Verhaltens-basierte Computerwurm-Erkennung

Sebastian Funke

TU Darmstadt - CASED
Mornewegstraße. 32, 64283 Darmstadt, Germany
http://www.seceng.informatik.tu-darmstadt.de/

Zusammenfassung Die Geschwindigkeit von Netzwerkverkehr, die Häufigkeit von Zero-Day-Lücken[1] und die Vielzahl neuer Netzwerkteilnehmer wie Handys, Fernseher, Industrieanlagen usw. nimmt immer mehr zu und damit auch der potentielle Schaden erfolgreicher Computerwürmer. Deshalb sind unter anderem die herkömmlichen Computerwurm-Erkennungsmethoden, wie z.B. Signatur-Erkennung, oft ein unzureichender Schutz. Aus diesen Gründen beschäftigt sich diese Arbeit damit, einen Überblick über flexible verhaltens-basierte Computerwurm-Erkennungsmethoden zu liefern. Diese analysieren den Netzwerkverkehr an einem Gateway auf Anomalien. Dazu werden die Techniken dieser Verfahren an den Beispieldetektoren TRW, RBS, TRW+RBS, DSC, PGD, MRW, SWORD2 und einem entropie-basierten Detektor von A. Wagner [1], erläutert. Sieben der Detektoren werden basierend auf der Doktorarbeit von J. S. Stafford mit einem vorgestellten Vergleichs-Framework evaluiert. Bei der Evaluation der Systeme werden außerdem verschiedene Netzwerkumgebungen und Wurmstrategien berücksichtigt, damit ein größtmöglicher Bezug zur Praxis hergestellt werden kann.

Keywords: Computerwürmer, Erkennung, Gegenmaßnahmen, verhaltensbasiert

1 Einleitung

Computerwurm-Programmierer entwickeln immer neuere Verschleierungstechniken um die schädliche Signatur ihrer Würmer vor Anti-Viren-Scannern zu verstecken. In einer Arbeit von Staniford et al.[11] wurden verschiedene Scan-Mechanismen katalogisiert, darunter: zufälliges Scannen, lokale Präferenz[2], topologisch[3] und Scannen nach Hit-Listen. Die lokalen, statischen Signatur-Verfahren vieler Anti-Virus-Scanner, sind häufig unzureichend und erkennen keine neuen und komplexen Wurmbedrohungen. Aber auch modernere, dynamische Verfahren, wie die Sandbox-Analyse, haben mit der rasanten Entwicklung von Wurmverschleierungsmaßnahmen zu kämpfen, sodass diese Ansätze nie einen vollständigen Schutz bieten können.

[1] Sicherheitslücken, welche noch nicht öffentlich bekannt sind und dementsprechend noch keine Gegenmaßnahmen vorhanden sind.
[2] Wurm scannt häufiger den Adresspräfix der Hostadresse.
[3] Wurm sammelt Informationen über Nachbarn auf Host, um diese zu befallen.

Welche Präventionsmaßnahmen kann man gegen diese Gefahr einsetzen? Um einen verbesserten Schutz gegen neue Computerwürmer zu gewährleisten, benötigt man, zusätzlich zu den erwähnten host-basierten Verfahren, flexible, netzwerk- bzw. gateway-basierte Verfahren. Deshalb werden in dieser Arbeit verschiedene verhaltens-basierte Verfahren verglichen, welche an einem Netzwerk-Gateway anwendbar sind. Dazu werden im Abschnitt 2 einige revolutionäre Computerwürmer als Fallstudien vorgestellt um die Bedrohung zu analysieren. In Abschnitt 3 werden im Überblick die verschiedenen Möglichkeiten zur Computerwurm-Erkennung abgebildet, sowie deren Vor- und Nachteile erläutert. Im Anschluss daran werden in Abschnitt 4 verhaltens-basierte Techniken zur Wurmerkennung unterschieden und Beispieldetektoren, die diese Techniken nutzen vorgestellt. Danach werden im Abschnitt 5 die vorgestellten Detektoren mithilfe der Qualitätsmetriken aus Teilabschnitt 5.1 in einem vorgestellten Evaluations-Framework evaluiert. Der Experimentablauf, der Evaluation wird in Teilabschnitt 5.2 erläutert. Im Teilabschnitt 5.3 werden die Ergebnisse visualisiert. Abschließend wird das Resultat der Arbeit in Abschnitt 6 zusammengefasst.

2 Motivation

Für einen Überblick über Wurm-Techniken, hat J.S. Stafford in seiner Doktorarbeit[5] wichtige Computerwürmer chronologisch nach ihren Charakteristiken analysiert. Die Tabelle 1 fasst diese Analyse zusammen.

