als legitim ein und würde Kali so nicht blocken da es den Traffic nicht analysiert (Anlage7 Einstellungen der LAN-Schnittstelle).

Deshalb erstellen wir eine Pass-List mit unseren LAN-Netzen (LAN, OPT1) die unter Home-Net geladen wird und mit unseren WAN\_Netzen (WAN, OPT2) die unter External Net geladen wir. Damit gehört OPT2 für Suricata zu den WAN netzen.

Als nächstes kommen die Einstellungen beim Reiter "Global Settings".

Ein wesentlicher Bestandteil der Konfiguration ist die Auswahl der Rulesets, die bestimmen, welche Bedrohungssignaturen aktiv sind. Dazu gehören unter anderem die Emerging Threats (ET Open) und die Snort GPLv2 Community rules, beide Community-basiertes Regelwerke, sowie das Snort VRT Ruleset (Subscriber Rules), welche die IPS Policy Funktionen bereitstellt.

Die ersten beiden Rule-Sets könne durch Anhaken ausgewählt und dann später geladen werden, bei den Snort VRT Ruleset muss man sich zuerst bei Snort anmelden, so erhält man den "Snort Oinkmaster Code" welcher unter Global Settings einzutragen ist und es muss der name des Rulearchive eingetragen werden, dabei ist zu beachten das die Version 3.0 Regeln nicht mit Suricata kompatible sind, deshalb wurde "snortrules-snapshot-29200.tar.gz" eingetragen (Anlage 10 Regelauswahl).

Außerdem setzen wir noch den Haken bei "ABUSE.ch SSL Blacklist -Rules" und "Feodo Tracker Botnet C2 IP-Rules" welche registrierte C&C Botnet IP-Adressen blockt sowie eine SSL Blacklist zum Blocken hinzufügt. Als nächstes kann man den Updateintervall festlegen, wie lange geblockte Host auf der Blocking-Liste bleiben sollen und ob man über Updates Benachrichtigungen bekommen will.

Danach werden die ausgewählten Rule-Sets über den Reiter Aktualisierungen heruntergeladen, wobei der download der SnortVRT-Rules das erste mal erzwungen werden muss. Nach abgeschlossen Download sollte das ganze wie unter Anlage 6 aussehen.

4. Testphase

4.1 Einstellungen zu den Tests

Um in der Testphase sowohl das LAN- als auch das OPT1-Netz verlässlich und vergleichbar auf Angriffe – insbesondere auf Port- und Netzwerkscans – zu überwachen, kommt Suricata in zwei Betriebsmodi zum Einsatz: zum einen als IDS (Intrusion Detection System), zum anderen als IPS (Intrusion Prevention System). Die zugrunde liegenden Regeln und Signaturen bleiben dabei identisch, sodass die Auswertung der Ereignisse konsistent ist. Der wesentliche Unterschied liegt darin, dass im IDS-Modus Angriffe lediglich erkannt und protokolliert (Alert) werden, während Suricata im IPS-Modus zusätzlich aktiv eingreift und den verdächtigen Traffic blockieren kann (“Alert/Block”). Im Folgenden ein Überblick über diese Vorgehensweise:

Identische Regelsets für IDS und IPS Damit die Testergebnisse für beide Modi direkt vergleichbar sind, wird auf LAN und OPT1 jeweils das gleiche Regelpaket aktiviert. In Suricata lassen sich die Regeln (z. B. Portscan- oder Scan-Erkennungsregeln) bei beiden Schnittstellen anwenden. Bei Tests gegen das LAN und gegen das OPT1-Netz werden somit identische Angriffsversuche gefahren (beispielsweise Nmap-Scans), während Suricata in der reinen IDS-Konfiguration nur Meldungen („Alerts“) generiert. Im IPS-Modus kommen dieselben Regeln zum Tragen, lösen aber über die Alert-Funktion hinaus eine Blockierung (“Drop“) des unerwünschten Datenverkehrs aus. Dadurch lässt sich genau festhalten, wie oft ein bestimmter Scan erkannt wird, und gleichzeitig untersuchen, inwiefern eine aktive Blockierung den Scan beeinflusst oder unterbindet.

Gleiche Einstellung, aber unterschiedliches Verhalten (IDS vs. IPS)

Im IDS-Modus horcht Suricata nur mit: Sobald ein verdächtiges Muster oder eine Angriffssignatur erkannt wird, protokolliert das System den Vorgang. Dies hat den Vorteil, dass das Produktionsnetzwerk (oder in diesem Falle die Testsysteme in LAN und OPT1) nicht beeinflusst wird: Alle Verbindungen bleiben technisch weiterhin bestehen, auch wenn das IDS einen Angriff erkennt. Der Administrator kann dann anhand der Alerts die Angriffsversuche nachvollziehen.

Im IPS-Modus greift Suricata proaktiv ein und blockiert verdächtigen Traffic, noch bevor er sein Ziel erreicht. Damit wird derselbe Angriff, der im IDS-Modus nur einen Alert erzeugt hätte, in der IPS-Konfiguration durch entsprechende Regeln unterbunden. Bei einem aktiven Portscan blockiert die IPS somit den Traffic zum Zielhost, was die Sichtbarkeit der Dienste für den Angreifer reduziert.

Fokussierung auf Scan-Regeln (Port-, Host- und Netzwerk-Scans) In beiden Modi liegt das Hauptaugenmerk hier auf den sogenannten „Scan“-Regeln, welche typische Verhaltensmuster von Port- und Netzwerkscans erkennen (z. B. ungewöhnlich viele SYN-Pakete an verschiedenen Zielports, Host-Discovery-Methoden etc.). Mithilfe dieser Scan-Regeln kann man gut testen, ob Suricata zuverlässig reagiert und in den Protokollen Einträge erstellt – beziehungsweise im IPS-Modus aktiv eingreift.

4.2 Testergebnisse

Gerade beim Testaufbau, in dem man von einer Kali-VM gezielt mit Nmap gegen die Windows-Rechner (LAN und OPT1) arbeitet, liefern die Scan-Regeln sehr schnell aussagekräftige Ergebnisse:

IDS: Man sieht klare Alerts in den Suricata-Logs (Anlage8).

IPS: Die Protokolle zeigen ebenfalls Meldungen, zudem wird das betroffene Paket (oder die Verbindung) blockiert (Anlage8).

6. Fazit

Suricata auf pfSense kann sowohl als rein passives IDS als auch als aktives IPS betrieben werden. Im IDS-Modus überwacht Suricata den Datenverkehr und meldet verdächtige Aktivitäten lediglich als Alarm, ohne in den Verkehr einzugreifen. Das heißt, Angriffe werden erkannt und geloggt, aber nicht automatisch blockiert. Im IPS-Modus hingegen greift Suricata aktiv ein: Verdächtiger Traffic wird noch während der Übertragung erkannt und direkt geblockt.

