

Implementieren eines IDS/IPS in einem beispielhaften Firmennetzwerk

Kai Kauffmann

Abgabedatum: 11.03.2025 Prüfungsjahr: 2025

# Inhaltsverzeichnis

Inhaltsverzeichnis .....	2
1. Projektbeschreibung .....	2
2. Konzeption & Planung .....	4
2.2 Zeitplan / Aufwand .....	5
3. Realisierungsphase.....	6
4. Testphase .....	13
6. Fazit.....	14
Anhang (Anlagen) .....	18
Anlage 3 - Übersicht Schnittstellen, Clients, Netzwerke .....	19
Anlage 5 – pfSense “Alerts” wegen asymmetrischem Routing .....	20
Regelkategorien mit aktiven SIDmgmt.....	27
SIDmgmt mit aktiven “enable-” und “drop List” .....	29
Anlage 8 – Testergebnisse auf das LAN-Netz IPS .....	29
.....	30
Anlage 8 – Testergebnisse auf das OPT1-Netz IDS .....	32
Anlage 9 – Beispiel Snort icmp Rules.....	32
(und wieviel der Regeln standardmäßig deaktiviert sind).....	32
Abkürzungsverzeichnis .....	33
Quellenverzeichnis .....	34

## 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-Clients

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 Einrichtung von pfsense über die Konsole abgeschlossen sein und der Zugang zur WebGUI, über den Windows-10-Clients 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 umfangreiches Fazit zum Implementieren 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 Netzwerkadaptoreinstellungen 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 Möglichkeit 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 (Anlage 7 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 wird. 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 kompatibel 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 böses 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, Fehllarme (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 Fehllarme 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 Fehllarme 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 Fehllarmen 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.

## Anlage 1 - Screenshot-Hyper-V freischalten

### Wie aktiviere ich Hyper-V unter Windows Home Edition?

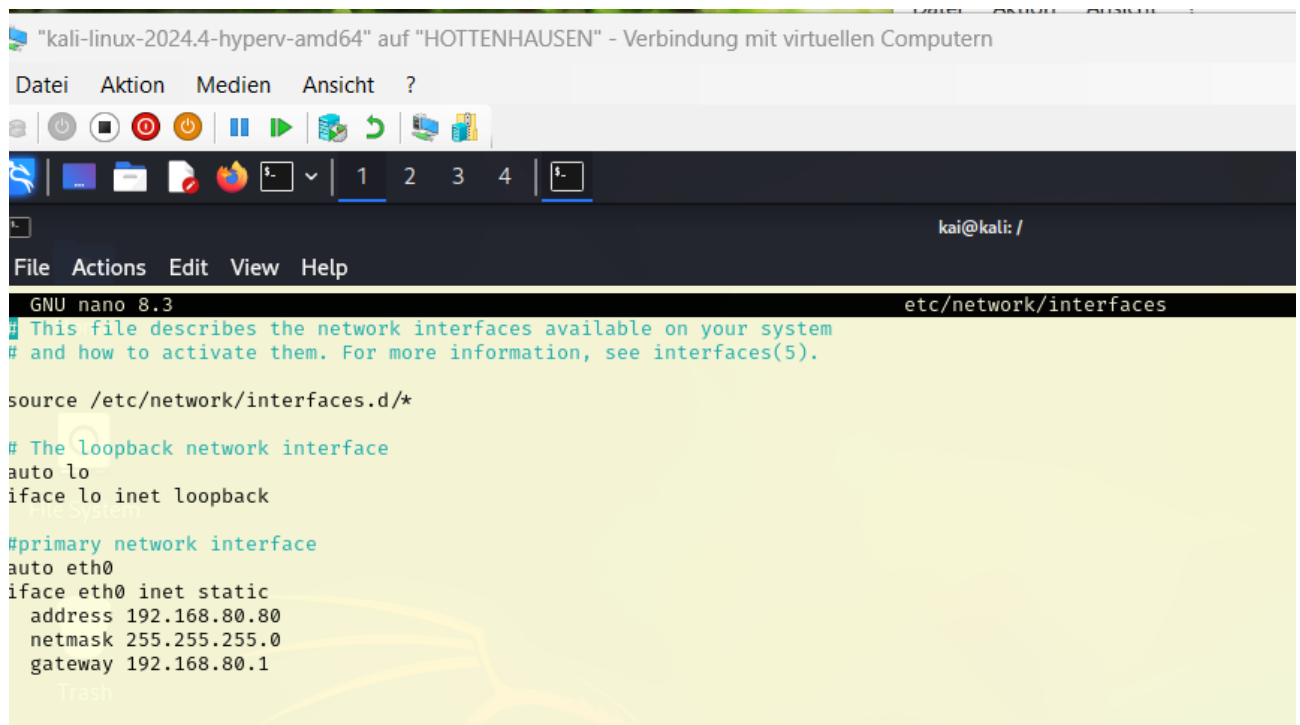
Öffnen Sie **PowerShell** mit **Administratorrechten**.

Bestätigen Sie die UAC-Eingabeaufforderung mit **Ja**.

Kopieren Sie das folgende Skript, fügen Sie es in PowerShell ein, und drücken Sie **die Eingabetaste**:

```
$file = "$env:TEMPHyper-V-Enabler.bat" $scriptContent = @" @echo off pushd "%~dp0" dir /b %SystemRoot%servicingPackages*Hyper-V*.mum >hyper-v.txt for /f %i in ('findstr /i . hyper-v.txt 2^>nul') do dism /online /norestart /add-package:"%SystemRoot%servicingPackages%i" del hyper-v.txt Dism /online /enable-feature /featurename:Microsoft-Hyper-V -All /LimitAccess /ALL pause "@ Set-Content -Path $file -Value $scriptContent Start-Process -FilePath $file -Verb RunAs
```

## Anlage 2 - Statische IP in KaliLinux



```
"kali-linux-2024.4-hyperv-amd64" auf "HOTTENHAUSEN" - Verbindung mit virtuellen Computern
Datei  Aktion  Medien  Ansicht  ?
GNU nano 8.3  etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

#primary network interface
auto eth0
iface eth0 inet static
    address 192.168.80.80
    netmask 255.255.255.0
    gateway 192.168.80.1
```

