INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Bachelorarbeit

Evaluierung und Migration der Secomat Funktionalität auf PfSense Firewalls

Benjamin Hacker

INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Bachelorarbeit

Evaluierung und Migration der Secomat Funktionalität auf PfSense Firewalls

Benjamin Hacker

Aufgabensteller:	Prof. Dr. Helmut Reiser
Betreuer:	Tobias Appel Bernhard Schmidt Helmut Tröbs Claus Wimmer
Abgabetermin:	07. November 2018

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 07. November 2018

. . . . . .
(Unterschrift des Kandidaten)

Abstract

Die Sicherheit vernetzter Systeme ist in heutiger Zeit von wesentlicher Bedeutung für den privaten und öffentlichen Bereich. Auf der einen Seite ermöglichen vernetzte Systeme einen schnellen Austausch von Informationen in Wissenschaft und Wirtschaft. Auf der anderen Seite werden diese auch immer öfter zu Zielen von Angriffen. Im Münchner Wissenschaftsnetz (MWN) werden Pakete von Hosts mit öffentlichen und privaten IP-Adressen geroutet. Um den privat-adressierten Hosts im MWN die Kommunikation mit Zielen außerhalb des MWN zu ermöglichen, wird der Secomat eingesetzt. Der Secomat ist ein NAT-Gateway mit mehreren Sicherheitsfunktionen.

Der Secomat schützt die Hosts mit einer privaten IP-Adresse im MWN vor Angriffen von Hosts außerhalb des MWN und in umgekehrter Richtung. Die Angriffe von außerhalb des MWN werden im Secomat durch eine Firewall blockiert. Kommt es zu Angriffen über den Secomat auf Ziele im Internet, so werden diese durch ein Intrusion Detection System erkannt und die IP-Adresse des Angreifers durch ein Intrusion Prevention System automatisch gesperrt. Die Nutzer gesperrter IP-Adressen werden durch eine Statusseite über ihre Sperrung informiert. Bleibt ein Host über einen längeren Zeitraum kontinuierlich gesperrt, so wird der zuständige Netzverantwortliche durch eine E-Mail darüber informiert.

Innerhalb des MWN schützen Organisationen und Institute ihre lokalen Netze mit pfSense Firewalls. Die pfSense Firewall schützt die Hosts im lokalen Netz vor Angriffen aus dem MWN und dem weltweiten Internet. Werden jedoch Angriffe aus dem lokalen Netz auf Ziele im MWN oder dem Internet versucht, werden diese nicht durch die pfSense Firewall erkannt und können daher auch nicht blockiert werden. Aus diesem Grund wird in dieser Arbeit evaluiert, ob die Funktionen des Secomats auf die pfSense Firewall migriert werden können, um die Sicherheit der Hosts im MWN und weltweiten Internet zu erhöhen und Angriffe bereits am lokalen Netzrand zu stoppen.

Inhaltsverzeichnis

1. Einleitung

In den vergangenen Jahren wurde immer wieder über Distributed Denial-of-Service (DDoS) Angriffe in den Medien berichtet [Wö14][Woo16][Wes18]. Die DDoS Angriffe werden durch Botnetze durchgeführt, deren Basis i. d. R. durch Schadsoftware infizierte Computer sind. Je mehr Bots Teil eines Botnetzes sind, desto mehr Schaden kann das Botnetz anrichten. Durch DDoS Angriffe wird die Verfügbarkeit von Diensten für berechtigte Nutzer eingeschränkt. Durch den kombinierten Einsatz eines Intrusion Detection Systems (IDS) und eines Intrusion Prevention Systems (IPS) können derartige Angriffe erkannt und erschwert werden.

In München wird das Münchner Wissenschaftsnetz (MWN) durch das Leibniz Rechenzentrum (LRZ) betrieben und verwaltet. Über Wireless Access Points können Studenten und wissenschaftliche Mitarbeiter ihre eigenen Computer mit dem MWN verbinden. Für die Sicherheit der Geräte sind die Eigentümer selbst verantwortlich. Es wird empfohlen, die Computer durch regelmäßige Updates und den Einsatz einer Firewall vor Angriffen im MWN zu schützen. Durch veraltete Software und unzureichend konfigurierte Firewalls kann es zur Ausbreitung von Schadsoftware kommen. Die infizierten Computer können dann für Denial-Of-Service Angriffe missbraucht werden.

Zum Schutz der Hosts im weltweiten Internet vor Hosts im MWN und der Hosts im MWN vor Hosts im weltweiten Internet, wird der Secomat eingesetzt. Der Secomat ist ein NAT-Gateway und ermöglicht den Hosts im MWN mit einer privaten IP-Adresse die Kommunikation mit Hosts außerhalb des MWN. Der Secomat schützt die privat-adressierten Hosts im MWN mit einer statefull Firewall. Dadurch werden nur Pakete an diese Hosts weitergeleitet, die einem bestehenden Verbindungsstatus zugeordnet werden können. Um Hosts außerhalb des MWN zu schützen, sind im Secomat zusätzlich zur Firewall auch ein IDS und ein IPS aktiv. Das IDS überwacht die Paketraten und protokolliert Überschreitungen. Kommt es gehäuft zu Verstößen durch einen Host im MWN, so wird dessen IP-Adresse durch das IPS gesperrt. Daraufhin wird die gesamte Kommunikation des gesperrten Hosts mit Zielen außerhalb des MWN durch die Firewall blockiert. Sobald der Nutzer einer gesperrten IP-Adresse eine Webseite im Internet aufrufen möchte, wird ihm eine Statusseite angezeigt. Die Statusseite informiert den Nutzer über die Sperrung seiner IP-Adresse. Durch die Sperrung von IP-Adressen werden Angriffe über den Secomat auf Ziele im Internet verhindert.

1.1. Motivation

Das LRZ stellt Instituten und Organisationen das Firewallsystem pfSense zum Schutz ihrer eigenen Netze zur Verfügung. Die pfSense schützt die Geräte im lokalen Netz durch eine Firewall vor Angriffen aus dem MWN oder dem Internet. Wird jedoch ein infiziertes Gerät in das lokale Netz eingebunden, so bietet die pfSense keinen Schutz und es kann zur Ausbreitung von Schadsoftware innerhalb des lokalen Netzes kommen. Die infizierten Geräte können versuchen Geräte außerhalb des lokalen Netzes über die pfSense anzugreifen oder zu infizieren.

Um in Zukunft derartige Angriffe nach dem Vorbild des Secomats bereits am lokalen Netzrand auf der pfSense Firewall erkennen und verhindern zu können, wird in dieser Arbeit untersucht, ob die Secomat Funktionalität auf die pfSense Firewall migriert werden kann.

1.2. Struktur der Arbeit

Im Kapitel Grundlagen werden die Funktionen des Secomats und der pfSense Firewall vorgestellt. Das Kapitel Grundlagen endet mit der Erklärung der Einsatzszenarien, bei denen Angriffe über die pfSense Firewall auf Ziele im MWN und im weltweiten Internet vorkommen können. Daran schließt sich die Anforderungsanalyse an, bei der Anforderungen anhand der Funktionen des Secomats aufgestellt werden. Im Kapitel Migration wird auf die Realisierung der, in der Anforderungsanalyse erarbeiteten Anforderungen, eingegangen. Im darauf folgenden Kapitel Evaluation werden die durchgeführten Tests und resultierenden Testergebnisse vorgestellt.

2. Grundlagen

In diesem Kapitel werden die Funktionen des Secomats und der pfSense Firewall vorgestellt.

2.1. Secomat

Das Intrusion Detection System (IDS) und das Intrusion Prevention System (IPS) des Secomats sind das Ergebnis eines Forschungsprojekts aus dem Jahr 2005 [DF06]. Bis dahin agierte der Vorgänger der Secomats lediglich als NAT-Gateway mit Firewall [LR12]. Die Implementierung eines IDS und IPS wurde notwendig, da das manuelle Sperren von IP-Adressen oder Port-Nummer nicht mehr ausreichte, um Angriffe zu unterbinden. Außerdem stieg der Managementaufwand aufgrund immer komplexerer Angriffsverfahren und Ausbreitungswege.

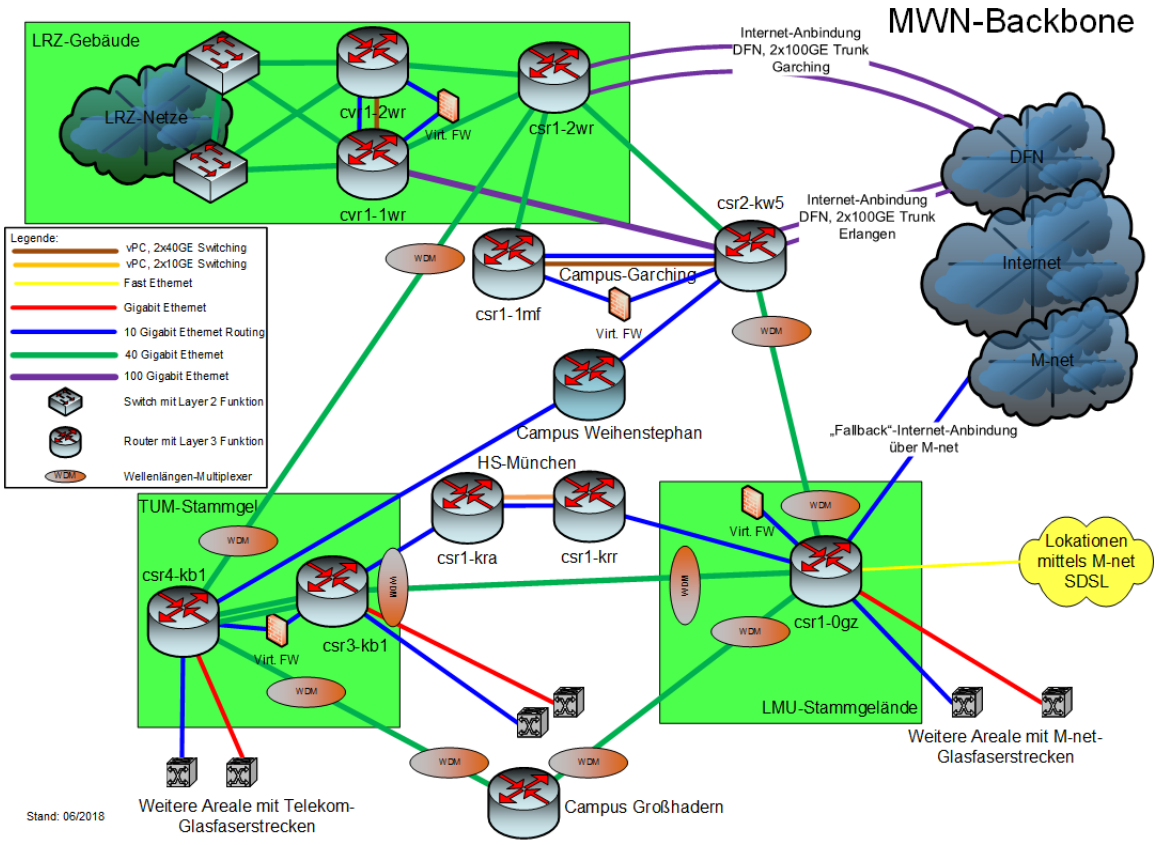

Abbildung 2.1.: Überblick über das Münchner Wissenschaftsnetz - Quelle: [LR18]

2.1.1. Münchner Wissenschaftsnetz

Der Secomat wird im Münchner Wissenschaftsnetz (MWN) eingesetzt. Das MWN ist ein Metropolitan Area Network und erstreckt sich über ganz München. Über das MWN sind diverse Forschungseinrichtungen miteinander verbunden, so u.a. Standorte der Ludwig-Maximilians-Universität, der Technischen Universität München, des Leibniz Rechenzentrums, mehrere Studentenwohnheime und weitere Einrichtungen. Das MWN wird durch das LRZ verwaltet und betrieben. Die Bandbreiten reichen von 100 Mbit/s bis zu 100 Gbit/s [LR18]. Die Sicherheit wird mithilfe von Port-Sperren am Übergang zum weltweiten Internet erhöht [LRZ16]. Innerhalb des MWN werden Pakete mit privaten und öffentlichen IP-Adressen geroutet. Alle IPv4 Pakete, deren Quell-IP eine private IP-Adresse ist und deren Ziel-IP nicht im MWN liegt, werden durch Policy-basiertes Routing zum Secomat geleitet [Gol16]. Dadurch kann mit dem Secomat die Kommunikation über IPv4 zwischen Hosts mit einer privaten IP-Adresse im MWN und Hosts außerhalb des MWN überwacht und gegebenenfalls unterbunden werden. Die Hosts im MWN können neben einer IPv4 Adresse auch eine IPv6 Adresse beziehen. Die Kommunikation über IPv6 wird nicht durch den Secomat überwacht und kann daher auch nicht durch ihn unterbunden werden.

2.1.2. Netfilter

Auf den Cluster-Knoten des Secomats ist das Betriebssystem SuSe Linux Enterprise Server 11 installiert [Gol16]. Dabei handelt es sich um eine Linux Distribution die das Kernel-Subsystem Netfilter enthält. Im Secomat wird Netfilter für die Network Address Translation, die Firewall und das Intrusion Detection System genutzt. Im Secomat erfolgt die Konfiguration von Netfilter anhand der Bash-Scripte setupnat und natomat.

Die Paketverarbeitung in Netfilter ist mit Tables, Hooks, Chains und Filterrules strukturiert [Pur04]. Im Secomat werden insbesondere die Tables raw, nat und filter für die Paketverarbeitung genutzt und durch das setupnat Script konfiguriert. Die Tables enthalten Hooks, die den Zeitpunkt der Verarbeitung festlegen. Die Hooks werden in der folgenden Reihenfolge abgearbeitet: PREROUTING, INPUT, FORWARD, OUTPUT, POSTROUTING. Jede Table enthält unterschiedliche Hooks, abhängig von ihrer Aufgabe. Jeder Hook ist fest mit einer gleichnamigen Chain verbunden. In eine Chain können entweder Filterrules zur Paketverarbeitung oder neue Chains eingefügt werden. Um die Übersichtlichkeit der Konfiguration zu erhöhen, wurden beim Secomat neue Chains definiert, die in die Chains der Hooks eingefügt sind. In die neuen Chains wurden dann die Filterrules eingefügt.

Jede Chain hat eine Default Policy, die aktiv wird, wenn ein Paket auf keine Regel in einer Chain zutrifft. Bei den Chains der Hooks ist die Standard Default Policy ACCEPT, d.h. das Paket wird durch Netfilter an seinen Ziel-Ort weiter geschickt. Beim Secomat wurde die Default Policy der Chains nicht verändert. Bei neu definierten Chains ist die Default Policy immer Return, d.h. es wird zur vorherigen Chain zurückgesprungen. Die Pakete werden in Netfilter von der ersten bis zur letzten Filterrule einer Chain abgeglichen. Wird von einer Chain auf eine andere referenziert, so werden erst die Filterrules der referenzierten Chain abgearbeitet und anschließend die Filterrules, die auf die referenzierte Chain folgen. Trifft ein Paket auf die Filterkriterien einer Filterrule zu, so wird dessen Maßnahme auf das Paket angewendet. In den Filterrules kann als Maßnahme ACCEPT, DROP, REJECT oder

RETURN angegeben werden. Bei ACCEPT wird das Paket an seinen Ziel-Ort weitergeschickt. Mit DROP wird das Paket verworfen. Bei REJECT wird das Paket verworfen und der Sender des Pakets darüber informiert. Mit RETURN wird die Default Policy der Chain angewendet.

Beim Secomat werden in Netfilter mehrere Platzhalter mit Ipset zum Speichern von gesperrten IP-Adressen angelegt. Durch Ipset können u.a. mehrere IP-Adressen in eine Variable gespeichert werden. In den Filterrules kann die Variable anstelle einer Konstanten, wie einer IP-Adresse, angegeben werden. Dadurch gilt die Filterrule für alle IP-Adressen in der referenzierten Variablen.

2.1.3. Network Address Translation

Source Network Address Translation

Um den Hosts im MWN mit einer privaten IP-Adresse die Kommunikation mit Hosts im weltweiten Internet zu ermöglichen, wird Source Network Address Translation (SNAT) genutzt. Diese Funktion hat der Secomat von seiner Vorgängerversion, dem NAT-o-MAT, übernommen. Bei SNAT wird die private Quell-IP von Paketen aus dem MWN durch eine öffentliche IP-Adresse ersetzt. Dies ist notwendig, da Pakete mit privaten IP-Adressen im weltweiten Internet nicht geroutet werden. Dem Secomat stehen mehrere Bereiche mit öffentlichen IP-Adressen für SNAT zur Verfügung. Nachdem die private Quell-IP durch eine der öffentlichen IP-Adressen ersetzt wurde, werden die Pakete über das externe Interface zurück in das MWN geschickt. Da die Pakete nun keine private Quell-IP mehr haben, können diese das MWN verlassen und an ihr eigentliches Ziel im weltweiten Internet geroutet werden. Der Host im Internet schickt die Antwort-Pakete zurück zum Secomat. Der Secomat ordnet die Pakete auf Basis einer NAT-Tabelle dem ursprünglich anfragenden Host im MWN zu und ändert die Ziel-IP der Antwort-Pakete entsprechend. Anschließend werden die Pakete an den Host im MWN weitergeschickt.

Destination Network Address Translation

Im Secomat wird Destination Network Address Translation (DNAT) genutzt, um gesperrte Nutzer automatisch über ihre Sperrung zu informieren. Dazu sind im Secomat DNAT-Regeln definiert, die HTTP-Requests von gesperrten Nutzern, also TCP-Pakete mit Ziel-Port 80 und gesperrter Quell-IP, an das interne Interface umleiten. Dazu wird die Ziel-IP der TCP-Pakete durch die IP-Adresse des internen Interface ersetzt, an dem sie ankamen. In Folge wird der HTTP-Request an den lokalen Webserver des verarbeitenden Secomat-Knotens geschickt. Der Webserver antwortet auf den HTTP-Request mit einem HTTP-Response. Bei den TCP-Paketen des HTTP-Response wird die Quell-IP durch die Ziel-IP des ursprünglichen HTTP-Requests ersetzt und weiter an den gesperrten Host im MWN geschickt.

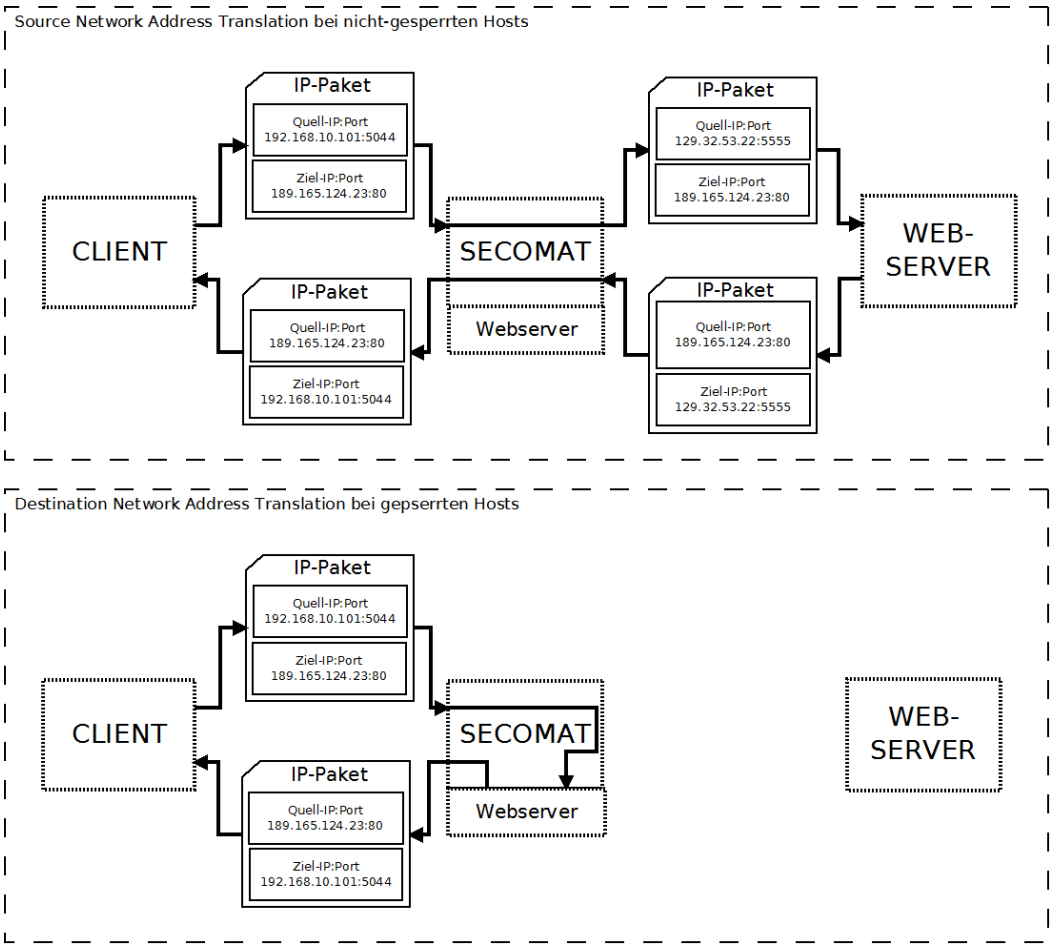

Abbildung 2.2.: Network Address Translation im Secomat

2.1.4. Firewall

Durch die Firewall wird der Paketfluss zwischen dem privat-adressierten MWN und dem weltweiten Internet gefiltert. In Netfilter des Secomats wird das Connection Tracking genutzt, damit nur Pakete in das MWN gelangen können, für die zuvor eine Anfrage geschickt wurde. Dazu wird für Pakete aus dem MWN, die ins weltweite Internet geschickt werden, ein Verbindungsstatus gespeichert. Alle Pakete, die den Secomat aus dem Internet erreichen, werden durch Filterregeln mit den gespeicherten Verbindungen verglichen. Pakete, die keinem gespeicherten Verbindungsstatus zugeordnet werden können, werden durch die Firewall verworfen. Das Connection Tracking wird für Pakete deaktiviert, deren Quell-IP gesperrt ist, da diese durch nachfolgende Filterregeln verworfen werden.