Auf pfSense erfolgt dies entweder im Legacy-Modus (dem klassischen IDS/IPS-Hybrid) oder im modernen Inline-Modus. Im Legacy-Modus wird jeder Paketstrom zunächst normal durch die Firewall geleitet und Suricata untersucht eine Kopie des Pakets parallel. Erkennt es einen Angriff, wird die Quelladresse nachträglich auf die pfSense-Blockliste gesetzt – dadurch können anfänglich ein oder zwei schädliche Pakete das Ziel noch erreichen, bevor der Angreifer blockiert wird. Im Inline-IPS-Modus dagegen hängt Suricata sich direkt in den Paketfluss ein, sodass Pakete schon beim Eintreffen geprüft und bei einem Regeltreffer verworfen werden, bevor sie ins Netzwerk gelangen. Dadurch „leckt“ kein bösartiges Paket mehr durch, was Inline zum echten IPS macht. Allerdings erfordert Inline den Netmap-Treiber in FreeBSD, der nicht von jeder Netzwerkkarte unterstützt wird. Insgesamt ist also der Hauptunterschied, dass im IDS-Modus keine automatische Sperrung erfolgt, während im IPS-Modus erkannte Bedrohungen unmittelbar blockiert werden, letzteres erhöht die Sicherheit, birgt aber auch neue Herausforderungen in der pfSense-Umgebung (Stichwort Netmap und NIC-Kompatibilität).

Beide Betriebsmodi haben spezifische Vor- und Nachteile. Der IDS-Modus hat den Vorteil, keine Auswirkungen auf legitimen Traffic zu verursachen, Fehlalarme (False Positives) führen nicht zum Abbruch von Verbindungen, da Suricata eben nur beobachtet. Dies sorgt für Stabilität und reduziert das Risiko, etwas “kaputtzukonfigurieren“. Außerdem eignet sich IDS-Modus gut zum Einstieg und Fein-Tuning, denn man kann das System zunächst beobachten, ohne dass versehentliche Blockierungen den Netzwerkbetrieb stören. Der Nachteil ist offensichtlich: Angriffe werden zwar gemeldet, aber nicht unterbunden. Ein Admin muss also zeitnah reagieren (z.B. manuell IPs sperren oder Regeln anpassen), damit die erkannten Bedrohungen nicht trotzdem Schaden anrichten. Im IPS-Modus hingegen liegt der große Vorteil darin, dass bekannte Angriffe sofort gestoppt werden – Suricata blockiert z.B. Exploit-Versuche oder Malware-Traffic in Echtzeit, bevor sie ins interne Netz eindringen können. Dies ermöglicht eine proaktive Durchsetzung der Sicherheitsrichtlinien: PfSense kann so als echte Schutzbarriere fungieren, anstatt nur Alarm zu schlagen. Allerdings geht IPS-Betrieb mit Risiken und Performancekosten einher. Zum einen können Fehlalarme nun fälschlicherweise legitimen Datenverkehr blockieren. Ohne sorgfältige Regelanpassung führt das dazu, dass etwa wichtige Updates fehlschlagen oder Dienste nicht funktionieren, ein Umstand, der im Heimnetz lästig und im Unternehmensnetz geschäftskritisch sein kann. Zum anderen benötigt IPS (insbesondere im Inline-Modus) mehr Ressourcen: Jedes Paket wird in Echtzeit gefiltert, was CPU und Durchsatz beansprucht. Hier zeigt sich, mehr Sicherheit durch IPS kann zu höherer Last und administrativem Aufwand führen, während IDS moderater ist, aber Angriffe nicht automatisiert stoppt. Um sowohl die Sicherheit hochzuhalten als auch Fehlalarme und Performanceprobleme zu minimieren, ist die individuelle Anpassung des Regelwerks unerlässlich. Suricata (bzw. Snort) nutzt Regelkategorien, die thematisch gruppiert sind (z.B. “scan“, “exploit“, “malware“ etc.). Wichtig zu wissen ist, dass nicht alle Regeln einer Kategorie standardmäßig aktiviert sind (Anlage9). Die Regel-Autoren liefern viele Signaturen vorsorglich deaktiviert aus (in den Regeldateien mit # auskommentiert), vor allem jene, die in den meisten Umgebungen zu häufigen Fehlalarmen führen könnten. Wenn man in pfSense eine Kategorie einschaltet, werden zunächst nur die vom Autor als „wichtig und zuverlässig“ eingestuften Regeln geladen, während potenziell problematische Regeln aus derselben Kategorie inaktiv bleiben. Deshalb darf man sich nicht in falscher Sicherheit wiegen, nur weil man beispielsweise die Scan-Kategorie aktiviert hat – tatsächlich sind dort standardmäßig oft nur wenige Signaturen aktiv, wodurch bestimmte Portscan-Techniken ohne Anpassung unerkannt bleiben könnten. Dieses Verhalten ist bewusst so gewählt, weil eine umfassende Scan-Erkennung sonst sehr viele harmlose Vorgänge meldet. Möchte man also alle Portscans detektieren, muss man zusätzliche Scan-Regeln von Hand einschalten (wohlwissend, dass dies die Anzahl der Alerts hochtreibt). Generell gilt: Ein IDS/IPS ist keine schlüsselfertige Plug-and-Play-Lösung, sondern muss an die eigene Umgebung angepasst werden. Man sollte nur relevante Regelkategorien aktivieren und innerhalb dieser Kategorien gezielt entscheiden, welche Einzelregeln benötigt werden. Umgekehrt empfiehlt es sich, Regeln zu deaktivieren, die für das eigene Netz bedeutungslos sind (z.B. Signaturen für Serverdienste, die man gar nicht betreibt), so spart man sich unnötige Last und Alarmflut.

Die Snort-Subscriber-Regeln bieten hierfür sogar vordefinierte Policy-Profile (Connectivity, Balanced, Security, Max Detect), die unterschiedliche „Aggressivitätsstufen“ des Regelwerks darstellen. Die Connectivity-Policy aktiviert nur die gravierendsten und eindeutigsten Bedrohungsregeln (weniger kritische bleiben deaktiviert oder nur als Alarm), während Balanced mehr Regeln einschließt und Security noch breiter fast alle Regeln durchsetzt, dies steigert die Abdeckung, führt aber unweigerlich zu deutlich mehr Fehlalarmen und erfordert intensive Betreuung. In der Praxis wird anfänglichen Nutzern geraten, mit einer konservativen Policy (z.B. Connectivity) oder reinem IDS-Modus zu starten, um das System zu beobachten, und erst nach einer Lernphase (ca. 1 Monat) schrittweise auf aggressivere Einstellungen oder IPS-Betrieb zu gehen. So gewinnt man ein Verständnis dafür, welche Alarmmeldungen im eigenen Netz auftauchen und kann anhand dessen feinjustieren, bevor man den automatischen Blockierungsmechanismus voll aktiviert. Eine solche abgestufte Anpassung der Regelkategorien ist essenziell, um die Balance zwischen Sicherheit und Praktikabilität zu halten. Auf pfSense steht für die Feinabstimmung insbesondere die SID-Management-Funktion zur Verfügung, welche eine Schlüsselrolle beim Optimieren des Regelwerks spielt. SID-Management erlaubt es dem Administrator, über Listen von Regel-IDs (SIDs) oder Muster effizient festzulegen, welche Regeln geladen und welche geblockt werden sollen. Anstatt jede Regel manuell an- oder auszuschalten, kann man in pfSense z.B. eine Disable-SID-Liste pflegen, die ganze Gruppen von unerwünschten Regeln per Schlagwort oder ID-Bereich deaktiviert. Ebenso gibt es Enable-Listen, um bestimmte standardmäßig inaktive Regeln dauerhaft einzuschalten. Diese Listen nutzen reguläre Ausdrücke und lassen sich gezielt auf Kategorien oder SID-Bereiche anwenden, etwa könnte man alle Regeln einer Kategorie mit einem Befehl deaktivieren oder alle SIDs eines bestimmten Bereichs aktivieren, ohne jede Zeile anfassen zu müssen. Der große Vorteil: Die Anpassungen bleiben auch nach Regel-Updates erhalten. Bei jedem Herunterladen neuer Snort/Suricata-Regeln wendet pfSense die SIDmgmt-Filter wieder an, sodass zum Beispiel störende Regeln konsequent deaktiviert bleiben und nicht durch ein Update wieder ungewollt aktiv werden. Darüber hinaus kann SID-Management auch genutzt werden, um die Aktion von Regeln zu ändern. Standardmäßig sind alle Signaturen auf ALERT gesetzt – nur alarmieren. Im IPS-Modus (wenn „Block Drop-only“ konfiguriert ist) blockiert Suricata allerdings nur solche Regeln, die als Aktion DROP definiert sind. Mit einer Drop-SID-Liste kann man gezielt bestimmten SIDs die Aktion auf “DROP” umstellen, sodass Suricata bei deren Treffer den Traffic tatsächlich verwirft.

