Fachhochschule
Münster University of
Applied Sciences

Fachbereich
Elektrotechnik und Informatik

Schadsoftware für vernetzte speicherprogrammierbare Steuerungen

Implementierung eines Computerwurms für Siemens SIMATIC S7-1200 Geräte

Masterarbeit
von
Maik Brüggemann

Erstprüfer Prof. Dr.-Ing. Sebastian Schinzel

Zweitprüfer Hendrik Schwartke

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Münster, 30. März 2016 Maik Brüggemann

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

DB DATA BLOCK, eine Programm-Organisationseinheit, welche einen globalen Speicher darstellt (Siemens spezifisch).

FB FUNCTION BLOCK, eine Programm-Organisationseinheit, welche ein Unterprogramm darstellt; Sie besitzt lokalen Speicher, der über mehrere Aufrufe erhalten bleibt.

FC FUNCTION, eine Programm-Organisationseinheit, welche ein Unterprogramm darstellt.

OB ORGANISATION BLOCK, eine Programm-Organisationseinheit, welche den Eintrittspunkt in das Programm darstellt (Siemens spezifisch).

POE Programm-Organisationseinheit, strukturiert ein SPS-Programm; vergleichbar mit Unterprogrammen.

SPS Speicherprogrammierbare Steuerung, digitales System zur Steuerung von industriellen Maschinen; (Englisch PLC).

TIA-Portal Totally Integrated Automation Portal, Entwicklungsumgebung zur Programmierung von Siemens SIMATIC Geräten.

1 Digitalisierung der Industrie

Informationstechnische Systeme sind ein wichtiger Bestandteil der Industrie geworden. Sie ermöglichen eine intelligente Vernetzung der einzelnen industriellen Prozesse und erlauben auf diese Weise einen hohen Grad der Automatisierung. Die zunehmende Vernetzung von Informationstechnologie und industrieller Produktion ist auf der Hannovermesse erkennbar. Die bedeutende Industriemesse setzt einen Themenschwerpunkt auf die „Digital Factory“ [1]. In der Politik wird diese Entwicklung als Industrie 4.0 bezeichnet und “für unseren Wohlstand von entscheidender Bedeutung” [2] gewertet. Es ist erkennbar, dass informationstechnische Systeme in der Industrie zukünftig weiter an Relevanz gewinnen werden. Gleiches gilt für einige Sektoren der kritischen Infrastrukturen wie der Energieerzeugung, dem Transport und Verkehrswesen und der Wasserversorgung.

Neben den Vorteilen einer weitgehenden Automatisierung entstehen neue Risiken, denen begegnet werden muss. Mit der Einführung von IT-Systemen sieht sich die Industrie zunehmend klassischen Bedrohungen aus diesem Bereich ausgesetzt. Hackerangriffe auf industrielle Produktionsstätten können hohen finanziellen Schaden verursachen und gefährlich für die Gesundheit der involvierten Personen werden. Die Effektivität solcher Angriffe demonstrierte im Jahr 2011 der Computerwurm Stuxnet [3]. Speicherprogrammierbare Steuerungen der Firma Siemens wurden mit dem Ziel manipuliert, das Atomprogramm des Iran zu verlangsamen bzw. zu stoppen. Der Wurm verbreitete sich in einer iranischen Urananreicherungsanlage, indem er mehrere Schwachstellen in Windows ausnutzte. Die Programme von speicherprogrammierbaren Steuerungen wurden gezielt verändert, um Zentrifugen, welche zur Urananreicherung nötig sind, zu zerstören. Der Computerwurm war für die Verbreitung auf ein klassisches Computersystem angewiesen. Nicht untersucht ist bislang die Frage, ob ein Computerwurm sich ausschließlich über speicherprogrammierbare Steuerungen ausbreiten kann.

Angesichts dieses Forschungsdefizits wird das Konzept eines Computerwurmes auf speicherprogrammierbare Steuerungen übertragen und an einem konkreten Beispiel umgesetzt. Mit der Einführung von IT-Systemen gewinnt die Ethernet-Schnittstelle zur Vernetzung von Industriegeräten an

Bedeutung. Die Arbeit konzentriert sich daher ausschließlich auf eine Verbreitung über diese Schnittstelle.

Im weiteren Verlauf der Arbeit werden zunächst die einzelnen Komponenten von klassischen Computerwürmern beschrieben und an Beispielen erläutert. Im dritten Kapitel wird die Norm IEC 61131 vorgestellt. Diese Norm beschreibt die Eigenschaften von speicherprogrammierbaren Steuerungen.

Weiterhin werden die möglichen Auswirkungen eines Computerwurms auf ein industrielles Netzwerk beleuchtet. Die Komponenten eines Computerwurms werden mit der Norm in Verbindung gesetzt und es wird die theoretische Möglichkeit aufgezeigt, einen Wurm für speicherprogrammierbare Steuerungen zu realisieren.

Um die Machbarkeit praktisch nachzuweisen wird ein Wurm implementiert. Die genaue Vorgehensweise wird dargestellt. Darüber hinaus werden die Auswirkungen des Computerwurms auf die speicherprogrammierbare Steuerung beschrieben und vorhandene Schutzmechanismen auf ihre Wirksamkeit überprüft.

