Proseminar Technische Informatik

Stuxnet

The first Cyberweapon?

Andreas Benzin

10. Juni 2011

Betreuer: Stephan Adler und Matthias Wählisch

Inhaltsverzeichnis

1 Die Signifikanz des Stuxnet-Wurms

Im November 2010 gab der iranische President Mahmoud Ahmadinejad in einer Pressekonferenz bekannt, dass die nationalen Anlagen zur Anreicherung von Uran von einer Schadsoftware sabotiert wurden[1]. Vieles deutete darauf hin, dass der Stuxnet Computerwurm für die Zerstörung von den ca. 1000 Uranzentrifugen verantwortlich war, die die iranische Regierung im angesprochenen Zeitraum ersetzen ließ [4, S. 1f]. Stuxnet war damit die erste Schadsoftware, die die sog. ,,air-gap“ [1, S. 8] überbrückte und eine real existierende, nicht mit dem Internet verbundene, Industrieanlage manipulierte und zumindest teilweise zerstört hat.

In der Vergangenheit beschränkten sich Cyberattacken stets auf Kommunikationsinfrastruktur, die direkt oder indirekt mit dem Internet verbunden war. Beispielsweise wurde im Kaukasus-Konflikt 2008 - zeitgleich mit russischen konventionellen Kriegsoperationen - die georgische Internet-Infrastruktur mit massiven Distributed Denial of Service (DDoS) Attacken angegriffen [1, S. 7]. Damit wurde Georgien für die Zeit des Angriffs vom Internet abgeschnitten. Mit Stuxnet jedoch, wurde ein in der realen physikalischen Welt vorhandenes Industrie-Equipment kompromittiert und zerstört. Die Cyberattacke hat damit evtl. sogar einen konventionellen Militärschlag ersetzt[2]. Insofern ist Stuxnet schon deswegen eine besondere Art von Computerwurm und grenzt sich von allem vorher dagewesenen ab.

Für die Erstellung von Stuxnet wurde ein extrem hoher Aufwand betrieben. So wurden insgesamt vier Zero-Day-Exploits in Windows Betriebssystemen, zwei gestohlene digitale Treiberzertifikate und ein Zero-Day-Exploit in der Siemens Software WinCC ausgenutzt [3]. Im Vergleich zu anderer Schadsoftware sind diese Zahlen sehr bemerkenswert, da Zero-Day-Exploits selten in Malware Verwendung finden [6, S. 1] und eher für viel Geld auf dem Schwarzmarkt an Softwarehersteller oder Comuptersicherheitsfirmen verkauft werden (siehe auch [3]). Selbst wenn in der Vergangenheit Zero-Day-Exploits für kriminelle Zwecke verwendet wurden, dann nicht gleichzeitig in dieser hohen Anzahl.

Zusätzlich wählt Stuxnet seine Ziele mit einer für Malware ungewöhnlichen Präzision aus [6, S. 1] und stellt damit sicher, dass nicht anderes Industrie-Equipment versehentlich zerstört wird . Die ungewöhnliche Strategie setzt sich auch bei der Manipulation der Industrieanlage selbst fort, die nach der Kompromittierung nicht sofort einfach zerstört wird, sondern in einem schleichenden und sehr langsamen Prozess in ihrer Leistung vermindert und schließlich unbrauchbar gemacht wird [2].

[1] BBC News: ,,Iran says nuclear programme was hit by sabotage“. http://www.bbc.co.uk/news/world-middle-east-11868596 (04.06.2011)

[2] NDR Fernsehen: ,,Stuxnet: Der erste echte Cyberwar-Angriff“ (Interview mit Ralph Langner). http://www.ndr.de/fernsehen/sendungen/45_min/hintergrund/internetkriminalitaet111.html (04.06.2011)

[3] Frank Rieger: ,,Der digitale Erstschlag ist erfolgt“. http://www.faz.net/-01i43d (04.06.2011)

Aufgrund der Signifikanz dieses Wurms, soll deshalb im Folgenden auf die Funktionen der einzelnen Teile des Schadprogramms eingegangen und die Auswirkungen auf das Zielsystem näher erläutert werden.

2 Funktionsbeschreibung

Das Hauptziel von Stuxnet ist es, ein Sabotageprogramm auf speicherprogrammierbaren Steuerungen (SPS) von Siemens zu installieren. Diese Controller werden beispielsweise dazu eingesetzt, Ventile, Motoren oder Leistungsschalter in industriellen Anlagen, wie z.B. Kraftwerken, Chemiewerken oder anderen Produktionsstätten zu steuern und zu regeln. Dabei sind nicht nur die Aktoren an die Steuereinheit angeschlossen, sondern auch alle möglichen Sensoren beispielsweise, die Messwerte aus dem Inneren der Maschine liefern. Auf Basis dieser können dann weitere Entscheidungen über die Steuerung und Regelung der Aktoren abgeleitet werden („MSR“ - Messen-Steuern-Regeln).

Dabei funktionieren die SPS nach dem gleichen Prinzip, wie ein Mikrocontroller, der an die speziellen Aufgaben in der Industrie angepasst ist. Es gibt einen zentralen Prozessor, Hauptspeicher, Programmspeicher und etliche Peripherie, wie Timer, AD-Umsetzer und Hardware-Implementierungen zum Ansprechen von Feldbussen. Die Industrieprozesse lassen sich mit einem solchen Steuerungssystem zu einem hohen Grad automatisieren und optimieren, und sind deswegen heutzutage aus dem Feld der industriellen Automation nicht mehr wegzudenken.