Alle Pakete, deren private Quell-IP nicht aus einem vergebenen Subnetz im MWN stammt, werden durch die Firewall verworfen. Dadurch wird verhindert, dass Pakete, deren Quell-IP offensichtlich durch IP-Spoofing manipuliert wurde und außerhalb der vergebenen privaten Subnetze liegt, das MWN nicht verlassen.

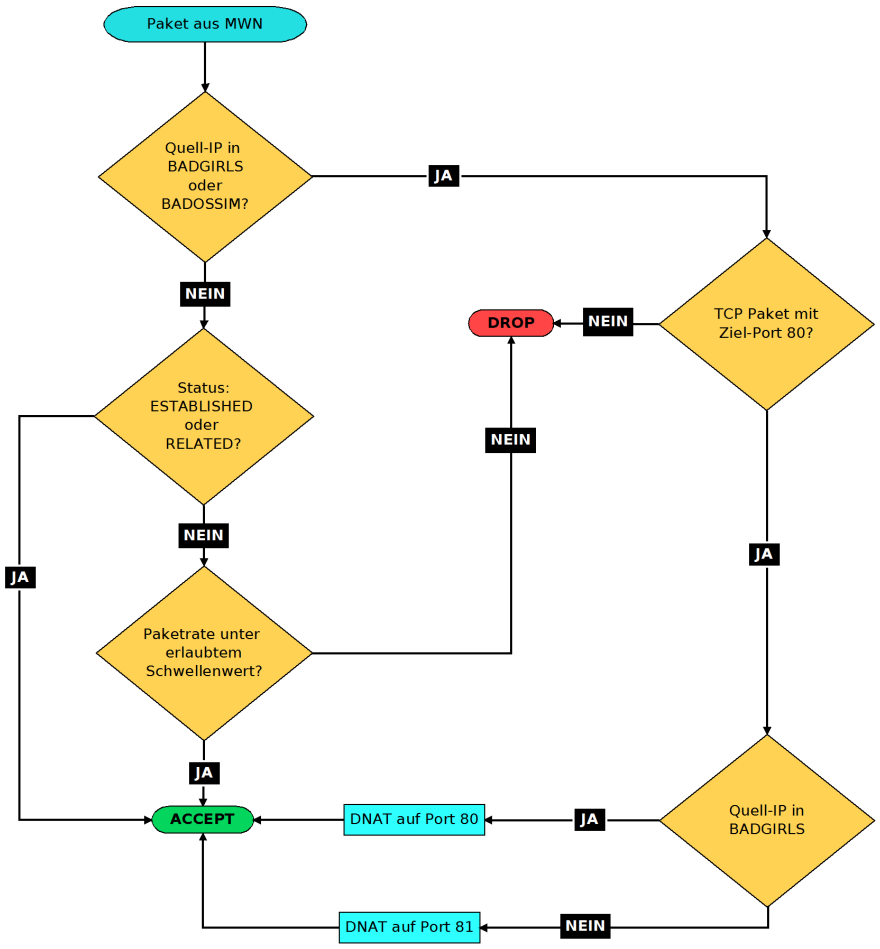

Abbildung 2.3.: Paketverarbeitung im Secomat - vom Host im MWN ins weltweite Internet

2.1.5. Eskalationsstufen

Die Sicherheitsmechanismen im Secomat zum Erkennen und Sperren infizierter Computer funktionieren nach dem Eskalationsprinzip. Das Intrusion Detection System deckt die oberen beiden Eskalationsstufen und das Intrusion Prevention System die unteren beiden Eskalationsstufen ab [DF06].

1. Stufe: Sie tritt ein, wenn es zu kurzzeitigen Überschreitungen der erlaubten Paketraten kommt. In diesem Fall wird das Burstsystem des IDS aktiv, d.h. die Pakete werden trotz Überschreitung ohne weitere Maßnahmen weitergeleitet.

2. Stufe: Diese Eskalationsstufe wird erreicht, wenn die Paketraten weit oder länger über einem Schwellenwert bleiben, sodass diese nicht durch das Burstsystem aufgefangen werden.

3. Stufe: Das Hard-Limit tritt in Kraft, wenn eine IP-Adresse in den vergangenen 15 Minuten mehr als 120 Log-Einträge verursacht hat [LR12].

4. Stufe: In schweren Fällen wird die organisatorische Eskalationsstufe erreicht. Diese

wird bei IP-Adressen aktiv, die permanent gesperrt sind. Die betroffenen IP-Adressen werden an den zuständigen Netzverantwortlichen gemeldet.

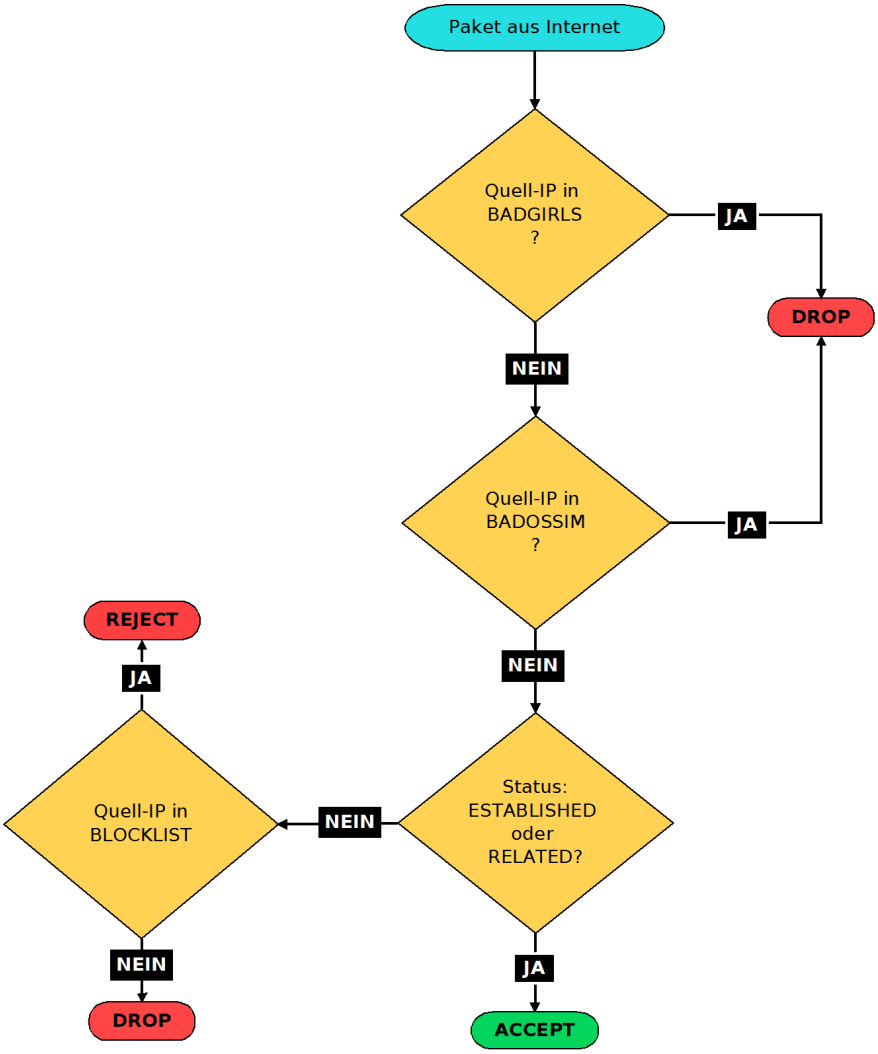

Abbildung 2.4.: Paketverarbeitung im Secomat - vom weltweiten Internet zum Host im MWN

2.1.6. Intrusion Detection System

Das Intrusion Detection System (IDS) im Secomat dient der Erkennung von hohen Paketraten und somit Denial-of-Service Angriffen. Das IDS wurde mit den Netfilter Modulen Hashlimit, Log, Nflog realisiert. Mit dem Modul Hashlimit werden die Paketraten für einzelne IP-Adressen und/oder Ports überwacht. Werden Überschreitungen der erlaubten Paketraten detektiert, werden die Pakete mit den Modulen Log und Nflog geloggt. In Hashlimit ist ein Burstsystem enthalten, durch das kurzzeitige Überschreitungen der erlaubten Paketraten toleriert und potentielle False-Positivs reduziert werden.

Das Burstsystem funktioniert mithilfe von Buckets, deren Größe pro Regel individuell gewählt werden kann. Die Buckets sind zu Beginn voll gefüllt. Solange sich die Paketrate unter dem definierten Schwellenwert befindet, bleibt das Bucket gefüllt. Das Zeitfenster für das der Schwellenwert gilt, kann pro Regel individuell angegeben werden. Sobald die Paketrate den Schwellenwert erreicht hat und weitere Pakete auf die Regel zutreffen, werden aus dem Bucket Punkte entfernt. Sobald der Wert des Buckets auf Null gefallen ist und weiterhin die Paketrate über dem Schwellenwert liegt, werden alle nachfolgenden Pakete, die auf die Regel zutreffen, als Verstoß detektiert. Sobald die Paketrate wieder unter den Schwellenwert fällt, werden dem Bucket pro Intervall wieder Punkte hinzugefügt bis es seinen Start-Füllstand erreicht hat.

In Netfilter des Secomats sind die folgenden Chains für die Überwachung der Paketraten aktiv:

- IDS_AC_DOS_SMTP: Erkennung von SPAM-Mails
- IDS_AC_DOS: Erkennung von Netzscans und DoS Angriffen, insbesondere auf DNS, HTTP, HTTPS Dienste im Internet
- IDS_AC_DDOS: Erkennung von DDoS Angriffen auf Ziele im Internet
- INPRULES: Erkennung von DoS Angriffen auf den Secomat Knoten

In allen Chains werden bis zu zehn Pakete nach einer Paketratenüberschreitung geloggt und gespeichert. Alle Pakete, die als Überschreitung detektiert wurden, werden durch nachfolgende Filterregeln mit der Maßnahme DROP verworfen.

2.1.7. Intrusion Prevention System

Beim Intrusion Prevention System des Secomats handelt es sich um ein Bash-Script, dass in Intervallen von einer Minute ausgeführt wird. Zu Beginn eines Durchlaufs werden die Log-Einträge des Intrusion Detection Systems der letzten 15 Minuten analysiert. Für jede auffällig gewordene IP-Adresse wird gezählt, wie oft diese Log-Einträge verursacht hat.

Nach Abschluss der Analyse, wird für jede IP-Adresse der Strafpunktestand ermittelt. Dieser ergibt sich aus der Summe der verursachten Log-Einträge. Übersteigt der Strafpunktestand den maximal erlaubten von 120, wird die IP-Adresse gesperrt. Die Sperrung erfolgt indem die zu sperrende IP-Adresse in die Variable BADGIRLS eingefügt wird. Alle IP-Adressen, die in einem vorherigen Durchlauf gesperrt wurden und deren Strafpunktestand wieder unter den maximalen gefallen ist, werden automatisch entsperrt. Dies erfolgt durch die Löschung der IP-Adresse aus BADGIRLS.

Für die IP-Adressen, die gesperrt werden, wird zusätzlich eine Datei in einem Verzeichnis abgelegt. Die Datei erhält die IP-Adresse als Namen und die angesammelten Verstöße gegen die einzelnen Regeln des IDS als Inhalt.

Bleibt eine IP-Adresse über einen längeren Zeitraum gesperrt, schickt das Intrusion Prevention System eine E-Mail an den zuständigen Netzverantwortlichen, um diesen darüber zu informieren. Die E-Mail-Adressen der Netzverantwortlichen sind in einer Datenbank hinterlegt.

2.1.8. Webserver und Statusseite

Auf jedem Knoten des Secomat Clusters ist ein Apache Webserver installiert. Auf dem Webserver sind zwei vHosts konfiguriert. Der eine vHost lauscht auf Port 80 und dient der Auslieferung der Statusseite für Nutzer, die durch das IPS im Secomat gesperrt wurden. Der andere vHost lauscht auf Port 81 und liefert die Statusseite für Nutzer, deren IP-Adresse aufgrund eines Log-Eintrags bei einem externen IDS gesperrt wurde. Zur Generierung der individuellen Statusseiten wird PHP genutzt.

Die Statusseiten enthalten folgende Informationen in deutscher und englischer Sprache:

- Status des Nutzers: gesperrt oder nicht-gesperrt
- Hinweise zu möglichen Ursachen der Sperrung
- Zeitpunkt der Sperrung
- gesperrte IP-Adresse des Nutzers
- Detaillierte Informationen zum aktuellen Grund der Sperrung

Warnhinweis / Warning Message

English version below...

No Internet

Lieber Nutzer,

Ihr Rechner wurde aufgrund exzessiver Überschreitung der erlaubten Paketrate **automatisch an der Nutzung des Internets gehindert**. Sehr wahrscheinlich ist Ihr Computer von einem **Wurm oder Virus befallen!** Auch P2P-Software (zum Filesharing, wie z.B. Gnutella, Kazaa, BitTorrent) kann in ungünstigen Fällen zu dieser Meldung führen.

Um wieder Zugriff auf die Internetdienste zu erhalten, beenden Sie eventuell laufende P2P-Software und versichern Sie sich bitte, dass Sie einen aktuellen Virenscanner auf Ihrem System installiert haben.

In seltenen Fällen kann auch eine Fehlkonfiguration oder die unkontrollierte Reaktion auf unerreichbare Server einer regulären Applikation zu einer Sperrung führen.

Weitere Informationen erhalten Sie unter: http://www.lrz.de/services/security/antivirus/ und http://www.lrz.de/services/netzdienste/secomat/

Dear User,

your computer has been **suspended from internet access** due to exceeding our packet rate limits. Most likely your computer is **infected by a worm or virus!** This message might also be caused by some P2P software used for file sharing like Gnutella, Kazaa, BitTorrent.

To regain internet access please disable any P2P software and make sure you have installed an up to date virus scanner.

In some rare cases a computer can be suspended due to a valid application's misconfiguration or runaway reaction in answer to unreachable servers.

Further information can be found on: http://www.lrz.de/services/security/antivirus/ and http://www.lrz.de/services/netzdienste/secomat/

Status Report for 10.156.200.21

Gesperrt seit / Blocked since 28.11.11 16:46

Überschreitungen Protokoll Zielport und Grund der Sperrung

Number of hits	Protocol	Destination port and suspension reason
176	TCP	25 - SMTP, Versenden von zu vielen Spam- oder Virenmails

Die Sperrung wird aufgehoben, sobald die Summe aller Überschreitungen unter 120 fällt. Technisch bedingt kann die automatische Freischaltung bis zu 15min dauern.

Internet access will be granted again if the total of all hit numbers falls below 120. Due to technical reasons re-enabling your access can take up to 15min.

powered by

Abbildung 2.5.: Statusseite für Nutzer, deren IP-Adresse durch das IPS im Secomat gesperrt wurde - Quelle: [LR12]

2.1.9. Hochverfügbarkeit und Lastverteilung

Der Secomat besteht aus einem Aktiv/Aktiv Cluster, d.h. alle Knoten sind im normalen Zustand aktiv und verarbeiten Pakete. Das Cluster enthält vier Knoten, die mit der Hochverfügbarkeitsoftware Heartbeat und Pacemaker hochverfügbar gemacht wurden. Fällt ein Knoten aus, so wird automatisch ein Failover durchgeführt, sodass die Arbeit des ausgefallenen Knotens durch einen verbleibenden Knoten übernommen wird.

Jeder Cluster-Knoten hat drei Interfaces:

- WAN-Interface: Kommunikation mit Hosts im Internet
- LAN-Interface: Kommunikation mit Hosts im MWN
- HA-Interface: Kommunikation zwischen Cluster-Knoten, zum Austausch von Statusinformationen

Alle Pakete von privat-adressierten Hosts mit einer Ziel-IP außerhalb des MWN werden zur IP-Adresse des Secomat-Clusters geleitet. Die IP-Adresse zeigt auf eine MAC-Multicast Adresse, sodass die Pakete an alle Cluster-Knoten des Secomats geschickt werden. Jeder Knoten verarbeitet nur die Pakete, die aus seinem Teil des MWN stammen und verwirft alle anderen Pakete. Dadurch wird die Last auf alle Knoten verteilt.

2.2. PfSense

Bei der pfSense Firewall handelt es sich um eine Modifikation des unixoiden Betriebssystems FreeBSD [CB13]. Sie hat ihren Ursprung im Embedded Computing und ist ein Fork von m0n0wall. Im Gegensatz zu m0n0wall kann die pfSense Firewall auch größere Rechen- und Speicherkapazitäten nutzen, als sie im Embedded Computing zur Verfügung stehen.

2.2.1. Packet Filter

Das Firewall Modul von FreeBSD und somit von pfSense Firewall ist Packet Filter (PF). Mit PF kann der Paketfluss gesteuert und auch NAT durchgeführt werden. Bei PF handelt es sich um eine statefull Firewall, d.h. sie speichert aktive Verbindungen in einer Statetable. Alle Pakete, die PF erreichen, werden zunächst implizit mit den Einträgen in der Statetable verglichen. Kann ein Paket einer Verbindung zugeordnet werden, wird es direkt an sein Ziel weitergeleitet. Kann ein Paket keinem Eintrag zugeordnet werden, wird es mit den Filterregeln in der Firewall verglichen. Die Konfiguration von PF erfolgt mit dem Werkzeug pfctl, das die Firewall-Konfiguration anhand einer Textdatei vornimmt. [Ope18b].

PF stellt folgende Konfigurationsmöglichkeiten bereit [Ope18c]:

- Macros: Variablen in denen IP-Adressen, Port-Nummern, Interfacenamen usw. gespeichert werden können.
- Tables: Variablen speziell für IP-Adressen mit effizienter Datenstruktur, um viele IP-Adressen in der Firewall zu verarbeiten.

- Filter Rules: Filterregeln werden entweder pro Interface oder Interface-Gruppe definiert. Dadurch gelten die Filterregeln entweder nur für ein Interface oder übergreifend für mehrere Interfaces in einer Interface-Gruppe.

- Options: Übergreifende Konfigurationseinstellungen der Firewall (Debug Modus, Statetable, ...)

Bei PF werden Pakete, die keiner bestehenden Verbindung zugeordnet werden können, in der Regel mit allen Filterregeln des Interface abgeglichen, an dem sie angekommen sind. Die Maßnahme der letzten Filterregel, auf die das Paket zutrifft, wird auf das Paket angewendet. Soll die Maßnahme einer Filterregel sofort auf ein passendes Paket anwendet werden und das Paket nicht weiter mit nachfolgenden Filterregeln verglichen werden, muss das Schlüsselwort „quick“ bei der Filterregel mit angegeben werden.

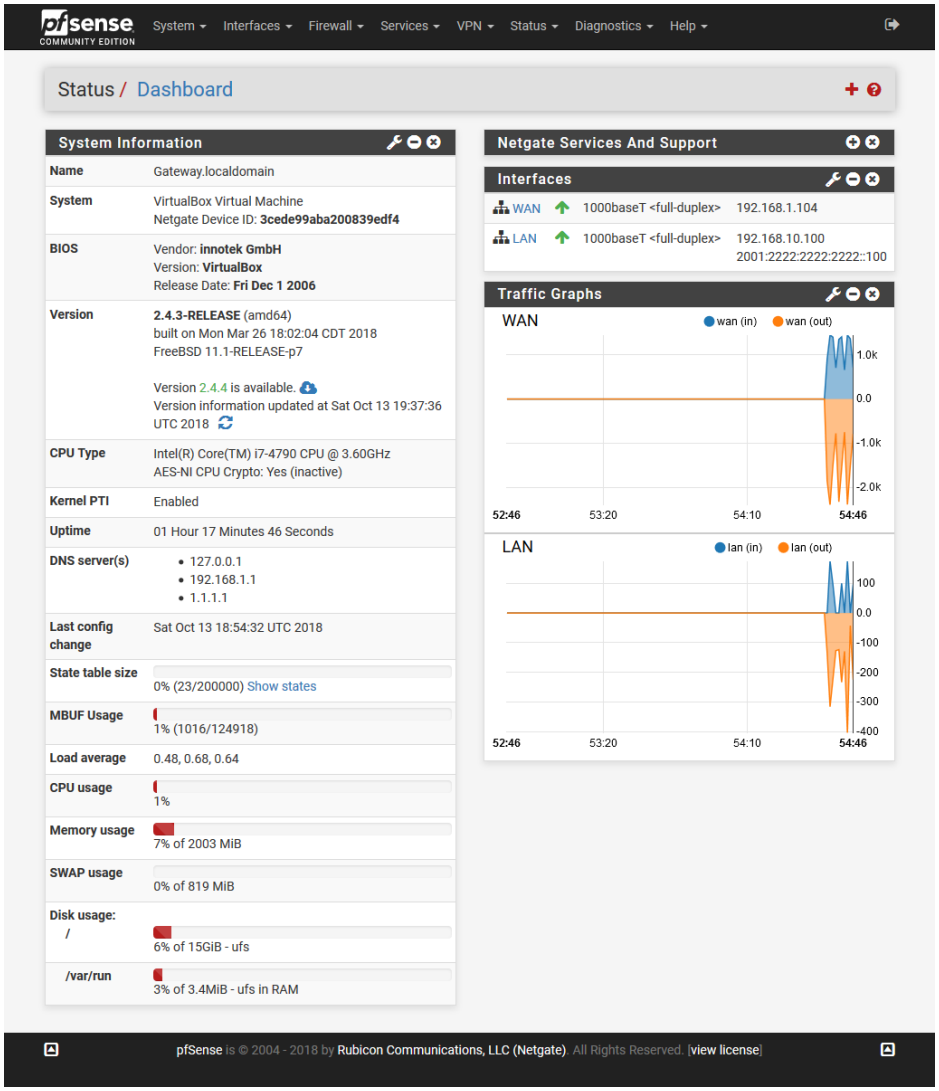

Abbildung 2.6.: Dashboard der pfSense Firewall

2.2.2. Weboberfläche und Konfiguration

Die Konfiguration der Firewall und anderer Dienste erfolgt in einer Weboberfläche, die durch einen nginx Webserver bereitgestellt wird. Die Weboberfläche kann mit HTTP oder HTTPS geöffnet werden. In der Standard-Konfiguration ist eine Weiterleitung von HTTP auf HTTPS aktiviert, welche bei Bedarf deaktiviert werden kann. Ebenfalls kann der Port, auf dem der Webserver lauscht, von Port 80 bzw. 443 auf einen individuellen Port geändert werden.

