Automatisierte Erkennung von Infection-Proxys mithilfe von statistischer Analyse

Nils Rogmann

Hochschule Darmstadt, Haardtring 100, 64295 Darmstadt, Deutschland,
nils.rogmann@stud.h-da.de

Abstract: Eine zuverlässige Erkennung von sogenannten Infection-Proxys, die immer häufiger zur automatisierten Manipulation von unverschlüsselt oder unzureichend verschlüsselt übertragenen Dateien eingesetzt werden, stellt trotz einiger bereits in der Fachwelt diskutierter Ansätze eine bisher nicht gelöste Herausforderung dar. Innerhalb dieser wissenschaftlichen Arbeit wird daher ein neuartiges statistisches Verfahren, welches eine Detektion dieser infizierenden Proxys ermöglichen soll, vorgestellt. Das entwickelte Verfahren wird darüber hinaus in Form eines Prototyps implementiert und der Allgemeinheit unter der GNU GPLv3 zur Verfügung gestellt.

Keywords: Infection-Proxy, Dateimanipulation, fortgeschrittene Angriffsszenarien, Angriffspunkte, statistische Analyse, Tor-Netzwerk, Man-in-the-Middle

1 Einleitung

In Unternehmen und staatlichen Einrichtungen kommt es trotz des Einsatzes von aktueller IT-Sicherheitsinfrastruktur und der Verwendung verschiedenster Schutzmaßnahmen immer häufiger zu massiven Sicherheitsvorfällen (vgl. [Bun15]). Eine Vielzahl von Angriffen auf unsichere Protokolle oder verwundbare Dienste können in der Regel durch traditionelle Systeme wie Firewalls, Proxys oder Intrusion Detection-Systeme bereits am Perimeter detektiert und unterbunden werden. Die Erkennung von neuartigen, fortgeschrittenen Angriffsszenarien, die unter anderem einen unmittelbaren Angriff auf die Kommunikation zwischen verschiedenen Systemen zum Ziel haben, stellt jedoch eine stetig zunehmende Herausforderung dar.

Bei einem dieser fortgeschrittenen Angriffsszenarien, welches unter anderem innerhalb des Tor-Netzwerks beobachtet wurde, handelt es sich um den Einsatz eines Infection-Proxys, der als Man-in-the-Middle (MitM) eine automatisierte Infizierung von unverschlüsselt oder unzureichend verschlüsselt übertragenen Dateien durchführt (vgl. [Lev14]). Hierbei wird im Allgemeinen ein transparenter Proxy verwendet. Dieser kann in der Regel unbemerkt in einem Netzwerk platziert werden, da im Gegensatz zu herkömmlichen Proxys keine explizite Eintragung innerhalb eines Clients notwendig ist (vgl. [mit16]).

Die Erkennung eines Infection-Proxys, der sich transparent innerhalb der Kommunikation platzieren und den Netzwerkverkehr mitlesen sowie automatisiert manipulieren kann, stellt trotz existierender Ansätze, die während dieser Untersuchung beleuchtet werden, eine bisher nicht zuverlässig lösbare Herausforderung dar.

Im Zuge dieser wissenschaftlichen Arbeit gilt es daher, einen neuen Ansatz zur zuverlässigen Detektion von Dateimanipulationen durch Infection-Proxys zu entwickeln. Hierbei ist es zunächst erforderlich, die bisher existierenden Ansätze zur Erkennung herauszuarbeiten. Darauf aufbauend sind die wesentlichen Angriffspunkte innerhalb der Kommunikation zwischen Client und Webserver zu beleuchten. Basierend auf den gesammelten Erkenntnissen soll dann die Entwicklung eines statistischen Verfahrens zur automatisierten Erkennung von Infection-Proxys erfolgen.

Das Ziel dieser wissenschaftlichen Arbeit besteht darin, ein auf statistischer Analyse basierendes Verfahren zur Erkennung von Infection-Proxys zu entwickeln. Ferner soll ein Prototyp, der eine automatisierte Detektion dieser infizierenden Proxys ermöglicht und eine Manipulation von unverschlüsselt und unzureichend verschlüsselt übertragenen Dateien erkennt, bereitgestellt werden. Zuletzt ist der implementierte Prototyp zur Anfertigung einer repräsentativen Messreihe, welche die Effektivität des implementierten statistischen Verfahrens aufzeigen soll, einzusetzen.

2 Angriffspunkte

Ein Infection-Proxy kann die Beobachtung und Manipulation von Datenverkehr grundsätzlich an verschiedenen physikalischen oder logischen Punkten innerhalb eines Netzwerks oder des Internets durchführen (vgl. [Pit16], [Aug16], S. 2). In Abhängigkeit von der jeweiligen Position sind unter Umständen verschiedene Verfahren, die eine zuverlässige Detektion von Dateimanipulationen durch bösartige Proxys ermöglichen sollen, zu implementieren. Zur Realisierung eines auf statistischer Analyse basierenden Ansatzes gilt es daher, zunächst die wesentlichen Angriffspunkte herauszuarbeiten und zu beleuchten. Diese sind in Abbildung 1 dargestellt und nachfolgend im Detail beschrieben:

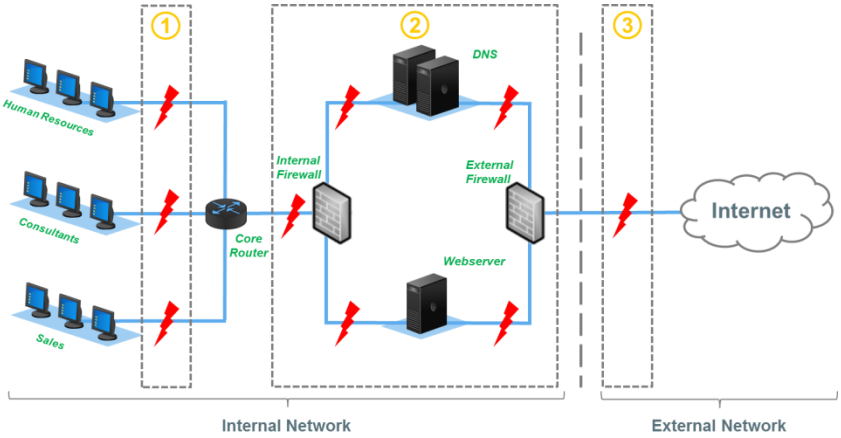

Abbildung 1: Übersicht der grundlegenden Angriffspunkte von Infection-Proxys

Wie der Abbildung zu entnehmen ist, lassen sich die in rot dargestellten Angriffspunkte innerhalb eines beispielhaften lokalen Unternehmensnetzwerks und des Internets im Wesentlichen in drei Kategorien zusammenfassen:

(1) Innerhalb des ersten Angriffspunkts wird ein Infection-Proxy im Allgemeinen in einem Client-Segment und somit in unmittelbarer Nähe der Clients eines Netzwerks platziert. Hierbei können sowohl spezielle Hardware als auch reine Software-Varianten dieser infizierenden Proxys zum Einsatz kommen. Bei Ersterer handelt es sich typischerweise um einen kleinen Computer wie einen Raspberry Pi, der durch einen physikalischen Zugriff innerhalb des Netzwerk-Segments angeschlossen wird (vgl. [Que14]). Die Etablierung der Software-Variante erfolgt hingegen häufig durch Angriffe über das Internet und findet demnach über einen ausschließlich entfernten Zugriff statt. Befindet sich ein Infection-Proxy innerhalb dieses lokalen Angriffspunkts, wird beispielsweise unter Einsatz von ARP-Spoofing (vgl. [DL04], [Pit16]) der Datenverkehr umgeleitet und die Manipulation ermöglicht.

(2) Eine weitere Kategorie von Angriffspunkten befindet sich in der eigentlichen Netzwerkinfrastruktur eines Unternehmens, also beispielhaft vor oder hinter einer Firewall, dem Web- oder DNS-Server. Auch eine unmittelbare Kontrolle dieser Infrastruktur-Komponenten oder die Simulation eines vermeintlichen Access Points (vgl. [Hak16]) ermöglichen eine Beobachtung und Manipulation durch Infection-Proxys. Neben der Platzierung von zusätzlicher Hardware durch einen physikalischen Zugriff ist hierbei ebenfalls grundsätzlich eine Etablierung des infizierenden Proxys über einen aus dem Internet durchgeführten Angriff denkbar.

(3) Während es sich bei den ersten beiden Kategorien um Angriffspunkte innerhalb des lokalen Netzwerks handelt, kann ein infizierender Proxy grundsätzlich auch auf dem Kommunikationsweg im Internet, also mit einer größeren Distanz auf die anzugreifenden Clients, platziert werden. Typische Möglichkeiten für eine Platzierung sind hierbei die Netzwerke und Infrastrukturen des Internet Service Providers oder des Webserver-Betreibers (vgl. [Aug16], S. 2-3). Für die Etablierung eines entfernten Infection-Proxys ist in diesem Fall kein physikalischer oder logischer Zugriff auf das lokale und häufig mit grundlegenden Schutzmaßnahmen ausgestattete Netzwerk erforderlich, weshalb hierbei eine Prävention im Allgemeinen kaum möglich ist.

Vor dem Hintergrund dieser drei Angriffspunkte ist zusammenfassend festzuhalten, dass ein Infection-Proxy zur Beobachtung und Manipulation von nicht oder nur unzureichend verschlüsseltem Datenverkehr an einer Vielzahl von verschiedenen Punkten innerhalb des Kommunikationswegs zwischen Client und Server platziert werden kann. Demzufolge stellt dieses fortgeschrittene Angriffsszenario eine erhebliche Bedrohung dar, welche die Entwicklung eines neuartigen Ansatzes zur zuverlässigen Erkennung erfordert.

3 Erkennungsansätze

Nachfolgend werden zwei von Sicherheitsexperten veröffentlichte Ansätze als mögliche Erkennungsmaßnahmen für das einleitend beschriebene Angriffsszenario vorgestellt. Während Ersterer eine Detektion von Dateimanipulationen zum Ziel hat, steht bei Letzterem die unmittelbare Erkennung eines Infection-Proxys im Vordergrund.

3.1 Hashsummen

Der Sicherheitsforscher Josh Pitts entdeckte im Jahr 2014 einen bösartigen Infection-Proxy innerhalb des Tor-Netzwerks, der bei der Manipulation von verschiedenen unverschlüsselt übertragenen Dateien beobachtet wurde (vgl. [Lev14]). Die Erkennung erfolgte hierbei unter Verwendung eines für den Tor-Scanner *exitmap*[1] entwickelten Python-Skripts. Dieses lädt automatisiert eine vordefinierte Datei eines Webservers über alle aktiven Exit-Knoten des Tor-Netzwerks herunter, berechnet die jeweilige sha512-Hashsumme und vergleicht diese mit einem zuvor kalkulierten Hash der angegebenen Datei. Stimmen die beiden Werte nicht überein, kann auf eine Manipulation und somit auf die Existenz eines Infection-Proxys innerhalb des Kommunikationswegs geschlossen werden.

Das beschriebene Vorgehen zur Detektion von manipulierten Dateien bringt jedoch ein nicht zu vernachlässigendes Defizit mit sich. Infiziert ein Proxy beispielsweise nur ausgewählte Datentypen oder sogar ausschließlich Downloads von spezifischen Domains, dann würde die im Skript hinterlegte Datei ggf. nicht manipuliert und demnach der Infection-Proxy nicht erkannt werden können.

3.2 Traceroute

Ein weiterer Ansatz zur Erkennung eines Infection-Proxys besteht in der Beobachtung der Route, die ein Paket zwischen zwei Systemen durch ein Netzwerk oder das Internet nimmt (vgl. [Cyb11]). Hierzu empfiehlt sich beispielsweise die Verwendung des Programms „Traceroute“ (vgl. [Ker16]). Dieses sendet wiederholt IP-Pakete zu einem Zielserver und erhöht hierbei kontinuierlich das TTL-Feld (Time to Live) innerhalb des IP-Headers um eins. An jedem Router, der das versendete Paket weiterleitet, wird dieses Feld wiederum automatisch um eins dekrementiert. Sobald der Wert „0“ erreicht wird, sendet der aktuelle Router eine Status-Nachricht an das Programm zurück und identifiziert sich hiermit selbst als einen Teil der Route.

Unter Verwendung dieses Programms ist zunächst die Annahme zu treffen, dass ein Proxy ausschließlich eine Manipulation der Web-Kommunikation auf den Standard-Ports 80 und 443 durchführt. Sind weitere offene Ports wie 21 oder 25 verfügbar, können

[1] Bei exitmap handelt es sich um ein Scan-Werkzeug, welches alle verfügbaren Exit-Knoten des Tor-Netzwerks mit verschiedenen Skripten auf maliziöses Verhalten testet kann (vgl. [Win16]).

verschiedene Traceroute-Anfragen gestellt und die Ergebnisse verglichen werden. Befindet sich ein Infection-Proxy innerhalb des Kommunikationswegs, weichen die Resultate voneinander ab und ermöglichen somit eine Erkennung.