## Anlage 3 - Übersicht Schnittstellen, Clients, Netzwerke

Netzwerkübersicht					
	Schnittstelle	IP-Adresse	Client	Netzwerk	Besonderheiten
1	WAN (hn0)	192.168.178.73/24	-	Heimnetz/Internet	WebGUI-Zugriff aus dem Heimnetz erlaubt
2	LAN (hn1)	192.168.50.1/24	Win10-Client (192.168.50.50)	Isoliertes Netz mit Internetzugang	Isoliert von OPT1, aber mit Internetzugang
3	OPT1 (hn2)	192.168.60.1/24	Win10-Client (192.168.60.60)	Isoliertes Netz mit Internetzugang	Isoliert von LAN, aber mit Internetzugang
4	OPT2 (hn3)	192.168.80.1/24	Kali-Client (192.168.80.80)	Isoliertes Netz ohne Internetzugang	Hat Zugriff auf LAN und OPT1, aber kein Internet

## Anlage 4 - Interface Zuordnung pfSense Konsole

```

Using username "admin".
Keyboard-interactive authentication prompts from server:
End of keyboard-interactive prompts from server
Microsoft Azure - Netgate Device ID: b4aefbbc83f6cc7c562d

*** Welcome to pfSense 2.7.2-RELEASE (amd64) on pfSense ***

WAN (wan)      -> hn0      -> v4/DHCP4: 192.168.178.73/24
LAN (lan)      -> hn1      -> v4: 192.168.50.1/24
OPT1 (opt1)    -> hn2      -> v4: 192.168.60.1/24
OPT2 (opt2)    -> hn3      -> v4: 192.168.80.1/24

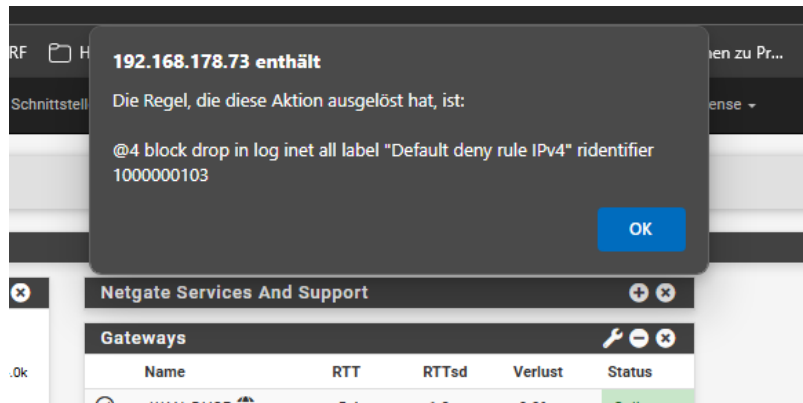
0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults  13) Update from console
5) Reboot system              14) Disable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell

Enter an option: █

```

## 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)



Status / Systemprotokollierung / Firewall / Normale Ansicht

System **Firewall** DHCP Authentifizierung IPsec PPP PPPoE/L2TP Server OpenVPN NTP Pakete Einstellungen

Normale Ansicht **Dynamische Ansicht** Zusammenfassung



Erweiterter Protokollfilter


Die letzten 500 Firewall Protokolleinträge. (Größtmögliche 500)

Aktion	Zeitpunkt	Schnittstelle	Regel	Quelle	Ziel	Protokoll
✗	Feb 20 15:27:24	LAN	Default deny rule IPv4 (1000000103)	192.168.50.50:27886	192.168.70.100:60132	TCP:RA
✗	Feb 20 15:27:24	LAN	Default deny rule IPv4 (1000000103)	192.168.50.50:56089	192.168.70.100:60132	TCP:RA
✗	Feb 20 15:27:24	LAN	Default deny rule IPv4 (1000000103)	192.168.50.50:17022	192.168.70.100:60132	TCP:RA
✗	Feb 20 15:27:24	LAN	Default deny rule IPv4 (1000000103)	192.168.50.50:58328	192.168.70.100:60130	TCP:RA
✗	Feb 20 15:27:24	LAN	Default deny rule IPv4 (1000000103)	192.168.50.50:24436	192.168.70.100:60130	TCP:RA
✗	Feb 20 15:27:24	LAN	Default deny rule IPv4 (1000000103)	192.168.50.50:40707	192.168.70.100:60130	TCP:RA
✗	Feb 20 15:27:24	LAN	Default deny rule IPv4 (1000000103)	192.168.50.50:37929	192.168.70.100:60130	TCP:RA
✗	Feb 20 15:27:24	LAN	Default deny rule IPv4 (1000000103)	192.168.50.50:36742	192.168.70.100:60130	TCP:RA

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

## Anlage 6 – Erfolgreicher Download “Rulesets”

System - Schnittstellen - Firewall - Dienste - VPN - Status - Diagnose - pfSense - 

Services / Suricata / Updates 

SchnittstellenGlobal SettingsAktualisierungenAlertsBlocksFilesPass ListsSuppressLogs ViewLogs Mgmt



SID MgmtSyncIP Lists

**INSTALLED RULE SET MD5 SIGNATURES**

Rule Set Name/Publisher	MD5 Signature Hash	MD5 Signature Date
Emerging Threats Open Rules	d10e77a2c8d4019584417e56f2753d1b	Tuesday, 11-Mar-25 10:00:39 CET
Snort Subscriber Rules	1dfcb7e9c30c90d758f7e6d377db0ca6	Monday, 10-Mar-25 10:00:56 CET
Snort GPLv2 Community Rules	1d67a8e987a3b4a3741715017c2e6ff7	Monday, 10-Mar-25 10:00:56 CET
Feodo Tracker Botnet C2 IP Rules	9426eb36f263fb54d6d4b4bcc800a1f3	Tuesday, 11-Mar-25 10:00:39 CET
ABUSE.ch SSL Blacklist Rules	53956841194322d3eefaed4b2456931c	Tuesday, 11-Mar-25 10:00:39 CET

**UPDATE YOUR RULE SET**

Last Update: Mar-11 2025 10:01  
Result: success

## Anlage 7 – Einstellungen LAN-Schnittstelle

Schnittstellen	Global Settings	Aktualisierungen	Alerts	Blocks	Files	Pass Lists	Suppress	Logs View	Logs Mgmt
SID Mgmt	Sync	IP Lists							
LAN Einstellungen	LAN Categories	LAN Regeln	LAN Flow/Stream	LAN App Parsers	LAN Variables	LAN IP Rep			

### Allgemeine Einstellungen

Aktivieren	<input checked="" type="checkbox"/> Checking this box enables Suricata inspection on the interface.
Schnittstelle	<div>LAN (hn1)</div> <div>Choose which interface this Suricata instance applies to. In most cases, you will want to choose LAN here if this is the first Suricata-configured interface.</div>
Beschreibung	<div>PFS IPS LAN</div> <div>Enter a meaningful description here for your reference. The default is the pfSense interface friendly description.</div>