Bevor die Konfigurationsoberfläche der pfSense aufgerufen werden kann, ist die Eingabe eines Benutzernamens und Passworts notwendig. Nach erfolgreicher Anmeldung wird das Dashboard angezeigt. Im Dashboard können Widgets konfiguriert werden, über die der Status einzelner Dienste und andere Informationen angezeigt werden können. Werden Änderungen an der pfSense Konfiguration über die Weboberfläche getätigt und gespeichert, so werden diese in die pfSense Konfigurationsdatei config.xml gespeichert. In dieser XML-Datei befinden sich die Konfigurationseinstellungen von pfSense und installierter Erweiterungen. Änderungen an den Einstellungen von PF werden zunächst nur in die Konfigurationsdatei gespeichert. Die pfSense erzeugt aus der Konfigurationsdatei die temporäre Firewallkonfigurationsdatei in /tmp/rules.debug, über die die eigentliche Konfiguration von PF erfolgt [LLC18a].

2.2.3. Network Address Translation

Source Network Address Translation

Die pfSense kann mit und ohne Source Network Address Translation (SNAT) genutzt werden. Die Einstellung erfolgt über die Weboberfläche unter *Firewall - NAT* beim Tab *Outbound*. Wird SNAT genutzt, so stehen drei verschiedene SNAT-Modi zur Auswahl:

- Im Modus Automatic Outbound NAT wird die IP-Adresse des WAN-Interfaces für SNAT genutzt. Die NAT-Regeln sind implizit definiert. Es wird das Interface, bei dem eine Gateway Adresse definiert ist, als WAN genutzt.
- Beim Hybrid Outbound NAT Modus können zusätzlich zu den impliziten NAT-Regeln auch eigene SNAT-Regeln definiert werden.
- Beim Advanced bzw. Manual Outbound NAT werden nur eigene SNAT-Regeln für SNAT genutzt.

Destination Network Address Translation

Auf der pfSense kann auch Destionation Network Address Translation (DNAT) genutzt und über die Weboberfläche konfiguriert werden. Die Einstellungen erfolgen unter *Firewall - NAT* beim Tab *Port Forwarding*. Dort können einzelne Weiterleitungsregeln angelegt und konfiguriert werden.

2.2.4. Firewall

Die Filterregeln der Firewall werden unter *Firewall - Rules* in Tabellen angelegt. Die Filterregeln können übergreifend für alle Interfaces als Floating Rules oder für einzelne Interfaces

und Interface-Gruppen als Filter Rule angelegt werden. Die Pakete werden zunächst mit den Floating Rules und anschließend mit den Filter Rules des Interfaces abgeglichen, an dem sie ankommen. Bei den Floating Rules kann das Schlüsselwort „quick“ deaktiviert bzw. entfernt werden. Dadurch werden Pakete, die auf eine solche Floating Rule zutreffen, auch mit nachfolgenden Regeln verglichen. In diesem Fall gilt immer die letzte Filterregel, auf die ein Paket zutraf. Bei den Filter Rules eines Interfaces gilt immer die Regel, auf die das Paket zuerst gepasst hat. Die Pakete werden beginnend mit der obersten Filterregel nacheinander bis zu untersten Filterregel in der Tabelle abgeglichen.

Soll eine Filterregel für mehrere IP-Adressen oder Ports gelten, kann dies durch die Referenzierung eines Alias realisiert werden. Die Aliases werden in der Weboberfläche unter *Firewall - Aliases* angelegt und konfiguriert. Unter dem Tab *IP* werden Aliases verwaltet, die einzelne IP-Adressen, Adressbereiche und/oder Subnetze enthalten können. Soll jedoch ein Alias mehreren Ports und/oder Portbereiche enthalten, kann dieser unter dem Tab *Ports* angelegt werden. Auf die Aliases kann nicht nur in den Filterregeln, sondern auch in den SNAT, DNAT-Regeln und in den virtuellen IP-Adressen referenziert werden.

2.2.5. Hochverfügbarkeit

Mit der Installation von pfSense stehen bereits Hochverfügbarkeitsfunktionen zur Verfügung. Diese sind auf die Aktiv/Passiv-Konfiguration spezialisiert, d.h. nur die Master-Instanz verarbeitet den Verkehr, die zweite Instanz steht passiv als Backup bereit. Fällt der Master aus, übernimmt die Backup-Instanz die Position des Masters und die Verarbeitung des Verkehrs.

Common Address Redundancy Protocol

Auf der pfSense Firewall können virtuelle IP-Adressen (virtual IPs) mit dem Common Address Redundancy Protocol (CARP) genutzt und unter *Firewall - virtual IPs* angelegt werden. Zwei oder mehr pfSense Instanzen bilden eine Gruppe und können sich so eine oder mehrere virtuelle IP-Adresse(n) teilen. Zu jedem Zeitpunkt wird eine virtuelle IP-Adresse nur einer pfSense Instanz zugeordnet, indem sie u.a. auf ARP-Requests antwortet. Das Teilen einer IP-Adresse funktioniert durch das CARP, über das die pfSense Firewalls Status-Nachrichten austauschen.

Die folgenden Parameter sind für die Nutzung einer virtuellen IP-Adresse vom Typ CARP in der pfSense anzugeben:

- virtuelle IP-Adresse
- Host-ID
- Passwort
- Base-Wert
- Skew-Wert

Die virtuelle IP-Adresse muss sich im gleichen Subnetz befinden, wie die IP-Adresse des Interfaces, auf dem sie genutzt wird. Befinden sich mehrere Gruppen in einem Netz, so können

deren Mitglieder anhand der Host-ID die Nachrichten der eigenen Gruppe von den Nachrichten der anderen Gruppen unterscheiden. Jede Gruppe bzw. virtuelle IP-Adresse bekommt eine eigene Host-ID zugewiesen und alle Instanzen in einer Gruppe haben die gleiche Host-ID. Mit dem Passwort werden die Nachrichten mit SHA1-HMAC verschlüsselt, sodass diese nicht gefälscht oder manipuliert werden können [Ope18a]. Der Base-Wert kann für jede pfSense Instanz individuell gewählt werden und gibt das Intervall in Sekunden an, in dem die pfSense Statusnachrichten über Multicast an die anderen Mitglieder der Gruppe geschickt werden. Der Skew-Wert kann auch individuell für jede pfSense Instanz gewählt werden und gibt an, wie stark die Instanz gegenüber anderen Mitgliedern in der Gruppe als Master bevorzugt werden soll. Je niedriger der Wert, desto mehr wird die pfSense Instanz für die Position des Masters bevorzugt.

pfsync

Auf der pfSense wird pfsync genutzt, um die Einträge der Statetable in der Firewall des Masters auf eine Backup-Instanz zu synchronisieren. Fällt die Master-Instanz aus, können die Pakete auf der Backup-Instanz bestehenden Verbindungen zugeordnet werden, sodass Verbindungen nicht neu aufgebaut werden müssen. In der Weboberfläche unter *System - High Avail. Sync* kann pfsync aktiviert werden. Zusätzlich sind das Interface, über das die Synchronisation durchgeführt wird, und die IP-Adresse der Backup-Instanz auf die synchronisiert werden soll, anzugeben.

XML-RPC sync

In der gleichen Konfigurationsseite wie pfsync kann auch XML-RPC Sync konfiguriert werden. Über XML-RPC erfolgt die Synchronisation der pfSense Konfiguration durch Verwendung von HTTP oder HTTPS von der Master- auf die Backup-Instanz. Dazu müssen der Benutzername und das Passwort, wie bei der Anmeldung über die Weboberfläche auf der Backup-Instanz, angegeben werden.

2.2.6. Package Manager

Der Funktionsumfang der pfSense Firewall kann über mehrere Package Manager erweitert werden. Über den Package Manager in der Weboberfläche der pfSense können Packages installiert werden, die für die pfSense optimiert sind. Diese Packages bringen eine Konfigurationsoberfläche mit, die über die Hauptnavigation der pfSense Weboberfläche aufgerufen werden kann. Über den Package Manager können die Pakete auch aktualisiert und wieder entfernt werden.

Zusätzlich können weitere Funktionen über den FreeBSD Package Manager installiert werden. Dieser ist bei pfSense Installation ab der Version 2.3 verfügbar. Im FreeBSD Package Manager sind die Paketquellen von pfSense und nicht die originalen Paketquellen von FreeBSD verlinkt. Es können auch Pakete aus den FreeBSD Paketquellen heruntergeladen werden. Bei diesen kann es zu Kompatibilitätsproblemen aufgrund des modifizierten FreeBSD Betriebssystems der pfSense kommen [LLC18b].

2.2.7. Einsatzszenarien im MWN

Das LRZ stellt Instituten und Organisationen zwei virtuelle pfSense Firewalls in Aktiv/Passiv Konfiguration zur Verfügung. In den folgenden Einsatzszenarien können Angriffe aus dem lokalen Netz der pfSense Firewall auf externe Ziele erfolgen.

Ein Teil der pfSense Firewalls wird als NAT-Gateway genutzt. Die Hosts im lokalen Netz sind mit einer privaten IP-Adresse adressiert. Die private Quell-IP von Paketen, die das lokale Netz verlassen, wird durch SNAT mit einer öffentlichen IP-Adresse ersetzt. Nachdem SNAT auf die Pakete angewendet wurde, werden diese über das externe Interface der pfSense Firewall in das MWN geleitet. Da die Pakete keine private Quell-IP-Adresse mehr haben, werden die Pakete mit Zielen außerhalb des MWN nicht über den Secomat geleitet. In diesem Szenario können deshalb sowohl Ziele im MWN als auch im weltweiten Internet durch die Hosts im lokalen Netz der pfSense Firewall angegriffen werden.

Die pfSense Firewall wird im MWN auch als Router ohne SNAT genutzt. In diesem Fall ist das lokale Netz ein Teil des MWN und die Hosts können private oder öffentliche IPv4-Adressen haben. Die pfSense Firewall routet die Pakete aus dem lokalen Netz unverändert in das restliche MWN weiter. Die Pakete von privat-adressierten Hosts und Zielen außerhalb des MWN werden daher durch Policy-basiertes Routing zum Secomat geleitet. Der Secomat kann in diesem Fall Angriffe auf Ziele außerhalb des MWN verhindern. Jedoch bietet der Secomat bei Angriffen auf Ziele innerhalb des MWN keinen Schutz.

Die Hosts mit öffentlichen IP-Adressen im lokalen Netz der pfSense Firewall können sowohl Angriffe auf Ziele innerhalb als auch außerhalb des MWN durchführen. Da deren Pakete keine private Quell-IP haben, werden die Pakete nicht über den Secomat in das Internet geroutet. Daher kann der Secomat in diesem Szenario keine Angriffe erkennen.

In einigen Netzen, die durch die pfSense Firewall geschützt werden, werden die Hosts zusätzlich zu ihrer IPv4-Adresse auch mit einer IPv6-Adresse adressiert. Da der Secomat aktuell auch keinen IPv6 Verkehr überwacht, können über IPv6 Angriffe auf Ziele im MWN und im Internet erfolgen.

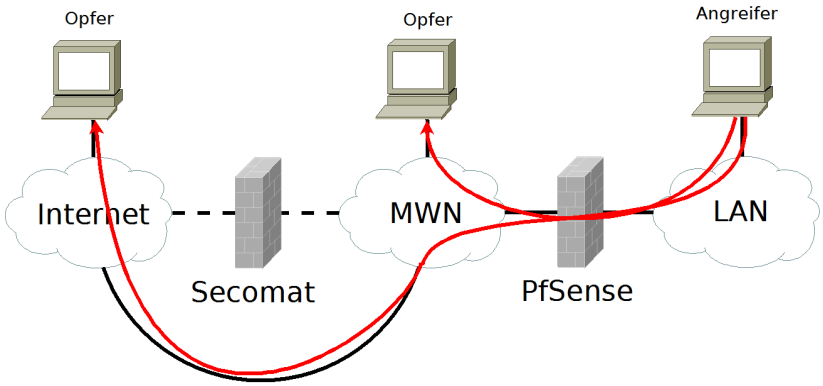

Abbildung 2.7.: Angriffe über die pfSense Firewall als Router - mit SNAT und privaten IPv4-Adressen im LAN - mit IPv6-Adressen im LAN - mit öffentlichen IPv4-Adressen im LAN

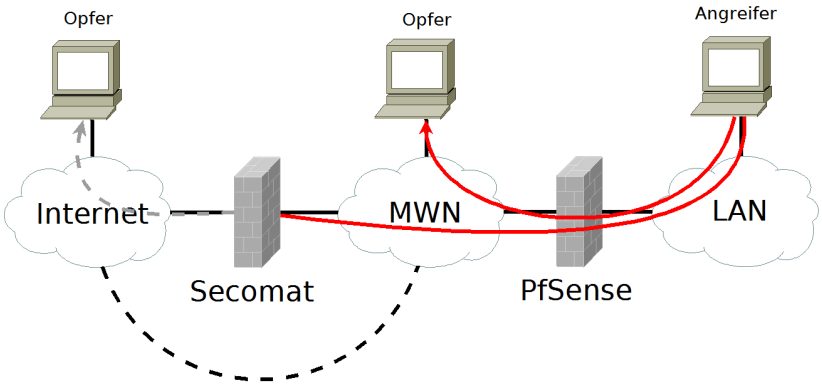

Abbildung 2.8.: Angriffe über die pfSense Firewall als Router ohne SNAT mit privaten IPv4-Adressen im LAN

3. Anforderungsanalyse

In der folgenden Anforderungsanalyse werden Anforderungen auf Basis der einzelnen Bereiche der Secomat Funktionalität aufgestellt, die in dieser Arbeit auf die pfSense Firewall migriert werden.

3.1. Destination Network Address Translation

Durch Destination Network Address Translation (DNAT) werden beim Secomat HTTP-Requests von gesperrten Nutzern an den lokalen Webserver umgeleitet.

Auf der pfSense Firewall wird DNAT benötigt, um gesperrte Nutzer automatisch über ihre Sperrung durch eine Statusseite zu informieren. In der DNAT-Regel muss nach der Quell-IP gefiltert werden, damit nur Pakete von gesperrten Nutzern umgeleitet werden. Damit nur HTTP-Anfragen umgeleitet werden, muss in der DNAT-Regel nach Protokoll TCP und Ziel-Port 80 gefiltert werden.

Da die lokalen Netze der pfSense Firewalls neben IPv4-Adressen zum Teil auch zusätzlich mit IPv6-Adressen versorgt werden, muss DNAT zusätzlich für IPv6 realisiert werden. Ist die IPv6-Adresse eines Nutzers gesperrt, so sind auch die HTTP-Requests über IPv6 an den lokalen Webserver umzuleiten.

3.2. Firewall

Mit Hilfe der Firewall werden im Secomat Pakete mit gesperrter Quell-IP verworfen, sodass keine weiteren Angriffe auf Ziele im Internet möglich sind. Die Pakete, die den Secomat aus dem Internet erreichen und eine gesperrte IP-Adresse als Ziel haben, werden ebenfalls durch die Firewall verworfen. Dazu werden im Secomat gesperrte IP-Adresse in einer Variablen BADGIRLS gespeichert, auf die in den Filterregeln referenziert wird.

Wie beim Secomat hat die Firewall von pfSense alle Pakete zu verwerfen, die von gesperrten Hosts geschickt und nicht über DNAT umgeleitet wurden. Dazu hat die Firewall das Speichern von IPv4- und/ oder IPv6-Adressen in einer Variablen zu unterstützen, damit nicht für jede zu sperrende IP-Adresse neue Filterregeln angelegt werden müssen. Dadurch bleibt die Firewall Konfiguration übersichtlich und die Pakete müssen nicht mit zahlreichen Filterregeln abgeglichen werden. In den Filterregeln muss auf die Variable mit den gesperrten IP-Adressen referenziert werden können, sodass nur Pakete verworfen werden, deren Quell-IP in der Variablen gespeichert ist.

3.3. Intrusion Detection System

Durch das Intrusion Detection System (IDS) des Secomats werden die Paketraten überwacht. Abhängig vom Ziel-Port und Protokoll eines Pakets, sind verschieden hohe Schwellenwerte für die Paketraten definiert. Die kurzzeitige Überschreitung eines Schwellenwertes wird durch das Burstsystem aufgefangen und führt zu keinen weiteren Maßnahmen. Wird ein Angriff durch das IDS detektiert, so werden die darauf folgenden Pakete verworfen und bis zu zehn Pakete geloggt.

Durch das IDS müssen Distributed Denial-of-Service (DDoS) Angriffe sowie Denial-of-Service (DoS) Angriffe wie beim Secomat detektieren werden. Dazu sind die Paketraten für einzelne Quell-IP- und Ziel-IP-Adressen zu überwachen. Das IDS muss zwischen Paketen unterscheiden, die einer bestehenden bzw. verwandten Verbindung oder einer neuen Verbindung angehören. Dies ist wichtig, damit nur Pakete von neuen Verbindungen in die Paketratenüberwachung einfließen. Die Schwellenwerte für die Paketraten sind für einzelne Regeln individuell zu definieren, um die Paketrate an den Dienst und somit den Port anzupassen. Dies ist notwendig, damit für das Senden von E-Mails ein anderer Schwellenwert definiert werden kann, als für das Aufrufen einer Webseite über HTTP. Durch ein Burstsystem sollen kurzzeitige Überschreitungen der erlaubten Paketraten toleriert werden, um False-Positivs zu reduzieren. Die Pakete, die nach Erreichen eines Schwellenwertes das IDS erreichen, müssen geloggt werden. Das Loggen ist notwendig, damit das IPS diese analysieren und daraufhin IP-Adressen sperren kann. Die Log-Einträge müssen begrenzt werden, um das „Fluten“ der Log-Datei mit Log-Einträgen durch sehr hohe Paketraten zu verhindern. Das IDS muss in der Lage sein, die Größe der Log-Datei zu begrenzen. Dies sehr wichtig, damit die Log-Datei nicht den gesamten Speicherplatz auf der pfSense Firewall belegt. Außerdem muss das IDS alle Pakete, die zu einer Überschreitung geführt haben, durch eine IPS-Komponente verwerfen. Dadurch werden Angriffe frühzeitig abgeschwächt bis das IPS die IP-Adresse vollständig sperrt. Das IDS muss die Paketraten für IPv4- und IPv6-Pakete überwachen, da beide Protokoll-Versionen genutzt werden.

3.4. Intrusion Prevention System

Um IP-Adressen vollständig zu sperren, wird im Secomat ein Intrusion Prevention System (IPS) in Intervallen von einer Minute ausgeführt. Das IPS berechnet für jede IP-Adresse, die Log-Einträge in den letzten 15 Minuten verursacht hat, einen Strafpunktestand. Übersteigt der Strafpunktestand einer IP-Adresse den Schwellenwert von 120 Punkten, so wird diese in die Variable BADGIRLS eingefügt. Ist eine IP-Adresse bereits gesperrt und der Strafpunktestand unter den Schwellenwert gefallen, so wird die IP-Adresse aus BADGIRLS entfernt. Zusätzlich wird gespeichert, wie oft die gesperrten IP-Adressen gegen eine der Regeln des IDS verstoßen hat. Diese Informationen werden dem gesperrten Nutzer auf einer Statusseite angezeigt.

Auf der pfSense wird ein IPS benötigt, um infizierte und angreifende Hosts vollständig zu sperren und dadurch weitere Angriffe zu unterbinden. Das IPS muss in regelmäßigen Abständen die Log-Einträge des IDS analysieren. Auf Basis der Log-Einträge sind für jede IP-Adresse die Anzahl der Vergehen pro IDS-Regel zu ermitteln. Nach Abschluss der

Analyse muss ein Strafpunktestand pro IP-Adresse auf Basis der Analyse ermittelt werden. Ist der Strafpunktestand einer IP-Adresse über dem Schwellenwert von 120, so hat das IPS die IP-Adresse mit Hilfe der Firewall vollständig zu sperren. Bereits gesperrte IP-Adressen, deren Strafpunktestand wieder unter 120 gefallen ist, müssen durch das IPS automatisch entsperrt werden. Bleibt eine IP-Adresse über einen längeren Zeitraum permanent gesperrt, so sind diese durch eine E-Mail an den Administrator der Firewall gemeldet. Dadurch soll verhindert werden, dass infizierte Hosts über einen längeren Zeitraum unerkannt bleiben und dadurch weiterhin Geräte im lokalen Netz angreifen können. Das IPS muss die Sperrung von IPv4- und IPv6-Adressen unterstützen, da beide Protokoll-Versionen genutzt werden. Das IPS soll mit einer Whitelist versehen werden, sodass die darin gespeicherten IP-Adressen nicht durch das IPS gesperrt werden.