Ein wesentlicher Nachteil dieses Ansatzes besteht in der notwendigen Verfügbarkeit von weiteren offenen Ports, die während der Messung als Referenzwerte dienen. Ferner ist es einem Infection-Proxy grundsätzlich möglich, eine Vielzahl von verschiedenen Ports zu beobachten und den Datenverkehr entsprechend zu manipulieren. In diesem Fall würde die Messung der Route durch die Angabe von zwei verschiedenen Ports ein identisches Ergebnis liefern und somit keine Erkennung ermöglichen.

Zusammenfassend erlauben die vorgestellten Ansätze zwar eine grundsätzliche Detektion von Infection-Proxys, funktionieren jedoch ausschließlich unter eingeschränkten Bedingungen und haben somit nicht zu vernachlässigende Grenzen, die im Allgemeinen keine zuverlässige Erkennung möglich macht. Vor dem Hintergrund dieser Erkenntnis ist die Entwicklung eines neuartigen Verfahrens, welches eine zuverlässige Detektion dieser bösartigen Proxys ermöglichen soll, essentiell.

4 Statistische Analyse

Die zuverlässige Detektion einer Dateimanipulation durch einen Infection-Proxy soll mithilfe des in diesem Kapitel beschriebenen statistischen Verfahrens, welches im Zuge der wissenschaftlichen Arbeit entwickelt wurde, realisiert werden. Zur Etablierung dieses Ansatzes war zunächst eine grundlegende Untersuchung des Netzwerkverkehrs, der über einen infizierenden Proxy manipuliert wurde, erforderlich. Nachfolgend werden daher einleitend die wesentlichen aus der Untersuchung resultierenden Beobachtungen beschrieben. Darauf aufbauend erfolgt die Erläuterung des zur Erkennung herangezogenen Merkmals innerhalb des Netzwerkverkehrs. Zuletzt wird das entwickelte statistische Verfahren vorgestellt und im Detail erläutert.

4.1 Erste Untersuchung

Eine auf Statistik basierende Erkennung von Anomalien innerhalb des Netzwerkverkehrs kann im Allgemeinen unter Verwendung verschiedener Merkmale erfolgen. Hierbei handelt es sich typischerweise um Informationen wie die Paketgröße, die Anzahl der Pakete oder deren Ankunftszeit (vgl. [DCG+08], S. 1738, [MP15], S. 786).

Zur Identifizierung eines geeigneten Merkmals, welches eine zuverlässige Detektion von Manipulationen durch Infection-Proxys ermöglicht, wurden zunächst die in Abbildung 2 beispielhaft dargestellten und nachfolgend beschriebenen Dateidownloads zur Auswertung visualisiert:

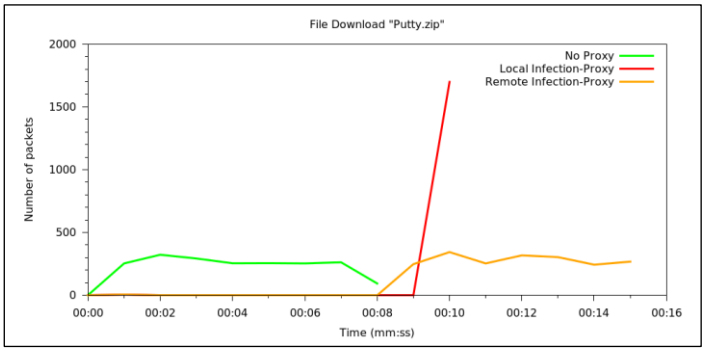

Abbildung 2: Dateidownload mit und ohne Infection-Proxy

Innerhalb der Abbildung wird die Anzahl der empfangenen Datenpakete über die gesamte Zeit eines beispielhaften Dateidownloads[2] dargestellt. Die Zeitmessung beginnt hierbei mit der HTTP-Anfrage eines Clients an den Webserver.

Während die grüne Linie die sekündlich ankommenden Datenpakete einer ohne Verwendung eines Proxys heruntergeladenen Datei repräsentiert, wird mit der roten und orangenen Linie der zeitliche Verlauf eines Downloads über einen lokalen bzw. einen entfernten Infection-Proxy (vgl. Kapitel 2) dargestellt. Von besonderer Bedeutung ist hierbei, dass die Anzahl der sekündlich gemessenen Datenpakete eines Downloads ohne bösartigen Proxy nahezu gleichmäßig verläuft. Die rote und orangene Linie verlaufen hingegen in den ersten Sekunden nahe der Nulllinie, bevor die eigentliche Übermittlung der Datenpakete und der damit einhergehende Anstieg auf der Y-Achse beginnen. Dies ist im Allgemeinen durch die grundlegende Arbeitsweise eines Infection-Proxys begründet. Die zu infizierende Datei ist zunächst vollständig durch den Proxy herunterzuladen. Erst dann besteht in der Regel die Möglichkeit zur automatisierten Manipulation und der anschließenden Weitergabe an den Client (vgl. [Pit16]).

Wie bereits im Kapitel „Angriffspunkte“ herausgearbeitet, befindet sich ein Infection-Proxy auf dem Kommunikationsweg zweier Systeme. Bei einem lokalen Angriff, wie zuvor in (1) und (2) innerhalb von Kapitel 2 dargestellt, ist der zurückzulegende physikalische Weg der Pakete zwischen Client und Infection-Proxy in der Regel deutlich geringer als der zwischen Client und Webserver. Infolgedessen ist häufig eine höhere Bandbreite, die einen extremen Anstieg der in rot dargestellten Paketübertragungsrate in Abbildung 2 zur Folge hat, verfügbar.

Einem entfernten Infection-Proxy, wie in (3) beschrieben, steht hingegen häufig aufgrund der entfernten Position im Internet keine höhere Bandbreite als dem eigentlichen Webserver zur Verfügung. Aus diesem Grund verläuft die Übertragung der Datenpakete nach erfolgter Manipulation und der damit verbundenen Wartezeit in der

[2] Bei dem beispielhaften Dateidownload handelt es sich um den SSH-Client „Putty“, der über die Adresse *https://the.earth.li/~sgtatham/putty/latest/x86/putty.zip* bezogen wurde.

