Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Seminarausarbeitung - Master Semester 3

André Harms

Simulation of malware propagation - Ein multiagentenbasierter Ansatz

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

André Harms

Simulation of malware propagation - Ein multiagentenbasierter Ansatz

Eingereicht am: 28. Februar 2013

Inhaltsverzeichnis

1 Einleitung

Zielgerichtete Angriffe auf IT-Systeme sind keine Seltenheit mehr, wie zum Beispiel der Angriff mittels Stuxnet auf das iranische Atomprogramm (Symantec Corporation, 2011) oder auch Red October, welches mittels Spear-Phishing Attacken initial verteilt wurde (Kaspersky Lab, 2012), veranschaulichen. Die Angriffsmöglichkeiten sind dabei vielfältig. Um eine Attacke zu realisieren, werden häufig mehrere Angriffsvektoren verwendet und miteinander kombiniert. Häufig werden initiale (sekundäre) Angriffe durchgeführt um an das primäre Ziel zu gelangen. Hierzu werden ausgewählte Systeme infiziert, um anschließen eine Schadsoftware in einem gesicherten Bereich, oder auch in nicht vernetzte Systeme, zu verbreiten. Dies ist unter anderem auch bei Stuxnet geschehen. Die Verwendung mobiler Geräte und eng vernetzter Systeme zum Datenaustausch - wie Cloud-Storage (DropBox, Google Drive, ...) oder Netzwerkshares - eröffnen dabei neue Verbreitungswege und werden aktiv zur Ausbreitung von Malware genutzt. (Symantec Corporation, 2011)(Wang und Stavrou, 2010)(Symantec Corporation, 2012)

Möchte man bestehende Systeme auf ihre Anfälligkeit von Angriffen testen, und empfehlenswerte Handlungsweisen im Falle eines Angriffs herleiten, besteht die Möglichkeit Penetration- und Vulnerability-Tests zur Analyse durchzuführen. Solche Tests kommen echten - aber abgesprochenen - Angriffen gleich. Somit besteht auch die Gefahr, ein System unbeabsichtigt bei so einem Test zu beschädigen und einen Ausfall hervorzurufen (Messner, 2011). Diese Tatsache birgt vor allem beim Testen von kritischer Infrastruktur - aufgrund ihrer Bedeutung - Gefahren. Auch ist solch ein Test im Kontext der Schadsoftwareausbreitung nicht gut geeignet, weil infizierte Systeme anschließend wieder bereinigt werden müssten. Daher bieten sich Simulationen von Angriffen an, um Erkenntnisse über die Anfälligkeit und empfehlenswerte Gegenmaßnahmen zu erlangen (Harms, 2012).
Bestehende mikroskopische Simulationen, die sich mit der Ausbreitung von Schadsoftware beschäftigen, greifen auf physische Replikation der zu testenden Umgebung zurück (vgl. (Leszczyna u. a., 2008)) oder benötigen manuelle Eingriffe (Harms, 2012). Dadurch ergeben sich Tests, die in ihrer Vorbereitung oder Durchführung zeit- und ressourcenintensiv sind. Hinzu kommt, dass die Dynamik eines Netzwerkes, die durch mobile Geräte erzeugt wird, nicht berücksichtigt wird. Gerade diese Geräteklasse wird aber häufig von Institutionen als potentielle Gefahrenquelle - auch im Kontext von „Bring Your Own Device“(BYOD) - angesehen (Deloitte, 2013) und für Angriffe instrumentalisiert. Um eine detaillierte Betrachtung des Ausbreitungsverhaltens zu ermöglichen und die Dynamik zu berücksichtigen, ist eine manuelle oder physische Simulation nicht geeignet.