3.5. Webserver

Im Secomat wird der Apache Webserver für die Bereitstellung der Statusseite genutzt.

Auf der pfSense wird somit auch ein Webserver für die Auslieferung der Statusseite benötigt. Der Webserver muss eine Server-Scriptsprache unterstützen, damit die Statusseite dynamisch generiert werden kann. Dies ist wichtig, damit einzelnen Nutzern der individuelle Grund der Sperrung angezeigt werden kann. Der Webserver muss mit eingeschränkten Systemrechten betrieben werden, da die gesperrten Nutzer direkt mit dem Webserver interagieren. Der Webserver muss auch HTTP-Redirection unterstützen, damit der Nutzer die Umleitung anhand der URL-Zeile im Browser erkennen kann.

3.6. Statusseite

Über eine Statusseite werden die Nutzer im MWN über die Sperrung oder Entsperrung ihrer IP-Adresse durch den Secomat informiert. Die Statusseite zeigt den Nutzern in einem allgemeinen Status an, ob diese Zugang zum Internet haben oder ob dieser gesperrt wurde. Ist der Zugang gesperrt, so gibt die Statusseite Hinweise zu möglichen Ursachen der Sperrung. Zusätzlich zu den allgemeinen Informationen zeigt die Statusseite auch für jeden Nutzer individuelle Informationen. Einem gesperrten Nutzer wird dessen IP-Adresse, der Zeitpunkt der Sperrung und der Grund der aktuellen Sperrung angezeigt.

Die Statusseite muss den Nutzern anzeigen, ob diese gesperrt oder nicht-gesperrt sind. Die Informationen sollen in deutscher und englischer Sprache angezeigt werden. Es sollen Hinweise auf mögliche Gründe der Sperrung gegeben werden. Es soll die IP-Adresse des gesperrten Nutzers angezeigt werden. Insbesondere muss die Seite den Grund der aktuellen Sperrung anzeigen, damit der gesperrte Nutzer gezielt die Ursache der Sperrung beheben kann.

3.7. Konfiguration und Synchronisation

Im Unterschied zum Secomat wird die pfSense Firewall durch verschiedene Institute und Organisationen genutzt. Aus diesem Grund soll die Statusseite über eine Konfigurationsoberfläche geändert werden können, sodass Kontaktinformationen hinterlegt und geändert werden können. Die Konfiguration und Verwaltung der pfSense Firewall wird zum Teil durch

die Institute selbst übernommen. Die pfSense Firewalls werden durch unterschiedliche Administratoren betreut. Aus diesem Grund soll die E-Mail Benachrichtigungsfunktion des IPS über eine Konfigurationssoberfläche angepasst werden können. Das IPS, IDS und der Webserver sollen über eine Konfigurationsoberfläche gestartet und gestoppt werden können, um u.a. Updates zu installieren oder Änderungen vorzunehmen.

Um inkonsistente Konfigurationen zu verhindern, müssen Änderungen an der Konfiguration des IPS, IDS, Webservers und der Statusseite automatisch auf die Backup-Instanz synchronisiert werden.

3.8. Package

Damit die bereits vorgestellten Anforderungen und Funktionalitäten mit geringem Zeitaufwand auf die pfSense Firewall übertragen werden können, muss ein Package entwickelt werden. Bei der Installation des Package sollen automatisch Komponenten hinzugefügt und Änderungen an der pfSense Konfiguration vorgenommen werden. Wird das Package wieder entfernt, sollen Änderungen an der pfSense Konfiguration automatisch rückgängig gemacht und die dazugehörigen Komponenten entfernt werden.

4. Migration

Im nachfolgenden Kapitel wird dargestellt, wie die Migration auf Basis der Anforderungsanalyse durchgeführt wurde.

4.1. Firewall

Um Angriffe von Hosts aus dem lokalen Netz durch die Firewall der pfSense zu blockieren, muss deren Konfiguration geändert werden. Zunächst wird ein Alias BADGIRLS angelegt. In diesen werden IPv4- und IPv6-Adressen von gesperrten Hosts gemeinsam gespeichert.

Damit die Pakete, deren Quell-IP in dem Alias BADGIRLS gespeichert ist, durch die Firewall verworfen werden, muss eine entsprechende Filterregel für das LAN-Interface definiert werden. Die Filterregel muss in der Tabelle für die Filterregeln des LAN-Interfaces über den bestehenden Filterregeln angelegt werden, sodass Pakete zunächst mit ihr und anschließend mit den restlichen Filterregeln des LAN-Interfaces abgeglichen werden. Als Maßnahme wird „Block“ausgewählt, damit alle Pakete verworfen werden, die von gesperrten Hosts gesendet wurden. Die Filterregel wird für IPv4 und IPv6 definiert, sodass sie die Pakete von gesperrten IPv4- und IPv6-Adressen blockiert. Um nur Pakete gesperrter IP-Adressen zu blockieren, wird in der Filterregel als Quell-IP der Alias BADGIRLS angeben. Dadurch gilt die Filterregel nur für Pakete, deren Quell-IP in BADGIRLS gespeichert ist.

Die gesperrten Nutzer geben i. d. R. eine Domain in ihren Browser ein, für die die IP-Adresse aufgelöst werden muss, bevor eine HTTP-Anfrage geschickt werden kann. Befindet sich ein DNS-Server im lokalen Netz, beantwortet dieser die DNS-Anfragen. Daraufhin schickt der Client die HTTP-Anfrage an den Webserver. Befindet sich kein DNS-Server im lokalen Netz, wird die DNS-Anfrage an die pfSense Firewall geschickt. Die Firewall verwirft die DNS-Anfragen der gesperrten IP-Adressen durch die zuvor beschriebene Filterregel, wodurch keine Auflösung der zur Domain gehörigen IP-Adresse erfolgt. Ohne Auflösung der Domain wird keine HTTP-Anfrage geschickt. Diese ist jedoch Voraussetzung, um dem gesperrten Nutzer die Statusseite zu schicken. Befindet sich also kein DNS-Server im lokalen Netz, muss eine weitere Filterregel definiert werden. Durch die DNS-Anfragen von gesperrten Hosts erlaubt und weitergeleitet werden.

4.2. Intrusion Detection System

Über den Package Manager in der Weboberfläche der pfSense können die IDS-Packages Snort und Suricata heruntergeladen und installiert werden. Nur Suricata unterstützt auf der pfSense Firewall ab Version 2.3 den Betrieb im Inline Modus [Mee16]. Im Inline Modus wird Suricata zwischen ein Interface und die Firewall geschaltet. Dadurch werden die Pakete vom Interface zu Suricata geschickt, das die Pakete entweder verwirft oder weiter an die

Firewall schickt. Die Firewall schickt Pakete ebenfalls an Suricata, das auch hier die Pakete verwirft oder an das Interface weiterschickt. Dadurch kann Suricata direkt in den Paketfluss eingreifen und Pakete auf Basis von Signaturen verwerfen. Ein weiterer Vorteil von Suricata ist die Unterstützung von Multi-Threading, wodurch es bei Mehrkernprozessoren die Last durch die Paketverarbeitung auf mehrere Prozessorkerne verteilen kann. Snort unterstützt weder den Betrieb im Inline Modus noch Multi-Threading und wurde aus diesen Gründen nicht eingesetzt.

4.2.1. Konfiguration

Die Konfiguration von Suricata erfolgt über die mitgelieferte Konfigurationoberfläche, die in der Hauptnavigation der pfSense Weboberfläche unter **Services - Suricata** aufgerufen wird.

Globale Einstellungen

Im Tab **Logs Mgmt** wird das *Auto Log Management* aktiviert. Daraufhin archiviert Suricata automatisch alle fünf Minuten die Log-Einträge der Log-Dateien, wenn die Größe einer Log-Datei einen bestimmten Schwellenwert überschritten hat. Der Schwellenwert kann im gleichen Tab für die verschiedenen Log-Dateien eingestellt werden. Innerhalb der fünf Minuten kann die Größe der Log-Dateien jedoch über den angegebenen Wert hinauswachsen. Im gleichen Tab wird auch der Speicherplatz übergreifend für das Log-Verzeichnis, in dem alle Log-Dateien gespeichert werden, begrenzt. Dadurch wird sichergestellt, dass die Log-Dateien nicht den gesamten Speicherplatz auf der pfSense Firewall belegen. Ist das Log-Verzeichnis voll, so werden automatisch ältere, archivierte Log-Dateien gelöscht, um aktuellere Log-Dateien zu archivieren.

Damit Konfigurationänderungen an Suricata und die Log-Einträge automatisch auf die Backup Instanz synchronisiert werden, wird im Tab **Sync** die Synchronisation über XML-RPC aktiviert und *Enable Sync* auf „Sync to configured system backup server“gestellt. Im gleichen Tab wird auch *Refresh Rule Sets* auf „Signal target host to refresh files“ gestellt, sodass geänderte Signaturen automatisch in Suricata geladen werden.

Interface Einstellungen

In Suricata wird der Paketfluss getrennt für einzelne Interfaces überwacht. Deshalb ist ein Eintrag unter dem Tab **Interfaces** für das LAN-Interface anzulegt. Anschließend werden folgende Einstellungen zur Überwachung des LAN-Interfaces vorgenommen:

- Im Tab **Settings** wird unter dem Abschnitt *General Settings* das LAN-Interface ausgewählt, damit Suricata den Verkehr über dieses Interface überwacht. Damit Suricata direkt in den Paketfluss eingreifen und Pakete verwerfen kann, wird im Abschnitt *Alert and Block Settings* der IPS-Mode auf „Inline Mode“ gestellt.
- Unter dem Tab **Rules** sind mit der Installation von Suricata bereits verschiedene Signaturen vordefiniert und aktiv. Diese Signaturen werden deaktiviert. Dafür werden im gleichen Tab neue Signaturen definiert.
- Da Suricata neben statistischer Überwachung der Paketraten auch Deep Packet Inspektion beherrscht, sind im Tab **App Parsers** bereits diverse Parser aktiviert. Diese

werden deaktiviert, um Rechenleistung zu sparen, da nur die Paketraten überwacht werden müssen.

4.2.2. Signaturen

Damit Suricata die Paketraten für einzelne Hosts überwacht, werden die Regeln des IDS im Secomat in Signaturen für Suricata übersetzt.

Die Signaturen von Suricata haben immer folgenden Aufbau:

1. Aktion
2. Protokoll
3. Quell-IP
4. Quell-Port
5. Flussrichtung des Pakets
6. Ziel-IP
7. Ziel-Port
8. Meta-Informationen und spezifische Filterkriterien

Die Aktion bestimmt, was passieren sollen, wenn ein Paket auf die Signatur zutrifft. Um Pakete zu verwerfen, wird die Aktion „drop“genutzt. An der Position Protokoll wird in der Signatur angegeben, welche Protokolle durch diese überwacht werden sollen. Soll eine Signatur für alle Protokolle gelten, so wird „ip“ angegeben. Es kann aber auch nach icmp, tcp, udp und weiteren Protokollen gefiltert werden. Mit Quell-IP und Quell-Port werden Pakete nach ihrer Quell-IP und ihrem Quell-Port gefiltert. Neben Konstanten können auch Variablen angegeben werden. In der Variablen *$HOME_NET* sind alle Subnetze enthalten, mit denen die pfSense Firewall über ihre Interfaces verbunden ist. Soll die Signatur für beliebige IP-Adresse und/oder Ports gelten, so wird *any* angegeben. Mit der Flussrichtung kann gefiltert werden, ob gesendete und/oder empfangene Pakete durch die Signatur überwacht werden. Mit Ziel-IP und Ziel-Port werden Pakete nach ihrer Ziel-IP und ihrem Ziel-Port gefiltert. Die Meta-Informationen umfassen u.a. folgende Informationen:

- msg: Log-Nachricht
- sid: Signatur-ID, die bei jeder Signatur einzigartig sein muss
- rev: Versionsnummer der Signaturen. Der Wert wird bei jeder Änderung um eins erhöht.
- classtype: Klassifikation der Regel - Dadurch kann angegeben werden, ob es sich um eine informelle oder sicherheitsrelevante Signatur handelt.

Damit eine Signatur nur aktiviert wird, wenn eine bestimmte Paketrate überschritten wurde, wird das spezifische Filterkriterium detection_filter genutzt. Mit detection_filter werden zusätzlich drei Parameter in der folgenden Syntax angegeben:

detection_filter: track <by_src/by_dst>, count <n>, seconds <m>

Mit *track* wird bestimmt, ob die Paketrate für den Sender (by_src) oder den Empfänger (by_dst) der Pakete überwacht werden soll. Sollen die Paketraten für einzelne Ports von einer Quell-IP überwacht werden, so muss für jeden Port eine neue Signatur angelegt werden. Mit *count* wird die Anzahl der Pakete angegeben, die die Signatur passieren, bis sie aktiv wird. Mit *seconds* wird das Zeitfenster angegeben, in dem die Pakete gezählt werden.

In der globalen Einstellung von Suricata ist festgelegt, dass nur Pakete in die Filterung einer Signatur mit einbezogen werden, die einer erfolgreich aufgebauten Verbindung zugeordnet werden können. Bei TCP entspricht dies einem erfolgreichen Drei-Wege-Handschlag. Bei UDP gilt eine Verbindung als erfolgreich aufgebaut, sobald Antwort-Pakete empfangen wurden [Fou18]. Damit nur Pakete von neuen Verbindungen in die Paketratenüberwachung einer Signatur einfließen, wird in den spezifischen Filterkriterien das Schlüsselwort *flow* angegeben. In den Signaturen wird *flow* zusammen mit to_server und not_established angegeben. Mit to_server werden nur Pakete getrackt, die zum Server geschickt werden. Durch not_established gilt die Signatur nur für Pakete, die keiner bestehenden Verbindung zugeordnet werden.

Suricata besitzt kein Burstsystem. Es kann somit nicht zwischen kurzzeitigen und langzeitigen Überschreitung der erlaubten Paketraten unterscheiden.

In Suricata werden alle Pakete geloggt, die durch Signaturen verworfen werden. Da alle Pakete verworfen werden sollen, sobald die Paketrate einen bestimmten Schwellenwert überschritten hat, ist eine Begrenzung der Log-Einträge pro Zeitfenster nicht realisierbar.

In Suricata werden Pakete mit allen Signaturen abgeglichen. Erfüllt ein Paket die Kriterien von mehreren Signaturen, so erzeugt Suricata für jede Signatur einen Log-Eintrag. Im Unterschied zum Secomat ist es in Suricata nicht möglich mit einer Signatur die Paketraten von einer Quell-IP zu unterschiedlichen Ziel-Ports getrennt zu überwachen. Um die Paketraten für einzelne Ziel-Ports zu überwachen, müsste für jeden Ziel-Port eine eigene Signatur angelegt werden. Da dies zu einem hohen Managementaufwand und leicht zu Fehlern in der Konfiguration führen kann, wurde dies nicht gemacht. Stattdessen werden die Paketraten übergreifend von einer Quell-IP zu mehreren Ziel-Ports durch eine Signatur überwacht.

Die Regeln im IDS des Secomats wurden durch folgende Signaturen umgesetzt:

Spam-Mails:
Mit der folgenden Signatur wird der Versandt von Spam-Mails erkannt und auf drei E-Mails pro Minute begrenzt werden.

```
drop tcp $HOME_NET any -> any 25 (msg:"SMTP SPAM-Mails";
flow:to_server, not_established;
classtype:bad-unknown; sid:1; rev:1;
detection_filter: track by_src, count 3, seconds 60;)
```

Ping-Scans und ICMP Flooding:
Durch diese Signatur werden Ping-Scans und ICMP-Flooding erkannt und erschwert.

```
drop icmp $HOME_NET any -> any any (msg:"ICMP too many pings";
classtype:bad-unknown; sid:2; rev:1;
detection_filter: track by_src, count 50, seconds 1;)
```

DoS auf DNS Dienste:
Durch diese Signatur werden DoS Angriffe auf DNS Dienste erkannt und abgeschwächt.

```
drop udp $HOME_NET any -> any 53 (msg:"DNS DoS Attack";
flow:to_server, not_established;
classtype:bad-unknown; sid:3; rev:1;
detection_filter: track by_src, count 60, seconds 1;)
```

DoS Angriffe auf HTTP Dienste
Mit dieser Signatur werden Angriffe auf Webserver mit HTTP erkannt und abgeschwächt.

```
drop tcp $HOME_NET any -> any 80 (msg:"HTTP DoS Attack";
flow:to_server, not_established;
classtype:bad-unknown; sid:4; rev:1;
detection_filter: track by_src, count 60, seconds 1;)
```

DoS Angriffe auf HTTPS Dienste
Mit dieser Signatur werden DoS Angriffe auf Webserver mit HTTPS erkannt und abgeschwächt. Beim Secomat sind hier sehr niedrige Schwellenwerte im Verhältnisse zum Burstwert angegeben. Da es bei Suricata kein Burstsystem gibt, wurde der Schwellenwert verdoppelt. Dadurch sollen False-Positivs reduziert werden.

```
drop tcp $HOME_NET any -> any 443 (msg:"HTTPS DoS Attack";
flow:to_server, not_established;
classtype:bad-unknown; sid:5; rev:1;
detection_filter: track by_src, count 60, seconds 1;)
```

DoS Angriffe auf QUIC Dienste
Diese Signatur erkennt Angriffe auf Webserver über das QUIC Protokoll. Auch bei dieser Regel ist im IDS des Secomats ein niedriger Schwellenwert im Verhältnis zum Burstwert definiert. Deshalb wurde auch hier der Schwellenwert verdoppelt, um False-Positivs zu reduzieren.

```
drop udp $HOME_NET any -> any 443 (msg:"QUIC DoS Attack";
flow:to_server, not_established;
classtype:bad-unknown; sid:6; rev:1;
detection_filter: track by_src, count 60, seconds 1;)
```

DDoS Angriffe und Port Scans
Diese Signatur dient der Erkennung und Abschwächung von DDoS und DoS Angriffen auf eine Ziel-IP. Diese Signatur wird immer aktiv, sobald mehr als tausend Pakete zu einer Ziel-IP fließen. Diese Signatur wird auch aktiv, wenn die Paketrate von einer Quell-IP zu einer Ziel-IP den angegeben Schwellenwert erreicht. Das führt dazu, dass wenn ein Host DoS Angriffe auf eine Ziel-IP durchführt, auch Pakete anderer Hosts im lokalen Netz zu dieser Ziel-IP durch Suricata geloggt und verworfen werden.

```
drop ip $HOME_NET any -> any any (msg:"(D)DoS Attack";
flow:to_server, not_established;
classtype:bad-unknown; sid:7; rev:1;
detection_filter: track by_dst, count 1000, seconds 1;)
```

DoS Angriffe
Durch diese Signatur werden die Paketraten von einer Quell-IP zu verschiedenen Ziel-Ports überwacht und begrenzt. Von der Überwachung ausgenommen sind die Ziel-Ports, für die eine eigene Signatur angelegt wurde. Da die Signatur für mehrere Ziel-Ports übergreifend gilt, wurde der Schwellenwert relativ hoch angesetzt.

```
drop ip $HOME_NET any -> any [1:65535, ![25,53,80,443]] (
msg:"DoS Attack";
flow:to_server, not_established;
classtype:bad-unknown; sid:8; rev:1;
detection_filter: track by_src, count 500, seconds 1;)
```

4.3. Intrusion Prevention System

Da kein IPS über die Package Manager der pfSense zur Verfügung steht, wurde dieses im Rahmen dieser Arbeit neu entwickelt.

4.3.1. Konfiguration

Die Konfiguration des IPS erfolgt über eine Konfigurationsoberfläche, die über die pfSense Weboberfläche unter *Services - SecomatIPS* aufgerufen werden kann. Das IPS wird über einen Start-Button aktiviert und über einen Stop-Button deaktiviert. Bei der Aktivierung wird ein Cron Job erstellt, der das IPS-Script jede volle Minute startet. Bei der Deaktivierung wird der Cron Job wieder entfernt.

Über die Konfigurationsoberfläche kann auch das E-Mail Benachrichtungssystem des IPS eingestellt werden. Es kann die E-Mail-Adresse konfiguriert werden, an die dauerhaft gesperrte IP-Adressen gemeldet werden. Es kann die Adresse des SMTP-Servers konfiguriert werden, über den die E-Mails versendet werden. Außerdem kann die Absender E-Mail-Adresse der E-Mails konfiguriert werden. Es kann auch der Schwellenwert in Tagen angegeben werden, bei dem dauerhaft gesperrte IP-Adressen per E-Mail gemeldet werden.

Über ein weiteres Eingabefeld können IP-Adressen angegeben werden, die nicht durch das IPS gesperrt werden dürfen. Die Einstellungen des IPS werden in der pfSense Konfigurationsdatei config.xml im Knoten *installedpackages* mit dem Unterknoten *secomatIPS* gespeichert.

Beim Speichern von Änderungen wird die geänderte Konfiguration durch XML-RPC auf die Backup-Instanz synchronisiert.