Um ein solches System zu programmieren, wird vom Entwicklungsingenieur zuerst das entsprechende Programm mit einer Software bzw. IDE - wie z.B. Step 7 für die Siemens Simatic SPS - geschrieben. Anschließend wird die kompilierte Software mit Hilfe eines Notebooks, beispielsweise dem sog. „Siemens Field PG“, vor Ort auf die SPS, die evtl. in einem Schaltschrank neben der Maschine steht, hochgeladen (siehe Abb. 1). Dabei laufen sowohl auf dem PC, auf dem die Software entwickelt wird, als auch auf dem Programmier-PC normale Versionen von Microsoft Windows als Betriebssystem. Um solche sensiblen Systeme vor Malwareangriffen zu schützen, wird meistens weder die SPS selbst, noch der Programmier-PC ans Internet, vielleicht sogar nicht einmal an das lokale Netzwerk angeschlossen [3, S. 3]. Damit Stuxnet auf der einen Seite die vernetzten Entwicklungsrechner infizieren kann und auf der anderen Seite die Lücke von den vernetzten PCs zum Programmier-PC („air-gap“) überbrücken kann, hat es einen sehr aufwendigen „malware-dropper“ [2] dabei, der sicherstellen soll, dass die „payload“ (das Rootkit und das Schadprogramm, welches auf der Siemens Simatic SPS laufen soll) auch bis zum Programmier-PC kommt. Ist Stuxnet einmal dort angekommen, werden die Treiber der Programmiersoftware Step 7 bzw. WinCC von Siemens manipuliert, damit anschließend das Sabotageprogramm in den Programmspeicher

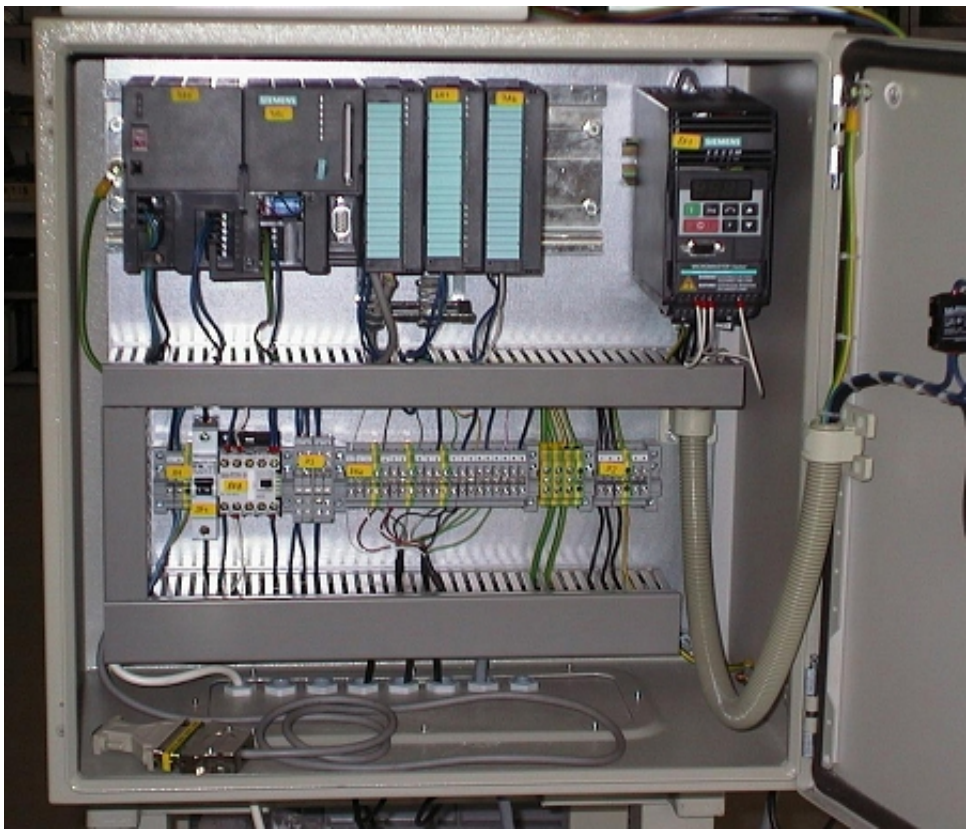

Abbildung 1: Siemens Simatic S7-314 SPS (oben links) in einem kleinen Schaltschrank. Die Programmierschnittstelle (D-SUB 9 Buchse, MPI-Schnittstelle) ist beim zweiten Modul von links zu erkennen (CPU-Modul). (Quelle: http://www.servicelab.co.uk)

der SPS geladen werden kann.

2.1 Der Malware-Dropper

Stuxnet hat eine reiche Anzahl von Methoden zur Selbstreproduktion, Selbstausführung und Verschleierung integriert. Es soll deshalb im Folgenden näher auf den komplexen Installationsvorgang von Stuxnet, das Windows-Rootkit und die Reproduktionsmethoden eingegangen werden.

Die folgenden Erläuterungen basieren hauptsächlich auf den Ausarbeitungen von Symantec, die unter [3] eine sehr ausführliche Dokumentation über die Funktionsweise und die Features von Stuxnet (ermittelt über Reengineering) zur Verfügung stellen.

2.1.1 Die Vorbereitung der Hauptinstallation und das Erlangen von Admin-Rechten

Wurde der Stuxnet-Dropper auf einem Windows-System gestartet (siehe Kapitel 2.1.3 für Reproduktionsmethoden), so wird der komplette Inhalt des Wurms - mit all seinen Funktionen - als eine Dynamic Link Library (DLL) in den Hauptspeicher geladen und die ,,Initial Entry Point"-Funktion [3, S. 12] aufgerufen. Anschließend wird gecheckt, ob sich der Wurm auf einem 32-Bit-Betriebssystem befindet und ob Admin-Rechte vorhanden sind. Konnte der Wurm von einem Account gestartet werden, der Admin-Rechte auf dem Windows-Betriebssystem hat, so wird überprüft, welche Antivirensoftware installiert ist, und anhand daran entschieden, mit welcher Methode und in welchen Windows Systemprozess (z.B. Lsass.exe, Winlogon.exe oder Svchost.exe) der Code für die Hauptinstallationsroutine injiziert werden soll[4].