Regel gleichmäßig und ist demnach vergleichbar zu einem Download ohne Proxy.

Zusammenfassend handelt es sich jedoch bei der initialen Wartezeit, die sowohl im Zuge der Manipulation durch einen lokalen als auch durch einen entfernten Infection-Proxy entsteht, um eine mit Statistik zu erfassende Anomalie. Diese soll unter Verwendung des statistischen Verfahrens mithilfe des Merkmals „*packet inter-arrival time*“ (dt. etwa Paketankunftszeit), welches nachfolgend im Detail beschrieben wird, erfasst werden.

4.2 Packet inter-arrival time

Bei der *packet inter-arrival time* (PIT) handelt es sich um die Zeitabstände zweier jeweils aufeinanderfolgenden Pakete innerhalb des Netzwerkverkehrs wie einem HTTP-Download (vgl. [Fra16], [DCG+08], S. 1738).

Wird die PIT zur Darstellung eines Downloadvorgangs ohne Proxy verwendet, ergibt sich die nachfolgende Abbildung:

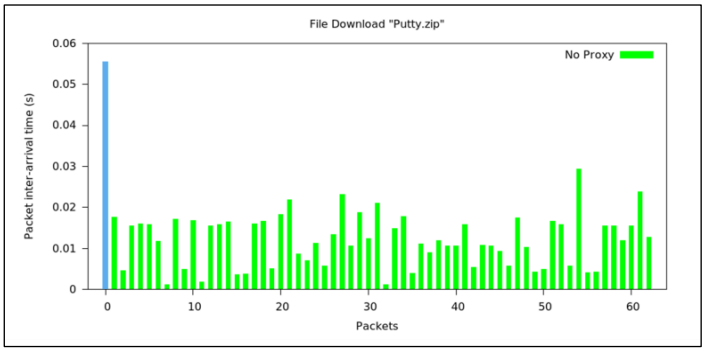

Abbildung 3: Download ohne Verwendung eines Infection-Proxys

In Abbildung 3 erfolgt jeweils die Darstellung der Zeitabstände zwischen den aufeinanderfolgenden Datenpaketen eines Dateidownloads[3]. Die erste, in blau dargestellte Säule repräsentiert hierbei die zeitliche Differenz zwischen der initialen Anfrage des Clients und dem tatsächlichen Beginn der Übermittlung von Datenpaketen an diesen. Obwohl die erste im Vergleich zu den übrigen Säulen durchschnittlich mehr als doppelt so groß ist, handelt es sich hierbei im Allgemeinen nicht um eine Anomalie. Dies wird durch eine unmittelbare Gegenüberstellung mit dem durch einen Infection-Proxy manipulierten Datenverkehr deutlich:

[3] Bei zusätzlichen Paketen, wie dem initialen TCP-Handshake sowie den Acknowledgement-Paketen der TCP-Kommunikation handelt es sich um redundante Informationen, die an dieser Stelle zur Übersichtlichkeit und im Zuge der Realisierung des statistischen Verfahrens zur Steigerung der Effizienz verworfen werden.

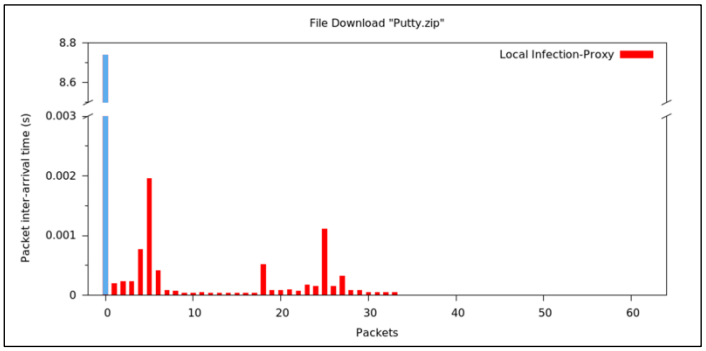

Abbildung 4: Download unter Verwendung eines lokalen Infection-Proxys

In Abbildung 4 werden die PITs einer Datei, die über einen lokalen Infection-Proxy heruntergeladen wurde, dargestellt. Die Platzierung des Proxys entspricht hierbei den in Kapitel 2 erläuterten Angriffspunkten (1) und (2).

An den differierenden Skalen der Y-Achsen beider Abbildungen ist zu erkennen, dass bei einem Download über einen lokalen Infection-Proxy mit Ausnahme der blauen Säule deutlich kleinere Werte gemessen werden als im Zuge des Herunterladens einer Datei ohne infizierenden Proxy. Hierbei handelt es sich um eine Besonderheit, die im Zuge einer Platzierung innerhalb der beiden lokalen Angriffspunkte auftritt. Wie einleitend kurz beschrieben, verfügt der lokal infizierende Proxy für die Weiterleitung der manipulierten Datei aufgrund der physikalischen Nähe zum Client über eine deutlich höhere Bandbreite und kann demnach die Datenpakete in der Regel schneller übertragen. Aufgrund dieser Erkenntnis wird bereits die Identifizierung eines lokalen Infection-Proxys ermöglicht.

Darüber hinaus stellt auch das Verhältnis zwischen der ersten und den restlichen Säulen eine deutliche Anomalie dar, die sich im Allgemeinen ebenfalls bei dem in Kapitel 2 beschriebenen entfernten Angriffspunkt (3) wiederfinden lässt:

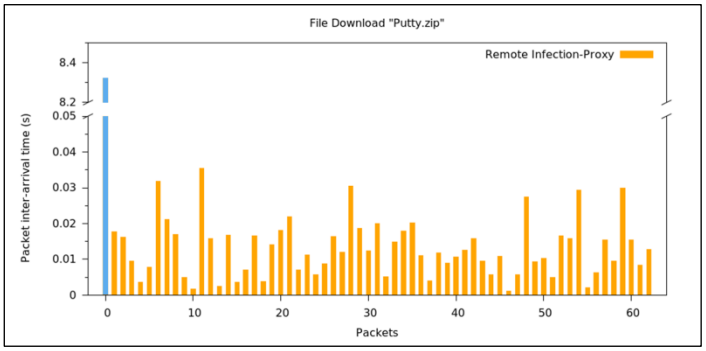

Abbildung 5: Download unter Verwendung eines entfernten Infection-Proxys

Auch in dieser Abbildung wird das stark divergierende Verhältnis zwischen der ersten und den übrigen Säulen ersichtlich. Diese Anomalie ist mit der technisch bedingten Arbeitsweise der Infection-Proxys zu begründen:

Während im Zuge des Downloads ohne Proxy die eigentliche Übertragung der Datei unmittelbar nach Ausführung der HTTP-Anfrage des Clients erfolgt, verstreicht bis zum tatsächlichen Beginn der Übermittlung einer modifizierten Datei eine im Verhältnis zu den übrigen Werten wesentlich größere Zeitspanne. Hierdurch wird eine auf Statistik basierende Detektion einer Dateimanipulation durch Infection-Proxys ermöglicht.