### Logging Settings

Send Alerts to System Log	<input checked="" type="checkbox"/> Suricata will send Alerts from this interface to the firewall's system log. NOTE: the FreeBSD syslog daemon will automatically truncate exported messages to 480 bytes max.
Log Facility	<div>LOCAL1</div> <div>Select system log Facility to use for reporting. Default is LOCAL1.</div>
Log Priority	<div>NOTICE</div> <div>Select system log Priority (Level) to use for reporting. Default is NOTICE.</div>
Enable Stats Collection	<input type="checkbox"/> Suricata will periodically gather performance statistics for this interface. Default is Not Checked.
Enable HTTP Log	<input checked="" type="checkbox"/> Suricata will log decoded HTTP traffic for the interface. Default is Checked.
HTTP Log File Type	<div>Regular</div> <div>Select "Regular" to log to a conventional file, or choose UNIX "Datagram" or "Stream" Socket to log to an existing UNIX socket. Default is "Regular"</div>
Append HTTP Log	<input checked="" type="checkbox"/> Suricata will append-to instead of clearing HTTP log file when restarting. Default is Checked.
Log Extended HTTP Info	<input checked="" type="checkbox"/> Suricata will log extended HTTP information. Default is Checked.
Enable TLS Log	<input checked="" type="checkbox"/> Suricata will log TLS handshake traffic for the interface. Default is Not Checked.
TLS Log File Type	<div>Regular</div> <div>Select "Regular" to log to a conventional file, or choose UNIX "Datagram" or "Stream" Socket to log to an existing UNIX socket. Default is "Regular"</div>
Append TLS Log	<input checked="" type="checkbox"/> Suricata will append-to instead of clearing TLS log file when restarting. Default is Checked.
Enable TLS Session Resumption	<input type="checkbox"/> Suricata will output TLS transactions where the session is resumed using a Session ID. Default is Not Checked.
Enable TLS Store	<input type="checkbox"/> Suricata will log and store TLS certificates for the interface. Default is Not Checked.
Log Extended TLS Info	<input checked="" type="checkbox"/> Suricata will log extended TLS info such as fingerprint. Default is Checked.
Enable File-Store	<input type="checkbox"/> Suricata will extract and store files from application layer streams. Default is Not Checked. WARNING: Enabling file-store will consume a significant amount of disk space on a busy network!
Enable Packet Log	<input type="checkbox"/> Suricata will log decoded packets for the interface in pcap-format. Default is Not Checked. This can consume a significant amount of disk space when enabled. Use the Packet Log Conditional setting below to select packets for capture.
Enable Verbose Logging	<input checked="" type="checkbox"/> Suricata will log additional information to the suricata.log file when starting up and shutting down. Default is Not Checked.

EVE Output Settings					
EVE JSON Log	<input checked="" type="checkbox"/> Suricata will output selected info in JSON format to a single file or to syslog. Default is Not Checked.				
EVE Output Type	<div>FILE</div> <div>Select EVE log output destination. Choosing FILE is suggested and is the default value. "Redis" is used for output to a Redis server, and the UNIX Socket options output to a user-created socket.</div>				
EVE HTTP XFF Support	<input type="checkbox"/> Log X-Forwarded-For IP addresses. Default is Not Checked.				
EVE Ethernet MAC	<input checked="" type="checkbox"/> Log Ethernet header in events when available. Default is Not Checked.				
EVE Log Alerts	<input checked="" type="checkbox"/> Suricata will output Alerts via EVE				
EVE Log Alert Payload Data Formats	<div>BOTH</div> <div>Log the payload data with alerts. Options are No (disable payload logging), Only Printable (lossy) format, Only Base64 encoded or Both. See Suricata documentation.</div>				
EVE Log Alert details	<input checked="" type="checkbox"/> Log a packet dump with alerts.	<input checked="" type="checkbox"/> Log additional HTTP data.	<input checked="" type="checkbox"/> Include App Layer metadata.	<input checked="" type="checkbox"/> Log final action taken on packet by the engine	<input checked="" type="checkbox"/> Log packets for rules using the "tag" keyword
EVE Log Drops	<input checked="" type="checkbox"/> Suricata will output Drops via EVE				
EVE Log Drops Options	<input checked="" type="checkbox"/> Log alerts that caused drops. Default is "Checked".	<input checked="" type="checkbox"/> Log final action taken on packet by the engine	<div>Alle</div> <div>"Start" logs only a single drop per flow direction. "All" logs each dropped pkt.</div>		
EVE Log Anomalies	<input checked="" type="checkbox"/> Suricata will log packet anomalies such as truncated packets, packets with invalid IP/UDP/TCP length values and other events that render the packet invalid for further processing. Networks with high rates of anomalies may experience packet processing degradation.				
EVE Log Anomaly Details	<input checked="" type="checkbox"/> Log packet decode anomaly events.	<input checked="" type="checkbox"/> Log packet stream anomaly events.	<input checked="" type="checkbox"/> Log packet applayer anomaly events.	<input checked="" type="checkbox"/> Log packet header for anomaly events.	
Select which details Suricata will use to enrich anomaly logging.					
EVE Logged Traffic	<input type="checkbox"/> BitTorrent	<input checked="" type="checkbox"/> DNS	<input checked="" type="checkbox"/> FTP	<input checked="" type="checkbox"/> HTTP	<input checked="" type="checkbox"/> HTTP2
	<input checked="" type="checkbox"/> IKE	<input checked="" type="checkbox"/> Kerberos	<input checked="" type="checkbox"/> NFS	<input type="checkbox"/> PostgreSQL	
	<input checked="" type="checkbox"/> QUICv1	<input checked="" type="checkbox"/> RDP	<input checked="" type="checkbox"/> RFB	<input type="checkbox"/> SIP	<input checked="" type="checkbox"/> SMB
	<input checked="" type="checkbox"/> SMTP	<input checked="" type="checkbox"/> TFTP	Choose the traffic types to log via EVE JSON output.		
EVE Logged Info	<input checked="" type="checkbox"/> DHCP Messages	<input checked="" type="checkbox"/> Flows	<input checked="" type="checkbox"/> MQTT	<input checked="" type="checkbox"/> Net Flows	<input type="checkbox"/> Perf Stats
	<input checked="" type="checkbox"/> SNMP	<input checked="" type="checkbox"/> SSH Handshakes	<input checked="" type="checkbox"/> TLS Handshakes	<input checked="" type="checkbox"/> Tracked Files	
Choose the information to log via EVE JSON output.					
EVE Logged Extended	<input checked="" type="checkbox"/> Extended HTTP Info	<input checked="" type="checkbox"/> Extended TLS Info	<input checked="" type="checkbox"/> Extended DHCP Info	<input checked="" type="checkbox"/> Extended SMTP Info	
Select which EVE logged events are supplemented with extended information.					
Extended HTTP Headers	<div>accept</div> <div>accept-charset</div> <div>accept-datetime</div> <div>accept-encoding</div> <div>Select HTTP headers for logging. Use CTRL + click for multiple selections.</div>				
Extended SMTP Fields	<div>bcc</div> <div>content-md5</div> <div>date</div> <div>importance</div> <div>Select SMTP fields for logging. Use CTRL + click for multiple selections.</div>				