Wurm	**Scan-Typ**	**Vektor**	**Schadcode**	**Opfer**
Code Red v2	zufällig	Bufferoverflow	Defacem.,DDOS	$\sim$ 359K
Code Red II	lokale Präferenz	Bufferoverflow	Backdoor	$\sim$ 359K
SQL Slammer	zufällig	Bufferoverflow	keinen	$\sim$ 75K
Witty	Hitlist, zufällig	Bufferoverflow	löschte HD	$\sim$ 12K
Santy	topologisch	Code-Injektion	Defacement	$<$ 20K
Conficker	verschiedene	Bufferoverflow	BotNet Client	$>$ 4M
IKEE.B	zufällig	Standard Passwort	BotNet Client	unbek.
StuxNet	verschiedene	mehrere	Industriesabotage	unbek.

Tabelle 1: Revolutionäre Computerwürmer[5]

Dabei wurde festgestellt, dass Würmer immer schneller und komplexer werden. Bestes Beispiel dafür ist der SQL Slammer und der StuxNet-Wurm. Der SQL Slammer konnte mit zustandslosen UDP Paketen etwa 75 000 Hosts in weniger als 10 Minuten befallen[2]. StuxNet verwendete mehrere Zero-Day-Lücken um komplexe Industrieanlagen zu sabotieren[12]. Nach einer Analyse von Weaver und Paxon[8], könnte ein erfolgreicher Wurm etwa 50 Milliarden Dollar Schaden und mehr anrichten. Erweiterte Polymorphie-Techniken[4] lassen Shellcode, wenn erwünscht, wie zufällige Bytes aussehen[5]. Damit wird es für Signatur-Scanner immer schwieriger sogar bereits bekannte Würmer zu erkennen. Lokal kann man, zusätzlich zu Signaturverfahren, effektive Buffer-Overflow-Erkennungsmechanismen entwickeln. Diese Mechanismen greifen allerdings tief in die

[4] Verschleierung der Signatur durch Codeänderung.

Ausführungsumgebung der Betriebssysteme ein und müssen auf allen Maschinen vorhanden sein. Würmer wie der IKEE.B[9], welcher eine Konfigurationsschwachstelle nutzt, oder Santy[7], der Code Injection nutzt, würden zudem mit diesen Methoden nicht erkannt werden. Da IKEE.B seinen Datentranfer noch über SSH verschlüsselt, wären auch inhalts-basierte Erkennungsmaßnahmen hilflos. Somit motivieren die Nachteile der anderen Verfahren den Einsatz verhaltens-basierter Erkennung, womit die Schwachstellen dieser Verfahren kompensiert werden können.

3 Erkennungsmaßnahmen im Überblick

Nach Stafford[5] kann man Wurmerkennungssysteme wie in Tabelle 2 unterscheiden. Dabei werden außerdem die Vor- und Nachteile der einzelnen Detektorkategorien, wie schon oben kurz erläutert, aufgezeigt und die Techniken dieser Kategorien mit ihren Erkennungs-Abdeckungen für verschiedene Wurm-Typen beschrieben.

Kategorie	positiv	negativ	Technik	Abdeckung
Host-basierend	- robust gegen Polymorphismus	- muss auf jeder Maschine vorhanden sein	Buffer-Overflows	Erkennt keine Injektionsangriffe auf Anwendungsebene, ansonsten gut
			Input Correlation	Erkennt keine Injektionsangriffe auf Anwendungsebene, ansonsten gut
			System Calls	gut
Inhalts-basierend	- an Gateway anwendbar - schnell bei nicht polymorphen Würmern	- anfällig gegen Polymorphie	Statische Signatur	Erkennt keine Zero-Day Würmer
			Dynamische Signatur	Erkennt keine polymorphen Würmer
			Fortgeschrittene Signatur	Erkennt eventuell keine polymorphen Würmer
Verhaltens-basierend	- an Gateway anwendbar - robust gegen Polymorphismus	- generiert keine Signatur	Connection Failure	Erkennt keine gezielten Scans mit geringer Verbindungsverlustrate
			Network Telescope	Erkennt keine gezielten Scans (z.B. topologische)
			Causation	gut bei Würmern mit nur einem Angriffsvektor
			Muster bei Zieladressen	gut bei Würmern mit höheren Scanraten als normaler Netzwerkverkehr
			Entropie	Gut bei Würmern mit Infektionsroutinen auf mehreren Ports

Tabelle 2: Vor- und Nachteile der Erkennungskategorien und Abdeckung der Kategorietechniken[5]

4 Verhaltens-basierende Erkennungstechniken und Beispieldetektoren

Verhaltens-basierte Wurmerkennungstechniken überwachen das Netzwerk, ohne die Nutzdaten der Datenpakete zu betrachten und versuchen daraus die Anwesenheit eines Wurmes zu erkennen. Ein großer Vorteil gegenüber Host-basierten Systemen, ist die leichte Installation auf Netzwerk-Gateways, sodass deutlich weniger Verwaltungsaufwand anfällt. Gegenüber inhalts-basierten Systemen liegt

der Vorteil in der Robustheit gegen Polymorphismus und verschlüsselte Daten. Allerdings erhält man typischerweise von verhaltens-basierten Ansätzen weniger Informationen. Sie können nur die Anwesenheit eines Wurmes erkennen und eventuell welche Hosts betroffen sind, was weitere Schadensbegrenzungsmaßnahmen einleiten kann. Aber sie können keine Wurm-Signaturen generieren oder den Befall direkt verhindern. Es folgen nun eine Reihe von Techniken dieser Kategorie und Beispieldetektoren, welche diese Techniken nutzen. Das wichtigste Wurm-Verhaltensmuster ist das schnelle Scannen verschiedener Adressen, sodass die folgenden Techniken alle dieses Muster als Grundlage verwenden und sich nur in der Herangehensweise unterscheiden.