4.3 Statistisches Verfahren

Wie im vorangegangenen Kapitel herausgearbeitet, kann im Allgemeinen durch die Berechnung der *packet inter-arrival time* aller Datenpakete einer unverschlüsselt oder unzureichend verschlüsselt übertragenen Datei die Manipulation durch einen Infection-Proxy detektiert werden. Zur automatisierten Unterscheidung zwischen einer tatsächlich durch Manipulation auftretenden Anomalie im Netzwerkverkehr und einem herkömmlichen Dateidownload ohne Proxy gilt es daher, basierend auf den Erkenntnissen der Untersuchung ein wohldefiniertes statistisches Verfahren zu entwickeln. Dieses beruht im Wesentlichen auf einer Berechnung der PIT, die unter anderem auch von verschiedenen Sicherheitsexperten zur Erkennung von Netzwerktunneln herangezogen wird (vgl. [DCG+08], [MP15]), und lässt sich in drei grundlegende Schritte unterteilen:

(a) Zu Beginn erfolgt eine absteigende Sortierung aller berechneten PITs der Datenpakete eines Dateidownloads ohne Proxy:

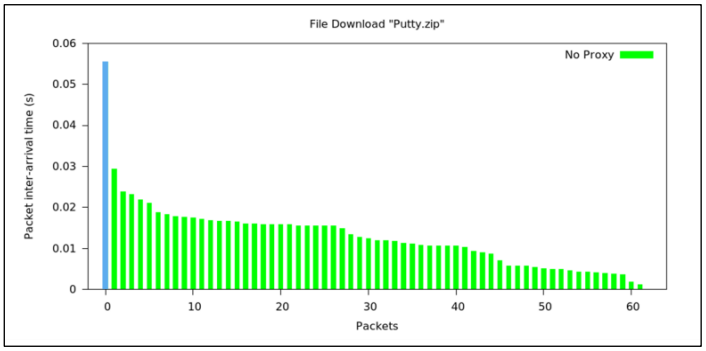

Abbildung 6: Absteigende Sortierung aller PIT-Werte

(b) Als nächstes sind die umsortierten Werte zu filtern, indem mit Ausnahme der größten zehn Prozent alle übrigen Säulen verworfen werden:

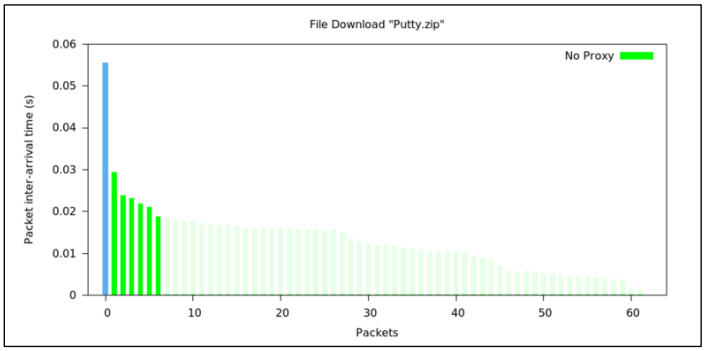

Abbildung 7: Filterung der größten PIT-Werte

(c) Zuletzt erfolgt die Berechnung mehrerer Werte, die zur Bestimmung einer Anomalie und somit zur Detektion einer Manipulation durch einen infizierenden Proxy heranzuziehen sind:

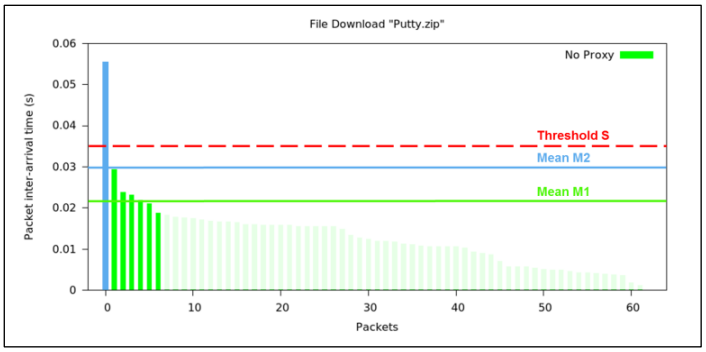

Abbildung 8: Berechnung der Mittelwerte und des Schwellwerts bei einem Download ohne Proxy

Unter Verwendung der grünen Säulen, also *ohne* die erste Zeitspanne zwischen der HTTP-Anfrage des Clients und der tatsächlichen Übermittlung des ersten Datenpakets, erfolgt die Berechnung des in grün dargestellten Mittelwerts (engl. mean) *M1*. Dieser repräsentiert die durchschnittliche *packet inter-arrival time* der größten Zeitspannen zwischen zwei jeweils aufeinanderfolgenden Datenpaketen eines Dateidownloads und dient ferner zur Bestimmung des in rot dargestellten Schwellwerts (engl. threshold) *S*. Der Schwellwert ist ein Vielfaches[4] des zuvor berechneten Mittelwerts.

Als nächstes wird unter Verwendung der blauen *und* grünen Säulen ein zweiter in blau dargestellter Mittelwert *M2* zum unmittelbaren Vergleich mit dem Schwellwert berechnet. Liegt dieser unterhalb des Schwellwerts, handelt es sich nach diesem

[4] Das bestmögliche Vielfache zur Bestimmung des Schwellwerts liegt bei 2,5 und wurde im Zuge der im nachfolgenden Kapitel „Anwendbarkeit“ durchgeführten repräsentativen Messreihe bestimmt.

entwickelten statistischen Verfahren um eine herkömmliche Dateiübertragung ohne die Einwirkung eines infizierenden Proxys. Falls jedoch der zuletzt berechnete Wert über dem Schwellwert liegt, wurde hiermit eine Anomalie erkannt, welche auf eine Manipulation durch einen Infection-Proxy schließen lässt.