<b>Enable Logging Magic for Tracked-Files</b>	<input checked="" type="checkbox"/> Suricata will force logging magic on all logged Tracked Files. Default is Not Checked.	
<b>Tracked-Files Checksum</b>	<div>Kein</div> <div>Suricata will generate checksums for all logged Tracked Files using the chosen algorithm. Default is None.</div>	
<b>Alert and Block Settings</b>		
<b>Block Offenders</b>	<input checked="" type="checkbox"/> Checking this option will automatically block hosts that generate a Suricata alert.	
<b>IPS Mode</b>	<div>Legacy Mode</div> <div>Select blocking mode operation. Legacy Mode inspects copies of packets while Inline Mode inserts the Suricata inspection engine into the network stack between the NIC and the OS. Default is Legacy Mode.</div> <div>Legacy Mode uses the PCAP engine to generate copies of packets for inspection as they traverse the interface. Some "leakage" of packets will occur before Suricata can determine if the traffic matches a rule and should be blocked. Inline mode instead intercepts and inspects packets before they are handed off to the host network stack for further processing. Packets matching DROP rules are simply discarded (dropped) and not passed to the host network stack. No leakage of packets occurs with Inline Mode. WARNING: Inline Mode only works with NIC drivers which properly support Netmap! Supported drivers include: bnxt, cc, cxgbe, cxl, em, ena, ice, igb, igc, ix, ixgbe, ixl, lem, re, vmx, vtnet. If problems are experienced with Inline Mode, switch to Legacy Mode instead.</div>	
<b>States entfernen</b>	<input checked="" type="checkbox"/> Checking this option will kill firewall states for the blocked IP. Default is Checked.	
<b>Which IP to Block</b>	<div>SRC</div> <div>Select which IP extracted from the packet you wish to block. Choosing BOTH is suggested, and it is the default value.</div>	
<b>Block On DROP Only</b>	<input checked="" type="checkbox"/> Checking this option will insert blocks only when rule signatures having the DROP action are triggered. When not checked, any rule action (ALERT or DROP) will generate a block of the offending host. Default is Not Checked.	
<b>IP Pass List</b>	<div>nicht gesetzt</div> <div>Choose the Pass List you want this interface to use. Addresses in a Pass List are never blocked. Select "none" to prevent use of a Pass List.</div> <div>The default Pass List adds Gateways, DNS servers, locally-attached networks, the WAN IP, VPNs and VIPs. Create a Pass List with an alias to customize whitelisted IP addresses. This option will only be used when block offenders is on. Choosing "none" will disable Pass List generation.</div>	<div>View List</div>
<b>Enable Passlist Debugging Log</b>	<input type="checkbox"/> Checking this option will enable detailed Passlist operations logging to file /var/log/suricata/suricata_hn150753/passlist_debug.log. Default is Not Checked.	
<b>Performance and Detection Engine Settings</b>		
<b>Run Mode</b>	<div>AutoFP</div> <div>Choose a Suricata run mode setting. Default is "AutoFP" and is the recommended setting for IDS-only and Legacy Blocking Mode. "Workers" uses multiple worker threads, each of which processes the packets it acquires through all the decode and detect modules. "Workers" runmode is preferred for Inline IPS Mode blocking because it offers superior performance in that configuration. "Single" uses only a single thread for all operations, and is intended for use only in testing or development instances.</div>	
<b>AutoFP Scheduler Type</b>	<div>Hash</div> <div>Choose the kind of flow load balancer used by the flow pinned autofp mode. "Hash" assigns the flow to a thread using the 5-7 tuple hash. "IP Pair" assigns the flow to a thread using addresses only. This setting is applicable only when the Run Mode is set to "autofp".</div>	
<b>Max Pending Packets</b>	<div>1024</div> <div>Enter number of simultaneous packets to process. Default is 1024. This controls the number of simultaneous packets the engine can handle. Setting this higher generally keeps the threads more busy. The minimum value is 1 and the maximum value is 65,000. Warning: Setting this too high can lead to degradation and a possible system crash by exhausting available memory.</div>	
<b>Detect-Engine Profile</b>	<div>Hoch</div>	

memory, but it offers lower performance. HIGH consumes a large amount of memory, but it offers the highest performance.

<b>Multi-Pattern Matcher Algorithm</b>	Auto	Choose a multi-pattern matcher (MPM) algorithm. Auto is the default, and is the best choice for almost all systems. Auto will use hyperscan if available.
<b>Single-Pattern Matcher Algorithm</b>	Auto	Choose a single-pattern matcher (SPM) algorithm. Auto is the default, and is the best choice for almost all systems. Auto will use hyperscan if available.
<b>Signature Group Header MPM Context</b>	Auto	Choose a Signature Group Header multi-pattern matcher context. Default is Auto. AUTO means Suricata selects between Full and Single based on the MPM algorithm chosen. FULL means every Signature Group has its own MPM context. SINGLE means all Signature Groups share a single MPM context. Using FULL can improve performance at the expense of significant memory consumption.
<b>Inspection Recursion Limit</b>	3000	Enter limit for recursive calls in content inspection code. Default is 3000. When set to 0 an internal default is used. When left blank there is no recursion limit.
<b>Delayed Detect</b>	<input type="checkbox"/> Suricata will build list of signatures after packet capture threads have started. Default is Not Checked.	
<b>Promiscuous Mode</b>	<input checked="" type="checkbox"/> Suricata will place the monitored interface in promiscuous mode when checked. Default is Checked.	
<b>Interface PCAP Snaplen</b>	1518	Enter value in bytes for the interface PCAP snaplen. Default is 1518. This parameter is only valid when IDS or Legacy Mode IPS is enabled. This value may need to be increased if the physical interface is passing VLAN traffic and expected alerts are not being received.