Hat der Wurm jedoch keine Admin-Rechte, so werden die ersten beiden Zero-Day-Exploits ausgepackt [3, S. 17]. Dabei ist es mit beiden Windows-Schwachstellen möglich, eine Rechteausweitung zu erreichen, sodass der Wurm anschließend in Prozessen läuft, die vollen Zugriff auf das Betriebssystem haben. Läuft Stuxnet auf einem Windows 2000 oder XP System, so wird eine (mittlerweile von Microsoft ausgebesserte[5]) Schwachstelle im win32k.sys Kernel-Mode-Driver ausgenutzt. Hat Stuxnet ein Windows 7 oder Vista Betriebssystem infiziert, so wird eine (noch nicht ausgebesserte) Schwachstelle im Windows Task-Scheduler verwendet, um die Admin-Rechte zu erlangen. Anschließend startet Stuxnet, nun mit vollem Zugriff auf das System ausgestattet, aus dem neuen Prozess heraus die oben erwähnte Injektionsroutine für die Hauptinstallation.

2.1.2 Die Hauptinstallation und das Windows-Rootkit

In der Hauptinstallation findet die eigentliche Infizierung des Rechners statt. Dabei werden die zahlreichen automatisieren Replizierungsmethoden von Stuxnet (siehe Kapitel 2.1.3), das Windows-Rootkit und die manipulierte Step 7 ,,s7otbxdx.dll" zur Absetzung des SPS-Programms (siehe Kapitel 2.2) installiert. Die jeweiligen Installationen finden stets über Code-Injektion in Systemprozesse statt, sodass niemals eigene Prozesse erstellt werden müssen und die komplette Entladung von Stuxnet in völliger Tarnung stattfindet. Im Folgenden wird das Windows-Rootkit erläutert, welches von Stuxnet installiert wird.

Um sicherzustellen, dass Stuxnet bei jedem Start des PCs ausgeführt wird, wird ein Treiber mit dem Namen ,,mrxcls.sys" installiert [3, S. 20] und als Service in der Windows Regis-

[4] siehe http://en.wikipedia.org/wiki/Code_injection für das Angriffsprinzip der Code-Injektion

[5] Microsoft Security Bulletin MS10-073. http://www.microsoft.com/technet/security/bulletin/ms10-073.mspx (06.06.2011)

try angemeldet. Damit Windows diese Treiberinstallation ohne Warnmeldung durchführt, haben die Stuxnetentwickler den Treiber zusätzlich mit einem gestohlenen Zertifikat von Realtek digital signiert (bei späteren Versionen von Stuxnet wurde sogar ein zweites gestohlenes Zertifikat von JMicron verwendet). So wird nun bei jedem Bootvorgang von Windows dieser Treiber geladen und dabei Stuxnet-Code in eine Reihe von System- und Step 7/WinCC-Prozesse injiziert (services.exe, explorer.exe, s7tgtopx.exe, ccProjMgr.exe), um eine ständige Ausführung in völliger Tarnung zu garantieren.

Während der ,,mrxcls.sys“ Treiber die ständige Persistenz und Prozessverschleierung des Wurms sicherstellt, muss zusätzlich das Auslesen von Daten auf Wechseldatenträgern, wie z.B. USB-Sticks, manipuliert werden. Eine der wichtigsten Replizierungsmethoden von Stuxnet ist die Infizierung von jedem in den PC eingesteckten Wechseldatenträger mit einer Kopie von sich selbst und einer Routine, die die Installation auf einem nicht-infizierten PC auslöst. Damit die von Stuxnet auf den USB-Stick geschriebenen Dateien nicht vom Benutzer gesehen werden, wird ein zweiter Treiber mit dem Namen ,,mrxnet.sys“ installiert. Dieser ist auch mit dem besagten gestohlenen Zertifikat signiert. Dieser Treiber fängt jegliche I/O-Request-Packets ab, die vom Windows-Kernel an Applikationen, wie z.B. den Windows-Explorer geschickt werden, damit diese z.B. Ordnerinhalte darstellen und auflisten können. Der Treiber kontrolliert, ob Dateien zu seiner Selbstreproduktion dargestellt werden sollen und löscht diese ggf. aus den Windows-Kernel Antworten heraus. Somit werden jegliche Dateien von Stuxnet auf dem USB-Stick verschleiert und der Benutzer kann keinen Verdacht schöpfen.

Neben den beschriebenen Rootkit-Funktionalitäten wird auch eine Routine gestartet, die über Injektion in einen Windows Internet Explorer Prozess einen Command & Control Server für den Wurm bereitstellt und Backdoor-Funktionalität (z.B. Updates, Statistiken, Kontrollbefehle) über HTTP ermöglicht [3, S. 22].

2.1.3 Reproduktionsmethoden

Stuxnet kann sich über folgende Wege selbst reproduzieren und damit andere PC-Systeme infizieren:

- Computernetzwerke
- Wechseldatenträger
- Step 7 Projektdateien

Die einzelnen Verbreitungswege werden im Folgenden näher erläutert. Abbildung 2 bietet eine vereinfachte Übersicht.