An dieser Stelle wird außerdem die Vorteil der in Schritt (**b**) durchgeführten Filterung kleiner PIT-Werte deutlich. Bei einer Berechnung der beiden Mittelwerte über den Wertebereich aller Datenpakete eines Dateidownloads, würde die Größe der blauen Säule im Zuge der Berechnung von *M2* nicht stark genug ins Gewicht fallen, um eine signifikante Änderung des Werts zu verursachen. Infolgedessen wären *M1* und *M2* nahezu identisch und keine sinnvolle Bestimmung von *S* möglich.

Die grundlegende Beschreibung des entwickelten statistischen Verfahrens erfolgte am Beispiel eines Dateidownloads ohne Verwendung eines Infection-Proxys. Die Anwendung auf einen manipulierten Netzwerkverkehr, wie zuvor beispielhaft in Abbildung 2 dargestellt, verdeutlicht die Effektivität des realisierten Verfahrens:

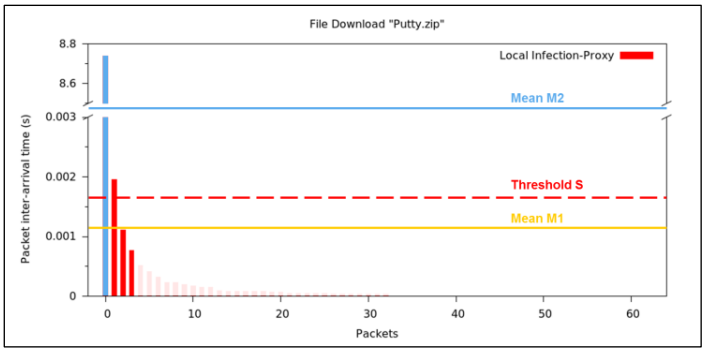

Abbildung 9: Anwendung des Verfahrens unter Verwendung eines lokalen Infection-Proxys

Nach erfolgter Umsortierung, Filterung von den größten PIT-Werten aller erfassten Datenpakete sowie der Berechnung von beiden Mittelwerten und des Schwellwerts ist anhand von Abbildung 9 eine Überschreitung von Letzterem zu erkennen. Somit handelt es sich um eine durch das statistische Verfahren detektierte Anomalie, welche im Zuge der Manipulation des Dateidownloads durch einen Infection-Proxy verursacht wurde.

5 Anwendbarkeit

Die Anwendbarkeit des entwickelten statistischen Verfahrens zur Erkennung von Dateimanipulationen durch einen Infection-Proxy soll innerhalb dieses Kapitels aufgezeigt werden. Zur Demonstration der Effektivität sowie zur Anfertigung einer repräsentativen Messreihe erfolgte die Implementierung des beschriebenen Verfahrens in Form eines Prototyps. Dieser wird einleitend hinsichtlich seiner grundlegenden

Funktionalität beschrieben. Weiterhin erfolgen eine Erläuterung des Vorgehens zur Anfertigung der repräsentativen Messreihe sowie die Darstellung der Messergebnisse.

5.1 Prototyp

Der entwickelte Prototyp, welcher unter der GNU GPLv3 auf Github veröffentlicht und somit für die Allgemeinheit zugänglich ist[5], wurde als Python-Programm implementiert. Das Programm soll unter Einsatz des statistischen Verfahrens eine zuverlässige Detektion von Dateimanipulationen durch Infection-Proxys ermöglichen. Aufgrund der verwendeten Programmiersprache *Python* ist im Allgemeinen eine Plattform-unabhängigkeit und somit die Möglichkeit zur Ausführung des Prototyps auf verschiedenen Betriebssystemen gegeben. Ferner erfolgt die Beobachtung des Netzwerkverkehrs mithilfe von Raw-Sockets (vgl. [Mac96]). Infolgedessen sind in der Regel keine Abhängigkeiten wie zusätzliche Netzwerk-Bibliotheken zu erfüllen.

Zur Detektion der in Kapitel 2 herausgearbeiteten Angriffspunkte ist grundsätzlich eine Platzierung auf dem jeweils zu überwachenden System zu empfehlen. Prinzipiell ist jedoch auch eine Etablierung der Software innerhalb der Netzwerkinfrastruktur und somit eine parallele Überwachung von mehreren Endgeräten möglich. Letztgenannter Ansatz wurde zur Erfassung der nachfolgend beschriebenen Messreihe gewählt.

5.2 Messreihe

Die Erhebung der Messreihe erfolgte unter Verwendung des realisierten Prototyps und eines weiteren Python-Programms, welches zum automatisierten Herunterladen von Dateien sowie zur zufälligen Aktivierung bzw. Deaktivierung des Infection-Proxys[6] entwickelt wurde. Startet dieses Programm einen Download, wird der jeweilige Status des Proxys protokolliert. Parallel beobachtet der Prototyp auf einem ebenfalls im Netzwerk befindlichen, unabhängigen System den Datenverkehr. Hierbei wird jeder Downloadvorgang mithilfe des entwickelten statistischen Verfahrens untersucht und das Ergebnis der Auswertung ebenfalls gespeichert. Erfolgt zuletzt die Zusammenführung der beiden unabhängig voneinander erfassten Datensätze, kann für jede heruntergeladene Datei nachvollzogen werden, ob eine Infizierung durch den Proxy erfolgt ist und ob diese mithilfe des statistischen Verfahrens erkannt wurde oder nicht.

Im Wesentlichen sollte die Repräsentativität der Messreihe durch eine Modifizierung von vier verschiedenen Parametern gewährleistet werden:

Bei dem ersten Parameter handelt es sich um die **Dateigröße**. Da das statistische Verfahren sowohl bei sehr kleinen als auch bei größeren Dateien zur Anwendung kommen soll, erfolgte eine Gruppierung in kleine (< 0,5 MB), mittlere (0,5 - 10 MB)

[5] Der entwickelte Prototyp ist unter der Adresse *https://github.com/0x71/infection-proxy-detector* abrufbar.
[6] Zur Anfertigung der Messreihe wurde der im August 2015 auf der Hackerkonferenz „Blackhat“ vorgestellte BDFProxy verwendet. Hierbei handelt es sich um einen quelloffenen Infection-Proxy (vgl. [Pit16]).

und große (> 10 MB) Dateien. Für jede dieser Gruppen wurden fünf bis zehn für die Messreihe wiederholt herunterzuladende Dateien ausgewählt. Zur Gewährleistung der Repräsentativität erfolgte die Auswahl über den Online-Dienst *Alexa.com*, der unter anderem eine stetig aktuelle Liste der weltweit meistbesuchten Webseiten bereitstellt[7].