2 Zielsetzungen der Masterarbeit

In Bezug auf das Ausbreitungsverhalten von Schadsoftware spielen mehrere Faktoren eine Rolle. Einer dieser Faktoren ist die Implementierung der Malware, die vorgibt, welche Propagationsmöglichkeiten überhaupt genutzt werden können. Hinzu kommt die Struktur des Systems, in dem sich die Malware ausbreitet. Hierzu zählen neben dem Aufbau des Netzwerkes und den logischen Verbindungen von Diensten auch die Patchlevel und Systemeinstellungen. Ebenfalls spielen die Benutzer eine wichtige Rolle bei der Ausbreitung von Schadsoftware. Sie beeinflussen durch ihr Verhalten Ausbreitungswahrscheinlich- und Häufigkeiten.
Unter Berücksichtigung der Vernetzung, der Mobilität der Benutzer und Systemeigenschaften soll eine Möglichkeit geschaffen werden, Informationen über die Ausbreitung von Schadsoftware zu gewinnen. Diese Informationen sollen sich für Resilienz-Analysen verwenden lassen, um die Anfälligkeit eines Systems oder einer Infrastruktur abschätzen zu können. Außerdem sollen anhand der gewonnenen Daten Abwehrstrategien im Falle eines Malware-Befalls hergeleitet werden.
Konkrete Fragen, die sich bei einem Befall ergeben, sind typischerweise, welche Segmente abgeschottet oder abgeschaltet werden sollten. Oder aber auch, wie es zu einer Infektion kommen konnte. Ziel ist es, diese Fragen mittels einer Simulation beantworten zu können. Außerdem soll versucht werden, Metriken in Bezug auf die Resilienz zu finden, indem durch szenarienbasierte Simulationen eine Korrelation zu Graphen-Eigenschaften hergestellt wird.

2.1 Abgrenzung

Eine Berücksichtigung psychologischer Aspekte bezüglich der Benutzer findet nicht statt, da sie den Rahmen der Masterarbeit sprengen würde. Vielmehr wird eine Grundlage für die angestrebten Untersuchungen geschaffen, weswegen das Nutzerverhalten nur rudimentär umgesetzt wird. Eine Betrachtung der Auswirkungen des Verhaltens soll aber durch entsprechende Erweiterungen möglich sein, welche aber nicht Gegenstand der Arbeit sind.

3 Vorgehensweise

Um die Dynamik eines komplexen Netzwerkes von Rechnern, Peripherie und Nutzern innerhalb einer Simulation zur Ausbreitung von Schadsoftware berücksichtigen zu können, hat sich herausgestellt, dass ein multiagentenbasierter Ansatz gut geeignet ist (Harms, 2012).

3.1 Ausbreitungssimulation mittels Multiagenten-Systemen

Bei diesem Ansatz gibt es mehrere Möglichkeiten der Modellierung. Eine ist, die Malware als Agenten zu modellieren und physische Rechner als Ausführungsumgebung zu verwenden. Somit ist es notwendig ein Replik der zu testenden Umgebung zu schaffen (Leszczyna u. a., 2008). Hier steigt der Hardwarebedarf allerdings proportional mit der Größe des zu testenden Systems.
Eine weitere - und hier verfolgte - Möglichkeit ist, zusätzlich zur Schadsoftware auch die weiteren Komponenten als Agenten zu modellieren. Dies hat den Vorteil, dass der Hardwareaufwand stark reduziert wird, da auf einen physischen Systemnachbau verzichtet wird. Stattdessen wird das zu testende System virtuell als Verbund von Agenten nachgebildet. Somit lassen sich auch größere Netze bei geringerem Hardwareeinsatz testen. Ein Nachteil ist der höhere Modellierungsaufwand, da nicht auf die physischen Systemeigenschaften zugegriffen werden kann.

3.2 Modellierung

In einem Multiagentensystem sind Agenten gleichberechtigte Kommunikationspartner, die in direkten Informationsaustausch treten können. Mögliche Formen sind Unicast, Multicast oder Broadcast. Routing, wie man es von Kommunikationsnetzen kennt, findet in der Regel keines statt. Um die Eigenschaften von realen Informationssytemen abbilden zu können, werden daher die Kommunikationswege der Agenten durch ein logisches Overlaynetzwerk eingeschränkt (Abb. 1). Dadurch lassen sich zum Beispiel auch Filterregeln von Firewalls, Zugriffskontrollen modellieren. Auch das Darstellen von Mobilität (Dynamik) wird hiermit ermöglicht. Hierfür wird der Overlaygraph, und somit mögliche Kommunikationspartner, geändert.