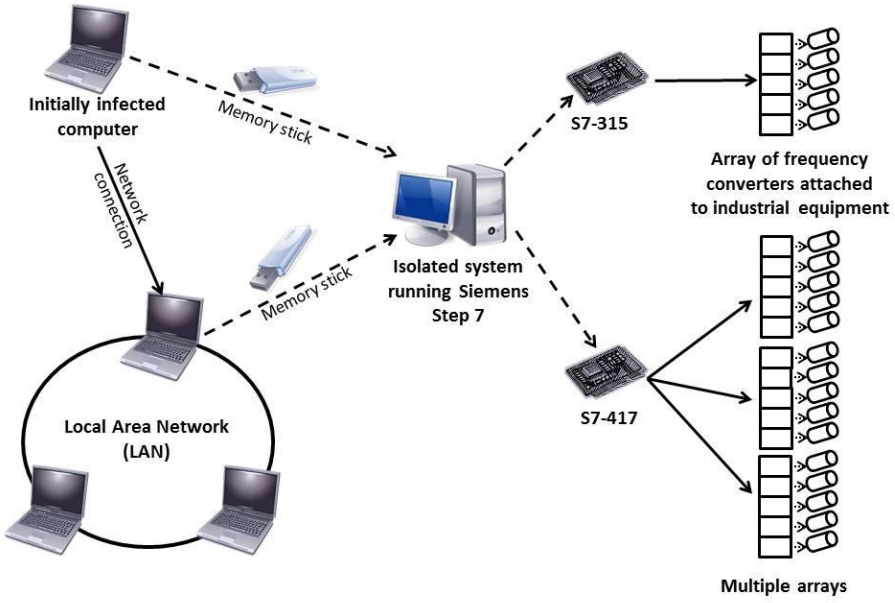

Abbildung 2: Verbreitungswege des Stuxnet-Wurms. Die an die SPS angeschlossenen Frequenzumrichter werden näher in Kapitel 2.2.1 behandelt. (Quelle: [1, S. 3])

Verteilte P2P-Updates Sobald Stuxnet ausgeführt wurde, läuft ein RPC-Server (Remote Procedure Call) im Hintergrund, der über Peer-to-Peer (P2P) Verbindungen zu anderen infizierten Rechnern verteilt Updates von Stuxnet bereitstellt (die wiederum selber diesen Service für andere anbieten). Dabei werden an alle infizierten Teilnehmer im lokalen Netzwerk Versionsinformationen über die momentan installierte Stuxnet-Version verschickt und bei Bedarf ein veralteter client mit einem Update versorgt. Da das RPC-Protokoll bzw. entsprechende Ports meistens (aber nicht unbedingt) von Internetfirewalls in Firmen geblockt werden, ist diese Methode dazu geeignet, auch PC-Systeme auf dem neusten Stand zu halten, die keinen direkten Zugang zum Internet haben und nur in einem Firmennetzwerk miteinander verbunden sind [3, S. 25f].

Verbreitung über Microsoft Dateifreigaben Stuxnet verteilt nicht nur automatisch Updates an alle infizierten Rechner in einem lokalen Netzwerk, sondern infiziert auch alle Windows-Rechner in einem Netzwerk, die Ressourcen über die Microsoft Dateifreigabe freigegeben haben. Ein Stuxnet-Dropper wird auf alle PCs kopiert, für die der Rechner, auf dem Stuxnet läuft, eine Schreibberechtigung hat. Zusätzlich wird mit Hilfe von Befehlen aus der Windows Management Instrumentation (WMI) versucht, Stuxnet an andere Rechner im Netzwerk zu verteilen [3, S. 27]. Über WMI-Befehle findet dann auch die anschließende Fernausführung von Stuxnet statt.

Verbreitung über Microsoft Druckerfreigaben Hier wird der dritte Zero-Day-Exploit ausgepackt. Hat ein PC in einem Netzwerk einen Drucker freigegeben, so können üblicherweise alle Teilnehmer in diesem LAN diesen Drucker benutzen. Mit der (mittlerweile von Microsoft behobenen[6]) „Printer Spooler Vulnerability“ ist es möglich, einen korrupten Druckauftrag an diesen Rechner zu schicken, dadurch eine Datei in den %System%-Ordner einzuschleusen und die beinhaltenden Dateien auf diesem Rechner fernauszuführen [3, S. 28].

Verbreitung über Windows Server Service Diese Schwachstelle ermöglicht die Fernausführung von Code auf einem ungepatchten Computer, indem ein spezieller String über das Server Message Block Protokoll verschickt wird. Diese Sicherheitslücke war zu der Zeit, als Stuxnet in Umlauf gebracht wurde (erste Infektion am 23. Juni 2009 [3, S. 8]), bereits von Microsoft geschlossen worden[7]. Werden von der Firmenfirewall RPC Anfragen nicht geblockt, funktioniert dieser Verbreitungsweg auch über das Internet.

Verbreitung über Siemens WinCC Datenbank Mit der WinCC Software von Siemens („Windows Control Center“) kann eine grafische Mensch-Maschine-Schnittstelle (Human Machine Interface - HMI) auf Windows PCs realisiert werden[8]. Sie wird meist zur Prozessvisualisierung benutzt. Zusätzlich bietet sie eine SQL-Datenbank an, um Messwerte für die nachträgliche Betrachtung zu speichern (z.B. zur Darstellung von Lastkurven über längere Zeiträume). In dieser WinCC Datenbanksoftware von Siemens nutzt Stuxnet einen Zero-Day-Exploit[9] aus. Über ein hartcodiertes Passwort in der Datenbanksoftware ist es möglich, Zugriff auf die komplette Datenbank zu erhalten. Stuxnet nutzt diese Schwachstelle aus, um den Rechner zu infizieren, auf dem diese Datenbank läuft [3, S. 26f] . Dazu trägt Stuxnet eine Kopie von sich als ASCII-String in die Datenbank ein, kopiert diesen String auf dem Server auf die Festplatte und führt anschließend das Programm auf dem Datenbank-Server aus. Zusätzlich verändert Stuxnet spezielle Routinen in der Datenbank, sodass sichergestellt wird, dass jedes Mal, wenn diese Routinen verwendet werden (wenn z.B. jemand etwas von der Datenbank abruft) Stuxnet auf dem Server ausgeführt wird.