4.1 Connection Failure

Diese Technik versucht Würmer im Netzwerk anhand einer hohen Verbindungsverlust-Rate zu erkennen. Besonders auffällig sind hierbei Würmer, die ein zufälliges Scanverhalten implementieren. Bei TCP-Verbindungen entspricht das einem unvollständigen Drei-Wege-Handshake nach einem gewissen Timeout. Für UDP-Verbindungen kann man die Annahme treffen, dass Verbindungsverlust bei fehlendem UDP Verkehr in umgekehrter Richtung vorliegt. Durch Analyse der Verbindungsverlustrate des normalen Netzverkehrs kann man geeignete Grenzwerte finden, um Fehlalarme zu minimieren. Der größte Nachteil dieser Technik ist das fehlende Erkennen von Würmern, die topologisch oder mit Hit-Listen scannen.

TRW (Threshold Random Walk) von Schechter et al. 2004[10] nutzt diese Technik. TRW stuft Hosts als infiziert ein sobald diese, beim Versuch Verbindungen aufzubauen, viele Verbindungsverluste produzieren. Dazu berechnet der Detektor die Likelihood-Wahrscheinlichkeit zwischen der Erfolgs-und Misserfolgsrate der Verbindungen eines Hosts und schlägt beim Überschreiten des Grenzwertes der Misserfolgsrate Alarm.

4.2 Network Telescope

Ähnlich zu der Verbindungsverlustanalyse von oben, versucht ein Netzwerkteleskop Verbindungsverluste zu beobachten. Hierbei werden große Bereiche von nicht erreichbaren (dunklen) Adressen überwacht, da diese keine Pakete erhalten sollten. Eine solche Analyse könnte durch eine zentrale Auswertung der Ergebnisse von verteilten Netzwerkteleskopen das Scanverhalten von Würmern im gesamten Internet beobachten. Da Netzwerkteleskope die gleichen Nachteile wie oben haben und schwer implementierbar sind, werden in dieser Arbeit keine Beispieldetektoren für diese Technik evaluiert.

4.3 Muster in Zieladressen

Im Vergleich zu den oberen Techniken, die das Scanverhalten von Würmern betrachten, kann man mit Mustern in Zieladressen auch unabhängig von der

Scanstrategie infizierte Hosts erkennen. Normaler Netzverkehr eines Hosts unterscheidet sich in dem Muster der Verbindungen zu eindeutigen Zieladressen von Netzverkehr eines infizierten Hosts. Je nach Umsetzung der Musteranalyse kann somit ein verbreitungswilliger Wurm, aufgrund der verursachten Musteränderung, auf kurz oder lang erkannt werden. Die einzige Ausnahme sind Würmer, die sich über Weitergabe von USB-Geräten verbreiten.

MRW (Multi-Resolution Worm Detector), wurde erstmals 2006 von Sekar et al. veröffentlicht[13]. Er basiert auf der Annahme, dass befallene Hosts viele neue Adressen kontaktieren, bis eine Sättigung erreicht ist. Dazu misst er über mehrere Zeitbereiche mit unterschiedlichen Grenzwerten die eindeutigen Verbindungen. Immer wenn ein Host eine neue Zieladresse kontaktiert, wird dessen vergangener Verkehr gegen eine Sammlung von Zeitfenstern und deren Grenzwerte, ausgewertet. Wenn die Anzahl neuer Adressen in dem jeweiligen Zeitfenster den zugehörigen Schwellwert überschreitet, wird ein Alarm ausgelöst.

RBS (Ratebased Sequential Hypothesis Testing), erstmals 2007 von Jung et al. veröffentlicht[4], misst die Rate von Verbindungen zu neuen Zielen, unter der Hypothese, dass wurmbefallene Hosts eine höhere Rate neuer Verbindungen aufweisen als normale Hosts. Diese Rate wird gemessen, indem die Zwischenankunftszeit neuer Zieladressen auf eine Exponentialverteilung abgebildet wird. Ähnlich dem TRW Detektor, wird der Alarm ausgelöst, wenn bei der Überwachung eines Hosts mit einer neuen Verbindung, der Grenzwert der Likelihood Wahrscheinlichkeit zwischen zwei Hypothesen überschritten wird.

TRW+RBS: Im Rahmen der Veröffentlichung von RBS wurde auch die Kombination aus TRW und RBS vorgeschlagen[4]. Dieser überwacht also beides, die Verbindungsverlustrate und die Rate der Verbindungen zu neuen Zielen.