4.3.2. Funktionsweise

Ist das IPS aktiviert, dann wird jede volle Minute ein Python Script durch einen Cron Job aufgerufen. Das Script lädt zu Beginn bei jedem Durchlauf seine Konfiguration aus der pfSense Konfiguration. Außerdem wird die Bezeichnung des LAN-Interface und der Universally

Unique Identifier (uuid) von Suricata aus der pfSense Konfiguration gelesen. Anhand dieser Werte wird der folgende Pfad zu den Log-Dateien von Suricata ermittelt:

/var/log/suricata/suricata_<interface><uuid>/

Nachdem das IPS konfiguriert und der Pfad zu den Log-Dateien ermittelt wurde, wird geprüft, ob es sich bei der pfSense Instanz, auf der das IPS ausgeführt wird, um den Master handelt. Der Status wird anhand der Interface Konfiguration des LAN-Interfaces ermittelt. Haben die darauf konfigurierten virtuellen IP-Adressen den Status „Master“, so handelt es sich um die Master-Instanz, ansonsten um die Backup-Instanz. Handelt es sich bei der pfSense Instanz um den Master, so wird mit der Analyse der Log-Dateien begonnen. Handelt es sich bei der pfSense Instanz jedoch um die Backup-Instanz, wird der aktuelle Durchlauf beendet.

Zu Beginn der Analyse der Log-Einträge wird der Startzeitpunkt erfasst, um zwischen aktuellen und veralterten Log-Einträgen zu unterscheiden. Dann werden die Log-Dateien von Suricata nacheinander verarbeitet. Es werden die aktuelle Log-Datei und die drei neuesten archivierten Log-Dateien eingelesen, soweit diese vorhanden sind. Die Log-Einträge in den Log-Dateien werden nacheinander eingelesen und verarbeitet. Dadurch muss nicht eine Log-Datei vollständig in den Arbeitsspeicher geladen werden. Ein Überlaufen des Arbeitspeichers wird so bei großen Log-Dateien verhindert.

Bei jedem Log-Eintrag wird zunächst der Zeitpunkt der Erstellung des Log-Eintrages mit String Slicing extrahiert. Der Zeitpunkt des Log-Eintrags wird mit den Startzeitpunkt der aktuellen Durchlaufs des IPS Scripts verglichen. Ist der Eintrag älter als 15 Minuten, so wird dieser übersprungen und nicht analysiert. Ist der Eintrag in den letzten 15 Minuten entstanden, wird anschließend die Quell-IP und somit die IP-Adresse des Absenders des geloggten Pakets aus dem Log-Eintrag extrahiert. Da sich die Quell-IP in den Log-Einträgen nach der Log-Nachricht der Signatur befindet und diese unterschiedlich lang sein kann, wird diese mit einer Regular Expression extrahiert. Die Quell-IP wird anschließend mit den IP-Adressen in der Whitelist verglichen. Ist die IP-Adresse in der Whitelist, dann wird der Log-Eintrag nicht weiter bearbeitet und zum nächsten Log-Eintrag gesprungen. Ist die IP-Adresse nicht in der Whitelist, so werden zusätzlich die Log-Nachricht und das Protokoll des Pakets aus dem Log-Eintrag mit Regualar Expressions extrahiert. Während der Analyse der Log-Einträge wird das JSON Objekt badgirls mit dem folgenden Struktur generiert:

```
{   <ip-address>: {
        totalHits: x,
        alerts: {
            <msg>: {
                hits: y,
                proto: z
            },
            ...
        }
    },
    ...
}
```

Ist noch kein Eintrag mit der extrahierten Quell-IP in dem JSON Objekt badgirls enthalten, so wird ein neuer Eintrag mit der IP-Adresse als Schlüssel angelegt. Mit totalHits wird gespeichert, wie viele Log-Einträge die IP-Adresse verursacht hat. Mit totalHits wird später ermittelt, ob eine IP-Adresse gesperrt oder entsperrt wird. Dazu wird totalHits mit Eins initialisiert. Für jeden weiteren Log-Eintrag, der der IP-Adresse zugeordnet wird, wird totalHits um eins erhöht. In alerts wird gespeichert, wie oft welche Log-Nachrichten verursacht wurden. Zusätzlich wird bei jeder Log-Nachricht das Protokoll des geloggten Pakets gespeichert.

Nach Abschluss der Analyse der Log-Einträge wird die Datei badgirlsstatistics.json aus dem Wurzelverzeichnis des Webservers geladen. Das, darin gespeicherte JSON Objekt, im folgenden als alreadyBadgirls bezeichnet, wird bei jedem Durchlauf des IPS Scripts aktualisiert und hat die folgende Struktur:

```
{
    <ip-address>: {
        blockedSince: n,
        emailsSend: o,
        alerts: {
            <msg>: {
                hits: y,
                proto: z
            },
            ...
        }
    },
    ...
}
```

Das JSON Objekt alreadyBadgirls enthält für jede bereits gesperrte IP-Adresse einen Eintrag. Für jede IP-Adresse ist der Zeitpunkt der Sperrung in blockedSince gespeichert. In emailsSend ist gespeichert, wie oft die jeweilige IP-Adresse bereits gemeldet wurde. Für die jeweiligen IP-Adressen sind in alerts die Ergebnisse der Analyse der Log-Einträge aus dem vorherigen Durchlauf des IPS-Scripts gespeichert.

Die Einträge der IP-Adressen, die nach der aktuellen Analyse der Log-Dateien weniger als 120 Log-Einträge verursacht haben, werden aus alreadyBadgirls entfernt. Ist eine IP-Adresse bereits gesperrt und werden ihr weiterhin über 120 Log-Einträge zugeordnet, dann wird der Inhalt von alerts durch den aus badgirls ersetzt. Für IP-Adressen, die noch keinen Eintrag in alreadyBadgirls haben und über 120 Log-Einträge verursacht haben, wird ein neuer Eintrag angelegt.

Darauf folgt die Berechnung der Dauer der Sperrung der IP-Adressen in alreadyBadgirls anhand des Datums in blockedSince und des aktuellen Startzeitpunkts des IPS-Scripts. Wird durch die Dauer der Sperrung einer IP-Adresse der konfigurierte Schwellenwert erreicht, wird die IP-Adresse per E-Mail an die angegebene Report E-Mail-Adresse gemeldet. Die E-

Mail wird über den SMTP-Server verschickt, dessen Adresse in der Konfiguration angegeben wurde. Erreicht die Dauer der Sperrung, seit der letzten Meldung, erneut den hinterlegten Schwellenwert, wird wieder eine E-Mail versendet. War das Senden einer E-Mail erfolgreich, wird emailsSend um eins erhöht.

Nach der Aktualisierung wird alreadyBadgirls wieder zurück in die Datei badgirlsstatistics.json geschrieben.

Damit die IP-Adressen in der Firewall gesperrt werden, müssen diese zunächst in den Alias BADGIRLS in der pfSense Konfigurationsdatei config.xml eingefügt werden. In der Konfigurationsdatei von pfSense wird das Schlüsselwort CDATA genutzt, um Strings zu kennzeichnen, die nicht durch den XML-Parser ausgewertet werden sollen. Auf der pfSense Firewall steht nur das Python Modul ElementTree für das Manipulieren von XML-Bäumen zur Verfügung. Das Modul ElementTree unterstützt jedoch nicht das Schlüsselwort CDATA, sodass es alle CDATA Schlüsselwörter entfernt, wenn der geänderte XML-Baum zurück in die pfSense Konfigurationsdatei geschrieben wird. Die pfSense Software erkennt die geänderte Konfiguration als invalide und ersetzt sie bei der nächsten Änderungen oder dem nächsten Neustart durch eine Backup Konfiguration. Der Installation eines zusätzlichen Moduls für das Editieren der pfSense Konfiguration wurde die Nutzung von bestehenden Funktionen von pfSense vorgezogen. Denn so kann der Quell-Code kurz gehalten und zum anderen werden so bestehende Routinen von pfSense für Änderungen an der Konfiguration, der Firewall und der Synchronisation genutzt und potentielle Fehlerquellen reduziert. Daraus folgt, dass die Sperrung der IP-Adressen über ein PHP Script erfolgt, das Funktionen der Bibliothek von pfSense nutzt, die in PHP geschrieben ist. Aus dem Python Script wird das PHP Script aufgerufen und die zu sperrenden IP-Adressen, durch ein Komma getrennt, als String übergeben. Durch das PHP Script werden die IP-Adressen in den Alias BADGIRLS innerhalb der pfSense Konfiguration eingefügt und anschließend die Konfigurationsdatei gespeichert. Die geänderte Konfiguration wird mit XML-RPC auf die Backup Instanz synchronisiert. Abschließend wird auf beiden pfSense Instanzen der Alias BADGIRLS mit den zu sperrenden IP-Adressen in die Firewall geladen. Dadurch werden alle Pakete neuer Verbindungen von gesperrten Hosts durch die Firewall verworfen.

Nachdem die IP-Adressen über die pfSense Konfiguration in PF eingefügt und somit gesperrt wurden, müssen noch deren Einträge aus der Statetable gelöscht werden, damit auch die Pakete von bestehenden Verbindungen verworfen werden. Die Löschung erfolgt mit dem Werkzeug pfctl. Bei IPv4-Adressen werden die States anhand deren IDs gelöscht. Dazu werden die Statetable geladen, die ID der States der gesperrten IP-Adressen extrahiert und daraufhin die States gelöscht. Dieses Verfahren kann bei IPv6 nicht genutzt werden. Das liegt daran, dass die IPv6 Adressen in der Statetable in kurzer Notation gespeichert werden, d.h. u.a. mehrere Blöcke mit Nullen werden zusammengefasst. Bei den Log-Einträgen von Suricata werden die IPv6-Adressen nicht gekürzt gespeichert, d.h. mehrere Blöcke mit Nullen werden ausgeschrieben. Deshalb können die IDs der States der IPv6-Adresse nicht ermittelt werden. Aus diesem Grund werden die States direkt anhand der IPv6-Adresse gelöscht. Dadurch werden alle States für die IP-Adresse unabhängig vom Port gelöscht.

Dieses Verfahren kann nicht bei IPv4 mit SNAT genutzt werden, da hier auch die States für die NAT-IPs gelöscht werden müssen. Da die States für alle Ports der IP-Adresse gelöscht

werden, werden somit auch die States der nicht gesperrten Hosts gelöscht. Dies würde daruch auch bei den nicht gesperrten Hosts zu einem Timeout führen. Da dies nicht passieren darf, werden die zwei zuvor beschriebenen Verfahren genutzt.

4.4. Destination Network Address Translation

Durch die pfSense Firewall wird Destination Network Address Translation für IPv4 unterstützt und kann über die Weboberfläche konfiguriert werden. Damit nun HTTP-Anfragen über IPv4 von gesperrten Nutzern auf den lokalen Webserver umgeleitet werden, muss eine entsprechende Regel angelegt werden.

In der DNAT-Regel ist das LAN-Interface angegeben, damit die Pakete von Hosts aus dem lokalen Netz umgeleitet werden. Damit nur Pakete von gesperrten Nutzern umgeleitet werden, ist als Quell-IP der Alias BADGIRLS angegeben. Damit nur HTTP-Anfragen von gesperrten Nutzern umgeleitet werden, ist in der DNAT-Regel das Protokoll TCP und als Ziel-Port 80 definiert. Um die Pakete an das lokale System und somit den lokalen Webserver umzuleitet, ist als NAT-IP die Loopback IP-Adresse 127.0.0.1 angegeben. Als neuer Ziel-Port für die umgeleiteten Pakete ist der Port 80 definiert. Damit die Pakete, die auf die DNAT-Regel zutreffen, direkt weitergeschickt und nicht durch die Filterregel für BADGIRLS des LAN-Interface verworfen werden, ist die Option *Filter rule association* auf Pass gesetzt. Alle Pakete, die auf die DNAT-Regel zutreffen, werden sofort weitergeleitet und nicht mit den Filterregeln des LAN-Interfaces verglichen.

Neben der Umleitung von HTTP-Anfragen über IPv4, ist auch die Umleitung von IPv6 gefordert. Es konnte keine Lösung gefunden werden, durch die es auf der pfSense Firewall möglich ist, HTTP-Anfragen über IPv6 von einzelnen Hosts umzuleiten. Die pfSense und ihr Firewall Modul PF unterstützen nicht das Umleiten von IPv6-Paketen auf eine neue Ziel-IP. Deshalb konnte diese Anforderung nicht migriert werden.

4.5. Webserver

Auf der pfSense wird der nginx Webserver für die Bereitstellung der pfSense Weboberfläche genutzt. Der nginx Webserver kann nicht für die Bereitstellung der Statusseite genutzt werden, da dieser mit den Systemrechten des root Benutzers ausgeführt wird. Dies würde ein großes Sicherheitsrisiko darstellen, da die gesperrten Hosts direkt mit dem Webserver interagieren. Ist es einem Angreifer möglich, die Kontrolle über den Webserver zu erlangen, kann dieser beliebige Änderungen an der pfSense Firewall vornehmen. Außerdem würde dies eine Änderung an der Konfiguration des Webservers erfordern, die zu Problemen bei Updates führen kann.

Aus den genannten Gründen wird ein weiterer Webserver auf der pfSense Firewall installiert. Die Wahl fiel auf den lighttpd Webserver, da dieser im Gegensatz zum Apache Webserver über die offiziellen Paketquellen von pfSense zur Verfügung steht. Dadurch ist die Kompatibilität des Webserver mit der pfSense sichergestellt, d.h. es gibt keine offenen Abhängigkeiten aufgrund des modifizierten FreeBSD Betriebssystems der pfSense.

4.5.1. Konfiguration

Die Konfigurationsoberfläche für den Webserver wurde neu entwickelt und kann nun über das Hauptmenü der pfSense Weboberfläche unter *Services - SecomatIPS* mit dem TAB *Webserver Einstellungen* aufgerufen werden. Dort kann der Webserver über einen Start-Button gestartet und über einen Stop-Button gestoppt werden. Außerdem lässt sich die Konfiguration des Webservers über ein Eingabefeld ändern.

4.5.2. Funktionsweise

Während der Installation des Webserver wird automatisch ein neuer Nutzer „www“ mit der Gruppe „www“ angelegt. In der Konfiguration des Webservers werden der Nutzer „www“ und die Gruppe „www“ angegeben. Dadurch wird der Webserver mit den Rechten des Nutzers „www“ gestartet. Der Nutzer „www“ hat nicht die Rechte des root Benutzers, deshalb kann dieser keine Änderungen an der Konfiguration von pfSense vornehmen.

Der Webserver unterstützt HTTP-Requests über IPv4 und IPv6. In der Konfiguration des Webservers werden Sockets mit dazugehörigem Deamon definiert, sodass der Webserver auf Anfragen an eine oder beliebige IP-Adressen mit zugehörigem Port lauscht. Damit der Webserver auf alle HTTP-Anfragen unabhängig von der Ziel-IP antwortet, wird 0.0.0.0 als IP-Adresse angegeben.

Durch die Einbindung der Module mod_redirect und mod_rewrite können der Hostname und der Pfad im HTTP-Header des HTTP-Requests modifiziert werden. In der Konfiguration des Webservers wird mit URL-Redirection eine Weiterleitung festgelegt, die alle Anfragen auf die virtuelle IP-Adresse des LAN-Interfaces umleitet, wenn der Hostname im HTTP-Header nicht mit der virtuellen IP-Adresse übereinstimmt. Aktualisiert der Nutzer die Seite, dann werden die Anfragen direkt an die aktive pfSense Instanz und den darauf installierten Webserver geschickt. Sollte die Master-Instanz ausfallen, werden die HTTP-Requests automatisch an die Backup-Instanz geschickt, da die Anfragen an die virtuelle IP-Adresse gerichtet sind. Enthält der Pfad im HTTP-Header Unterverzeichnisse, wie beispielsweise /pfad/zur/datei/, werden diese durch eine Rewrite Regel entfernt. Dadurch richten sich alle Anfragen immer an das Wurzelverzeichnis (/), in dem sich die Statusseite befindet. Ansonsten würde der Webserver den HTTP-Statuscode 404 zum gesperrten Nutzer zurückschicken, da die angefragte Ressource nicht auf dem Webserver gefunden werden konnte.

Abschließend wird in der Konfiguration PHP aktiviert. Dazu wird zunächst das Modul mod_fastcgi eingebunden. Anschließend wird der Pfad zum PHP Interpreter und fastcgi Socket angegeben.

Bevor der lighttpd Webserver gestartet werden kann, muss noch die Weiterleitung von Port 80 auf 443 des nginx Webservers in der pfSense Weboberfläche deaktiviert werden. Unter **System - Advanced** wird WebGUI redirect aktiviert und dadurch die Weiterleitung deaktiviert. Dadurch wird der Port 80 nicht mehr durch den nginx Webserver belegt und kann für die Bereitstellung der Statusseite genutzt werden.

4.6. Statusseite

4.6.1. Konfiguration

Für die Konfiguration der Statusseite wurde eine weitere Weboberfläche erstellt. Diese kann über das Hauptmenü unter *Services* - *SecomatIPS* beim TAB *Statusseite* aufgerufen werden. Die Konfiguration der Statusseite erfolgt über ein Eingabefeld. Dort kann die Statusseite vollständig individualisiert werden. Änderungen an der Statusseite auf der Master-Instanz werden über XML-RPC auf die Backup-Instanz synchronisiert.

4.6.2. Funktionsweise

Die Statusseite wird mit PHP dynamisch generiert. Mit PHP wird die Datei badgirlsstatistics.json geladen und geprüft, ob ein Eintrag für die IP-Adresse des anfragenden Clients in dem JSON Objekt gespeichert ist. Ist die IP-Adresse des anfragenden Clients nicht im JSON Objekt gespeichert, ist diese auch nicht gesperrt. Dem Nutzer wird dann der Status „Internet access“ angezeigt, wie in Abbildung 4.2 zu sehen ist. Als zusätzliche Information wird dem Nutzer seine IP-Adresse angezeigt. Befindet sich die IP-Adresse jedoch in dem JSON Objekt, wird dem Nutzer der Status „No Internet Access“ angezeigt. Ist der Nutzer gesperrt, werden ihm die Hinweistexte in deutscher und englischer Sprache angezeigt, wie beim Secomat. Im unteren Abschnitt der Seite wird dem Nutzer seine gesperrte IP-Adresse dargestellt. Anhand der IP-Adresse des Nutzers wird diesem der Zeitpunkt seiner Sperrung angezeigt, damit dieser mögliche Ursachen eingrenzen kann. Des Weiteren wird der aktuelle Grund der Sperrung in Tabellenform ausgegeben. Die Statusseite eines gesperrten Nutzers ist in Abbildung 4.1 zu sehen.

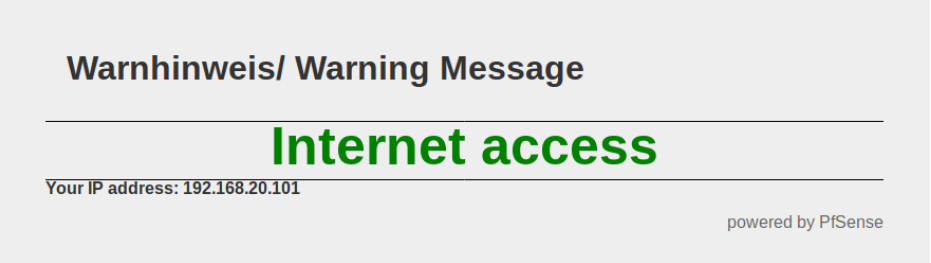

Abbildung 4.1.: Default Statusseite für nicht-gesperrte Nutzer

4.7. Package

Es wurde ein FreeBSD Package entwickelt, dass die Migration einzelner Funktionen und deren Konfiguration automatisiert. Die Installation erfolgt über den FreeBSD Package Manager.

Die Konfiguration des Package ist in einem Manifest gespeichert, das sich im Wurzelverzeichnis des Package befindet. Darin werden Metadaten, Abhängigkeiten, zu erstellende Verzeichnisse, zu kopierende Dateien und Shell-Scripte, die vor und nach der Installation oder Deinstallation ausgeführt werden sollen, angegeben. Die im Manifest angegebene Verzeichnisstruktur und Dateien sind identisch mit denen im Package vorhandenen Verzeichnissen und Dateien.

In dem Package wurde der lighttpd Webserver als Abhängigkeit angegeben, sodass das Package nur installiert wird, wenn der Webserver bereits installiert ist. Sollte der lighttpd Webserver noch nicht installiert sein, so wird eine Fehlermeldung ausgegeben und das Package nicht installiert.

Im Manifest werden die zu kopierenden Dateien zusammen mit einem Hashwert angegeben. Bei einer späteren Deinstallation des Package, werden die Dateien mit den, in einer Datenbank hinterlegten Hashwerten verglichen und nur Dateien entfernt, deren Hashwert mit dem bei der Installation identisch ist. Dadurch bleiben Konfigurationsänderungen an der Statusseite auch nach der Deinstallation erhalten.

Mit den Shell-Scripten im Manifest werden PHP-Scripte nach der Installation und vor der Deinstallation aufgerufen. Durch diese werden bei der Installation der Alias BADGIRLS, die Filterregel für die Firewall, die DNAT-Regel für das Umleiten von HTTP-Anfragen, die Konfiguration des IPS und des Package in die pfSense Konfiguration eingefügt. Wird das Package wieder deinstalliert, werden die Änderungen an der pfSense Konfiguration wieder rückgängig gemacht.

Um die Erzeugung des Package zu automatisieren, wurde ein Python Script entwickelt. Mit dem Python Script werden die Verzeichnisse des Package ermittelt und im Manifest aktualisiert. Durch das Script werden auch die Dateien im Package ermittelt und der dazugehörige Hashwert berechnet. Anschließend werden die Pfade zu den Dateien und die Hashwerte im Manifest aktualisiert. Nach der Aktualisierung wird das Package erzeugt, das auf der pfSense Firewall installiert werden kann.