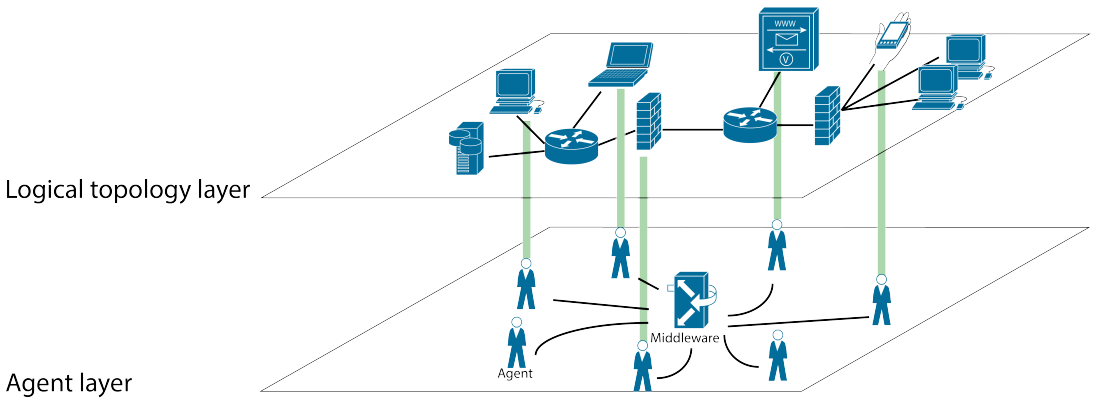

Abbildung 1: Overlay über Agenteninfrastruktur mit zentraler Kommunikationsmiddleware

Die einzelnen Agenten erhalten je nach Realvorbild entsprechende Eigenschaften und Fähigkeiten, die sich nach der installierten Software richten. Diese werden als Agentbehaviours implementiert (z.B. MailserverBehaviour). Durch die unterschiedlichen Behaviours ergibt sich ein Baukastensystem, mit dem Agenten ad-hoc zusammengesetzt werden können. Dies ermöglicht eine hohe Flexibilität beim Erstellen eines Versuchsaufbaus.
Um eine Simulation durchführen zu können, werden Daten benötigt, die die Versuchsumgebung beschreiben. Hierzu gehört die zu untersuchende Infrastruktur mit ihrer Topologie, Diensten und Einstellungen. Diese Informationen können dann vom Simulator dazu genutzt werden, um die nötige Versuchsumgebung aufzubauen. Der Vorgang zum Aufbau wird durch Bestandteile des Gesamtprozesses (Abb. 2) unterstützt . Dieser setzt sich folgendermaßen zusammen:

1. Das Information-Gathering [1] beschreibt den Teilprozess des Erfassens, des Ist-Zustands eines Netzes, und findet weitgehend automatisiert statt. Es werden neben der Topologie auch angebotene Dienste und Softwareversionen (Patchlevel) ermittelt (Krauß, 2012). Die Automatisierung dieses Teilprozesses wurde von Robert Krauß im Rahmen seiner Bachelorarbeit vorgenommen.

2. Anschließend an das automatische Information-Gathering werden die gesammelten Daten überprüft und gegebenen Falls bearbeitet. Dies geschieht mittels eines grafischen Editors, welcher in einer studentischen Arbeit durch Stephan Paulsen und Erhan Yilmaz implementiert wurde.

3. Die zusammengestellten Daten werden mittels eines Konverters in eine formale Beschreibung umgewandelt. Diese wird genutzt, um die Konfiguration der einzelnen Agenten vorzunehmen. Der so konfigurierte Testfall kann dann simuliert werden.