Auf diese Weise lässt sich ein maßgeschneidertes Blockierungsverhalten umsetzen: Weniger verlässliche Regeln bleiben auf „Alert only“, während hochkritische oder vertrauenswürdige Signaturen per SID-Mgmt zu aktiven Block-Regeln hochgestuft werden. SIDmgmt ist somit das Werkzeug, um das Regelset optimal an die eigene Netzwerkumgebung und Risikotoleranz anzupassen, ohne bei jedem Regelupdate erneut von vorn anfangen zu müssen.

Die konkrete Netzwerkarchitektur (WAN, LAN, OPT1, OPT2) beeinflusst ebenfalls den optimalen Einsatz von Suricata. In einem pfSense-System mit mehreren Netzsegmenten stellt sich die Frage, auf welchen Interfaces Suricata mitlaufen soll. Auf dem WAN-Interface platziert, sieht Suricata den Verkehr an der Außengrenze des Netzwerks – also typischerweise eingehende Verbindungen aus dem Internet sowie ausgehenden Traffic nach NAT. Der Vorteil: Mit einer einzigen Suricata-Instanz auf WAN kann man den gesamten externen Datenstrom überwachen, was bei vielen internen Netzen effizienter sein kann als für jedes ein separater Prozess. Allerdings arbeitet Suricata auf WAN vor der Firewall: Es untersucht dort auch Pakete, die die pfSense-Firewall später ohnehin blockieren würde (z.B. unerwünschte Scans auf geschlossene Ports). Zudem kann Suricata auf dem externen Interface die internen Zieladressen nicht erkennen, da es nur die öffentlichen IPs (und die eigene WAN-IP als Ziel bei eingehendem Traffic) sieht.

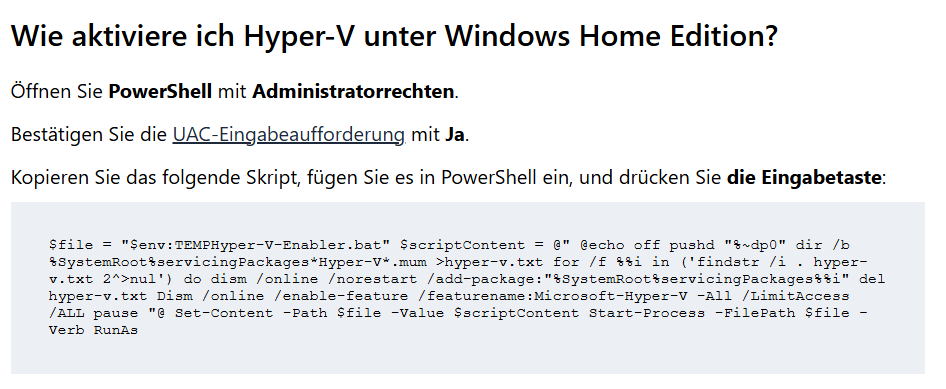
Man verliert also etwas an Kontext, etwa welcher interne Host angesprochen wurde. Auf den LAN/OPT-Interfaces installiert, überwacht Suricata hingegen den Verkehr innerhalb der lokalen Netze bzw. zwischen Internetwork und Firewall. Es sieht die echten internen IP-Adressen als Quelle oder Ziel und nur den Traffic, der die Firewall tatsächlich passiert. Dadurch erhält man präzisere Informationen, welcher interne Client oder Server an einer verdächtigen Kommunikation beteiligt ist.

Abschließend denke ich, dass die Entscheidung, ob Suricata als IPS oder IDS betrieben wird, mehr als nur eine einfache Wahl ist. Es fließen zahlreiche Faktoren in diesen Entscheidungsprozess ein, von den grundlegenden Funktionsweisen bis hin zu der Möglichkeit die verschiedenen IPS Policy zu nutzen. Hinzu kommen die umfassenden Konfigurationsoptionen des SIDmgmt, die es ermöglichen, eine wirklich flexible und tragbare Lösung zu entwickeln.

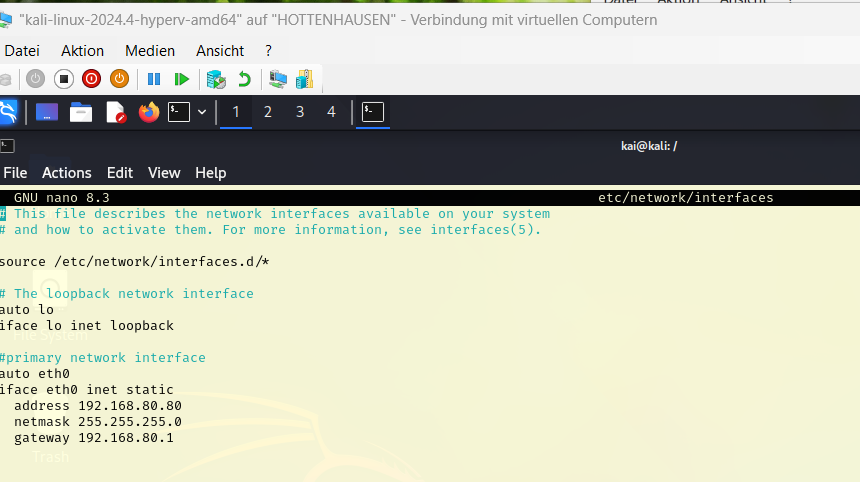
Wenn man all diese Aspekte berücksichtigt, schafft man die Basis für einen kontinuierlichen Entwicklungsprozess, der sich an wechselnde Anforderungen anpassen kann. Dieser Ansatz motiviert dazu, immer wieder neue Optimierungen vorzunehmen und die Balance zwischen Standardisierung und Flexibilität zu finden.