**Networks Suricata Should Inspect and Protect**

<b>Home Net</b>	only_LAN_HOMENET	<a href="#">View List</a>
Choose the Home Net you want this interface to use.  Default Home Net adds only local networks, WAN IPs, Gateways, VPNs and VIPs. Create an Alias to hold a list of friendly IPs that the firewall cannot see or to customize the default Home Net.		
<b>External Net</b>	all_ExNET	<a href="#">View List</a>
Choose the External Net you want this interface to use.  External Net is networks that are not Home Net. Most users should leave this setting at default. Create a Pass List and add an Alias to it, and then assign the Pass List here for custom External Net settings.		

**Alert Suppression and Filtering**

<b>Alert Suppression and Filtering</b>	Standard	<a href="#">View List</a>
Choose the suppression or filtering file you want this interface to use. Default option disables suppression and filtering.		

**Arguments here will be automatically inserted into the Suricata configuration**

<b>Advanced Configuration Pass-Through</b>	<div></div>
Enter any additional configuration parameters to add to the Suricata configuration here, separated by a newline	

Speichern

## Regelkategorien mit aktiven SIDmgmt.

SchnittstellenGlobal SettingsAktualisierungenAlertsBlocksFilesPass ListsSuppressLogs viewLogs mgmt

SID MgmtSyncIP Lists

LAN EinstellungenLAN CategoriesLAN RegelnLAN Flow/StreamLAN App ParsersLAN VariablesLAN IP Rep

Automatic flowbit resolution

Resolve Flowbits

☒ Auto-enable rules required for checked flowbits  
Default is Checked. Suricata will examine the enabled rules in your chosen rule categories for checked flowbits. Any rules that set these dependent flowbits will be automatically enabled and added to the list of files in the interface rules directory.

View rules

View

Click to view auto-enabled rules required to satisfy flowbit dependencies

Note: Auto-enabled rules generating unwanted alerts should have their GID:SID added to the Suppression List for the interface.

Snort IPS Policy selection

Use IPS Policy

☒ Use rules from one of three pre-defined Snort IPS policies  
Note: You must be using the Snort rules to use this option.  
Selecting this option disables manual selection of Snort rules categories in the list below, although Emerging Threats categories may still be selected if enabled on the Global Settings tab. These will be added to the pre-defined Snort IPS policy rules from the Snort rules set.

IPS Policy Selection

Security

Connectivity blocks most major threats with few or no false positives. Balanced is a good starter policy. It is speedy, has good base coverage level, and covers most threats of the day. It includes all rules in Connectivity. Security is a stringent policy. It contains everything in the first two plus policy-type rules such as Flash in an Excel file. Maximum Detection encompasses vulnerabilities from 2005 or later with a CVSS score of at least 7.5 along with critical malware and exploit kit rules. The Maximum Detection policy favors detection over rated throughput. In some situations this policy can and will cause significant throughput reductions.

IPS Policy Mode

Policy

When Policy is selected, this will automatically change the action for rules in the selected IPS Policy from their default action of alert to the action specified in the policy metadata (typically drop, but may be alert for some policy rules).

Select the rulesets (Categories) Suricata will load at startup

🟢 - Category is auto-enabled by SID Mgmt conf files

🔴 - Category is auto-disabled by SID Mgmt conf files

Select AllUnselect AllSave

AktiviertRuleset:

☐ Snort GPLv2 Community Rules (Talos-certified)

☐ Feodo Tracker Botnet C2 IP Rules

☐ ABUSE.ch SSL Blacklist Rules

Aktiviert	Ruleset: Default Rules	Aktiviert	Ruleset: ET Open Rules	Aktiviert	Ruleset: Snort Text Rules
<input type="checkbox"/>	app-layer-events.rules	<input type="checkbox"/>	emerging-activex.rules	<input checked="" type="checkbox"/>	snort_app-detect.rules
<input type="checkbox"/>	decoder-events.rules	<input type="checkbox"/>	emerging-adware_pup.rules	<input type="checkbox"/>	snort_attack-responses.rules
<input type="checkbox"/>	dhcp-events.rules	<input checked="" type="checkbox"/>	emerging-attack_response.rules	<input type="checkbox"/>	snort_backdoor.rules
<input type="checkbox"/>	dnp3-events.rules	<input checked="" type="checkbox"/>	emerging-botcc.portgrouped.rules	<input type="checkbox"/>	snort_bad-traffic.rules
<input type="checkbox"/>	dns-events.rules	<input checked="" type="checkbox"/>	emerging-botcc.rules	<input type="checkbox"/>	snort_blacklist.rules
<input type="checkbox"/>	files.rules	<input type="checkbox"/>	emerging-chat.rules	<input type="checkbox"/>	snort_botnet-cnc.rules
<input type="checkbox"/>	ftp-events.rules	<input type="checkbox"/>	emerging-ciarmy.rules	<input type="checkbox"/>	snort_browser-chrome.rules

Schnittstellen

Global Settings

Aktualisierungen

Alerts

Blocks

Files

Pass Lists

Suppress

Logs View

Logs Mgmt

SID Mgmt

Sync

IP Lists

LAN Einstellungen

LAN Categories

LAN Regeln

LAN Flow/Stream

LAN App Parsers

LAN Variables

LAN IP Rep

Available Rule Categories

Kategorie

emerging-scan.rules

View All

Select the rule category to view and manage.

Rule Signature ID (SID) Enable/Disable Overrides

SID Actions

Apply

Reset All

Reset Current

Disable All

Enable All

When finished, click APPLY to save and send any SID state/action changes made on this tab to Suricata.