Verbreitung über Wechseldatenträger Dieser Verbreitungsweg ist ein essentieller Bestandteil von Stuxnet, da er sicherstellt, dass PC-Systeme infiziert werden, die an kein Netzwerk angeschlossen sind. Erst hierdurch kann Stuxnet bis zu den isolierten Notebooks

[6] Microsoft Security Bulletin MS10-061. http://www.microsoft.com/technet/security/bulletin/ms10-061.mspx (06.06.2011)
[7] Microsoft Security Bulletin MS08-067. http://www.microsoft.com/technet/security/bulletin/ms08-067.mspx (06.06.2011)
[8] http://www.automation.siemens.com/mcms/human-machine-interface/de/visualisierungssoftware/Seiten/Default.aspx (06.06.2011)
[9] CVE-2010-2772. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-2772 (06.06.2011)

(„Siemens Field PGs“, siehe Abb. 3) gelangen, die an die SPS angeschlossen werden.

Abbildung 3: Russischer Techniker im iranischen Kernkraftwerk Buschehr mit einem Siemens Field PG Notebook (Quelle: http://www.upi.com)

Bei der Installation von Stuxnet wird auch eine Routine gestartet, die permanent überwacht, ob Wechseldatenträger eingesteckt werden und sich bei neu erkannten Geräten sofort auf diese kopiert. Die kopierten Dateien sind durch das installierte Rootkit zum Maskieren von Stuxnet-Dropper-Dateien vom Benutzer nicht sichtbar. Zusätzlich zur Stuxnet-Kopie wird auch eine .lnk-Datei mitgeliefert, die die Installation beim Einstecken des Wechseldatenträgers auf einem nicht-infizierten PC auslöst. Hier kommt der vierte und letzte Windows-Zero-Day-Exploit zur Anwendung, wobei dieser auch einer der mächtigsten ist. Diese Sicherheitslücke erlaubt Code-Execution, wenn die Windows Shell (GUI) versucht das Icon, einer speziell präparierten Link- bzw. Shortcut-Datei (.lnk) darzustellen[10]. Wird also der infizierte USB Stick auf einem nicht-infizierten System eingesteckt und sein Inhalt, beispielsweise im Windows-Explorer, betrachtet, so wird der Stuxnet-Dropper sofort ausgeführt. Dabei wird vor der üblichen Installationsroutine ein quick-and-dirty Rootkit installiert, um die Stuxnet-Dateien auf dem USB-Stick so schnell wie möglich auf dem noch nicht infizierten Rechner zu verbergen [3, S. 30]. Anschließend wird die normale Installationsroutine von Stuxnet wie gewohnt ausgeführt.

[10] Microsoft Security Bulletin MS10-046. http://www.microsoft.com/technet/security/bulletin/ms10-046.mspx (06.06.2011)

Verbreitung über Step 7 Projektdateien Zusätzlich zu den beschriebenen direkten Verbreitungswegen infiziert Stuxnet auch Step 7 Projekte, die von den Entwicklungsingenieuren zum Abspeichern ihrer Quellcodes für die SPS verwendet werden. Dabei wird ein Stuxnet-Dropper in den Projektordner kopiert und die Projetdatei so modifiziert, dass der Stuxnet-Dropper bei jedem Projektstart ausgeführt wird [3, S. 33]. Wird nun das Step 7 Projekt auf einem anderen Rechner ausgeführt, z.B. um die kompilierten Binärdateien des Projekts auf eine SPS zu flashen, wird während des Öffnens Stuxnet auf diesem Rechner installiert.

2.2 Die Payload

Die folgenden Erläuterungen basieren hauptsächlich auf den Reengineering-Ergebnissen und Erkenntnissen von Symantec [3] und Ralph Langner [2, 7].

Das eigentliche Ziel von Stuxnet ist es, ein Schadprogramm in eine SPS einzuschleusen und damit im Endeffekt das an die SPS angeschlossene Industrie-Equipment zu zerstören bzw. zu sabotieren. Um das Schadprogramm auf die SPS flashen zu können, tauscht Stuxnet auf jedem infizierten Rechner auch immer den Step 7 Treiber „s7otbxdx.dll“ aus. Dieser Treiber bietet normalerweise Funktionen an, die es dem Programmierprogramm Step 7 von Siemens ermöglichen, z.B. Daten auf die SPS zu schreiben oder Daten von der SPS zu lesen [3, S. 38]. Der Stuxnet Treiber implementiert diese Funktionen ebenfalls, verändert diese jedoch so, dass er die volle Kontrolle über den kompletten I/O-Verkehr zur SPS hat.

Dabei werden im Wesentlichen von diesem Treiber die folgenden zwei essentiellen Funktionalitäten realisiert:

1. Es wird eine Art Rootkit-Funktionalität implementiert, die beim Auslesen des Programmspeichers der SPS die von Stuxnet auf die SPS geflashten Programmdaten verschleiert und nicht anzeigt [3, S. 38]. Somit wird sichergestellt, dass, falls der Programmspeicher der SPS durch den Benutzer ausgelesen wird, dieser nicht wegen der unbekannten Programminhalte Verdacht schöpft.
2. Die Infektionsroutine, mit der Stuxnet sein Sabotageprogramm in den Programmspeicher der SPS schreibt. Die speziellen Sabotageprogramme, die für bestimmte Siemens SPS-Modelle geschrieben wurden, befinden sich ebenfalls (vorerst verschlüsselt) in der „s7otbxdx.dll“ von Stuxnet [7].

Der Infizierungsprozess und die Funktionsweise von Stuxnets Sabotageprogrammen für die Siemens Steuerungen werden im Folgenden näher erläutert.

2.2.1 Infektion der speicherprogrammierbaren Steuerungen

Sobald Stuxnet eine angeschlossene SPS gefunden hat, wird das Sabotageprogramm nicht einfach wahllos kopiert, sondern es werden vorerst einige Überprüfungen vorgenommen und Informationen über das angeschlossene System gesammelt.