Anhang (Anlagen)

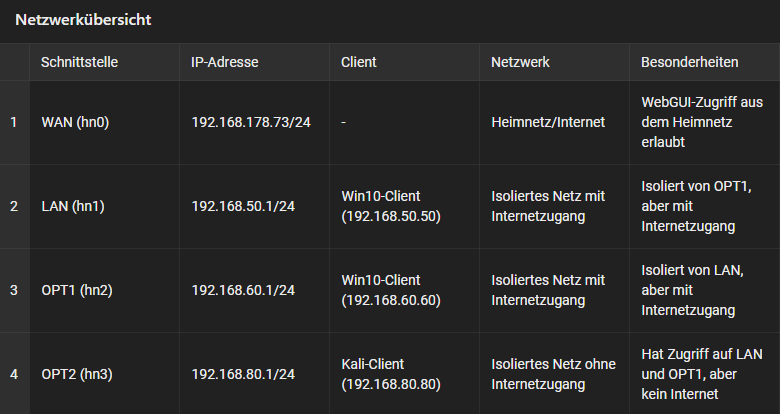
Anlage 1 - Screenshot-Hyper-V freischalten



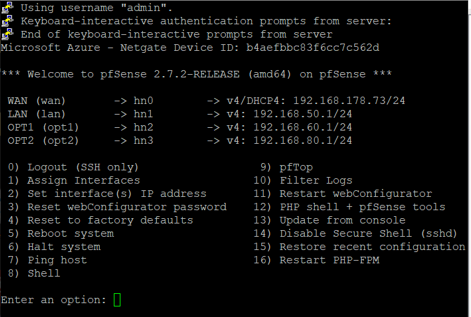
Anlage 2 - Statische IP in KaliLinux



Anlage 3 - Übersicht Schnittstellen, Clients, Netzwerke

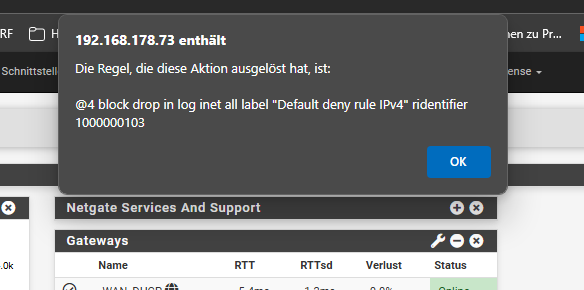


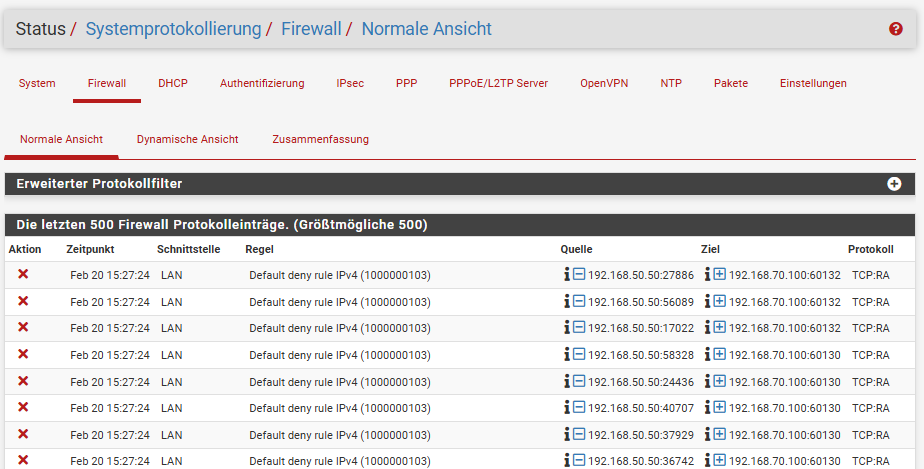
Anlage 4 - Interface Zuordnung pfSense Konsole

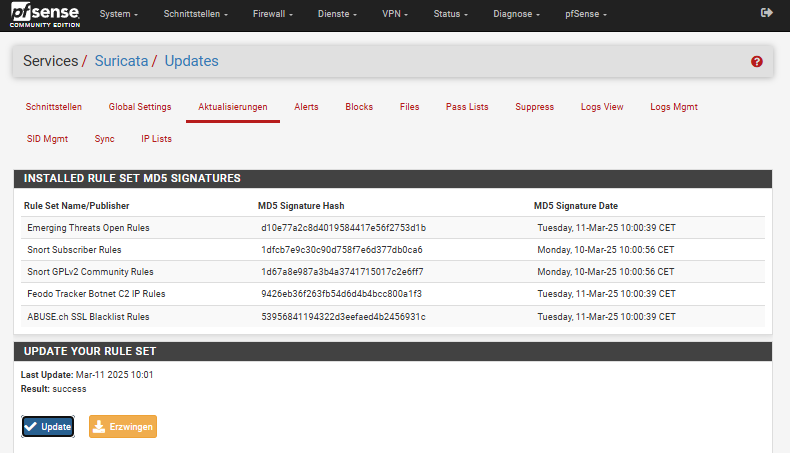


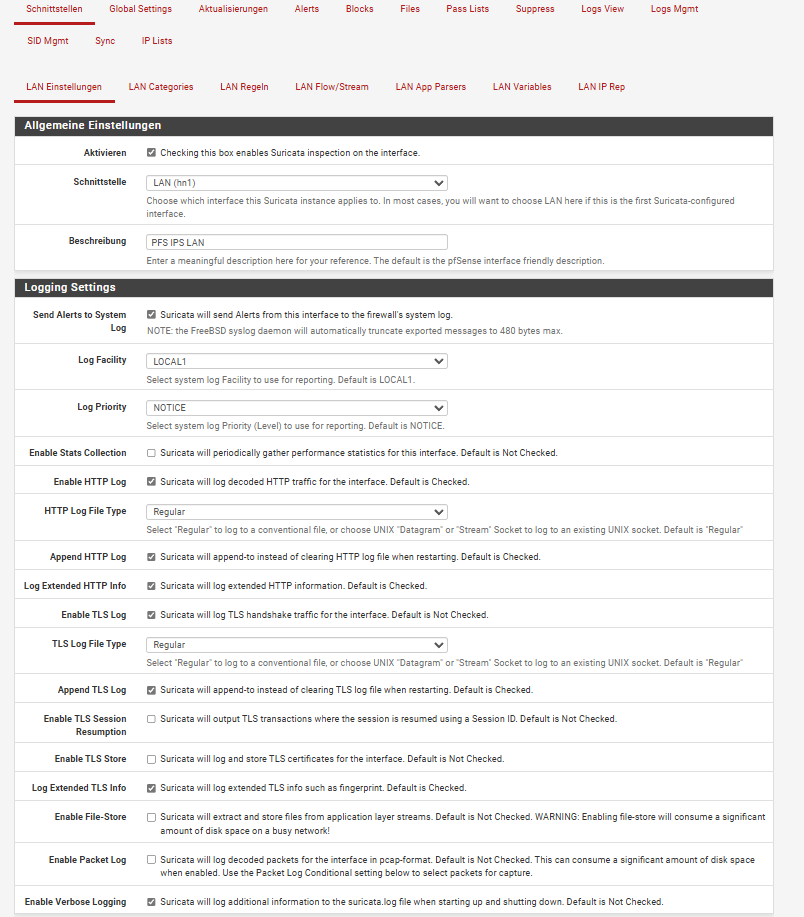
Anlage 5 – pfSense “Alerts” wegen asymmetrischem Routing