Rules View Filter

Rule Signature ID (SID) Enable/Disable Overrides

Legend:

Default Enabled

Enabled by user

Auto-enabled by SID Mgmt

Action/content modified by SID Mgmt

Rule action is alert

Rule contains noalert option

Default Disabled

Disabled by user

Auto-disabled by SID Mgmt

Rule action is drop

State	Aktion	GID	SID	Proto	Quelle	SPort	Ziel	DPort	Nachricht
		1	2010371	tcp	SEXTERNAL_NET	any	\$HOME_NET	any	ET SCAN Amap TCP Service Scan Detected
		1	2010372	udp	SEXTERNAL_NET	any	\$HOME_NET	any	ET SCAN Amap UDP Service Scan Detected
		1	2008414	udp	SEXTERNAL_NET	any	\$HOME_NET	69	ET SCAN Cisco Torch TFTP Scan
		1	2010642	tcp	SEXTERNAL_NET	any	\$HOME_NET	21	ET SCAN Multiple FTP Root Login Attempts from Single Source - Possible Brute Force Attempt
		1	2010643	tcp	SEXTERNAL_NET	any	\$HOME_NET	21	ET SCAN Multiple FTP Administrator Login Attempts from Single Source - Possible Brute Force Attempt
		1	2000575	icmp	SEXTERNAL_NET	any	\$HOME_NET	any	ET SCAN ICMP PING IPTools
		1	2008560	udp	SEXTERNAL_NET	any	\$HOME_NET	1434	ET SCAN NNG MS02-039 Exploit False Positive Generator - May Conceal A Genuine Attack

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

Services / Suricata / SID Management

SchnittstellenGlobal SettingsAktualisierungenAlertsBlocksFilesPass ListsSuppressLogs ViewLogs Mgmt

SID MgmtSyncIP Lists

### Allgemeine Einstellungen

**Enable Automatic SID State Management** ☒ Enable automatic management of rule state and content using SID Management Configuration Lists. Default is Not Checked.

When checked, Suricata will automatically enable/disable/modify text rules upon each update using criteria specified in SID Management Configuration Lists. The supported configuration list format is the same as that used by PulledPork and Oinkmaster. See the included sample conf lists for usage examples. Either upload existing configurations to the firewall or create new ones by clicking ADD below.

### SID Management Configuration Lists

SID Mods List Name	Last Modified Time	List Actions
disablesid-sample.conf	Jan-27 2025 4:25 pm	
icmp	Mar-05 2025 12:41 am	
enablesid-sample.conf	Jan-27 2025 4:25 pm	
alert ->drop modi	Mar-05 2025 12:36 am	
modify von ex nets zu any	Mar-04 2025 1:02 am	
example modifysid.conf	Mar-04 2025 5:48 pm	
emerging rules add list + icmp	Mar-05 2025 12:42 am	

+ Hinzufügen
Importieren
Runterladen

### Interface SID Management List Assignments

Neu aufbauen	Schnittstelle	SID State Order	Enable SID List	Disable SID List	Modify SID List	Drop SID List	Reject SID List
<input type="checkbox"/>	WAN	Disable, Er ▼	Kein ▼	Kein ▼	Kein ▼	N/A	N/A
<input type="checkbox"/>	LAN	Disable, Er ▼	emerging rules add lis ▼	Kein ▼	Kein ▼	emerging rules add li ▼	N/A
<input type="checkbox"/>	OPT1	Disable, Er ▼	Kein ▼	Kein ▼	Kein ▼	Kein ▼	N/A
<input type="checkbox"/>	OPT2	Disable, Er ▼	icmp ▼	Kein ▼	alert ->drop modi ▼	N/A	N/A

Speichern
Remember to save changes before exiting this page

## Anlage 8 – Testergebnisse auf das LAN-Netz IPS

Services / Suricata / Alerts

SchnittstellenGlobal SettingsAktualisierungenAlertsBlocksFilesPass ListsSuppressLogs ViewLogs Mgmt

SID MgmtSyncIP Lists

Alert Log View Settings

Instance to View

(LAN) PFS IPS LAN

Choose which instance alerts you want to inspect.

Save or Remove Logs

Runterladen

All alert log files for selected interface will be downloaded

Zurücksetzen

Clear the currently active Alerts log file

Save Settings

Speichern

Save auto-refresh and view settings

Aktualisieren

Default is ON

313

Number of alerts to display. Default is 250

Alert Log View Filter

Last 313 Alert Entries. (Most recent entries are listed first)

Note: Alerts triggered by DROP rules that resulted in dropped (blocked) packets are shown with highlighted rows below.

Datum	Aktion	Pri	Proto	Class	Src	SPort	Dst	DPort	GID:SID	Beschreibung
03/11/2025 13:48:52		2	UDP	Potentially Bad Traffic	192.168.50.50	60536	192.168.50.1	53	1:2002752	ET INFO Reserved Internal IP Traffic
03/11/2025 13:48:48		2	TCP	Potentially Bad Traffic	192.168.80.80	63565	192.168.50.50	3306	1:2010937	ET SCAN Suspicious inbound to mySQL port 3306
03/11/2025 13:48:48		2	TCP	Attempted Information Leak	192.168.80.80	63565	192.168.50.50	110	1:2009582	ET SCAN NMAP -sS window 1024
03/11/2025 13:48:48		2	TCP	Potentially Bad Traffic	192.168.80.80	63565	192.168.50.50	110	1:2002752	ET INFO Reserved Internal IP Traffic

Services / Suricata / Blocked Hosts

SchnittstellenGlobal SettingsAktualisierungenAlertsBlocksFilesPass ListsSuppressLogs ViewLogs Mgmt

SID MgmtSyncIP Lists

Blocked Hosts Log View Settings

Save or Remove Hosts

Runterladen

All blocked hosts will be saved

Zurücksetzen

All blocked hosts will be cleared

Save Settings

Speichern

Save auto-refresh and view settings

Aktualisieren

Default is ON

500

Number of blocked entries to view. Default is 500

Last 500 Hosts Blocked by Suricata

Note: Only blocked IP addresses from Legacy Mode interfaces are shown! For inline IPS mode interfaces, dropped IP addresses are highlighted on the ALERTS tab.

Blocked IP	Block Date/Time	Block Alert Description	Block Rule GID:SID	Remove Block
192.168.80.80	03/11/2025 13:48:48	ET INFO Reserved Internal IP Traffic	1:2002752	
	03/05/2025 00:56:57	PROTOCOL-ICMP Squid Pinger IPv6 denial of service attempt	1:36650	
	03/05/2025 00:56:28	PROTOCOL-ICMP PING undefined code	1:365	
	03/05/2025 00:56:11	ET SCAN NMAP -sS window 1024	1:2009582	
	03/05/2025 00:55:56	ET SCAN Suspicious inbound to Oracle SQL port 1521	1:2010936	
	03/05/2025 00:55:41	ET SCAN Suspicious inbound to MSSQL port 1433	1:2010935	
	03/05/2025 00:55:30	ET SCAN Suspicious inbound to PostgreSQL port 5432	1:2010939	
	03/05/2025 00:55:09	ET SCAN NMAP -sS window 1024	1:2009582	
	03/05/2025 00:52:13	ET SCAN NMAP -sS window 1024	1:2009582	
	03/05/2025 00:48:56	ET INFO Reserved Internal IP Traffic	1:2002752	
	03/05/2025 00:31:41	ET INFO Reserved Internal IP Traffic	1:2002752	
	03/05/2025 00:08:44	ET INFO Reserved Internal IP Traffic	1:2002752	
	03/05/2025 00:02:42	ET INFO Reserved Internal IP Traffic	1:2002752	
	03/04/2025 20:50:29	PROTOCOL-ICMP Squid Pinger IPv6 denial of service attempt	1:36650	
	03/04/2025 15:38:37	PROTOCOL-SNMP AgentX/tcp request	1:1421	
	03/04/2025 15:13:57	ET SCAN Suspicious inbound to mySQL port 3306	1:2010937	
	03/04/2025 15:10:39	ET SCAN Suspicious inbound to mySQL port 3306	1:2010937	
	03/04/2025 15:03:06	ET SCAN Suspicious inbound to mySQL port 3306	1:2010937	
	03/04/2025 14:57:23	ET SCAN Suspicious inbound to mySQL port 3306	1:2010937	