4.4 Causation

Diese Technik basiert auf der Annahme, dass jede Wurm-Verbindung eine weitere verursacht. Für diese Technik werden meist Graphen eingesetzt, in der Annahme, dass sich Würmer häufig „baumartig“verbreiten. Verwenden Würmer allerdings mehrere Angriffsvektoren, wird es schwer ausgehende Verbindungen in Beziehung zu setzen, da diese sich dann unterscheiden können. Verbindungen kann man zum Beispiel über die Nutzdaten (Payload) der Pakete in Beziehung setzen, was bei polymorphen Würmern jedoch zu Problemen führt, da sich dadurch der Payload unterscheiden kann. Wird diese Technik an einem Netzwerk-Gateway angewendet, können außerdem interne Infektionsverbindungen nicht erkannt werden. Allgemein setzt diese Technik voraus, dass Zugriff auf Großteile des gesamten Netzwerkverkehrs besteht, was in der Praxis für Administratoren selten möglich ist.

DSC (Destination-Source-Correlation) wurde von Gu et al. 2004[3] veröffentlicht und ist ein klassischer Vertreter der Causation-Technik. Er erkennt infizierte Hosts, indem er eine eingehende Verbindung, auf einem Port, mit einer darauffolgenden Menge ausgehender Infektionsverbindungen, von diesem Port, in Beziehung setzt. Überschreitet die Rate der ausgehenden Verbindungen einen Grenzwert wurde ein Wurm erkannt. Dazu überwacht er einen Host mit eingegangener Verbindung über einen bestimmten Zeitrahmen.

PGD Protocol Graph Detector wurde 2007 von Collins und Reiter eingeführt[6]. Er wurde entwickelt, um Würmer mit Hit-Listen und topologischen Verbreitungsstrategien zu erkennen, sowie Würmer die sich langsam verbreiten. Dazu erstellt der Algorithmus, anhand des Netzwerkverkehrs, protokollspezifische Graphen, in denen jeder Knoten ein Host und jede Kante eine Verbindung mit diesem Protokoll zwischen zwei Hosts symbolisiert. Über kurze Zeitintervalle sollte die Anzahl an Hosts in den Graphen normal verteilt sein. Im Gegensatz dazu können jedoch bei einer Wurminfektion abnormale Graphen-Formen und Knotenzahlen beobachtet werden.

SWORD2: (Selfpropagating Worm-Observation and Rapid Detection) wurde 2006 von Stafford et al. entwickelt und wurde zur Verbesserung aller oben genannten Detektoren von Stafford in seiner Doktorarbeit 2012 überarbeitet und evaluiert[5]. Seine Erkennungsstrategie beruht auf einer Kombination aus Causation-Technik, der Muster in Zieladressen-Technik und der Analyse von Verbindungspausen. Die Komponente, welche die Muster in Zieladressen-Technik einsetzt, misst die Verteilung der Anzahl der Besuche jeder Zieladresse gegen die Popularität der Zieladresse. Die Komponente, welche die Causation-Technik nutzt, erstellt einen kausalen Verbindungsgraph. Die Knoten des kausalen Verbindungsgraph stellen Verbindungen dar und die gerichteten Kanten potentiellen Beziehungen zwischen den Verbindungen. Eine neue Verbindung wird zum Kind eines existierenden Knotens, wenn für diesen die Lamport („passierte-davor“) Bedingung gilt und somit der Elternknoten diese Verbindung verursacht haben könnte. Weiterhin wird der Knoten dabei auf Ähnlichkeit, im Bezug auf Verbindungsattribute wie Protokoll, Zielport und TCP Flags, mit den anderen Knoten verglichen. Dadurch, dass also der Payload nicht betrachtet wird, ist der SWORD2 Detektor robust gegen polymorphe Würmer. Wird nun ein bestimmter Grenzwert an ähnlichen Vorgängerverbindungen überschritten, wird die Kindverbindung als verdächtig eingestuft. Falls auch die zusätzlich eingesetzte Ziel-Adressen-Technik verdächtige Verbindungen meldet, werden mithilfe eines „Sliding Windows“[5] mehrere zusammenhängende verdächtige Verbindungen gesucht und bei ausreichender Anzahl wird ein Wurm-Alarm ausgelöst. Außerdem ermöglicht SWORD2 die Gruppierung von Hosts in Aktivitäts-Profile.

[5] Beim Sliding-Window-Verfahren wird ein Bereich mit einem Such-Fenster abgesucht, dessen Auflösung mit jedem Durchgang verkleinert oder vergrößert wird.