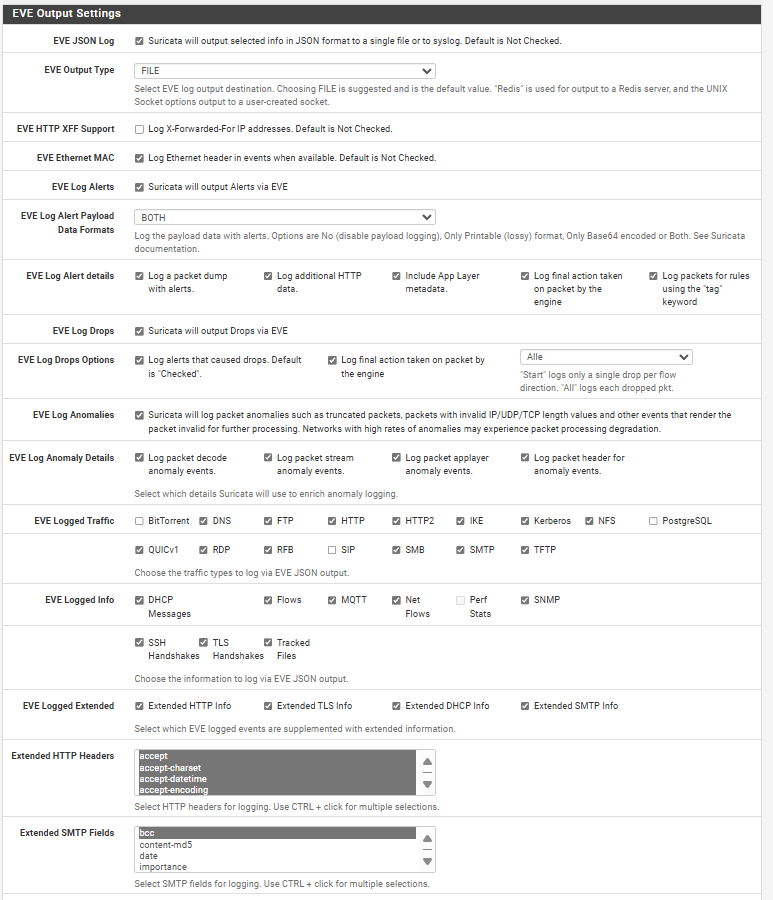
(bevor der Fehler behoben war, hatte die Kali-VM, also die VM noch auf dem Cloudcommand-Laptop die IP-Adresse 192.168.70.100)

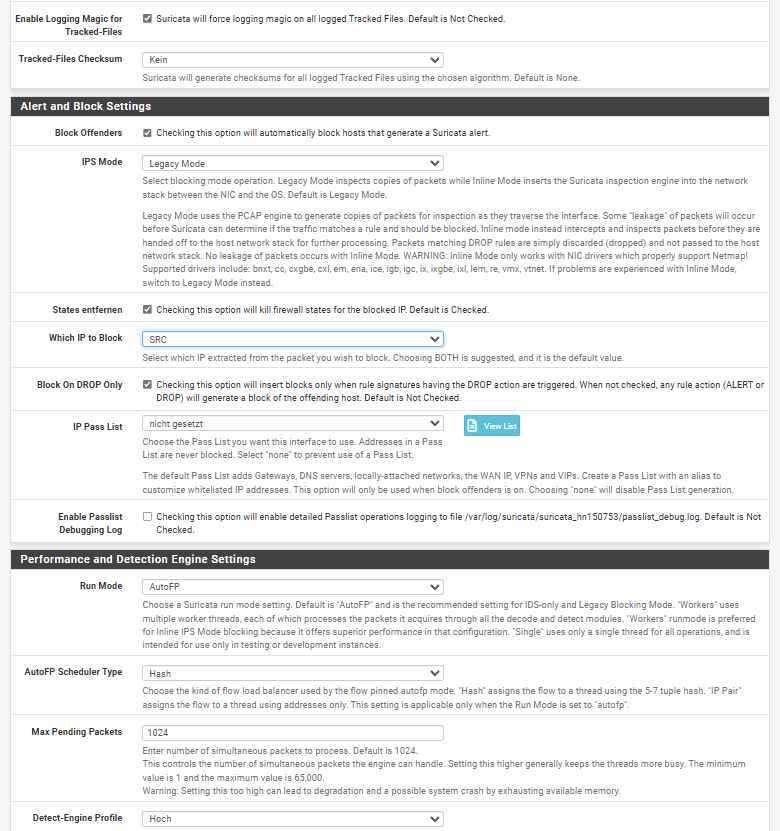


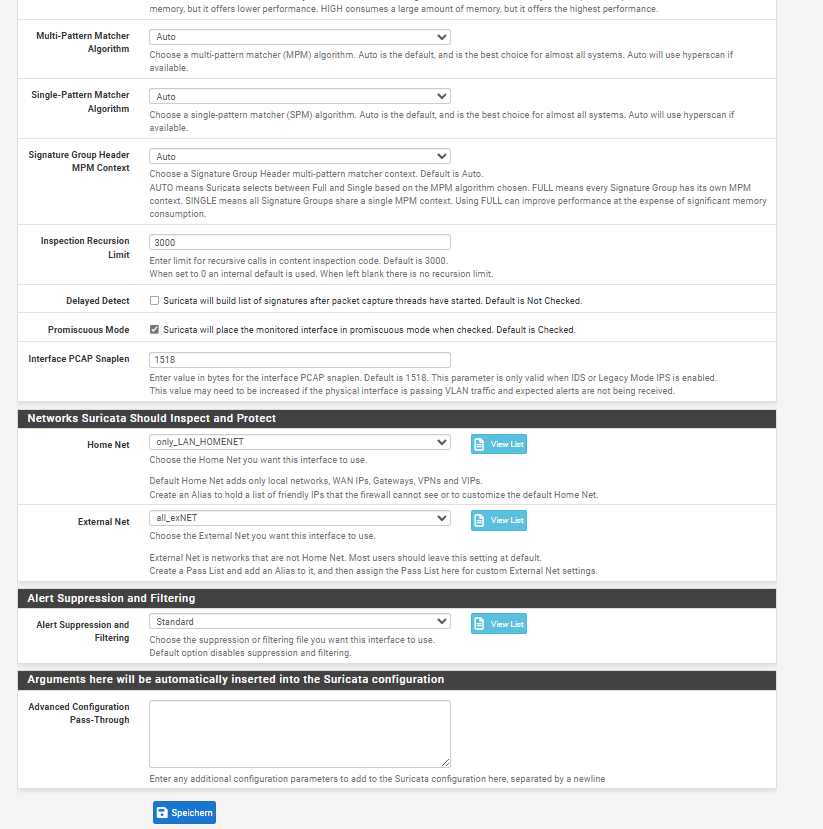
Dadurch wurde der Traffic von pfSense geblockt, bevor Suricata ihn ”sah”.

Anlage 6 – Erfolgreicher Download “Rulesets”

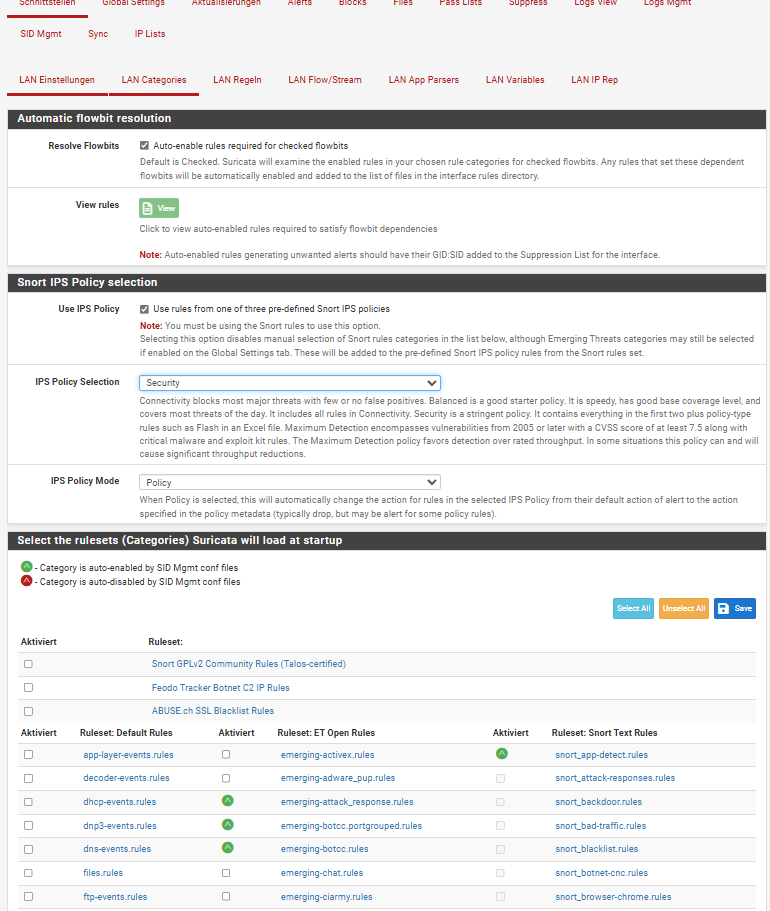
Anlage 7 – Einstellungen LAN-Schnittstelle