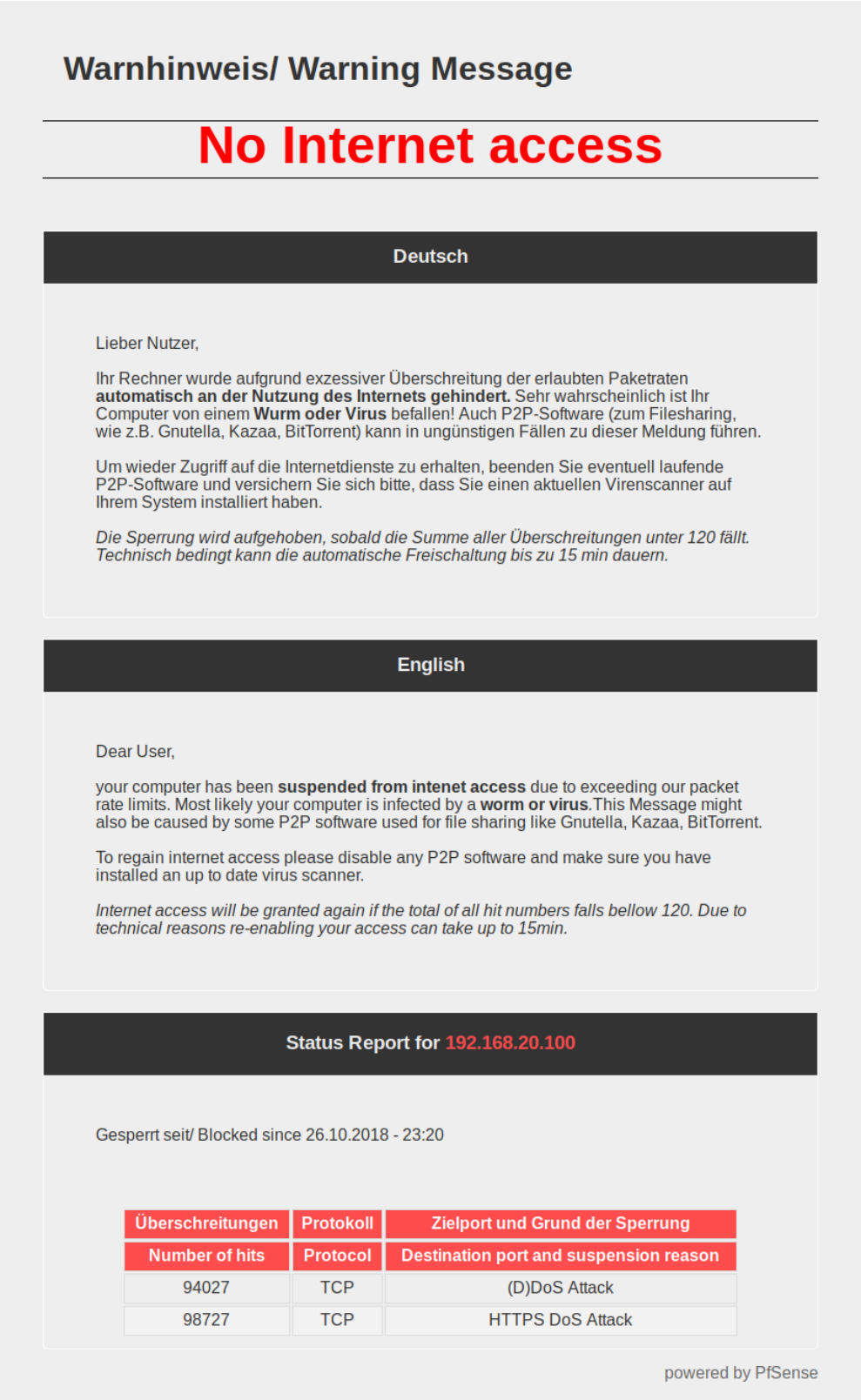

Abbildung 4.2.: Default Statusseite für gesperrte Nutzer

5. Evaluation

In der Evaluation wurden die in der Migration erarbeiteten Funktionen getestet und gegenüber den Anforderungen evaluiert.

5.1. Testumgebung und Konfiguration

Die Testumgebung wurde in Virtual Box aufgebaut und ist in der Abbildung 5.1 zu sehen. Sie besteht aus drei virtuellen pfSense Firewall Instanzen und zwei virtuellen Ubuntu Instanzen.

Die zwei pfSense Instanzen, Master und Backup genannt, werden für die Evaluierung der migrierten Funktionen genutzt und haben beide drei Interfaces. Über ihr LAN-Interface sind die beiden pfSense Firewalls mit dem virtuellen Netz LAN verbunden. Im LAN befindet sich ebenso eine Ubuntu Instanz, die die Rolle eines Angreifers übernimmt.

Über das WAN-Interface sind beide pfSense Instanzen mit dem virtuellen Netz WAN verbunden, mit dem auch die zweite Ubuntu Instanz und die dritte pfSense Firewall verbunden ist. Das WAN steht stellvertretend für das MWN und Internet. Über die, in der Abbildung als Gateway bezeichnete pfSense Instanz, haben insbesondere die pfSense Instanzen Master und Backup Zugang zum Internet, um den Webserver und die Packages auf die pfSense Firewall herunterzuladen und zu installieren. Auf die Ubuntu Instanz im WAN wurden die Angriffe durchgeführt. Diese steht stellvertretend für potentielle Opfer eines Angriffs aus dem lokalen Netz der pfSense Firewall auf Ziele im MWN oder Internet.

Die pfSense Instanzen Master und Backup sind jeweils über ein drittes Interface mit dem Netz HA verbunden, über das die Konfiguration von der Master- auf die Backup-Instanz synchronisiert wird.

Auf den beiden pfSense Instanzen Master und Backup wurden jeweils eine virtuelle IPv4 und eine virtuelle IPv6-Adresse für das LAN- und WAN-Interface konfiguriert. Bei den Ubuntu Instanzen sind jeweils die virtuellen IP-Adressen aus ihrem Netz als Gateways konfiguriert.

Die Weiterleitung von Port 80 auf 443 wurde auf beiden pfSense Instanzen deaktiviert, damit der Port 80 für den lighttpd Webserver genutzt werden kann.

Um während der Tests auf die Konsole der pfSense Firewall zugreifen zu können, wurde SSH auf beiden pfSense Instanzen aktiviert.

Auf beiden pfSense Instanzen wurde eine Filterregel für das WAN-Interface definiert, sodass während der Tests mit der Ubuntu Instanz im WAN auf die pfSense Instanzen zugegriffen werden kann.

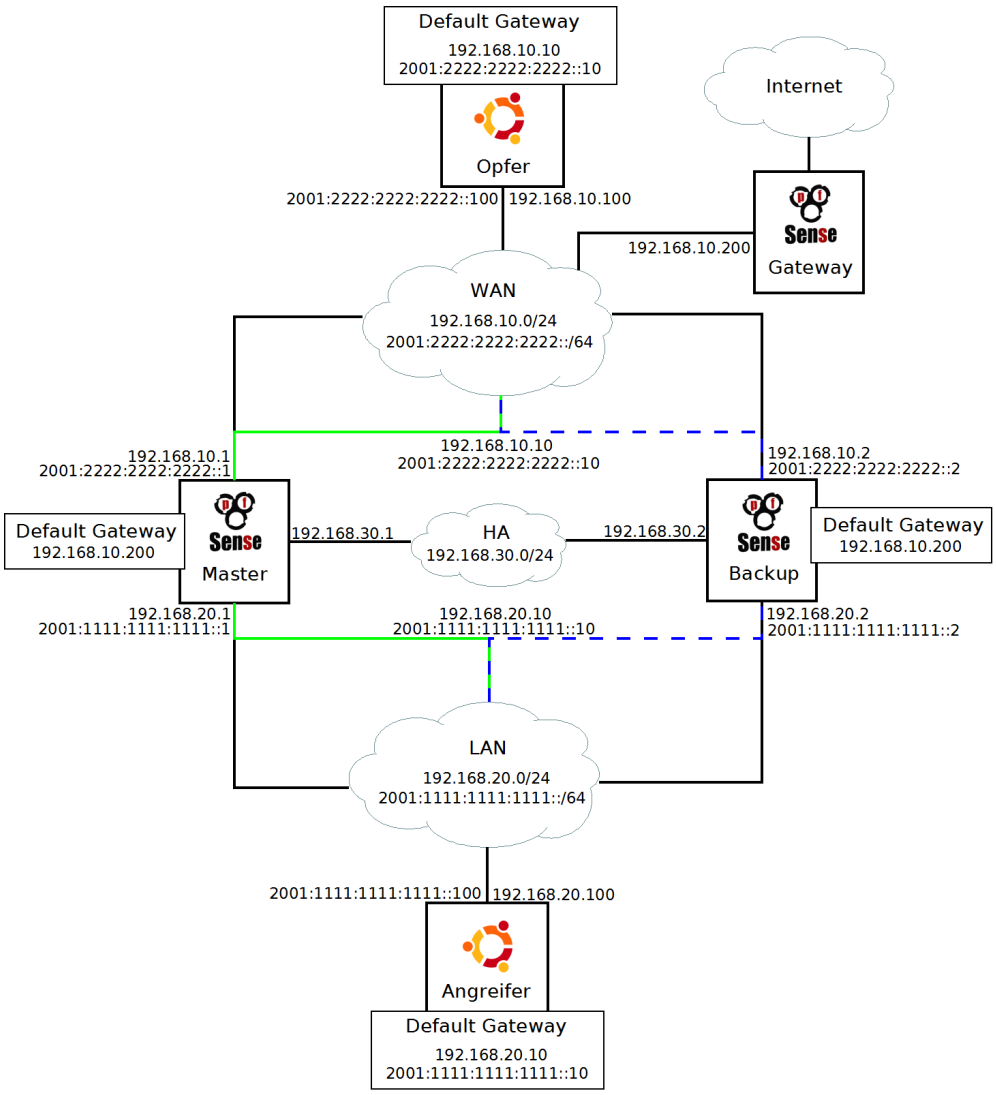

Abbildung 5.1.: Testumgebung

5.2. Testdurchführung und Testergebnisse

5.2.1. Installation, Konfiguration und Synchronisation

Zu Beginn wurden das Package Suricata, der Webserver und das IPS-Package zunächst auf die Backup-Instanz und anschließend auf die Master-Instanz installiert.

Dazu wurde das Package Suricata über den Package Manager in der Weboberfläche heruntergeladen und installiert. Anschließend wurde der lighttpd Webserver über den FreeBSD Package Manager installiert. Die Installation erfolgte über einen Shell Befehl, der über die pfSense Weboberfläche unter *Diagnostics -> Command Prompt* eingegeben und daraufhin ausgeführt wurde. Darauf folgte die Installation des erstellten IPS-Package, das ebenfalls über den FreeBSD Packager Manager durch einen Shell Befehl installiert wurde.

Die Anforderung, den Funktionsumfang der pfSense Software anhand von Packages zu erweitern, konnte somit erfüllt werden.

Nach der Installation erfolgte die Konfiguration der Funktionen auf der Master-Instanz. Bei Suricata wurden die in der Migration vorgestellten Einstellungen vorgenommen und die Signaturen angelegt. Die Konfiguration des Webservers wurde über die erstellte Konfigurationsseite vorgenommen. Dort wurden die Umleitungen der HTTP-Requests über IPv4 und IPv6 auf die virtuellen IP-Adressen des LAN-Interfaces geändert. Bei der Statusseite wurden Kontaktinformationen über die erstellte Konfigurationsseite ergänzt. Beim Intrusion Prevention System wurde das E-Mail-Benachrichtigungssystem und die Whitelist über die erstellte Konfigurationsseite eingestellt. Die Anforderung, dass die Konfiguration des IDS, des IPS, des Webservers und der Statusseite über die Konfigurationsseiten vorgenommen werden können, wurde somit erfüllt.

Nach der Konfiguration wurde geprüft, ob die Funktionen auf die Backup-Instanz synchronisiert wurden. Dazu wurde die pfSense Weboberfläche auf der Backup-Instanz geöffnet und die vorgenommenen Einstellungen abgeglichen. Die Einstellungen stimmten mit denen auf der Master-Instanz überein. Das Starten und Stoppen von Suricata, des Webservers und des IPS erfolgte getrennt auf beiden pfSense Instanzen und erfüllt damit nicht die Anforderungen an die Synchronisation. Damit die Synchronisation des Webservers auf der Backup-Instanz übernommen wird, muss dieser dort einmal gestoppt und wieder gestartet werden. Die Anforderungen an die Synchronisation konnte daher nur teilweise erfüllt werden.

5.2.2. Intrusion Detection System

Bereits während der Migration hat sich gezeigt, dass nicht alle Anforderungen durch Suricata auf der pfSense Firewall erfüllt werden können. Suricata hat kein Burstsystem, d.h. es werden alle Pakete bei Überschreitung eines Schwellenwertes unabhängig von der Dauer geloggt und verworfen. Eine weitere wesentliche Einschränkung ist, dass alle Pakete geloggt werden, die durch das IDS verworfen werden. Dies führt dazu, dass die Log-Einträge nicht pro Zeitfenster beschränkt werden und somit ein „Fluten“ der Log-Datei mit Log-Einträgen nicht verhindert werden kann.

Um die mit Suricata umgesetzten Funktionen zu evaluieren, wurden ICMP-, UDP- und SYN-Flooding Angriffe über IPv4 und IPv6 von der Ubuntu Instanz im LAN auf die im WAN durchgeführt. Bei IPv4 wurden die Tests mit und ohne SNAT durchgeführt, dabei konnten jedoch keine wesentlichen Unterschiede bei den Testergebnissen festgestellt werden. Die Paketraten wurden auf beiden Ubuntu Instanzen mit dem Paketsniffer Wireshark ausgewertet.

Für die Angriffe über IPv4 wurde das Penetration Testing Tool hping3 genutzt, das mit dem Parameter flood in einen Flooding Modus geschaltet werden kann, bei dem so viele Pakete wie möglich generiert werden. In diesem Modus konnten in der Testumgebung Paketraten um die 30.000 Pakete/Sekunde generiert werden. Bei längeren und wiederholten Tests kam es in der Testumgebung zeitgleich zum Ausfall des LAN-Interface bei der Master- und der Backup-Instanz. Auch nach Abbruch eines Angriffs waren das LAN-Interface der Master- und der Backup-Instanz nicht mehr erreichbar. Durch Aufruf der Weboberfläche über das

WAN-Interface auf der Ubuntu Instanz im WAN wurde festgestellt, dass bei beiden pfSense Instanzen die virtuellen IP-Adressen den Status Master angenommen haben. Dieses Problem trat bei Tests mit geringeren Paketraten mit bis zu 20.000 Paketen/Sekunde nicht auf.

Um die Erkennung von ICMP-Flooding über IPv4 zu testen, wurden Echo-Requests in verschieden hohe Paketraten zum Opfer geschickt. Suricata hat die Überschreitungen erkannt und die Pakete nach Erreichen des Schwellenwertes verworfen. Bei längeren Tests mit Paketraten um die 18.000 Pakete/Sekunde traten in unregelmäßigen Abständen Schwankungen auf, wie in Abbildung 5.2 zu sehen ist. Die Schwankungen traten bei niedrigeren Paketraten seltener oder nicht auf. Daraus wurde geschlossen, dass diese wahrscheinlich am unbegrenzten Loggen durch Suricata liegen.

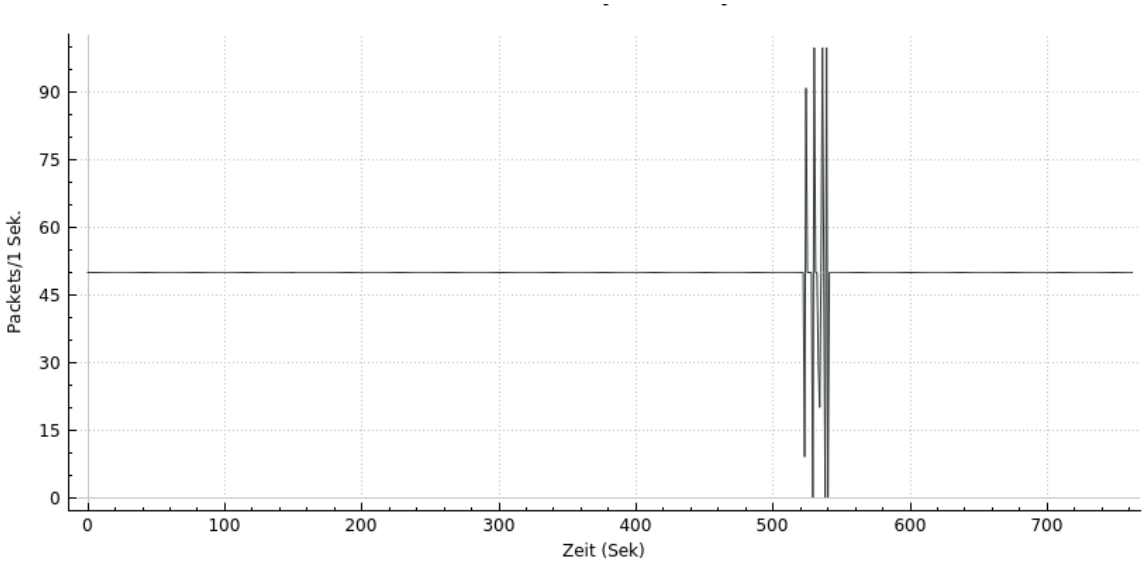

Abbildung 5.2.: ICMP-Flooding - Paketraten beim Opfer

Zum Simulieren eines ICMP-Flooding Angriffs über IPv6 wurde der Befehl „ping“ genutzt. Andere Tools wurden getestet, jedoch konnte mit diesen keine höhere Paketrate als 130 Pakete/Sekunde generiert werden. Die gesendeten Echo-Requests wurden nicht durch die definierte Signatur zur Erkennung von ICMP-Flooding als Überschreitung detektiert und es erfolgte keine Limitierung der Paketraten. Suricata speichert bei ICMPv6 über IPv6 einen Status, sobald ein Echo-Reply für einen gesendeten Echo-Request empfangen wurde. Alle weiteren gesendeten Echo-Requests werden der Verbindung zugeordnet und fließen dadurch nicht in die Paketratenüberwachung und -begrenzung ein. Um neben Ping-Scans auch ICMP-Flooding zu erkennen, wurden die bestehenden Signaturen um folgende Signatur erweitert:

Ping-Scans und ICMP-Flooding über IPv6:

```
drop icmp $HOME_NET any -> any any (msg:"ICMPv6 too many pings";
classtype:bad-unknown; sid:9; rev:1;
flow: stateless;
detection_filter: track by_src, count 50, seconds 1;)
```

Durch das Schlüsselwort flow mit dem Wert stateless wird erreicht, dass alle ICMPv6 Pakete eines Hosts im lokalen Netz in die Paketratenüberwachung einbezogen werden - Auch die Pakete, die einer bestehenden Verbindung zugeordnet werden. Diese Signatur gilt jedoch nur für ICMPv6 und nicht für ICMP bei IPv4, d.h. durch diese werden keine ICMP Pakete über IPv4 getrackt.

Um UDP-Flooding zu testen, wurde Netcat auf der Ubuntu Instanz im WAN gestartet und so konfiguriert, dass es nach UDP-Paketen über IPv4 und IPv6 mit Ziel-Port 443 lauscht. Anschließend wurden UDP-Flooding Angriffe über IPv4 und IPv6 auf den Ziel-Port 443 simuliert. Bei IPv4 und Paketraten bis zu 16.000 Paketen/Sekunde hat Suricata die Paketraten begrenzt. Bei einigen Durchläufen kam es zu kurzzeitigen Schwankungen, bei denen die Paketraten für den Bruchteil einer Sekunde auf das Doppelte des Schwellenwertes anstiegen, wie in Abbildung 5.3 zu sehen ist. Da die Schwankungen in unregelmäßigen Abständen und vor allem bei hohen Paketraten mit über 17.000 Paketen/Sekunde auftraten, liegt die Ursache der Schwankungen wahrscheinlich am unbegrenzten Loggen durch Suricata.

Für die Simulation von UDP-Flooding wurde bei IPv6 das Werkzeug nping genutzt. Dabei konnten beim Angreifer Paketraten bis zu 6.000 Pakete/Sekunde generiert werden. Bei UDP-Paketen über IPv6 wurden die Paketraten durch Suricata auf 60 Pakete/Sekunde begrenzt und alle weiteren Pakete verworfen.