1 host IP address is currently being blocked.

Services / Suricata / Logs View

Schnittstellen Global Settings Aktualisierungen Alerts Blocks Files Pass Lists Suppress Logs View Logs Mgmt

SID Mgmt Sync IP Lists

**Logs Browser Selections**

Instance to View (LAN) PFS IPS LAN  
Choose which instance logs you want to view.

Log File to View eve.json  
Choose which log you want to view.

Status/Result File successfully loaded.  
Log File Path: /var/log/suricata/suricata\_hm150753/eve.json  
[Aktualisieren](#)

**Log Contents**

Normale Ansicht Dynamische Ansicht Zusammenfassung

**Erweiterter Protokollfilter**

Die letzten 500 Firewall Protokolleinträge. (Größtmögliche 500)

Aktion	Zeitpunkt	Schnittstelle	Regel	Quelle	Ziel	Protokoll
✗	Mar 11 13:51:19	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63567	192.168.50.50:5190	TCP-S
✗	Mar 11 13:51:19	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63565	192.168.50.50:5190	TCP-S
✗	Mar 11 13:51:19	WAN	Default deny rule IPv4 (1000000103)			IGMP
✗	Mar 11 13:51:19	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63567	192.168.50.50:62078	TCP-S
✗	Mar 11 13:51:19	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63565	192.168.50.50:62078	TCP-S
✗	Mar 11 13:51:19	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63567	192.168.50.50:1166	TCP-S
✗	Mar 11 13:51:19	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63565	192.168.50.50:1166	TCP-S
✗	Mar 11 13:51:19	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63567	192.168.50.50:4129	TCP-S
✗	Mar 11 13:51:19	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63565	192.168.50.50:4129	TCP-S
✗	Mar 11 13:51:19	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63567	192.168.50.50:3690	TCP-S
✗	Mar 11 13:51:19	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63565	192.168.50.50:3690	TCP-S
✗	Mar 11 13:51:18	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63567	192.168.50.50:8042	TCP-S
✗	Mar 11 13:51:18	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63798	192.168.50.50:23	TCP-S
✗	Mar 11 13:51:18	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63565	192.168.50.50:8042	TCP-S
✗	Mar 11 13:51:18	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63567	192.168.50.50:981	TCP-S
✗	Mar 11 13:51:18	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63565	192.168.50.50:981	TCP-S
✗	Mar 11 13:51:18	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63567	192.168.50.50:6005	TCP-S
✗	Mar 11 13:51:18	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63565	192.168.50.50:6005	TCP-S
✗	Mar 11 13:51:18	WAN	Default deny rule IPv4 (1000000103)			IGMP
✗	Mar 11 13:51:18	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63567	192.168.50.50:9207	TCP-S
✗	Mar 11 13:51:18	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63565	192.168.50.50:9207	TCP-S
✗	Mar 11 13:51:18	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63567	192.168.50.50:6566	TCP-S
✗	Mar 11 13:51:17	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63565	192.168.50.50:6566	TCP-S
✗	Mar 11 13:51:17	OPT2	Block snort2c hosts (1000000109)	192.168.80.80:63567	192.168.50.50:1271	TCP-S



## Anlage 8 – Testergebnisse auf das OPT1-Netz IDS

The screenshot displays a network security monitoring interface. On the left, a table lists test results for various services and protocols. On the right, a web browser shows the 'Blocked Hosts' page of a pfSense interface, which includes settings for blocked hosts and a list of blocked IP addresses.

Date/Time	Alert ID	Protocol	Service	Source IP	Destination IP	Port	Score	Rule	Action
03/11/2025 13:59:41	1	TCP	Web Application Attack	192.168.80.80	192.168.60.60	5357	1.2009358	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Blocked
03/11/2025 13:59:41	3	TCP	Generic Protocol Command Decode	192.168.80.80	192.168.60.60	5357	1.2260002	SURICATA Applayear Detect protocol only one direction	Blocked
03/11/2025 13:59:41	1	TCP	Web Application Attack	192.168.80.80	192.168.60.60	5357	1.2009358	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Blocked
03/11/2025 13:59:41	1	TCP	Web Application Attack	192.168.80.80	192.168.60.60	5357	1.2009358	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Blocked
03/11/2025 13:59:41	1	TCP	Web Application Attack	192.168.80.80	192.168.60.60	5357	1.2009358	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Blocked
03/11/2025 13:59:41	1	TCP	Web Application Attack	192.168.80.80	192.168.60.60	5357	1.2009358	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Blocked
03/11/2025 13:59:41	1	TCP	Web Application Attack	192.168.80.80	192.168.60.60	5357	1.2009358	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Blocked
03/11/2025 13:59:41	1	TCP	Web Application Attack	192.168.80.80	192.168.60.60	5357	1.2009358	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Blocked

The right side of the screenshot shows the pfSense 'Blocked Hosts' page. It includes a 'Blocked Hosts Log View Settings' section with options to 'Save or Remove Hosts' and 'Save Settings'. Below this is a table titled 'Last 500 Hosts Blocked by Suricata' with columns for 'Blocked IP', 'Block Date/Time', 'Block Alert Description', 'Block Rule GID:SID', and 'Remove Block'. The table currently shows no hosts are blocked.