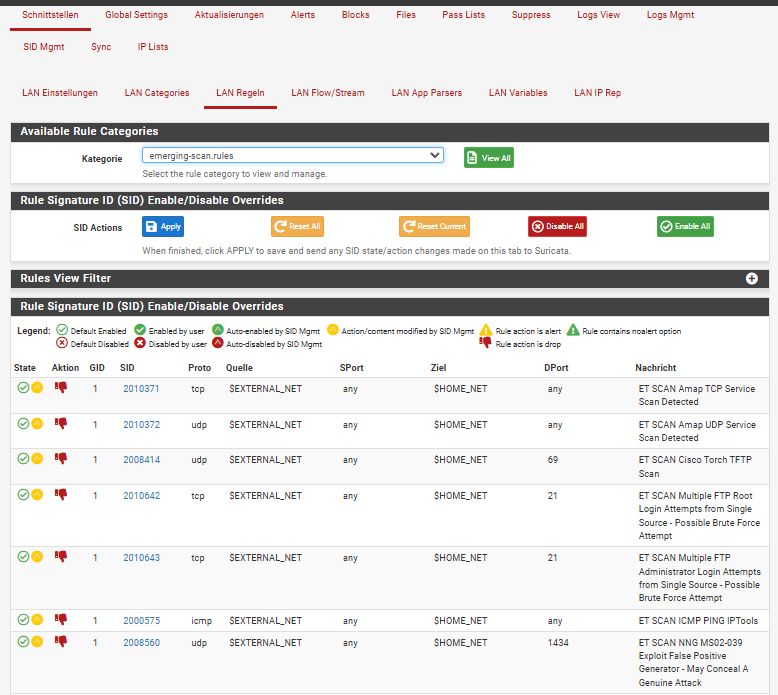




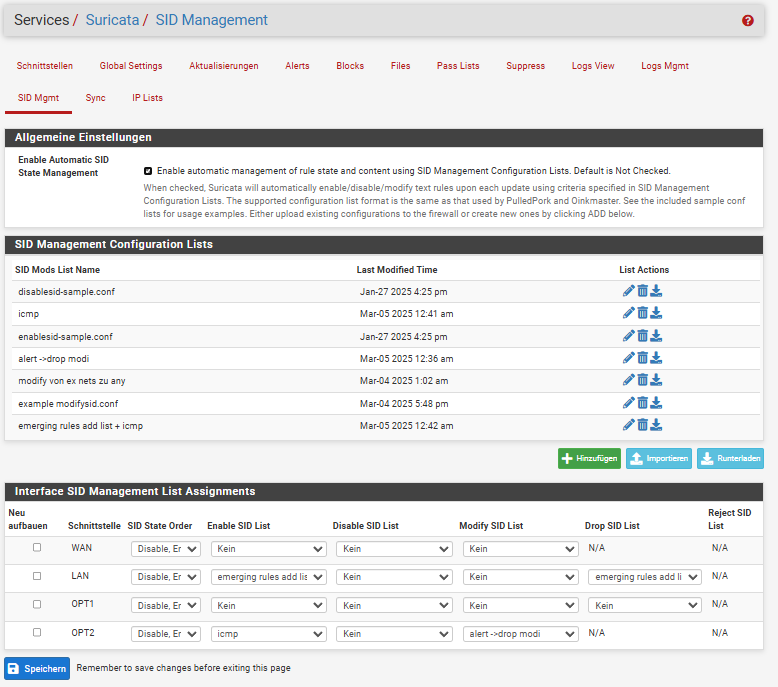


Regelkategorien mit aktiven SIDmgmt.

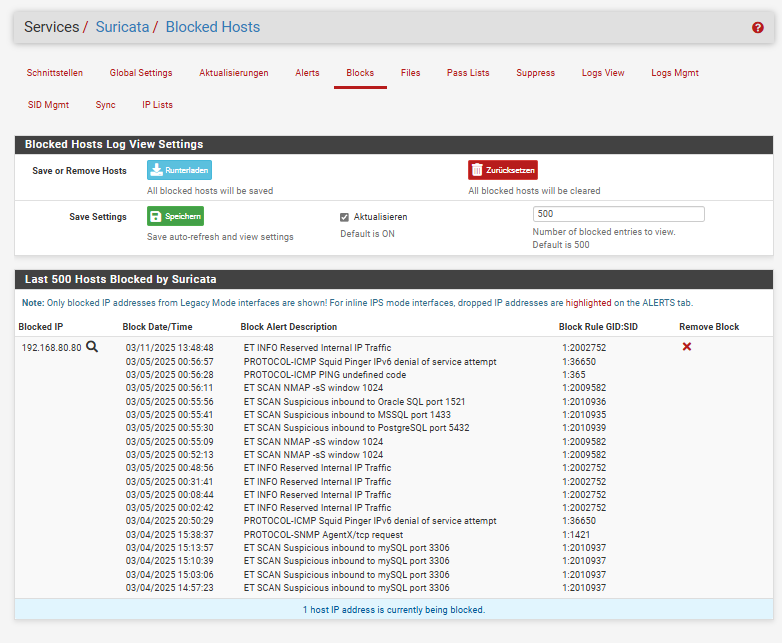
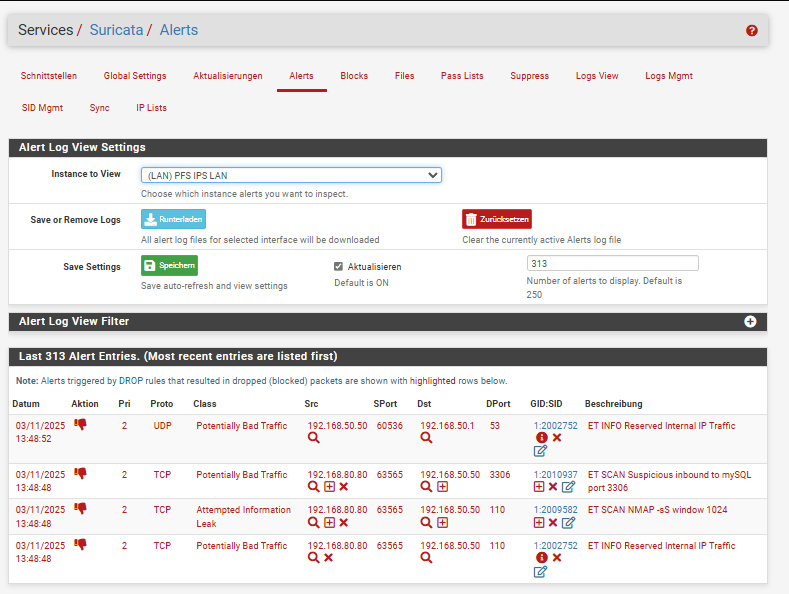