4.5 Entropie

Hier nun ein kleiner Exkurs in das entropie-basierte System von A. Wagner von 2005[1]. Entropie ist das Maß, wie viel Information in einer Datenmenge steckt bzw. wie viel Zufälligkeit. Je höher die Entropie, desto zufälliger die Datenmenge. Dazu wird die sequentielle und binäre Form der Daten „mathematisch perfekt“ komprimiert und die Größe des Ergebnisses gemessen. Der Kernzusammenhang zwischen der Entropie und der Wurmerkennung, liegt in der uniformen oder strukturellen Änderung des Netzwerkverkehrs bei Scanverhalten. Da Quelladressen dann in vielen Netzwerkströmen vorhanden sind, enthalten diese weniger Entropie pro Adresse als in normalen Netzverkehr. Im Gegensatz dazu kommen Zieladressen dann deutlich zufälliger vor. Ähnlich verhält es sich auch mit den Zielports, deren Entropie dabei signifikant sinkt. Um genauere Aussagen zu treffen, werden die Entropieschätzungen verschiedener Datenintervalle (Quell-, Ziel-Adressen/Ports) zueinander verglichen. In Abbildung 3 wird dieses Verhalten experimentell mit der LZO-Komprimierung[6], anhand des Blaster-Wurms, visualisiert.

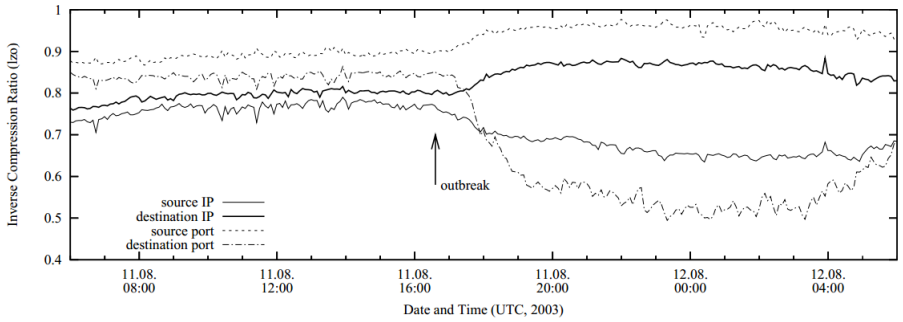

Abbildung 3: Blaster - TCP Adressparameter Komprimierbarkeit.[1]

Da die Evaluation dieses Ansatzes aber in der Arbeit von A. Wagner mehr oder weniger nur auf zwei zufällig scannenden Würmern beruht, wird dieser Ansatz hier nicht weiter betrachtet.

5 Verhaltensbasierte Detektoren im Vergleich

Die Wahl der zu vergleichenden Detektoren, fiel auf die oben vorgestellten Detektoren: TRW[10], RBS[4], PGD[6], TRWRBS[4], DSC[3], MRW[13] und SWORD2[5]. Diese Detektoren wurden aufgrund ihrer Robustheit gegen verschiedene Wurmstrategien, der leichten Installation im Netzwerk und der Fähigkeit, die infizierten Hosts zu identifizieren, gewählt. Detektoren, welche die Netzwerkteleskop-Technik nutzen, kann man ausschließen, da in der Praxis nicht genügend unerreichbare Adressbereiche überwacht werden können. Des Weiteren sind diese

[6] Lempel Ziv Oberhumer: Echtzeit Kompressionsalgorithmus.

Systeme „Network Flow“-Detektoren[7], für welche Stafford[5] ein dynamisches Evaluations-Framework entwickeln konnte. Dieses Framework erlaubt das Importieren von aufgezeichneten Netzwerkverkehr in Datenbanken. Damit kann man dann mit gezielten Abfragen verschiedenen normalen und wurmbehafteten Netzwerkverkehr in einer Evaluationsumgebung vermischen, um darin die Detektoren zu evaluieren. Der Wurm-Verkehr wird mit einem Simulator simuliert und ermöglicht es, Würmer mit den von Staniford et al. beschriebenen Scanstrategien[11] *zufällig, lokale Präferenz* und *topologisch* zu generieren. Um die gewählten Detektoren zu kalibrieren, wurden diese mit den empfohlen Standardeinstellungen versehen und danach mit dem jeweils experiment-spezifischen normalen Netzwerkverkehr feinjustiert.

5.1 Vergleichsmetriken

Um die Stärken und Schwächen der Detektoren zu unterscheiden, muss man geeignete Eigenschaften eines Detektors untersuchen. Die wichtigsten Eigenschaften sind das Erkennen der Präsenz eines Wurmes (F-), die Geschwindigkeit der Erkennung und die damit in Beziehung stehende Rate an Fehlalarmen (F+). Deshalb werden die vier interessantesten Detektoreigenschaften: F-, F+ nach Hosts und nach Zeit und die Detektionslatenz als Evaluations-Metriken eingesetzt. Die folgende Tabelle 4 erläutert diese genauer.

F-	Prozentanteil der Experimente in denen Wurmverkehr vorhanden war, aber nicht erkannt wurde in einer Zeitspanne τ
F+ nach Hosts	Anzahl Fehlalarme in einer Zeitspanne τ, begrenzt auf einen Fehlalarm pro Host
F+ nach Zeit	Prozentanteil von Minuten während einer Zeitspanne τ, in der von mind. einem Host ein Fehlalarm gemeldet wurde
Detektionslatenz	Anzahl von ausgehenden Wurmverbindungen von einem infizierten Netzwerk bevor der Wurm erkannt wurde

Tabelle 4: Metriken zur Evaluation der Detektoren[5]

Weitere Eigenschaften, wie Laufzeit- und Speicherkosten oder Installations- und Wartungsaufwand, wurden im Rahmen der Arbeit von J.S. Stafford nicht untersucht. Die Detektionslatenz wird mit erfolgreich ausgehenden Wurmverbindungen gemessen, da ausgehende Verbindungen weiteren Schaden im Internet verursachen.