## Anlage 9 – Beispiel Snort icmp Rules

(und wieviel der Regeln standardmäßig deaktiviert sind)

hliessen

Schliessen

Rule file

```
# Copyright 2001-2025 Sourcefire, Inc. All Rights Reserved.
#
# This file contains (i) proprietary rules that were created, tested and certified by
# Sourcefire, Inc. (the "VRT Certified Rules") that are distributed under the VRT
# Certified Rules License Agreement (v 2.0), and (ii) rules that were created by
# Sourcefire and other third parties (the "GPL Rules") that are distributed under the
# GNU General Public License (GPL), v2.
#
# The VRT Certified Rules are owned by Sourcefire, Inc. The GPL Rules were created
# by Sourcefire and other third parties. The GPL Rules created by Sourcefire are
# owned by Sourcefire, Inc., and the GPL Rules not created by Sourcefire are owned by
# their respective creators. Please see http://www.snort.org/snort-team/ for a
# list of third party owners and their respective copyrights.
#
# In order to determine what rules are VRT Certified Rules or GPL Rules, please refer
# to the VRT Certified Rules License Agreement (v2.0).
#
#-----
# PROTOCOL-ICMP RULES
#-----

# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP record route rr denial of service attempt"; ipopts:rr; icode:0; ityp
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP Microsoft Windows IPv6 DNSSEC option record denial of service attempt";
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP Microsoft Windows IPv6 stack remote execution attempt"; itype:134; ic
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP Squid Pinger IPv6 denial of service attempt"; icode:0; itype:>160; m
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP Squid Pinger IPv6 denial of service attempt"; icode:0; itype:11<>127
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP FreeBSD rtsock dname_labeldec stack buffer overflow attempt"; itype:
# alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"PROTOCOL-ICMP Unusual Microsoft Windows 7 Ping detected"; icode:0; itype:8; dsiz
# alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"PROTOCOL-ICMP Unusual PING detected"; icode:0; itype:8; fragbits:!!M; content:!!ABC
# alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"PROTOCOL-ICMP Unusual Microsoft Windows Ping detected"; icode:0; itype:8; dsiz
# alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"PROTOCOL-ICMP Unusual L3retriever Ping detected"; icode:0; itype:8; dsiz:>32; con
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP IPv6 0xfacebabe ICMP ping attempt"; itype:128; icode:0; icmp_id:6420
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP Truncated ICMPv6 denial of service attempt"; content:"|60 58 58 5
# alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP invalid ICMPv6 header attempt"; dsiz:32; content:"|3A 01 66 00 66 00
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP IPv6 multicast neighbor add attempt"; itype:135; icode:0; reference:
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP IPv6 multicast neighbor delete attempt"; itype:132; icode:0; referen
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP IPv6 MLD multicast listener query attempt"; itype:130; icode:0; refe
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP IPv6 invalid router advertisement attempt"; itype:134; icode:0; cont
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP IPv6 0xdeadbeef ICMP ping attempt"; itype:128; icode:0; icmp_id:5700
```



## Anlage 10 – Global Settings, Regelauswahl

Schnittstellen	Global Settings	Aktualisierungen	Alerts	Blocks	Files	Pass Lists	Suppress	Logs View	Logs Mgmt
SID Mgmt	Sync	IP Lists							

Please Choose The Type Of Rules You Wish To Download

Install ETOpen Emerging Threats rules	<input checked="" type="checkbox"/> ETOpen is a free open source set of Suricata rules whose coverage is more limited than ETPro.	<input type="checkbox"/> Use a custom URL for ETOpen downloads
Enabling the custom URL option will force the use of a custom user-supplied URL when downloading ETOpen rules.		
Install ETPro Emerging Threats rules	<input type="checkbox"/> ETPro for Suricata offers daily updates and extensive coverage of current malware threats.	<input type="checkbox"/> Use a custom URL for ETPro rule downloads
The ETPro rules contain all of the ETOpen rules, so the ETOpen rules are not required and are disabled when the ETPro rules are selected. <a href="#">Sign Up for an ETPro Account</a> . Enabling the custom URL option will force the use of a custom user-supplied URL when downloading ETPro rules.		
Install Snort rules	<input checked="" type="checkbox"/> Snort free Registered User or paid Subscriber rules <a href="#">Sign Up for a free Registered User Rules Account</a> <a href="#">Sign Up for paid Snort Subscriber Rule Set (by Talos)</a>	<input type="checkbox"/> Use a custom URL for Snort rule downloads
Enabling the custom URL option will force the use of a custom user-supplied URL when downloading Snort Subscriber rules.		
Snort Rules Filename	<input type="text" value="snortrules-snapshot-29200.tar.gz"/> Enter the rules tarball filename (filename only, do not include the URL.) Example: snortrules-snapshot-29200.tar.gz DO NOT specify a Snort3 rules file! Snort3 rules are incompatible with Suricata and will break your installation!	
Snort Oinkmaster Code	<input type="text" value="b37a6ca77481526d27a6c2a6e3372abb806ca4eb"/> Obtain a snort.org Oinkmaster code and paste it here.	
Install Snort GPLv2 Community rules	<input checked="" type="checkbox"/> The Snort Community Ruleset is a GPLv2 Talos-certified ruleset that is distributed free of charge without any Snort Subscriber License restrictions.	<input type="checkbox"/> Use a custom URL for Snort GPLv2 rule downloads
This ruleset is updated daily and is a subset of the subscriber ruleset. If you are a Snort Subscriber Rules customer (paid subscriber), the community ruleset is already built into your download of the Snort Subscriber rules, and there is no benefit in adding this rule set separately.		
Install Feodo Tracker Botnet C2 IP rules	<input checked="" type="checkbox"/> The Feodo Botnet C2 IP Ruleset contains Dridex and Emotet/Heodo botnet command and control servers (C&Cs) tracked by Feodo Tracker.	
Install ABUSE.ch SSL Blacklist rules	<input checked="" type="checkbox"/> The ABUSE.ch SSL Blacklist Ruleset contains the SSL cert fingerprints of all SSL certs blacklisted by ABUSE.ch.	
Hide Deprecated Rules Categories	<input checked="" type="checkbox"/> Hide deprecated rules categories in the GUI and remove them from the configuration. Default is Not Checked.	
Download Extra Rules	<input type="checkbox"/> Download Extra Rules Download extra rules file or tar.gz archive with rules. If "Check MD5" is set, the code will assume a matching filename exists at the same URL with an additional extension of ".md5".	

Rules Update Settings

## 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](https://forum.netgate.com) zu bekommen, teilweise von den Entwicklern von pfSense selbst- Stand 11.03.2025

[Forum.netgate.com](https://forum.netgate.com)

 [https://forum.netgate.com/topic/154152/suricata-ids-and-ips?\\_=1741698840615](https://forum.netgate.com/topic/154152/suricata-ids-and-ips?_=1741698840615)