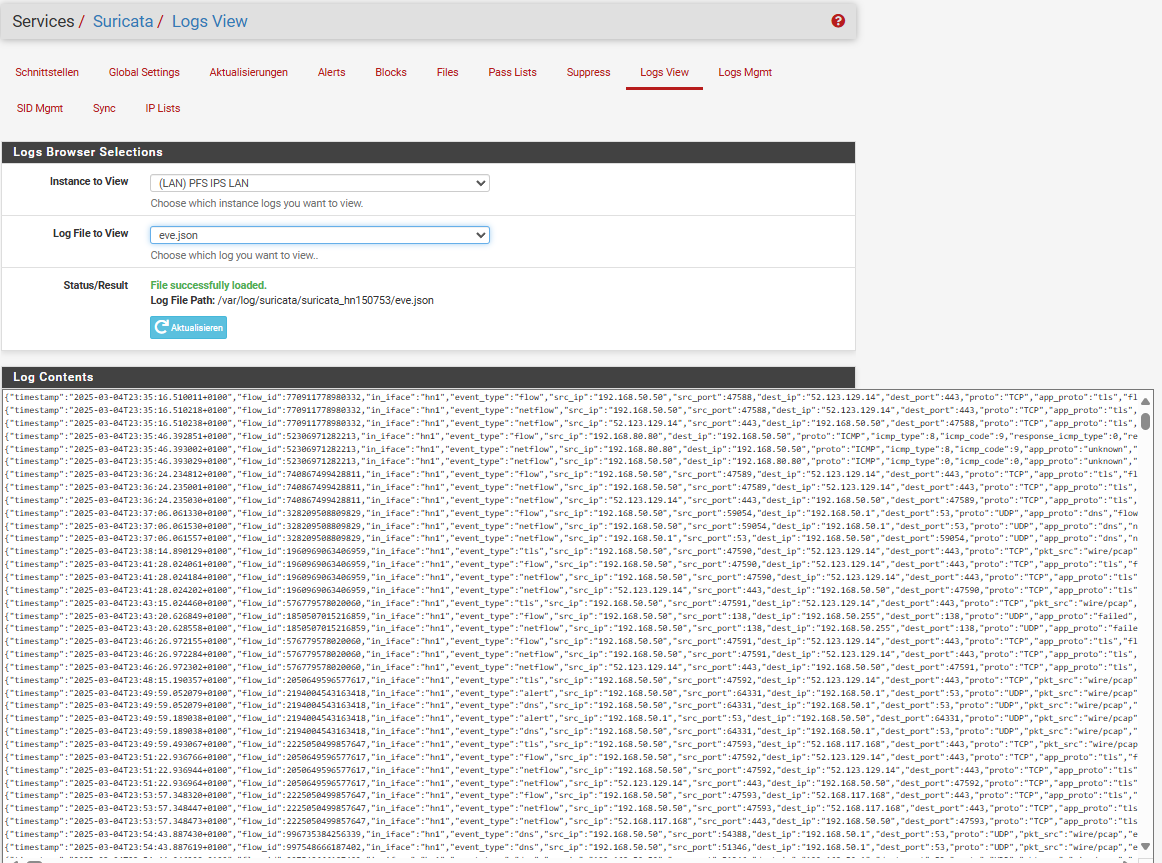


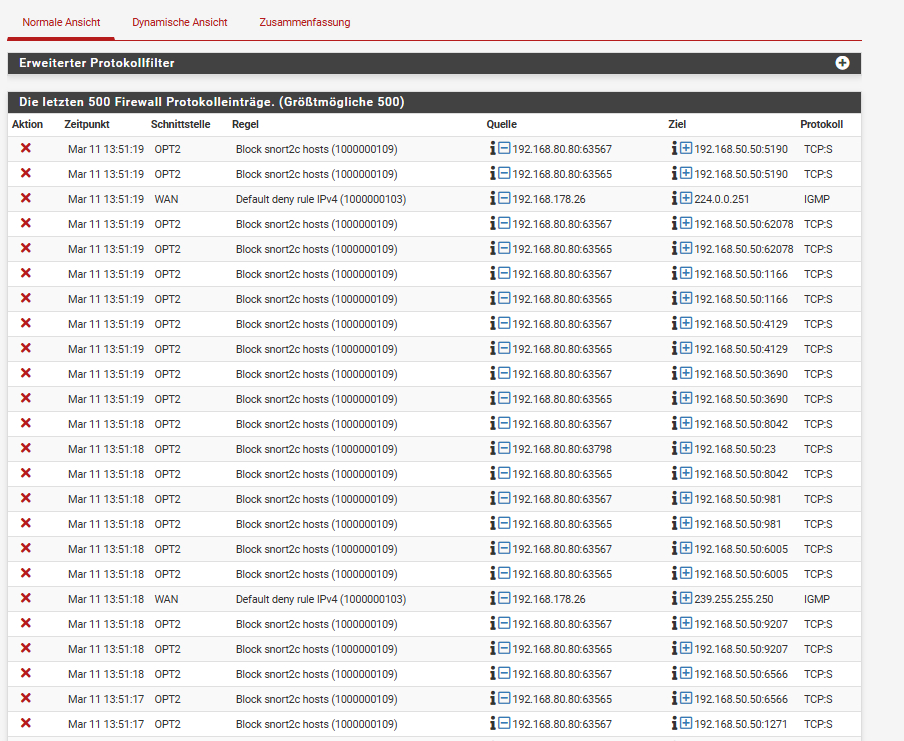
SIDmgmt mit aktiven “enable-” und “drop List”