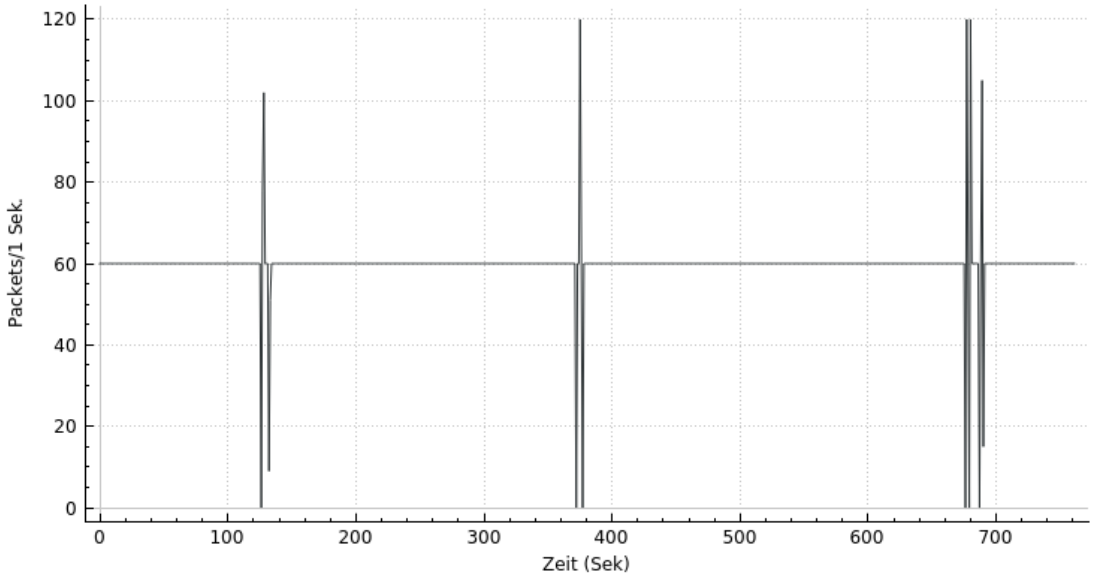

Abbildung 5.3.: UDP-Flooding - Paketraten beim Opfer

Für das Testen von SYN-Flooding wurde erneut hping3 für IPv4 genutzt und SYN-Pakete mit verschieden hohe Paketraten zum Opfer geschickt. Bei der Durchführung von SYN-Flooding Angriffen hat Suricata die TCP-Pakete bereits einer bestehenden Verbindung zugeordnet, sobald ein Antwort-Paket vom Opfer empfangen wurde. Ist der Ziel-Port geschlossen, antwortet das Opfer mit TCP-Paketen die das Flag RST/ACK haben. Im oberen Graph der Abbildung 5.4 ist zu sehen, dass die Paketraten mit der Sendung eines SYN-Paketes mit einem bereits genutzten Quell-Port stufenweise angestiegen. Sendet der Angreifer erneut TCP-Pakete mit dem Flag SYN und gleichem Quell-Port, werden die Pakete nicht getrackt, sodass die Paketraten beim Opfer steigen. Ist beim Opfer ein Dienst, wie in der Testumgebung ein Apache Webserver auf dem Port aktiv, werden durch das Opfer TCP-Pakete mit dem Flag SYN/ACK zurückgeschickt, um den Verbindungsaufbau zu bestätigen. Auch in diesem Fall stiegen die Paketraten beim Opfer sobald SYN-Pakete mit bereits genutztem Quell-Port empfangen wurden, wie im unteren Graph der Abbildung 5.4 zu sehen ist. Auch bei SYN-Flooding über IPv6 mit dem Werkzeug nping sind die Paketraten ebenfalls gestiegen, sobald SYN-Pakete mit bereits genutztem Quell-Port erneut gesendet wurden.

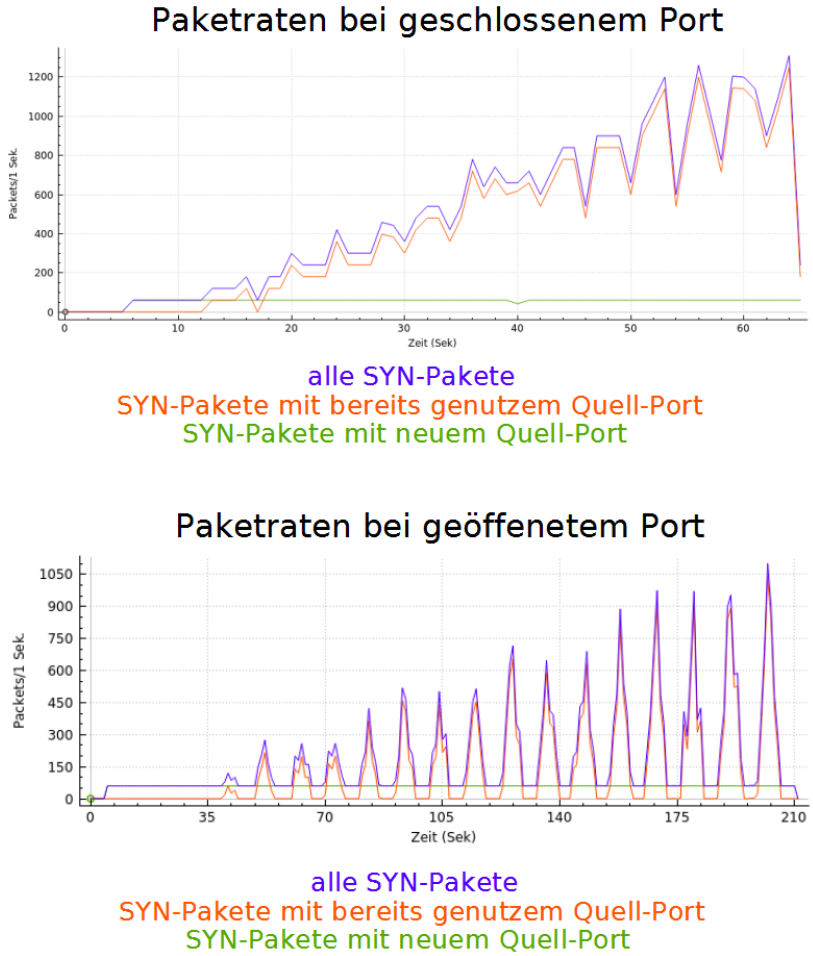

Abbildung 5.4.: SYN-Flooding - Signatur mit flow: not_established

Um die Paketraten zuverlässig zu begrenzen, wurden die Signaturen zur Erkennung von SYN-Flooding angepasst. Bei dem Schlüsselwort flow wird statt not_established, stateless eingetragen. Zusätzlich wird mit dem Schlüsselwort flags und dem Wert S definiert, dass durch die Signatur nur TCP-Pakete mit dem Flag SYN getrackt werden. Dadurch werden die Paketraten für alle TCP-Pakete mit dem Flag SYN unabhängig vom Verbindungsstatus überwacht. Die Änderung wird anhand der Signatur zur Erkennung von SYN-Flooding Angriffen auf Ziel-Port 80 gezeigt:

geänderte Signatur zur Erkennung von SYN-Flooding:

```
drop tcp $HOME_NET any -> any 80 (msg:"HTTP DoS Attack";
flow:to_server, stateless; flags: S;
classtype:bad-unknown; sid:4; rev:1;
detection_filter: track by_src, count 60, seconds 1;)
```

Anschließend wurden erneut SYN-Flooding Angriffe auf den Port 80 simuliert. Durch die Änderung der Signatur wurden die Paketraten erfolgreich begrenzt, jedoch sind die Paketraten bei IPv4 teilweise vollständig auf Null gefallen, siehe Abbildung 5.5. Diese Einbrüche traten bei SYN-Flooding Angriffen über IPv6 mit gleichen Paketraten um die 6000 Pakete/Sekunde nicht auf. Dort wurden beim Opfer 60 Pakete/Sekunde empfangen.

Mit Suricata werden somit ICMP-, UDP- und SYN-Flooding Angriffe über IPv4 und IPv6 erkannt, aber die Paketraten nicht immer zuverlässig begrenzt. Suricata erfüllt damit nicht immer die Anforderung, dass alle Pakete nach erreichen des Schwellenwertes verworfen werden.

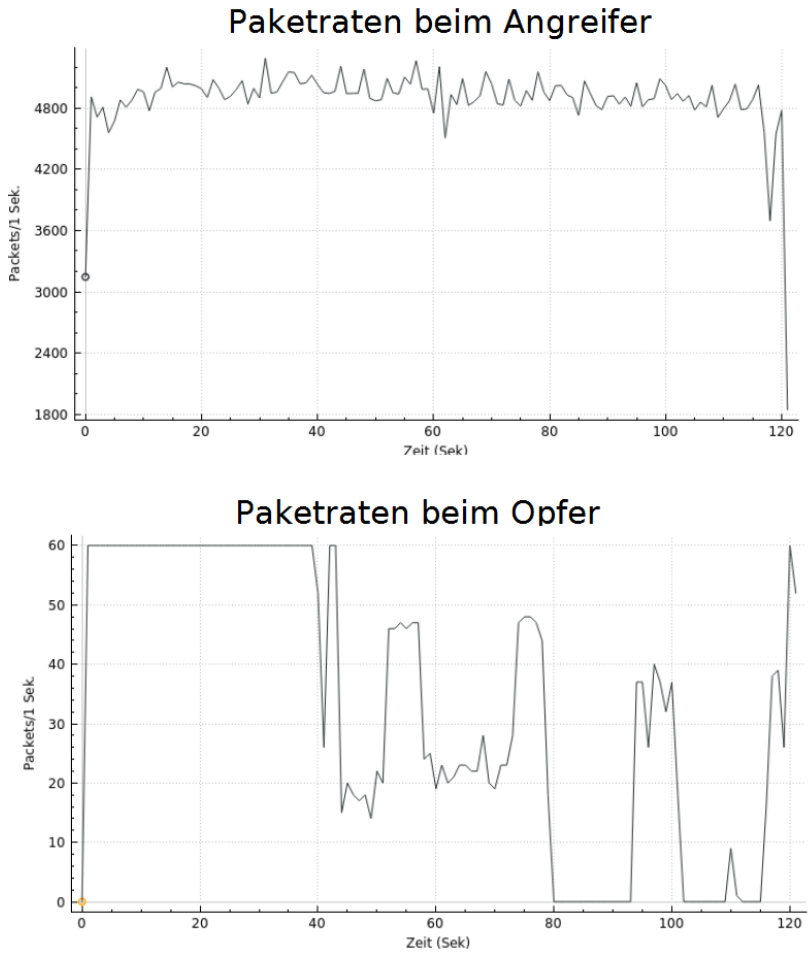

Abbildung 5.5.: SYN-Flooding - Signatur mit flow: stateless und flags: S

5.2.3. Intrusion Prevention System und Firewall

Zu Beginn der Evaluierung wurde die Leistungsfähigkeit des IPS-Scripts getestet. Durch hohe Paketraten wurden nacheinander mehrere Log-Dateien mit der Größe eines halben Gigabytes erzeugt und jeweils anschließend das IPS-Script über SSH gestartet.

Anhand von Zeitstempeln, die vor und nach der Verarbeitung der Log-Einträge ausgegeben werden, wurde die Dauer der Analyse berechnet. Für eine Log-Datei mit der Größe eines halben Gigabytes und Log-Einträge innerhalb der 15 Minuten vor dem Start des IPS-Scripts, benötigte es ungefähr 35 Sekunden. Für jedes weitere halbe Gigabyte verlängert sich die Dauer der Verarbeitung der Log-Datei(en) um weitere 35 Sekunden.

Anschließend wurde geprüft, wie lange für die Verarbeitung von Log-Einträgen benötigt wird, wenn diese älter als 15 Minuten sind. Für die Analyse einer Log-Datei mit veralterten Log-Einträgen und der Größe eines halben Gigabytes benötigte das Script neun Sekunden. Für jedes weitere halbe Gigabyte verlängerte sich die Verarbeitung um weitere neun Sekunden.

Eine Log-Datei mit der Größe eines halben Gigabytes entspricht ungefähr drei Millionen Log-Einträgen. Da Suricata alle Pakete nach der Überschreitung eines Schwellenwertes loggt, können die Log-Dateien bei hohen Paketraten mehrere Gigabyte groß werden. Durch die lange Laufzeit bei der Analyse der Log-Einträge und durch das Starten des IPS-Scripts durch den Cron Job jede Minute, werden immer mehr Prozesse mit dem IPS-Script gleichzeitig ausgeführt. Da die Rechenkapazitäten begrenzt sind, verlängert sich die Laufzeit durch jeden weiteren Start des IPS-Scritps weiter. Dieser Vorgang wiederholt sich solange bis das System nicht mehr reagiert und möglicherweise abstürzt. Um dies zu verhindern, wurde das IPS-Script um eine Abfrage zu Beginn des Script erweitert, die anhand eines pid-file prüft, ob bereits ein IPS-Script aktiv ist. Das pid-file wird zu Beginn des Scripts erstellt und am Ende des Durchlaufes wieder gelöscht. Ist das pid-file bei der Prüfung vorhanden, dann wird der Durchlauf beendet. Daraus folgt, dass das Script nicht mehr jede Minute vollständig ausgeführt wird, sobald die Laufzeit des Scripts eine Minute überschreitet. Dadurch wird auf der einen Seite die Anforderung nicht mehr erfüllt, dass das IPS jede Minute durchlaufen wird, aber auf der anderen Seite wird so der Absturz der pfSense Firewall verhindert. Um die Laufzeit des Script auch bei Log-Dateien, die zusammen mehr als ein Gigabyte groß sind auf unter eine Minute zu senken, kann zunächst die Auswertung der Log-Einträge optimiert werden. Da das Log-Format von alerts.log nicht geändert werden kann, muss entweder die Auswertung durch die Regular Expressions optimiert werden oder es wird „EVE JSON Output“ von Suricata genutzt. Dabei werden die Log-Einträge im JSON Format gespeichert. Dieses wurde in der vorliegenden Arbeit nicht genutzt, da das Loggen im JSON Format einen Overhead aufgrund der zusätzlichen Schlüssel bei jedem Log-Eintrag bedeuten würde. Außerdem ist die alerts.log fest in das Suricata Package integriert, d.h. das Loggen konnte nicht durch Konfigurationsänderungen deaktiviert werden. Es ist zwar möglich die Log-Datei alters.log zu löschen, wodurch keine Log-Einträge darin gespeichert werden. Diese Log-Datei wird jedoch bei einem Neustart von Suricata für das überwachte Interface neu erstellt, wodurch darin wieder Log-Einträge gespeichert werden. Eine weitere Optimierung am IPS-Script ist die Verteilung der Auswertung der Log-Dateien auf mehrere Threads, sodass die Log-Einträge parallel ausgewertet werden. Aber auch dann besteht dass Problem, dass das Skript nicht jede Minute ausgeführt wird, wenn eine Log-Datei so groß ist, dass diese nicht innerhalb einer Minute verarbeitet werden kann.

Die Ursache des Problems liegt bei Suricata, durch das alle verworfenen Pakete in „alerts.log“ geloggt werden. Um die Log-Einträge zu reduzieren gibt es zwei Optionen, bei denen jedoch nicht mehr alle Pakete nach Erreichen des Schwellenwertes verworfen werden. Dadurch wird die Anforderung nicht mehr erfüllt, dass alle Pakete durch das IDS verworfen werden. Bei der ersten Option, werden nur die Pakete verworfen und geloggt, sobald der Schwellenwert oder ein Vielfaches des Schwellenwertes erreicht wird. Dadurch würde die Paketrate abhängig vom Schwellenwert nur leicht, aber die Zahl der Log-Einträge stark, reduziert. Alternativ können nur die Pakete geloggt und verworfen werden, wenn ein Schwellenwert erreicht wurde. Dadurch werden die Paketraten nicht mehr durch das IDS begrenzt, aber die Zahl der Log-Einträge würde sehr stark reduziert. Bei dieser Variante müssten jedoch auch der Schwellenwert im IPS angepasst werden, ab dem eine IP-Adresse gesperrt wird, da nur noch ein Log-Eintrag pro Zeitfenster erzeugt wird.

Neben der langen Laufzeit durch große Log-Dateien, ergibt sich aus der Begrenzung des Spei-

cherplatzes für das Log-Verzeichnis und somit der Log-Dateien eine weitere Einschränkung. Wird der Speicherplatz für das Log-Verzeichnis so klein gewählt, dass alle bereits archivierten Log-Dateien durch die Archivierung einer aktuellen, größeren Log-Datei überschrieben werden, dann kann das IPS-Script nicht mehr alle Log-Einträge der letzten 15 Minuten auswerten kann, da die Log-Datei(en) und somit die Log-Einträge gelöscht wurden. In diesem Fall ist die Anforderung nicht erfüllt, dass das IPS alle Log-Einträge der letzten 15 Minuten auswertet.

Um zu testen, ob das IPS-Script nur auf der paketverarbeitenden pfSense Instanz ausgeführt wird, wurde über SSH auf beide pfSense Instanzen zugegriffen. Anschließend wurde auf beiden Instanzen manuell das IPS-Script ausgeführt. Anhand der Ausgaben durch das IPS-Script in der Konsole wurde festgestellt, dass das Script auf der Master-Instanz vollständig durchlaufen und auf der Backup-Instanz nach der Abfrage des Status der virtuellen IP-Adresse im LAN beendet wurde. Anschließend wurde über die Weboberfläche auf der Master-Instanz CARP deaktiviert, sodass die Backup-Instanz die virtuellen IP-Adressen übernimmt und somit der Status von „BACKUP“ auf „MASTER“ wechselt. Nun wurde das IPS-Script erneut auf der Master- und der Backup-Instanz ausgeführt. Anhand der Ausgaben in der Konsole wurde erkannt, dass das IPS-Script auf der Master-Instanz nach der Abfrage des CARP Status der virtuellen IP-Adresse im LAN beendet und auf der Backup-Instanz vollständig durchlaufen wurde. Während dieser Tests hat sich in einigen Fällen nicht bei allen virtuellen IP-Adresse der Status bei der Deaktivierung von CARP geändert. Sobald eine virtuelle IP-Adresse vom LAN-Interface nicht den Status von „BACKUP“ auf „MASTER“ geändert hat, wurde das IPS-Script nicht ausgeführt.

Um die Sperrung der IP-Adressen für IPv4 und IPv6 durch das IPS-Script zu prüfen, wurden ein ICMP-Flooding Angriff mit ca. 1000 Paketen/Sekunde auf die Ubuntu Instanz im WAN gestartet. Die eintreffenden Echo-Requests des ICMP-Flooding Angriffs wurden mit Wireshark beobachten. Anschließend wurde das IPS über die Weboberfläche aktiviert. Nach Durchlauf des IPS-Scripts befand sich die IP-Adresse in dem Alias BADGIRLS, jedoch wurden weiterhin Echo-Requests von der Ubuntu Instanz im LAN empfangen. Um die Sperrung der IP-Adresse durch das Einfügen in den Alias zu überprüfen, wurde der ICMP-Flooding Angriff beendet und neu gestartet. Anschließend wurden keine weiteren Echo-Requests durch die Ubuntu Instanz im WAN empfangen.

Daraufhin wurden die IP-Adressen aus dem Alias entfernt und der Test wiederholt, der Flooding Angriff jedoch nicht nach der Sperrung der IP-Adresse durch das IPS gestoppt. Nach dem zweiten Durchlauf des IPS-Script wurden keine weiteren Echo-Requests durch die Ubuntu Instanz im WAN empfangen.

Daraus wurde geschlossen, dass die IP-Adressen im Alias BADGIRLS nicht unmittelbar in der Firewall eingefügt werden. Das IPS-Script löschte zwar die States aus der Statetable, da die IP-Adressen aus dem Alias aber nicht sofort in die Firewall geladen wurden, wurden neue States in der Statetable angelegt. Durch sukzessives Erhöhen des Zeitraums zwischen dem Einfügen der IP-Adresse in den Alias und dem Löschen der States aus der Statetable wurde ermittelt, dass die IP-Adresse nach einer Sekunde durch die Firewall gesperrt ist. Somit erfüllt das IPS die Anforderung, dass es die IP-Adressen anhand automatisch vollständig sperrt.

Um die Funktion der Whitelist sicherzustellen wurden zunächst Log-Einträge über IPv4 und IPv6 verursacht. Daraufhin wurden die IP-Adressen durch das IPS-Script gesperrt. Anschließend wurden die IPv4- und IPv6-Adresse der Ubuntu Instanz im LAN in die Whitelist eingefügt und jeweils ein Ping zur IPv4- und zur IPv6-Adresse der Ubuntu Instanz im WAN gestartet. Da die IP-Adressen gesperrt waren, wurden keine Echo-Replies empfangen. Nach dem nächsten Durchlauf des IPS-Scripts, wurden Echo-Replies empfangen und somit beide IP-Adressen entsperrt. Daraus folgt, dass die IP-Adressen in der Whitelist durch das IPS nicht gesperrt und bereits gesperrte IP-Adressen durch das IPS entsperrt werden. Damit wird diese Anforderung erfüllt.

Um das E-Mail-Benachrichtigungssystem des IPS zu testen, wurde zunächst das IPS-Script angepasst. Der Schwellenwert für das Melden einer IP-Adresse wurde von Tagen auf Minuten geändert. In der Konfigurationsseite des IPS wurde dann ein Schwellenwert von 20 konfiguriert, d.h. eine IP-Adresse die über 20 Minuten anstelle von 20 Tagen dauerhaft gesperrt ist, wird per E-Mail gemeldet. Als Adresse für den SMTP Server wurde die IP-Adresse der Ubuntu Instanz im WAN angegeben. Auf der Ubuntu Instanz im WAN wurde die Paketspeicherung durch Wireshark gestartet. Außerdem wurde Netcat genutzt, um auf dem Port 25 zu lauschen, damit ein vollständiger Verbindungsaufbau durch das IPS-Script beim Versuch des Sendens einer E-Mail vollzogen werden kann. Anschließend wurde das IPS aktiviert und über 120 Log-Einträge durch einen kurzen Flooding Angriff verursacht. Nach 10 Minuten wurden erneut über 120 Log-Einträge verursacht, sodass die IP-Adresse für weitere 15 Minuten und somit insgesamt 25 Minuten gesperrt blieb. 20 Minuten nach Sperrung der IP-Adresse wurden durch das IPS Script TCP-Pakete mit Ziel-Port 25 zur Ubuntu Instanz geschickt. Der Verbindungsaufbau war erfolgreich und nach einem Timeout wurde die Verbindung durch das IPS-Script wieder abgebaut. Eine weitere Evaluierung der E-Mail-Benachrichtigungsfunktion war nicht möglich, da kein SMTP-Server zur Verfügung stand und auch nicht erfolgreich konfiguriert werden konnte. Anhand dieses Tests ist nicht die volle Funktionsfähigkeit der E-Mail-Benachrichtigungsfunktion bestätigt.