Zusätzlich sollte die Zuverlässigkeit des statistischen Verfahrens unter Verwendung von verschiedenen **Internetgeschwindigkeiten** erprobt und somit aufgezeigt werden, dass sowohl bei langsamen als auch bei schnellen Anbindungen eine Detektion möglich ist. Die Messungen wurden zu diesem Zweck jeweils an einem 2 Mbit/s und einem 50 Mbit/s DSL-Anschluss durchgeführt.

Weiterhin wurde eine zufällige Veränderung des **Verbindungswegs** angestrebt. Während einige Downloads der Messreihe jeweils über den direkten DSL-Anschluss durchgeführt wurden, sind zusätzlich Dateiübertragungen über das Tor-Netzwerk und infolgedessen mit nicht-deterministischen Latenzzeiten und Wegen durch das Internet erfolgt. Hierbei sollte die Robustheit des Verfahrens gegen verschieden zuverlässige Verbindungen getestet werden.

Zuletzt erfolgte in Anlehnung an Kapitel 2 eine Veränderung des **Angriffspunkts** eines Infection-Proxys. Hierbei wurden die in Abbildung 1 dargestellten Angriffspunkte (1) und (2) zusammengefasst und somit im Wesentlichen zwischen einem lokal und einem entfernt infizierenden Proxy (3) differenziert.

Zusammenfassend ergibt sich für die nachfolgend dargestellte Messreihe eine Variation von Parametern aus drei Gruppen von Dateigrößen, zwei Internetgeschwindigkeiten, zwei variierenden Verbindungswegen und zwei Angriffspunkten. Für jede dieser Variationen sollten hierbei 100 repräsentative Dateien heruntergeladen werden.

5.3 Messergebnis

Im Zuge der Messreihe erfolgten 1800 automatisierte Dateidownloads unter Veränderung der zuvor beschriebenen Parameter. Hierbei wurden mithilfe eines Zufallsgenerators insgesamt 929 Dateien ohne und 871 über einen infizierenden Proxy heruntergeladen. Die Ergebnisse werden innerhalb der beiden nachfolgenden Tabellen dargestellt und erläutert:

Dateigröße	Anzahl Downloads	Mit Proxy	Ohne Proxy	False Positives	False Negatives	Erkennungsrate
< 0,5 MB	400	206	194	2	0	99,5 %
0,5 - 10 MB	400	189	211	0	0	100 %
> 10 MB	400	183	217	0	0	100 %

Tabelle 1: Messreihe zur Erkennung eines lokalen Infection-Proxys

[7] Die zur Durchführung der Messreihe genutzten Dateien wurden am 10.02.2016 aus der Kategorie „Freeware“ ausgewählt (*http://www.alexa.com/topsites/category/Top/Computers/Software/Freeware*).

Tabelle 1 stellt die Messreihe unter Verwendung eines lokalen Infection-Proxys dar. Insgesamt wurden 400 Downloads pro Dateigruppe durchgeführt. Dies umfasst jeweils 200 heruntergeladene Dateien pro Internetanbindung (2 Mbit/s und 50 Mbit/s) und Verbindungsart (DSL und Tor).

Wie der Tabelle zu entnehmen ist, sind im Zuge der Messungen lediglich bei der kleinsten Dateigruppe fehlerhafte Alarme (*False Positives*) aufgetreten. Diese können beispielsweise während einer Überlastung der Internetverbindung und ggf. daraus resultierenden Verbindungsstörungen oder -abbrüchen entstehen. Als negative Begleiterscheinung einer solchen Überlastung können mehrere große PITs zwischen aufeinanderfolgenden Datenpaketen verursacht werden. Dies kann wiederum trotz der Berechnung von Mittelwerten und dem daraus resultierenden Schwellwert zu einer vermeintlichen Erkennung von Dateimanipulationen durch das statistische Verfahren führen, obwohl kein Proxy zur Infizierung der Datei eingesetzt wurde.

Die Anzahl der nicht erkannten Dateimanipulationen (*False Negatives*) liegt hingegen durchgehend bei null. Demnach wurden alle innerhalb der Messreihe durchgeführten Infizierungen erkannt. Insgesamt ergibt sich hiermit für die 1200 getätigten Downloads eine Gesamterkennungsrate von 99,83 Prozent.

Auch die Messungen unter Verwendung eines entfernten Infection-Proxys zeigen die Funktionalität des entwickelten Verfahrens auf:

Dateigröße	Anzahl Downloads	Mit Proxy	Ohne Proxy	False Positives	False Negatives	Erkennungsrate
< 0,5 MB	200	113	87	8	0	96,0 %
0,5 - 10 MB	200	101	99	3	0	98,5 %
> 10 MB	200	79	121	0	0	100 %

Tabelle 2: Messreihe zur Erkennung eines entfernten Infection-Proxys

Die in Tabelle 2 aufgelisteten Downloads wurden ausschließlich über das Tor-Netzwerk heruntergeladen, da der Infection-Proxy zur Durchführung der Messungen von entfernten Manipulationen als Exit-Knoten innerhalb des Tor-Netzwerks betrieben wurde. Infolgedessen war der Proxy nicht über den zuvor beschriebenen Verbindungsweg „DSL“ zu erreichen. Für jede Dateigruppe wurden daher über das Tor-Netzwerk 100 Downloads pro zur Verfügung stehender Internetanbindung (2 Mbit/s und 50 Mbit/s) durchgeführt und insgesamt pro Dateigruppe 200 Dateien heruntergeladen.

Auch in dieser Messreihe sind keine False Negatives durch das Verfahren erzeugt worden. Bei der kleinen und mittleren Dateigruppe sind hingegen von 200 Downloads insgesamt elf False Positives entstanden. Diese sind im Allgemeinen durch die teilweise extremen Latenzschwankungen innerhalb des Tor-Netzwerks zu begründen und können durch den im nachfolgenden Kapitel beschriebenen Einsatz von sogenannten Kontrollverbindungen reduziert werden. Zusammenfassend ergibt sich für die automatisierte Detektion von Dateimanipulationen durch entfernte Infection-Proxys eine Gesamterkennungsrate von 98,17 Prozent.