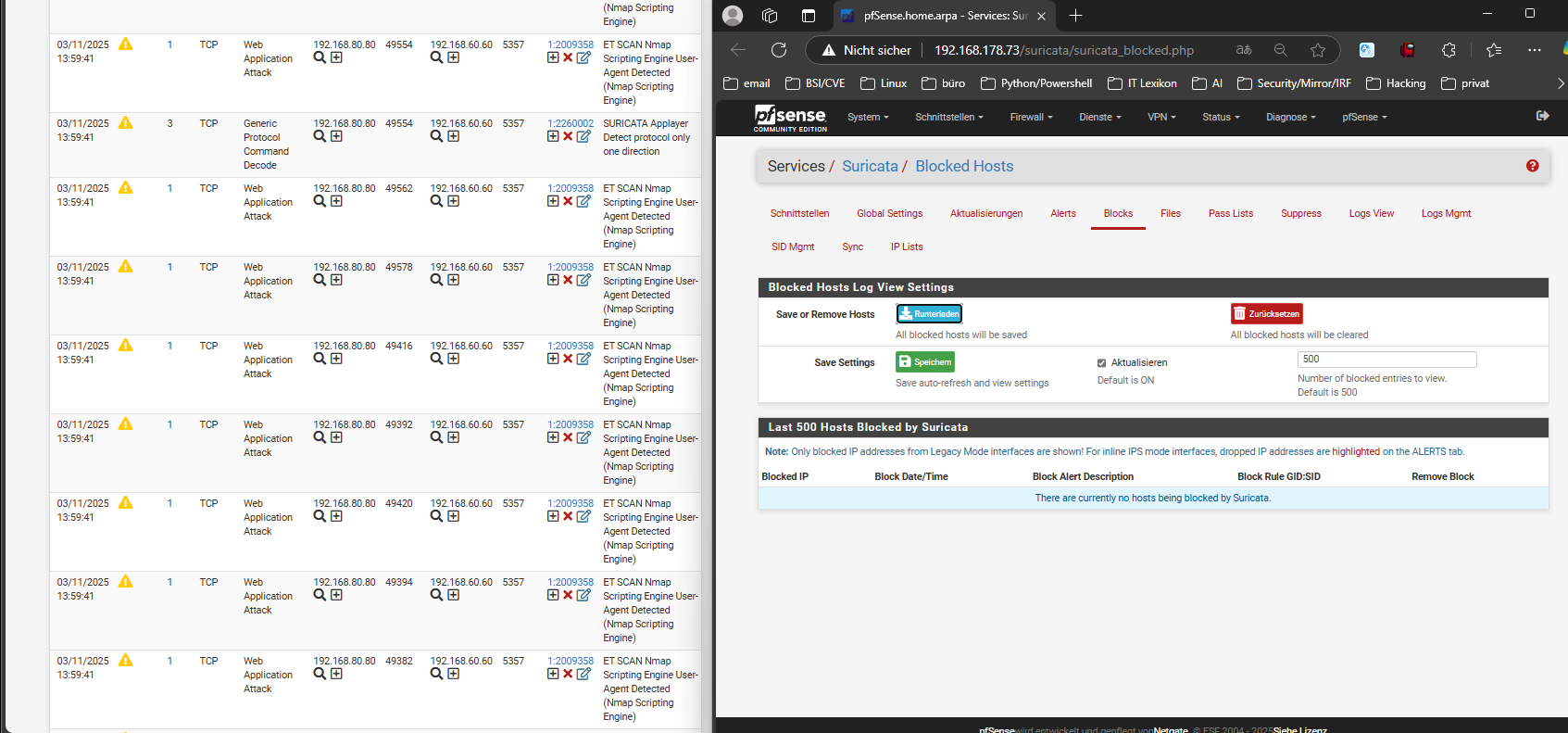
Anlage 8 – Testergebnisse auf das LAN-Netz IPS





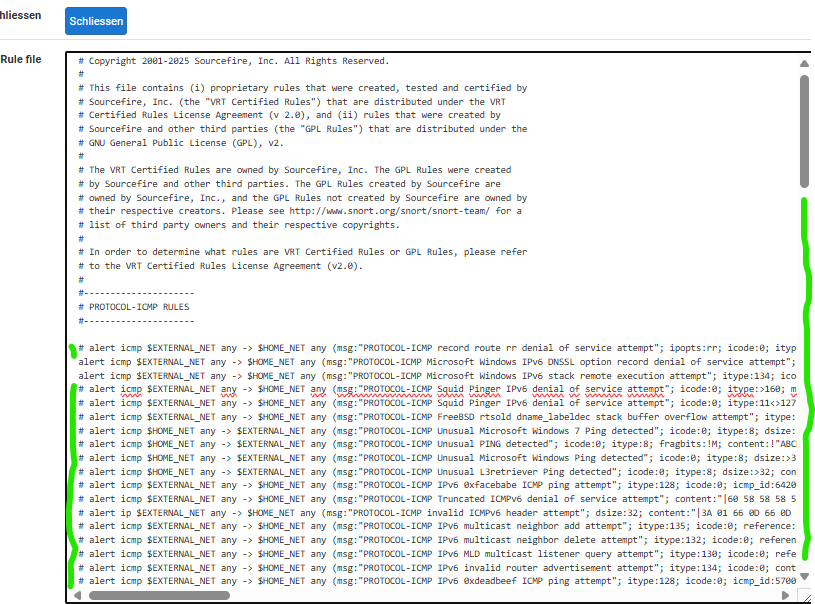


Anlage 8 – Testergebnisse auf das OPT1-Netz IDS

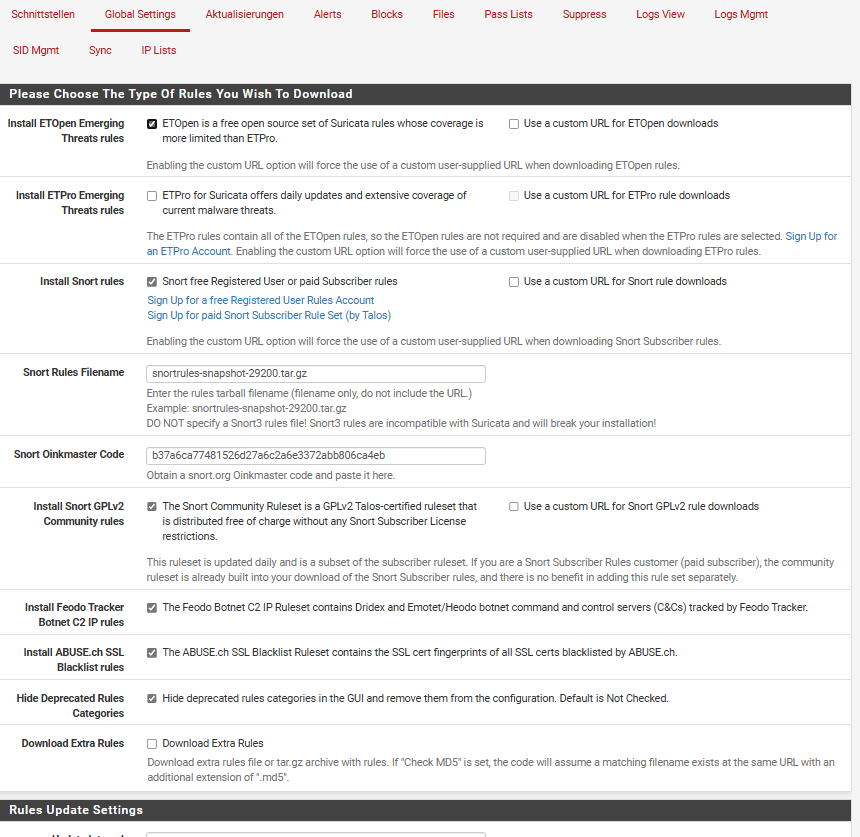


Anlage 9 – Beispiel Snort icmp Rules

(und wieviel der Regeln standardmäßig deaktiviert sind)



Anlage 10 – Global Settings, Regelauswahl



Abkürzungsverzeichnis

ADDS – Active Directory Domain Services

DHCP – Dynamic Host Configuration Protocol

DNS – Domain Name System

DoS – Denial of Service

GUI – Graphical User Interface

IDS – Intrusion Detection System

IPS – Intrusion Prevention System

LAN – Local Area Network

NAT – Network Address Translation

NIC – Network Interface Card

NTP – Network Time Protocol

SIDmgmt – Suricata ID Management

SSH – Secure Shell

VM – Virtuelle Maschine

WAN – Wide Area Network

Quellenverzeichnis

-Hyper-V auf Windows 11 Home freischalten - Stand 06.03.2025

https://assistouest.fr/en/enabling-hyper-v-on-windows-home-edition-with-powershell/

-Die meisten Informationen sind unter forum.netgate.com zu bekommen, teilweise von den Entwicklern von pfSense selbst- Stand 11.03.2025

Forum.netgate.com