25 Minuten nach Sperrung wurde die IP-Adresse durch das IPS wieder entsperrt bzw. diese aus dem Alias BADGIRLS entfernt. Dies wurde anhand eines Pings zur Ubuntu Instanz im WAN festgestellt. Daher erfüllt das IPS die Anforderung IP-Adressen automatisch wieder zu entsperren, sobald deren Strafpunktestand unter den Schwellenwert von 120 gefallen ist.

5.2.4. Destination Network Address Translation

Voraussetzung für das Testen der DNAT Funktionalität ist die Sperrung der IP-Adresse der Ubuntu Instanz im LAN. Über die Ubuntu Instanz im WAN wurde SNAT auf der pfSense Firewall aktiviert, indem in den Manual Outbound NAT Modus gewechselt wurde. Anschließend wurde eine SNAT-Regel angelegt, die die Quell-IP der Pakete aus dem lokalen Netz durch die virtuelle IPv4-Adresse des WAN-Interfaces ersetzt. Anschließend wurde ein HTTP-Request von der Ubuntu Instanz im LAN an eine IP-Adresse außerhalb des lokalen Netzes geschickt. Die TCP Pakete des HTTP-Requests wurden anschließend zum lokalen Webserver auf die pfSense Firewall umgeleitet, da die Statusseite angezeigt wurde. Somit hat die Umleitung auf den lokalen Webserver über IPv4 mit SNAT funktioniert und die Anforderung ist mit SNAT erfüllt.

Daraufhin wurde SNAT auf der pfSense Firewall deaktiviert und erneut eine HTTP-Anfrage von der Ubuntu Instanz im LAN an eine IP-Adresse außerhalb des lokalen Netzes geschickt. Auch in diesem Fall wurde die Statusseite angezeigt und folglich die TCP-Pakete des HTTP-Requests durch DNAT an den lokalen Webserver umgeleitet. Somit ist auch die Anforderung an die Umleitung von HTTP-Requests mit deaktiviertem SNAT erfüllt.

Das Umleiten von HTTP-Anfragen über IPv6 konnte aufgrund fehlender Funktionalität der pfSense Firewall und des Moduls PF nicht realisiert werden. Die Umsetzung dieser Funktion würde einen tiefen Eingriff in das System bedeuten, das zu Kompatibilitätsproblemen bei zukünftigen Updates der pfSense führen kann. Diese funktionale Anforderung konnte im Rahmen dieser Arbeit nicht erfüllt werden.

5.2.5. Webserver und Statusseite

War die IPv4-Adresse gesperrt, so wurden die TCP-Pakete eines HTTP-Requests durch DNAT an den lokalen Webserver umgeleitet. Der Webserver hat die HTTP-Requests daraufhin an die virtuelle IP-Adresse des LAN-Interface umgeleitet. Dies wurde an der URL-Zeile im Browser festgestellt, da dort nicht mehr die ursprünglich eingegebene IP-Adresse stand, sondern die virtuelle IPv4-Adresse des LAN-Interfaces der pfSense Instanzen.

Um die Umleitung von HTTP-Requests für IPv6 zu testen, wurde zunächst eine Filterregel für das LAN-Interface definiert, die TCP-Pakete mit Ziel-Port 80 über IPv6 weiterleitet. Ansonsten würden die TCP-Pakete mit den HTTP-Anfragen durch die Filterregel verworfen, die die Kommunikation von gesperrten Hosts vollständig sperrt. Anschließend wurde ein HTTP-Request an die IPv6-Adresse der Master Instanz geschickt. Anhand der URL-Zeile im Browser konnte festgestellt werden, dass der Webserver die HTTP-Anfrage an die virtuelle IPv6-Adresse des LAN-Interfaces weitergeleitet hat. Damit erfüllt der Webserver die Anforderung an HTTP-Redirection für IPv4 und IPv6.

Nachdem die Umleitung der HTTP-Anfragen für IPv4 und IPv6 erfolgreich war, wurde geprüft, ob über den Webserver andere Verzeichnisse oder Dateien aufgerufen oder anzeigt werden können. Dazu wurde zunächst versucht, die Datei badgirlsstatistics.json, die sich im gleichen Verzeichnis wie die Statusseite befindet, aufzurufen. Dazu wurde der IP-Adresse in der URL-Zeile des Browsers der Dateiname angehängt und der HTTP-Request zum Server geschickt. Der Webserver hat weiterhin die Statusseite und nicht die angegebene Datei zurückgeschickt. Anschließend wurde versucht ein anderes Verzeichnis aufzurufen. Dazu wurde der IP-Adresse in der URL-Zeile ein Pfad angehängt und der HTTP-Request zum Server geschickt. Auch in diesem Fall, wurde die Statusseite vom Webserver zurückgeschickt und keine Fehlermeldung ausgegeben.

Um die Funktionsweise der Statusseite zu überprüfen, wurden einmal über IPv4 und IPv6 Log-Einträge bei Suircata verursacht. Nach der Sperrung der jeweiligen IP-Adresse durch das IPS-Script, wurde die Statusseite aufgerufen. Die Statusseite hat in beiden Fällen die Sperrung durch „No Internet access“ angezeigt. Darunter wurden die Hinweistexte angezeigt. Im unteren Bereich der Statusseite wurde in beiden Fällen die gesperrte IP-Adresse korrekt dargestellt. Das Datum und die Uhrzeit der Sperrung haben mit dem der Sperrung durch

das IPS übereingestimmt. Darunter wurde die gleiche Zahl an Log-Einträgen angezeigt, die zuvor verursacht wurde.

Nachdem 15 Minuten seit der Sperrung vergangen waren, wurde die Statusseite erneut aufgerufen. Sowohl bei IPv4 als auch bei IPv6 wurde der Status auf „Internet access“ geändert.

Somit erfüllt die Statusseite alle an sie gestellten Anforderungen.

6. Fazit und Ausblick

In dieser Arbeit konnte eine Lösung erarbeitet werden, die es auf der pfSense ermöglicht, Angriffe nach dem Vorbild des Secomats zu erkennen und zu blockieren. Bevor die erarbeitete Lösung jedoch auf den pfSense Firewalls der Institute und Organisationen genutzt werden kann, müssen weiterführende Anpassungen und Optimierungen vorgenommen werden.

Nach Erarbeiten der Funktionen des Secomats und der pfSense Firewall wurden Einsatzszenarien ermittelt, bei denen Angriffe auf externe Ziele versucht werden können. Die Anforderungsanalyse, in der die einzelnen Bereiche der Funktionalität des Secomats betrachtet und daraus Anforderungen an die pfSense aufgestellt wurden, war Basis für die Durchführung der Migration.

Durch die Installation und Konfiguration des IDS-Package Suricata werden Angriffe erkannt und durch die Begrenzung der Paketrate aktiv behindert. Dazu wird Suricata in den „Inline Modus“ geschaltet, sodass es direkt in den Paketfluss eingreifen und gegebenenfalls Pakete verwerfen kann. In Suricata wurden Signaturen angelegt, nach denen die Paketraten für einzelne IP-Adressen überwacht werden. Übersteigt die Paketrate den definierten Schwellenwert einer Signatur, so werden die nachfolgenden Pakete geloggt und verworfen. Im Unterschied zum Secomat hat Suricata kein Burstsystem. Deshalb konnte die Funktionsweise des IDS des Secomats nicht vollständig migriert und die erste Eskalationsstufe nicht auf der pfSense Firewall realisiert werden. Außerdem ist eine Begrenzung der Log-Einträge pro Zeitfenster nicht möglich, d.h. jedes Paket wird nach Überschreitung eines Schwellenwertes geloggt.

Die Log-Einträge von Suricata werden durch ein im Rahmen dieser Arbeit entwickeltes IPS-Script ausgewertet. Das IPS-Script ermittelt wie viele Log-Einträge in den letzten 15 Minuten durch IP-Adressen verursacht wurden. Die IP-Adressen, die mehr als 120 Log-Einträge verursacht haben, werden mithilfe der Firewall von pfSense vollständig gesperrt. Für die gesperrten IP-Adressen wird gespeichert, gegen welche Signaturen wie oft verstoßen wurde. Diese Informationen werden den gesperrten Nutzern über eine Statusseite angezeigt. Bleiben IP-Adressen über einen längeren Zeitraum gesperrt, werden diese durch eine E-Mail an den Administrator der pfSense Firewall gemeldet.

Um Nutzer automatisch über ihre Sperrung zu informieren, werden HTTP-Requests über IPv4 durch DNAT auf den lokal installierten Webserver umgeleitet. Eine Umleitung von HTTP-Requests über IPv6 konnte aufgrund fehlender Unterstützung durch die pfSense nicht realisiert werden. Alle anderen Pakete werden durch die Firewall verworfen, sodass keine weiteren Angriffe auf Hosts im MWN oder Internet möglich sind. Für die Bereitstellung der Statusseite wurde der lighttpd Webserver installiert und konfiguriert. Dieser sendet auf HTTP-Requests von gesperrten Nutzern die Statusseite. Über die Statusseite werden die Nutzer über den Grund und den Zeitpunkt ihrer Sperrung informiert. Das IPS-Script aktualisiert die Zahl der verursachten Log-Einträge einer IP-Adresse bei jedem Durchlauf, sodass

der gesperrte Nutzer den Status seiner Sperrung über die Statusseite verfolgen kann. Wurde der Nutzer durch das IPS wieder entsperrt, so wird ihm dies bei der nächsten Aktualisierung der Statusseite angezeigt. In der Migration wurden Konfigurationsseiten für die pfSense Weboberfläche erstellt, über die Änderungen an den Einstellungen des IPS, des Webservers und der Statusseite vorgenommen werden können und auf die Backup-Instanz synchronisiert werden. Die Migration wurde durch die Erstellung eines IPS-Packages abgeschlossen, dass auf der pfSense Firewall installiert werden kann.

Nach der Migration wurden die einzelnen Funktionen in einer Testumgebung evaluiert. Während der Prüfung des entwickelten IPS-Scripts hat sich gezeigt, dass die Verarbeitung von Log-Dateien, die zusammen eine Größe von mehreren Gigabyte haben, über einer Minute dauert. Da durch Suricata alle verworfenen Pakete geloggt werden, können die Log-Dateien bei hohen Paketraten mehrere Gigabyte groß werden. Durch die getrennte Migration und Evaluation einzelner Komponenten mit dem Ziel, möglichst die gleiche Sicherheit auf der pfSense wie beim Secomat zu erreichen, wurde dieses Problem erst in der Evaluation erkannt. Ein möglicher Lösungsansatz, bei dem jedoch die Paketraten nicht mehr durch Suricata begrenzt werden, ist nur die Pakete beim Erreichen eines Schwellenwertes zu verwerfen und zu loggen. Dadurch werden auf der einen Seite Angriffe erst durch die Sperrung der IP-Adresse durch das IPS-Script gestoppt. Auf der anderen Seite ist so eine geringere Laufzeit des IPS-Scripts aufgrund der Begrenzung der Log-Einträge sichergestellt. Außerdem kann auch die Verarbeitung der Log-Dateien optimiert werden, um so die Laufzeiten zu reduzieren. Die Verkürzung der Laufzeit könnte entweder durch eine Optimierung des aktuellen Verfahrens erfolgen oder durch die Entwicklung eines neuen Verfahrens, bei dem nur neue Log-Einträge ausgewertet werden, statt alle Log-Einträge der letzten 15 Minuten. Die Laufzeit des aktuellen Verfahrens kann wahrscheinlich durch die parallele Auswertung der Log-Dateien über mehrere Threads und durch effizientere Regular Expressions reduziert werden.

Neben Anpassungen und Optimierungen ergeben sich auch weitere Fragestellungen und funktionale Erweiterungen aus der Arbeit. Die Höhe der Schwellenwerte zur Erkennung von (Distributed) Denial-of-Service Angriffen sollten in einer Evaluation überprüft werden. Insbesondere ohne ein Burstsystem und die übergreifende Überwachung der Paketraten für mehrere Ziel-Ports kann es durch zu niedrige Schwellenwerte häufig zu False-Positivs kommen. Diese würden zu einem hohen Managementaufwand führen und im schlimmsten Fall den Administrator der pfSense Firewall dazu bewegen, das IDS oder IPS zu deaktivieren. Dadurch würden dann keine weiteren Angriffe mehr erkannt und verhindert. Der Funktionsumfang des IPS könnte erweitert werden, sodass es nicht nur Sperrungen anhand der Anzahl der Log-Einträge, sondern auch anhand der Art der Log-Einträge durchführt. Werden in Suricata weitere Signaturen für Deep Packet Inspection genutzt, dann wäre denkbar, dass das IPS bereits bei einem Log-Eintrag von diesen Signaturen die verursachende IP-Adresse sperrt.

Abschließend muss festgestellt werden, dass einzelne Bereiche der Secomat-Funktionalität nur mit Einschränkungen auf der pfSense realisiert und genutzt werden können. Mit Suricata können die Paketraten des Angreifers bereits vor der vollständigen Sperrung durch das IPS begrenzt werden. Das Loggen aller verworfenen Pakete durch Suricata führt jedoch zu einer langen Laufzeit bei der Auswertung der Log-Einträge durch das IPS. So kann aktuell nicht die Begrenzung der Paketraten durch Suricata genutzt werden, um dafür die Laufzeit

des IPS im nutzbaren Bereich zu halten. Aus diesem Grund kann aber nicht das gleiche Sicherheitsniveau wie beim Secomat erreicht werden, da Angriffe erst durch die Sperrung des Angreifers durch das IPS blockiert werden. Sollte es in Zukunft mit Suricata auf der pfSense Firewall möglich sein, die Log-Einträge zu begrenzen und gleichzeitig Pakete bei Überschreitung eines Schwellenwertes zu verwerfen, dann könnte annähernd das Sicherheitsniveau wie beim Secomat auch auf der pfSense Firewall erreicht werden.

A. Anhang

A.1. Installation

1. Suricata:
    a) Menü: System -> Package Manager -> Available Packages
    b) Suricata per Klick auf “Install” installieren
2. Webserver:
    a) Menü: Diagnostics -> Command Prompt
    b) Execute Shell Command: pkg install -y lighttpd
    c) Klick auf Execute
3. IPS-Package:
    a) Menü: Diagnostics -> Command Prompt
    b) Execute Shell Command: cd /; pkg add <link-to-package>
    c) Klick auf Execute
    d) PfSense Weboberfläche neu laden

A.2. Konfiguration

1. PfSense:

    1. Menü: System -> Advanced, Tab: Admin Access
        a) “WebGUI redirect” aktivieren
    2. Menü: System -> Advanced, Tab: Networking
        a) “Hardware Checksum Offloading” aktivieren
        b) “Hardware TCP Segmentation Offloading” aktivieren
        c) “Hardware Large Receive Offloading” aktivieren

2. Suricata:

    1. Menü: Services -> Suricata, Tab: Interfaces
        a) Einen neuen Eintrag für das LAN-Interface durch Klick auf „Add“ anlegen
            i. Settings
                A. „Interface“: LAN
                B. „Enable HTTP Log“: deaktivieren

C. „Block Offenders“: aktivieren =¿„IPS Mode“: Inline Mode

ii. Rules

A. „Category“: Active Rules

B. Durch Klick auf Disable All vordefinierte Signaturen deaktivieren

C. „Category“: custom.rules

D. Neue Signaturen anlegen

iii. App Parser

A. alle Parser deaktivieren

2. Menü: Services -> Suricata, Tab: Logs Mgmt

a) „Auto Log Management“: aktivieren

b) „Log Directory Size Limit“: aktivieren

c) „Log Limit Size in MB“: maximalen Speicherplatz für das Log-Verzeichnis einstellen

d) (Optional)„Log Size und Retention Limits“: Größe der Log-Dateien, ab der sie archiviert werden, einstellen.

3. Menü: Services -> Suricata, Tab: Sync

a) „Enable Sync“: Sync to configured system backup server

b) „Refresh Rule Sets“: Signal target host to refresh rules files

3. Webserver:

1. Menü: Services -> SecomatIPS, Tab: Webserver

2. „Konfiguration“: IP-Adresse bei„url.redirect“ durch die virtuelle IP-Adresse des LAN-Interfaces ersetzen

3. Änderungen durch Klick auf „Save“ speichern

4. Webserver über den Button „start“ starten

4. IPS:

1. Menü: Services -> SecomatIPS, Tab: IPS Einstellungen

2. Report E-Mail-Adresse: E-Mail-Adresse, an die dauerhaft gesperrte IP-Adressen gemeldet werden sollen, eintragen

3. Server-Adresse: Adresse des SMTP-Servers

4. Absender E-Mail-Adresse: E-Mail-Adresse, die als Absender für die E-Mails genutzt wird

5. Melde IP ab: Zeitraum in Stunden angeben, nach dem eine permanent gesperrte IP-Adresse per E-Mail gemeldet werden soll

6. Whitelist: IP-Adresse, deren Verkehr zwar durch Suricata überwacht wird, die aber nicht durch das IPS gesperrt werden darf.
7. Änderungen durch Klick auf „Save“ speichern
8. IPS durch Klick auf „start“ aktivieren/starten

4. Statusseite:
   1. Menu: Services -> SecomatIPS, Tab: Statusseite
   2. Sperrseite: Änderungen vornehmen
   3. Änderungen durch Klick auf „Save“ speichern

A.3. Deinstallation

1. IPS-Package:
   a) IPS und Webserver stoppen
   b) Menü: Diagnostics -> Command Prompt
   c) Execute Shell Command: pkg remove -y secomatIPS
   d) Deinstallation durch Klick auf „Execute“ ausführen
2. Webserver:
   a) Menü: Diagnostics -> Command Prompt
   b) Execute Shell Command: pkg remove -y lighttpd
   c) Deinstallation durch Klick auf „Execute“ ausführen
3. Suricata:
   a) Menü: System -> Package Manager -> Installed Packages
   b) Suricata per Klick auf das Mülltonnensymbol deinstallieren

Abbildungsverzeichnis

Literaturverzeichnis

[CB13] Christopher Buechler, Jim P.: *pfSense: The Definitive Guide Version 2.1.* PfSense, 2013

[DF06] Detlef Fliegl, Helmut Reiser Bernhard S. Timo Baur B. Timo Baur: *Ein generisches Intrusion Prevention System mit dynamischer Bandbreitenbeschränkung.* http://www.nm.ifi.lmu.de/pub/Publikationen/fbrs06a/PDF-Version/fbrs06a.pdf. Version: 2006. – abgerufen am 28. Juni 2018

[Fou18] Foundation, Open Information S.: *4.7.2. Flow.* https://suricata.readthedocs.io/en/suricata-4.0.4/rules/flow-keywords.html. Version: 2018. – abgerufen am 02. Oktober 2018

[Gol16] Goldber, Viktor: *A Revision of LRZ's Security Gateway Secomat with particular Focus on Load-Balancing and Redundancy.* TUM, 2016

[LLC18a] LLC, Rubicon C.: *Editing the pf ruleset.* https://www.netgate.com/docs/pfsense/firewall/editing-the-pf-ruleset.html?highlight=tmp. Version: 2018. – abgerufen am 13. August 2018

[LLC18b] LLC, Rubicon C.: *Installing FreeBSD Packages.* https://www.netgate.com/docs/pfsense/packages/installing-freebsd-packages.html. Version: 2018. – abgerufen am 23. September 2018

[LR12] Leibniz-Rechenzentrum: *Security- und NAT-Gateway für das Münchner Wissenschaftsnetz (MWN).* https://www.lrz.de/services/netzdienste/secomat/. Version: 2012. – abgerufen am 20. Juni 2018

[LR18] Leibniz-Rechenzentrum: *Überblick über das Münchner Wissenschaftsnetz (MWN).* https://www.lrz.de/services/netz/mwn-ueberblick/. Version: 2018. – abgerufen am 02. November 2018

[LRZ16] LRZ: *Beschränkungen und Monitoring im Münchner Wissenschaftsnetz.* https://www.lrz.de/services/netz/einschraenkung/. Version: 2016. – abgerufen am 29. Juni 2018

[Mee16] Meeks, Bill: *Suricata true inline IPS mode coming with pfSense 2.3 – here is a preview.* https://forum.netgate.com/topic/96482/suricata-true-inline-ips-mode-coming-with-pfsense-2-3-here-is-a-preview. Version: 2016. – abgerufen am 02. Oktober 2018

[Ope18a] OpenBSD: *OpenBSD PF - Firewall Redundancy (CARP and pfsync).* https://www.openbsd.org/faq/pf/carp.html. Version: 2018. – abgerufen am 29. Juni 2018

[Ope18b] OpenBSD: *OpenBSD PF - Getting Started*. `https://www.openbsd.org/faq/pf/config.html`. Version: 2018. – abgerufen am 29. Juni 2018

[Ope18c] OpenBSD: *PF - User's Guide*. `https://www.openbsd.org/faq/pf/`. Version: 2018. – abgerufen am 29. Juni 2018

[Pur04] Purdy, Gregor: *Linux iptables Pocket Reference: Firewalls, NAT and Accounting*. 1. Auflage. O'Reilly Media, Inc., 2004

[Wes18] Westernhagen, Olivia von: *Hacker legten RWE-Webseite lahm*. `https://www.heise.de/newsticker/meldung/Hacker-legten-RWE-Webseite-lahm-4172195.html`. Version: 2018. – abgerufen am 26. September 2018

[Woo16] Woolf, Nicky: *DDoS attack that disrupted internet was largest of its kind in history, experts say*. `https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet`. Version: 2016. – abgerufen am 23. September 2018

[Wö14] Wölbert, Christian: *Playstation- und Xbox-Dienste nach Angriff wieder online*. `https://www.heise.de/newsticker/meldung/Playstation-und-Xbox-Dienste-nach-Angriff-wieder-online-2506810.html`. Version: 2014. – abgerufen am 23. September 2018