5.4 Kontrollverbindungen

Zur Reduktion der beschriebenen fehlerhaften Alarme wurde im Zuge der Entwicklung des Prototyps eine sogenannte Kontrollverbindung realisiert. Wird durch das implementierte statistische Verfahren eine Manipulation detektiert, erfolgt automatisiert die Berechnung der sha512-Hashsumme der heruntergeladenen Datei. Darauf aufbauend wird über eine gesicherte SSL-Verbindung die Download-Adresse der Datei an eine im Internet etablierte Kontrollinstanz gesendet. Diese führt ebenfalls den Dateidownload über die zur Verfügung stehende Internetverbindung aus und berechnet die sha512-Hashsumme. Zuletzt wird der Wert zurück an den Prototyp übertragen und mit der zuvor berechneten Hashsumme verglichen. Stimmen die beiden Werte nicht überein, ist im Allgemeinen von einer Dateimanipulation auszugehen. Andernfalls handelt es sich in der Regel um eine durch das statistische Verfahren ausgelöste Falschmeldung, die jedoch mithilfe der implementierten Kontrollverbindung abgefangen werden kann.

6 Fazit und Ausblick

Innerhalb dieser wissenschaftlichen Arbeit wurde ein neuartiger Ansatz zur Erkennung von Manipulationen durch Infection-Proxys an unverschlüsselt oder unzureichend verschlüsselt übertragenen Dateien entwickelt. Die zuverlässige Detektion dieses fortgeschrittenen Angriffsszenarios der automatisiert infizierenden Proxys stellte trotz einigen vielversprechenden und in der Fachwelt diskutierten Erkennungsansätzen eine bisher nicht gelöste Herausforderung dar. Im Gegensatz zu diesen bereits bestehenden Ansätzen kommt bei dem neuartigen Verfahren Statistik zum Einsatz. Hierbei erfolgt im Wesentlichen unter Verwendung der *packet inter-arrival time* und somit ohne eine Betrachtung der Paketinhalte eine Erkennung von Anomalien innerhalb eines Dateidownloads. Weiterhin wurde der entwickelte Ansatz zur statistischen Analyse in Form eines Prototyps, welcher für die Allgemeinheit zur Verfügung gestellt wird, implementiert. Zuletzt galt es, unter Verwendung dieser entwickelten Software die Funktionalität und Zuverlässigkeit des implementierten statistischen Verfahrens im Zuge der Durchführung einer repräsentativen Messreihe aufzuzeigen.

Der realisierte und auf der Plattform Github veröffentlichte Prototyp kann zukünftig innerhalb von produktiven Netzwerkumgebungen eingesetzt werden und erlaubt somit eine Detektion von Manipulationen an heruntergeladenen Dateien. Hiermit wird die Möglichkeit gegeben, Angriffe auf das Netzwerk sowie die darin befindlichen Systeme frühzeitig zu erkennen. Weiterhin soll die entwickelte Software zukünftig um Funktionen zur Benachrichtigung von Sicherheitssystemen in einer bestehenden Infrastruktur erweitert werden. Dies ist beispielsweise in Form von Syslog-Nachrichten, die an SIEM-Systeme gesendet und dort weiter verarbeitet werden können, realisierbar.

Literaturverzeichnis

[Aug16] Auger, Robert: *thesecuritypractice.com* : Socket Capable Browser Plugins Result in Transparent Proxy Abuse. http://www.thesecuritypractice.com/the_security_practice/TransparentProxyAbuse.pdf, zuletzt besucht am 07. März 2016.

[Bun15] Bundesamt für Sicherheit in der Informationstechnik: *Die Lage der IT-Sicherheit in Deutschland*. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Lageberichte/Lagebericht2015.pdf?__blob=publicationFile&v=3, erstellt im November 2015, zuletzt besucht am 02. Februar 2016.

[Cyb11] Cybersis: *How to Detect Transparent Proxies* : Cybersis blog. http://blog.cyberis.co.uk/2011/06/how-to-detect-transparent-proxies.html, erstellt am 16. Juni 2011, zuletzt besucht am 22. Februar 2016.

[DCG+08] Dusi, Maurizio; Crotti, Manuel; Gringoli, Francesco; et al.: *Detection of Encrypted Tunnels Across Network Boundaries*. In: ICC '08. IEEE International Conference on Communications, 2008, IEEE, Beijing, 2008, S. 1738-1744.

[DL04] Demuth, Thomas; Leitner, Achim: *Angriffstechnik im lokalen Netz: ARP-Spoofing und -Poisoning*. In: Linux-Magazin 06(2004).

[Fra16] Fratto, Mike: *quora.com* : What is the mean Inter-packet Arrival Time?. https://www.quora.com/What-is-the-mean-Inter-packet-Arrival-Time, zuletzt besucht am 17. März 2016.

[Hak16] Hak5: *wifipineapple.com* : WiFi Pineapple Wiki. http://wiki.wifipineapple.com/#!index.md, zuletzt besucht am 10. März 2016.

[Ker16] Kerrisk, Michael: *man7.org* : traceroute(8). http://man7.org/linux/man-pages/man8/traceroute.8.html, zuletzt besucht am 09. März 2016.

[Lev14] Leviathan Security Group: *leviathansecurity.com* : The Case of the modified binaries. http://www.leviathansecurity.com/blog/the-case-of-the-modified-binaries, erstellt am 23. Oktober 2014, zuletzt besucht am 28. Januar 2016.

[Mac96] MacKinnon, Cameron: *man7.org* : What raw sockets are for. http://www.tldp.org/LDP/khg/HyperNews/get/khg/18/1.html, erstellt am 08. August 1996, zuletzt besucht am 19. März 2016.

[mit16] mitmproxy Project: *docs.mitmproxy.org* : Transparent Proxying. http://docs.mitmproxy.org/en/stable/transparent.html, zuletzt besucht am 22. Februar 2016.

[MP15] Mujtaba, Ghulam; Parish, David: *A Statistical Framework for Identification of Tunnelled Applications using Machine Learning*. In: The International Arab Journal of Information Technology, Vol. 12, No. 6A, Zarqa University, Jordan, 2015, S. 785-790.

[Pit16] Pitts, Josh: *github.com* : The Backdoor Factory Proxy. https://github.com/secretsquirrel/BDFProxy#attack-scenarios-all-with-permission-of-targets, zuletzt besucht am 08. März 2016.

[Que14] Quesnelle, Jeffrey: *jeffq.com* : Setting up a man-in-the-middle device with Raspberry Pi. http://jeffq.com/blog/setting-up-a-man-in-the-middle-device-with-raspberry-pi-part-1/, erstellt am 01. Februar 2014, zuletzt besucht am 09. März 2016.

[Win16] Winter, Philipp: *github.com* : exitmap. https://github.com/NullHypothesis/exitmap, zuletzt besucht am 03. März 2016.