Jede Siemens SPS hat einen Speicherbereich, in dem Konfigurationsdaten hinterlegt sind (der sog. SDB - „System Data Block"). Bevor der Stuxnet-Wurm eine SPS infiziert, wird dieser Konfigurationsdatenblock ausgelesen und anhand dessen entschieden, ob die SPS infiziert werden soll.

Da Stuxnet nur zwei spezielle Modelle von Siemens SPS befällt (S7-315-2 und S7-417 Controller [7, 3]), ist dies der erste Wert, den Stuxnet anhand der Konfigurationsdaten überprüft. Ist eine andere SPS angeschlossen, wird Stuxnet diese nicht infizieren. Wurde einer der besagten SPS-Typen gefunden, so wird bei S7-315-2 Controllern zusätzlich überprüft, ob das Zusatzmodul CP342-5 (stellt zusätzliche Profibus-Kommunikation zur Verfügung) vorhanden ist. Wird dieses Modul benutzt, so wird weiter überprüft, ob mindestens 33 Frequenzumrichter entweder von der finnischen Firma Vacon oder der iranischen Firma Fararo Paya an den Profibus angeschlossen sind [3, S. 39]. Diese Frequenzumrichter werden meist zur Steuerung der Umdrehungsgeschwindigkeit von Drehstrommotoren benutzt. Ist eine dieser Bedingungen nicht erfüllt, so wird die SPS nicht infiziert. Der Selektionsvorgang für S7-417 Controller ist nicht vollständig bekannt, da bei den vorliegenden Code-Samples von Symantec der Code für S7-417 Controller deaktiviert bzw. nicht vollständig war [3, S. 46].

Bei diesem Prozess wird sehr deutlich, dass die Stuxnet-Entwickler nur eine ganz bestimmte Industrieanlage [7] im Visier hatten und durch diese Checks verhindern wollten, dass andere SPS infiziert werden und ein größerer Kollateralschaden entsteht. Es kann außerdem daraus geschlossen werden, dass die Stuxnet-Entwickler ausführliche Kenntnisse über die Konfiguration und Anlagenpläne des Ziels gehabt haben müssen [7]. Wie im Folgenden noch erläutert wird, weist vieles darauf hin, dass die Urananreicherungsanlage in Natanz im Iran das Ziel der Stuxnet-Entwickler war (siehe Kapitel 2.2.4).

2.2.2 Funktionsweise des Sabotageprogramms für S7-315-2 Controller

Wurde ein den obigen Ausführungen entsprechendes Setup für S7-315-2 Controller gefunden, so wird das Sabotageprogramm auf die SPS geladen. Dabei wird das ursprüngliche Programm nicht überschrieben, sondern Stuxnet lediglich hinzugefügt. Stuxnet schleust Code in die Main-Entry Interrupt Service Routine und die Watchdog-ISR ein (zwei Einträge im Interruptvektor der SPS (bei der Siemens SPS „OB - Organization Blocks" genannt),

sodass das Schadprogramm beim Starten sofort ausgeführt wird und über den Watchdog zu jeder Zeit die Kontrolle übernehmen kann. Zusätzlich wird auch die Funktion mit Stuxnet Code ersetzt, die zum Empfangen und Auswerten von Profibus Datenpaketen benutzt wird. Die Originalroutine wird dabei an eine andere Stelle im Speicher kopiert.

Das Sabotageprogramm zerstört nicht einfach das angeschlossene Equipment, indem es z.B. die Motorendrehzahlen extrem erhöht. Stattdessen wurde ein komplizierter Zustandsautomat implementiert, der zunächst die von den Frequenzumrichtern gemeldeten Drehzahlen überwacht und sonst das normale Steuerprogramm vom Entwicklungsingenieur ausführt. Stellt Stuxnet über einen längeren Zeitraum fest, dass die Motoren mit einer Frequenz zwischen 807 und 1210Hz angesteuert werden [3, S. 41], so wird das normale Steuerprogramm einfach angehalten und die Manipulationsroutine von Stuxnet gestartet, die die Motordrehzahlen in einem Bereich von 2 bis 1410Hz [1] für eine bestimmte Zeitdauer von bis zu 50 Minuten [7] variiert. Anschließend wird der Zustandsautomat wieder neu gestartet.

Auch hier wird deutlich, dass die Stuxnetentwickler nur ein ganz bestimmtes System sabotieren wollten, und dies nicht nur durch einen Konfigurationscheck, sondern auch durch Überwachung der Betriebsparameter sicherstellen.

2.2.3 Funktionsweise des Sabotageprogramms für S7-417 Controller

Das Programm für die High-End S7-417 SPS ist sehr viel komplexer als das Programm für die kleinere S7-315-2 SPS. Das Sabotageprogramm funktioniert ähnlich wie das für die S7-315-2, hat jedoch einen komplizierteren Zustandsautomaten. Dabei wird, während die Sabotageroutine aktiv ist, ein Man-in-the-middle-Angriff durchgeführt [7], um angeschlossene Prozessvisualisierungssoftware (und damit die Betriebstechniker) zu täuschen und die Manipulation der Industrieanlage durch Stuxnet zu verschleiern. Dazu wird zuerst für 21 Sekunden [3, S. 47] der Daten-Input von allen an die SPS angeschlossenen Sensoren aufgezeichnet und in den Speicher der SPS geschrieben. Anschließend wird dieses aufgezeichnete Prozessimage von Eingangsdaten dem normalen Programm während der Ausführung von Stuxnets Sabotageprogramm statt der echten Eingangsdaten übergeben. Dabei ist bemerkenswert, dass das legitime Programm während der Sabotageattacke nicht gestoppt wird, sondern ganz normal weiterläuft, jedoch mit falschen Sensordaten [7]. Im Folgenden wird dann die Kontrolle über den kompletten I/O-Verkehr der SPS dem normalen Programm entzogen und Stuxnets Sabotageroutine übernimmt die Kontrolle über die Industrieanlage. Dabei werden, nicht wie bei der S7-315-2 Routine Frequenzumrichter über den Profibus angesteuert, sondern digitale Relais-Ausgänge angesteuert. Durch Stuxnet wird hier also ein Denial-of-Control- und Denial-of-View-Angriff kombiniert [7], ohne dass das normale SPS-Programm selbst und die Benutzer der Anlage davon etwas merken.