4. Ist die Simulation beendet, können die Ergebnisse (Ereignisse zur Simulationszeit) analysiert werden.

Abbildung 2: Gesamtprozess

[1] Dieser Schritt kann optional sein, wenn fiktive Umgebungen untersucht werden. In solch einem Fall wird mit 2. begonnen.

4 Aktueller Stand

Im Projekt 1 wurden konzeptionelle Arbeiten durchgeführt, die unter anderem Überlegungen in Abschnitt 3 mit einbeziehen. Wie bereits erwähnt, wurde der automatische Information-Gathering Prozess durch Robert Krauß im Rahmen seiner Bachelorarbeit implementiert. Auch der grafische Editor ist durch Stephan Paulsen und Erhan Yilmaz im Rahmen einer studentischen Arbeit bereits fertiggestellt worden. Die hierfür nötigen Anforderungen und Schnittstellen wurden ebenfalls in Projekt 1 ermittelt und festgehalten.

Für den Simulator wurde im Rahmen vom Projekt 1 evaluiert, welche Werkzeuge sich für eine Umsetzung am besten eignen. Die Wahl fiel auf Python, da es sich für ein schnelles Prototyping gut eignet und ein Framework - SPADE2[2] - zur Multi-Agenten-Enwicklung für Python existiert. Dies hält sich an den FIPA-Standard[3] und lässt sich daher für zukünftige Weiterentwicklungen auch mit anderen FIPA kompatiblen Agenten, welche in in anderen Programmiersprachen implementiert sein können, nutzen (Gregori u. a., 2006).
Im Projekt 2 wurden die Grundlagen für eine modulare Agenten-Architektur auf Basis von SPADE2 geschaffen. Agenten lassen sich hiermit anhand von formalen JSON Beschreibungen konfigurieren und mit bestimmten Verhalten ausstatten. Dies ermöglicht es, die Daten aus dem Information-Gathering- und Bearbeitungsprozess zu konvertieren und flexibel verarbeiten zu können. Zudem wurde das Kommunikationsoverlay (Abb.1 aus 3.2) realisiert. Hierzu wurde ein einfaches Source-Routing mittels Agentbehaviours implementiert.

5 Ausblick

Um die in 2 gesteckten Ziele zu erreichen und generische Aussagen über die Widerstandsfähigkeit eines Netzes gegenüber der Ausbreitung von Schadsoftware treffen zu können, wird ein empirisches Vorgehen angestrebt. Hierfür soll das entwickelte Toolkit aus Information-Gatherer, Editor und Simulator weiterentwickelt und verwendet werden. Insbesondere ist hier noch die Implementierung von Mailserver-Agenten und einfach Benutzer-Agenten nötig.
In definierten Experimentierumgebungen (Netzen) wird ein Malware-Agent mit aktuellen Ausbreitungsmethoden getestet. Mehrere Simulationsläufe, in denen die Netzeigenschaften verändert werden, dienen der Messdatengewinnung. Die gewonnen Daten werden in Bezug auf Eigenschaften der Netzgraphen untersucht.
Interessante Metriken, die hier betrachtet werden sollen, sind die Infektionsrate (Neu-Infektionen

[2] https://github.com/javipalanca/spade
[3] http://www.fipa.org

pro Zeiteinheit), die Replikationshäufigkeit (wie viele Infektionen gehen von einem Knoten aus), Durchsetzung (prozentuale Bestimmung der Gesamtinfektionen) und die Infektions-Distanz (über wie viele Knoten breitet sich Malware im Mittel aus). In Bezug auf die Graphentheorie sind Valenz, Zusammenhangszahl und Clusterkoeffizient interessante Kennzahlen, die Einfluss auf die Propagation von Malware haben könnten. Ob hier Korrelationen vorliegen wird mithilfe der gewonnen Daten untersucht. Die daraus resultierenden Erkenntnisse sollen bei Incident Management Prozessen und Resilienz-Analysen unterstützen.