5.2 Experimentaufbau

Zum einen werden die Experimente nach vier verschiedenen Netzwerkumgebungen mit unterschiedlichem Netzverkehr unterteilt. Und zum anderen in die verschiedenen Scanstrategien und unterschiedlichen Scanraten der simulierten Würmer. Bei der *topologisch*en Scanstrategie unterscheidet man zusätzlich die verschiedenen Implementierungen der Nachbarfindung. Dabei ist eine Implementierung mit *100 (Topo100)*, *1000 (Topo1000)* und eine mit *unbegrenzt (TopoAll)* vielen Nachbarn vorgesehen, wobei der topologische Wurm zufällig weiter scannt,

[7] nach J.S. Stafford: Detektoren deren Heuristiken nicht den Payload berachten[5].

nachdem keine Nachbarn mehr gefunden wurden. Die *Scanraten* variieren zwischen *10 Verbindungen pro Sekunde* bis nur noch zu *einer Verbindung in 200 Sekunden*. Die größte Schwierigkeit in der Evaluation von verhaltens-basierten Wurmdetektoren liegt in den großen Unterschieden des Netzverkehrs von verschiedenen Netzwerkumgebungen. In Tabelle 5 sind die unterschiedenen Netzwerkumgebungen aufgeführt.

Name	**Aktive Hosts**	**Anzahl Verbindungen**	**Ausgehender Anteil**	**TCP Anteil**
Unternehmen	139	25042	76,3%	50,6%
Campus	117	22935	66,2%	86,4%
Fachbereich	92	29634	53,0%	48,0%
WLAN	313	120032	72,3%	59,8%

Tabelle 5: Untersuchte Netzwerkumgebungen mit ihren Eigenschaften[5]

5.3 Ergebnisse

Fehlalarme bei normalem Netzverkehr: Die Fehlalarmraten, nach Zeit und Hosts bei normalen Netzverkehr fallen besonders hoch in der WLAN-Umgebung aus. Der PGD und DSC weisen die geringsten F+ Raten auf.

Zufälliger Wurm: In Abbildung 6 wird die Fähigkeit von TRW und MRW, zufällig scannende Würmer bei steigenden Scanraten zu erkennen, verglichen. Es fällt auf, dass langsam scannende Würmer in allen Umgebungen die meisten Detektoren täuschen konnten und erst ab durchschnittlicher Scanrate erkannt wurden. Hierbei fällt auf, dass der TRW langsamere Würmer am besten erkennt.

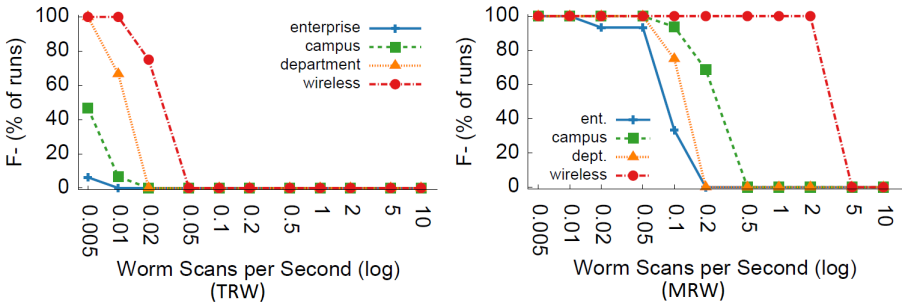

Abbildung 6: F-: Prozentanteil der Experimente in denen der Wurm **nicht** erkannt wurde (je geringer desto besser der Detektor). Verglichen in den jeweiligen Umgebungen, bei zufälligem Scanverhalten.[5]

In Abbildung 7 wird die Detektionslatenz von TRW und MRW gemessen. Dazu wird die Anzahl von erfolgreichen Verbindungen nach außen in Abhängigkeit zur Scanrate visualisiert. Bei DSC und TRW besteht anscheinend kaum eine Verbindung zwischen der Latenz und der Scanrate, jedoch unterscheiden sich die Umgebungen bei diesen stark. Die geringsten Latenzen, in den meisten Umgebungen, erreichten der MRW und der PGD bei durchschnittlich bis schnell scannenden Würmern.