2.2.4 Verbindungen vom Stuxnet Code zur iranischen Urananreicherungsanlage in Natanz

Viele Hinweise in Stuxnets Datenstrukturen und Verhaltensweisen deuten darauf hin, dass das Ziel des Sabotageprogramms die iranische Urananreicherungsanlage in Natanz war.

Um Kernbrennstoff für Kernreaktoren oder Kernwaffen herzustellen, wird angereichertes Uran benötigt. Um dieses herzustellen, werden Uranzentrifugen mit natürlichem Uranhexafluorid befüllt und diese Zentrifugen in sog. Kaskaden angeordnet, um das von einer Zentrifuge angereicherte Material von einer Zentrifuge in die nächste zu füllen und so den Anreicherungsprozess zu optimieren. Um einen technisch nutzbaren Prozentsatz von angereichertem Uran in einer realistischen Zeitdauer zu erzeugen, müssen sehr viele Zentrifugen vorhanden sein (siehe Abb. 4). Dabei müssen einerseits die Geschwindigkeiten der Zen-

Abbildung 4: Urananzentrifugen in der iranischen Urananreicherungsanlage in Natanz (Quelle: http://www.upi.com)

trifugenrotoren und andererseits die Ventile zum Weiterleiten des angereicherten Urans gesteuert werden[11]. Zusätzlich müssen die Zustandsvariablen, wie z.B. Temperatur und Druck überwacht werden.

Einen ersten Anhaltspunkt liefern die Datenstrukturen aus den Sabotageroutinen für die S7-315-2 und S7-417 Steuerungen. Aus dem Code ist ersichtlich, dass die Schadroutine

[11] Ralph Langner: http://www.langner.com/en/2011/01/30/ (10.06.2011)

für die S7-417 SPS mit einem 6x164 Input-Daten-Array arbeitet[12]. Dabei ist aus einem Bericht der IAEA (Internationale Atomenergie-Organisation) bekannt[12], dass in Natanz 6 Kaskaden mit jeweils 164 Zentrifugen betrieben werden [4, S. 2]. Des Weiteren ist aus dem S7-315-2 Code ersichtlich, dass Befehle an 6 Profibusmodule geschickt werden, an die jeweils maximal 31 Frequenzumrichter angeschlossen sind [3, S. 44]. Es ergibt sich die in Abbildung 5 dargestellte Anlagenkonfiguration[13]. Dabei steuert jeweils eine der sechs klei-

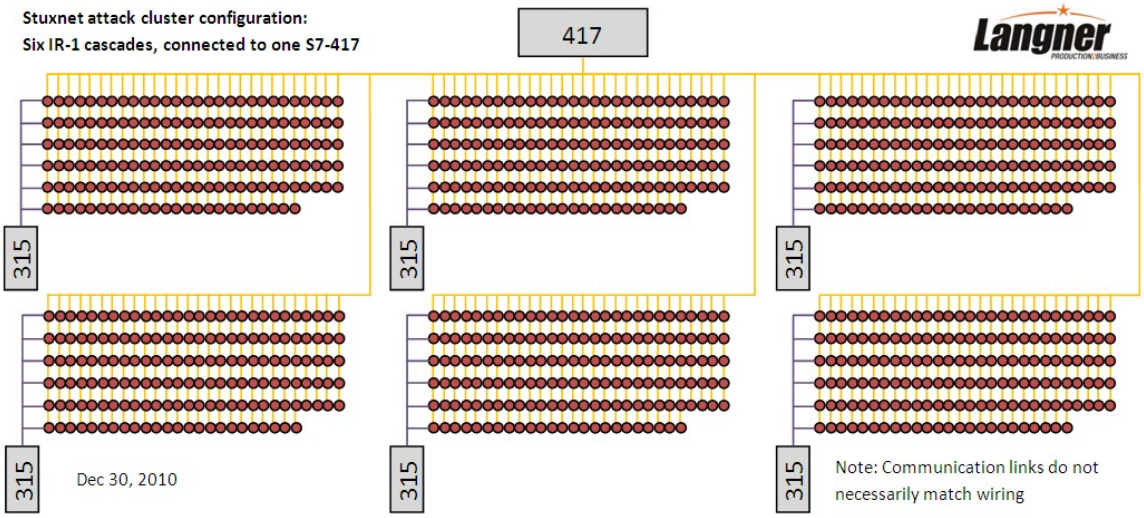

Abbildung 5: Wahrscheinliche Anlagenkonfiguration der Urananreicherungsanlage in Natanz, basierend auf Stuxnet Code-Analysen und Berichten der IAEA (Quelle: http://www.langner.com/en/2010/12/30)

neren S7-315-2 SPS die Rotorengeschwindigkeiten der 164 Zentrifugen in einer Kaskade. Die große S7-417 SPS steuert die kompletten sechs Kaskaden, also insgesamt 984 Zentrifugen. Sie übernimmt die Steuerung der Pumpen und Ventile zur Transferierung des Uranhexafluoridgases zwischen den Zentrifugen, die Überwachung der Betriebsparameter und die Steuerung des Sicherheitssystems[11]. Die Urananreicherungsanlage wurde von Stuxnet also von zwei verschiedenen Seiten angegriffen und es wurde durch die Manipulation der S7-417 Steuerung zusätzlich sichergestellt, dass das Sicherheitssystem (z.B. zur Notabschaltung bei abweichender Drehgeschwindigkeit der Zentrifugenrotoren) nicht funktionierte und die Anzeigen der Betriebsparameter keine Veränderungen zeigten.