6 Zusammenfassung

Schadsoftware und ihr Ausbreitungsverhalten spielen bei Angriffen auf Informationssysteme eine wichtige Rolle. Die von Malware ausgehende Gefahr ist durch die gestiegene Mobilität von Geräten und durch das Einsetzen von BYOD-Richtlinien noch gestiegen.
Möchte man erfolgreiche Abwehrstrategien entwickeln oder mögliche Auswirkungen ermitteln, bieten sich Simulationen zum Testen an.

Es wurde ein Ansatz vorgestellt, wie solch eine Simulation realisiert werden kann und welche Faktoren dabei berücksichtigt werden. Zudem wurde gezeigt, welche Problemstellungen mit diesem Ansatz im Rahmen der Masterarbeit untersucht werden sollen und welche Methodik verwendet wird. Die hieraus resultierenden Ergebnisse sollen dann unter anderem zum Erstellen für Incident Manangement Prozessen und Resilenz-Analysen verwendet werden können.

7 Literatur

[Deloitte 2013] Deloitte: Blurring the lines: 2013 TMT Global Security Study / Deloitte Touche Tohmatsu Limited. 2013. – Forschungsbericht

[Gregori u. a. 2006] Gregori, Miguel E. ; Cámara, Javier P. ; Bada, Gustavo A.: A jabber-based multi-agent system platform. In: *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA : ACM, 2006 (AAMAS '06), S. 1282–1284. – URL http://doi.acm.org/10.1145/1160633.1160866. – ISBN 1-59593-303-4

[Harms 2012] Harms, André: Ausbreitungssimulation von Schadsoftware - Simulation of malware propagation. 2012. – Forschungsbericht

[Kaspersky Lab 2012] Kaspersky Lab: Kaspersky Lab Identifies Operation "Red October", an Advanced Cyber-Espionage Campaign Targeting Diploma-

tic and Government Institutions Worldwide / Kaspersky Lab. URL http://www.kaspersky.com/about/news/virus/2013/Kaspersky_Lab_Identifies_Operation_Red_October_an_Advanced_Cyber_Espionage_Campaign_Targeting_Diplomatic_and_Government_Institutions_Worldwide, 2012. – Forschungsbericht. Abgerufen 17.02.2013

[Krauß 2012] Krauss, Robert: Analyse und Aufbereitung sicherheitsrelevanter Informationen von Netzwerkknoten zur Simulation von IT-Angriffen. 2012. – Forschungsbericht

[Leszczyna u. a. 2008] Leszczyna, Rafał ; Fovino, Igor N. ; Masera, Marcelo: MAISim: mobile agent malware simulator. In: *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*. ICST, Brussels, Belgium, Belgium : ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008 (Simutools '08), S. 35:1–35:6. – URL http://dl.acm.org/citation.cfm?id=1416222.1416262. – ISBN 978-963-9799-20-2

[Messner 2011] Messner, Michael: *Metasploit: Das Handbuch zum Penetration-Testing-Framework*. 1. Auflage. d.punkt Verlag, 2011. – 13–58 S

[Symantec Corporation 2011] Symantec Corporation: W32.Stuxnet Dossier / Symantec Corporation. 2011. – Forschungsbericht. abgerufen 01.04.2011

[Symantec Corporation 2012] Symantec Corporation: W32.Disttrack / Symantech Corporation. URL http://www.symantec.com/security_response/writeup.jsp?docid=2012-081608-0202-99, 2012. – Forschungsbericht. abgerufen 17.02.2013

[Wang und Stavrou 2010] Wang, Zhaohui ; Stavrou, Angelos: Exploiting smart-phone USB connectivity for fun and profit. In: *Proceedings of the 26th Annual Computer Security Applications Conference*. New York, NY, USA : ACM, 2010 (ACSAC '10), S. 357–366. – URL http://doi.acm.org/10.1145/1920261.1920314. – ISBN 978-1-4503-0133-6