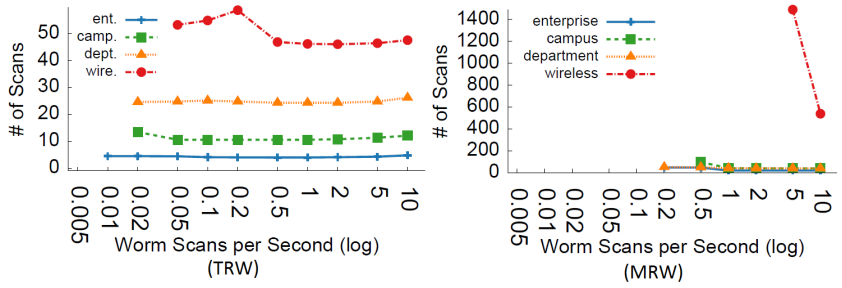

Abbildung 7: Detektionslatenz vom Beginn der Infektion bis zur Erkennung, der Detektoren TRW und MRW, in den jeweiligen Umgebungen, bei zufälligem Scanverhalten des Wurms. Gemessen an der Anzahl ausgehender Verbindungen aus dem Netzwerk vor der Erkennung.[5]

Wurm mit lokaler Präferenz: Bei Würmern mit lokaler Präferenz sinkt erwartungsgemäß die Sensitivität und Erkennungslatenz der Detektoren ein wenig. Da der Unterschied jedoch nur geringfügig ist, werden die Ergebnisse hier nicht aufgeführt. Nur der PGD Detektor zeigte bessere Leistungen. TRW+RBS, RBS, DSC und MRW zeigen außerdem eine schlechtere Detektionslatenz in allen Umgebungen als bei dem zufälligen Wurm.

Topologischer Wurm: Da die topologische Scanstrategie nur existierende, aktive und anfällige Hosts infiziert, ändert diese Strategie nur etwas bei TRW und TRW+RBS, welche Verbindungsverluste analysieren. In der Abbildung 8 wird das Verhalten der beiden Detektoren bei unbegrenzt vielen Nachbarn (TopoAll) visualisiert. TRW erkennt den Wurm erst, wenn dieser keine Nachbarn mehr findet und zufällig weiter scannt, was bei unbegrenzt vielen Nachbarn nicht eintritt. Dadurch benötigt er auch länger, bevor er diesen Wurm erkennt bzw. bei TopoAll wird er gar nicht erkannt. Die Umgebungen nehmen einen geringeren Einfluß auf die Wurmerkennungsrate. Der TRW+RBS, welcher nicht nur auf Connection Failures beruht, wie der TRW, zeigt bessere Detektionslatenzen und erkennt alle topologischen Würmer in den meisten Umgebungen. Kann ein Wurm also eine Hit-Liste aktiver Hosts erstellen, die groß genug ist, hat der TRW keine Chance.

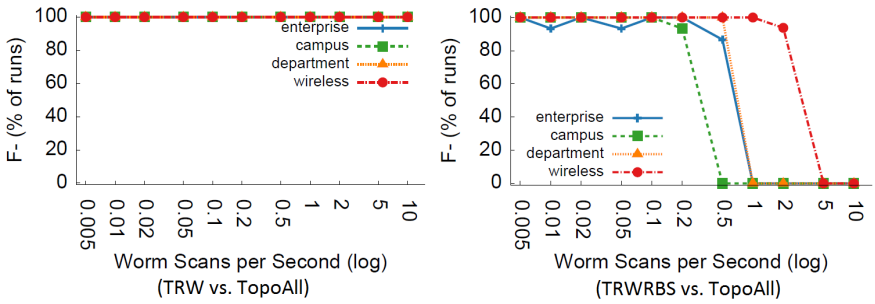

Abbildung 8: F-: Prozentanteil der Experimente in denen der Wurm **nicht** erkannt wurde (je geringer desto besser der Detektor). Verglichen in den jeweiligen Umgebungen, bei topologischer Scanstrategie unbegrenzt vielen Nachbarn.[5]

SWORD2 Verglichen mit den anderen Detektoren: Zum Vergleich mit den anderen Detektoren folgt nun die Wurmerkennungsrate von SWORD2 beim zufälligen Wurm. Nur in zwei von vier Umgebungen zeigt der TRW Detektor leicht bessere Ergebnisse als der SWORD2 Detektor. Ansonsten reagiert der SWORD2 deutlich sensitiver als alle anderen Detektoren auf Würmer mit langsamen bis schnellen Scanraten und macht ihn damit zum Sieger über die verglichenen Detektoren.

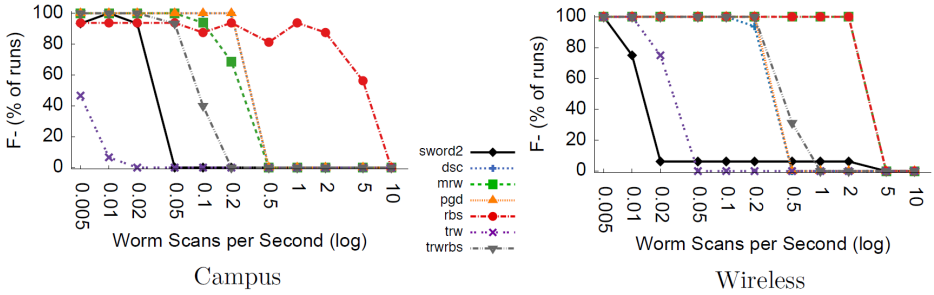

Abbildung 9: F-: Prozentanteil der Experimente in denen der Wurm **nicht** erkannt wurde (je geringer desto besser der Detektor). Verglichen in den jeweiligen Umgebungen, bei allen Detektoren, für den Wurm mit zufälliger Scanstrategie.[5]