Zusätzlich zu der Bestätigung der Zahlen durch den Aufbau der Anlage wird der Verdacht auch durch die Ereignisse nach der Erstinfektion durch Stuxnet am 23. Juni 2009 erhärtet: Der Iran tauschte Ende 2009 ca. 1000 seiner Uranzentrifugen aus [4, S. 1]. Außerdem gab der Präsident später bekannt, dass die Urananreicherungsanlage in Natanz durch Schadsoftware sabotiert wurde[1].

[12] Ralph Langner: http://www.langner.com/en/2010/12/27/ (10.06.2011)
[13] Ralph Langner: http://www.langner.com/en/2010/12/30/ (10.06.2011)

Einen weiteren Hinweis zu Natanz liefert der S7-315-2 Code zur Manipulation der Drehzahlen der Rotoren. Während die Schadroutine aktiv ist und die Drehzahlen der Motoren variiert werden, fährt Stuxnet die Motoren auch immer wieder auf die Nominalfrequenz von 1064Hz zurück [4, S. 3]. Nach [4, S. 4] werden die Zentrifugen in Natanz genau mit dieser Drehzahl betrieben.

Auch die maximale Frequenz, mit der Stuxnet die Motoren drehen lässt, gibt einen Hinweis auf Natanz. Die maximale Belastbarkeit einer Zentrifuge in Natanz beträgt 1400Hz bis 1432Hz [4, S. 4]. Stuxnet erhöht die Frequenz auf maximal 1410Hz. So werden die Zentrifugen maximal belastet und mehr oder weniger langsam zerstört. Aus den Berichten von [4, S. 4] ist bekannt, dass die verwendeten Zentrifugen in Natanz sehr sensibel auf zu starke Erschütterungen bzw. zu hohe Rotorgeschwindigkeiten reagieren.

Des weiteren hat Ralph Langner unter [11] gezeigt, dass der Programmcode für die S7-417 die Verteilung des Uranhexafluoridgases auf die Zentrifugen so verändert, dass die Ausbeute an angereichertem Uran extrem vermindert wird.

3 Fazit

Der Stuxnet-Wurm ist der aggressivste und revolutionärste Wurm, der in den letzten 20 Jahren entdeckt wurde[14]. Er ist das erste Schadprogramm, das eine vom Internet isolierte industrielle Anlage aus der virtuellen Welt heraus sabotiert und zerstört hat. Die Präzision des Angriffs glich einer militärischen Kommandoaktion. Zudem wurde mit Stuxnet der erste erfolgreiche Man-in-the-Middle-Angriff bei einer Industriesteuerung durchgeführt.

Stuxnet kann deswegen zu Recht als die erste echte Cyberweapon bezeichnet werden. Eventuell wurde mit Stuxnet der Beginn eines Cyberkrieges markiert, bei dem nicht nur Computer- oder Telekommunikationsnetzwerke, sondern militärische und industrielle Anlagen in der realen, physikalischen Welt mögliche Ziele von Cyberangriffen sind[15].

Obwohl Stuxnet nur für ein bestimmtes Angriffsziel programmiert wurde, sind viele der revolutionären Angriffsprinzipien von Stuxnet generisch und können von anderen kopiert und adaptiert werden[16]. Es sind deswegen in den nächsten Monaten und Jahren unbedingt Vorsichtsmaßnahmen zu treffen, um gegen weitere Cyberweapons à la Stuxnet gewappnet zu sein.

[14] ARTE Reportage vom 07.06.2011: „Vom Digitalangriff zum Cyberkrieg?", Interview mit Alex Gostov von Kaspersky Labs.
[15] ARTE Reportage vom 07.06.2011: „Vom Digitalangriff zum Cyberkrieg?"
[16] Ralph Langner: http://www.langner.com/en/2011/06/07/ (10.06.2011)

Literatur

[1] Shakarian, Paulo:
Stuxnet: Cyberwar Revolution in Military Affairs.
http://smallwarsjournal.com/blog/journal/docs-temp/734-shakarian3.pdf
(04.05.2011)
Small Wars Journal, 15. April 2011

[2] Langner, Ralph:
Cracking Stuxnet, a 21st-century cyber weapon.
http://www.ted.com/talks/ralph_langner_cracking_stuxnet_a_21st_century_cyberweapon.html?awesm=on.ted.com_Langner (04.05.2011)
TED Talks, März 2011

[3] Falliere, Nicolas; O Murchu, Liam; Chien, Eric:
W32.Stuxnet Dossier Version 1.4 (February 2011).
http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf (04.05.2011)
Symantec Corporation, Februar 2011

[4] Albright, David; Brannan, Paul; Walrond, Christina:
Did Stuxnet Take Out 1,000 Centrifuges at the Natanz Enrichment Plant? (ISIS Report).
http://isis-online.org/uploads/isis-reports/documents/stuxnet_FEP_22Dec2010.pdf (04.05.2011)
Institute for Science and International Security, 22. Dezember 2010

[5] Chen, Thomas:
Stuxnet, the real start of cyber warfare? [Editor's Note].
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5634434&isnumber=5634431 (04.05.2011)
Network, IEEE , vol.24, no.6, pp.2-3, November-December 2010
doi: 10.1109/MNET.2010.5634434

[6] Chen, T.M.; Abu-Nimeh, S.:
Lessons from Stuxnet.
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5742014&isnumber=5741994 (04.05.2011)
Computer , vol.44, no.4, pp.91-93, April 2011
doi: 10.1109/MC.2011.115

[7] Langner, Ralph:
How to Hijack a Controller - Why Stuxnet Isn't Just About Siemens' PLCs.
http://www.controlglobal.com/articles/2011/IndustrialControllers1101.html (04.05.2011)
Control Magazine, 13. Januar 2011