6 Fazit und Ausblick

Der von Stafford vorgeschlagene, verbesserte SWORD2 Detektor hebt sich in den meisten Netzwerkumgebungen deutlich von allen anderen Detektoren bei naiven Wurmstrategien ab. Der TRW erzielte als zweitbester die besten Ergebnisse bei Würmern mit naiven Strategien, versagte aber bei topologischen Würmern. Zusätzlich konnte er langsame Würmer besser erkennen als alle anderen. Der PGD, als drittbester Detektor, erkannte zwar in allen Umgebungen alle durchschnittlich schnellen Würmer, hatte dafür aber recht hohe Detektionslatenzen. Der TRW+RBS zeigte ähnliche Ergebnisse wie der PGD, aber war schlechter bei topologischen Würmern. RBS konnte lediglich schnell scannende Würmer gut erkennen. Der MRW hatte große Probleme bei Würmern mit lokaler Präferenz in der WLAN-Umgebung. Der DSC Detektor war recht gut, erkannte aber keine Wurminfektionen, die aus dem inneren Netzwerk stammen. Jeder Detektor würde in allen Netzwerkumgebungen kläglich an einem raffinierten, topologischen Wurm mit einer geringen Scanrate scheitern. Allgemein stellen kabellose Netzwerkumgebungen die größte Herausforderung dar, da fast alle Detektoren darin ungenaue Ergebnisse bei hoher Erkennungslatenz zeigten. Es ist

immer noch unklar, ob aktuelle Detektoren wirklich alle neuen Würmer entdecken können. Ebenso ist trotz der Evaluation unklar, wie verhaltens-basierte Detektoren korrekt verglichen werden können, da es in der Praxis viel zu viele verschiedene Abhängigkeiten gibt. Daher werden verhaltens-basierte Systeme nie die klassischen Systeme ersetzen können. Sie können sie aber sinnvoll ergänzen, da diese Verfahren essentielles Wurmverhalten erkennen können. In Fällen von sehr dynamischen Netzwerkumgebungen wie WLAN oder bei sehr speziellen Scan-Strategien der Würmer, wie Hit-Listen und langsamen Scanraten, müssen diese Detektoren aber von Systemen der anderen Kategorien ergänzt werden, wie die Evaluation der bestehenden Systeme bewiesen hat.

Literatur

1. Arno Wagner: *Entropy-Based Worm Detection for Fast IP Networks*, Swiss Federal Institute of Technology Zurich, Diss., 2008
2. D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, und N. Weaver: Inside the slammer worm. In: *IEEE Security and Privacy, vol. 1* (2003)
3. G. Gu, M. Sharif, X. Qin, D. Dagon, W. Lee, und G. Riley: Worm detection,early warning and response based on local victim information. In: *Annual Computer Security Applications Conference. Washington, DC: IEEE Computer Society* (2004)
4. J. Jung, R. Milito, und V. Paxson: On the adaptive real-time detection of fast-propagating network worms. In: *Journal on Computer Virology, vol. 4, no. 1* (2008)
5. John Shadrach Stafford: *Behavior-based Worm Detection*, University of Oregon, Diss., 2012
6. M. P. Collins und M. K. Reiter: Hit-list worm detection and bot identification in large networks using protocol graphs. In: *Symposium on Recent Advances in Intrusion Detection. Berlin, Heidelberg: Springer-Verlag* (2007)
7. N. Provos, J. McClain, und K. Wang: Search worms. In: *Workshop on Rapid Malcode. New York, NY: ACM Press* (2006)
8. N. Weaver und V. Paxson: A worst-case worm. In: *Workshop on Economics and Information Security* (2004)
9. Phillip Porras, Hassen Saidi und Vinod Yegneswaran: *An analysis of the ikee.b (duh) iPhone botnet*. Online: `http://mtc.sri.com/iPhone/` [zuletzt gelesen am: 17.01.2013], 2009
10. S. E. Schechter, J. Jung, und A. W. Berger: Fast detection of scanning worm infections. In: *Symposium on Recent Advances in Intrusion Detection. Berlin, Heidelberg: Springer-Verlag* (2004)
11. S. Staniford, V. Paxson, und N. Weaver: How to 0wn the Internet in your spare time. In: *USENIX Security Symposium. Berkeley, CA* (2002)
12. Symantec Corp.: N. Falliere, L. O Murchu, und E. Chien: *W32.stuxnet dossier*. Online: `http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf` [zuletzt gelesen am: 17.01.2013], 2011
13. V. Sekar, Y. Xie, M. K. Reiter, und H. Zhang: A multi-resolution approach for worm detection and containment. In: *International Conference on Dependable Systems and Networks. Washington, DC: IEEE Computer Society* (2006)