Aus den Erkenntnissen werden Maßnahmen für einen besseren Schutz von speicherprogrammierbaren Steuerungen abgeleitet und weitere Forschungsfragen aufgestellt.

2 Computerwürmer

Computerwürmer sind Programme, welche sich automatisch in IT-Systemen verbreiten, indem sie Schwachstellen ausnutzen. Sie zählen zu der Familie der Schadprogramme (Malware). Diese Arbeit definiert einen Computerwurm wie folgt:

> A "worm" is a program that can run independently, will consume the resources of its host from within in order to maintain itself, and can propagate a complete working version of itself on to other machines [4].

Der erste Computerwurm verbreitete sich im Jahr 1988 und wurde unter dem Namen Morris-Wurm bekannt [5]. Wie jeder Computerwurm besteht er aus vier grundlegenden Komponenten [6]: der Zielerkennung, dem Übertragungsmechanismus, der Ausführungsfunktion und einer Schadfunktion.

2.1 Zielerkennung

Ein Computerwurm beginnt die Infektion mit der Zielerkennung. Er durchsucht ein Netzwerk nach einem bestimmten Merkmal, das geeignet ist, um verwundbare Systeme zu erkennen. Dabei kann es sich z.B. um einen offenen Port handeln, hinter dem ein bestimmter Netzwerkdienst vermutet wird. Das Verfahren, nach dem die Ziele durchsucht werden, beeinflusst die Verbreitungsgeschwindigkeit eines Computerwurms.

Das simpelste Verfahren baut in einer zufälligen Reihenfolge Verbindungen zu IP-Adressen auf. Der Wurm Code Red [7] verbreitete sich im Jahr 2001 durch dieses Verfahren. Eine Schwachstelle in Microsofts IIS Webserver erlaubte dem Wurm, sich innerhalb von 14 Stunden auf 359.000 Computern zu verteilen. Dieses Verfahren ist besonders effektiv, wenn potenzielle Ziele mit einer hohen Geschwindigkeit gescannt werden können. Durch den TCP-Verbindungsaufbau entstandene Latenzen führten dazu, dass Code Red nur einen geringen Teil der verfügbaren Bandbreite

für die Zielerkennung nutzte. Schneller konnte sich der Wurm SQL-Slammer [8] verbreiten. Er sendete im Jahr 2003 ein einziges UDP-Paket, das einen Pufferüberlauf in Microsofts SQL-Server auslöste und die Infektion ermöglichte. Durch die Verbindungslosigkeit von UDP konnten die Pakete in hoher zeitlicher Abfolge versendet werden. Innerhalb 30 Minuten infizierte er 75.000 Opfer. Das Verfahren kann weiter optimiert werden, indem verschiedene Wurm-Kopien unterschiedliche IP-Adress-Bereiche durchsuchen. Existieren nur wenige verwundbare Ziele, sind jedoch die meisten Infektionsversuche nicht erfolgreich, wodurch das Verfahren nur eine langsame Verbreitungsgeschwindigkeit erreicht.

Im Jahr 1988 verbreitete sich der Morris-Wurm. Zu diesem Zeitpunkt waren 56.000 Computer [9] mit dem Internet verbunden. Damit bei dieser geringen Anzahl an möglichen Zielen eine effektive Verbreitung stattfinden konnte, nutzte der Morris-Wurm ein anderes Verfahren [10]. Er wertete die Datei `/etc/hosts` des UNIX-Systems aus, welche IP-Adressen in einer tabellarischen Form enthält. Das Scannen zufälliger IP-Adressen ist dadurch nicht nötig. Kann ein Wurm Informationen auf einem infizierten System nutzen, um weitere Ziele zu bestimmen, erreicht der Wurm eine höhere Ausbreitungsgeschwindigkeit.

Ein weiteres Verfahren setzt eine genaue Kenntnis des Wurm-Autors über die anzugreifenden Ziele voraus. Der Wurm wird mit einer genauen Liste alle Ziele vorprogrammiert. Dieses Verfahren erlaubt eine sehr effektive Verbreitung des Wurms. Die Herausforderung besteht für den Wurm-Autor darin, diese Ziel-Liste zu generieren.

2.2 Übertragungsmechanismus

Nachdem der Wurm ein Ziel identifiziert hat, überträgt er eine Kopie von sich selbst auf das Zielsystem. Für diesen Vorgang ist ein Übertragungsmechanismus erforderlich.

Weil es sich bei einem Computerwurm um ein selbständiges Programm handelt, kann der Übertragungsmechanismus von dem Computerwurm implementiert werden. Der MyDoom Wurm [11] verbreitete sich im Jahr 2004, indem er ein infiziertes System nach E-Mail-Adressen durchsuchte und sich als Anlage an diese sendete. Die notwendige SMTP-Funktionalität war Bestandteil des Wurms.

Einige Würmer implementieren den Übertragungsmechanismus nicht selbst, sondern benutzen einen zweiten Kanal. Der Wurm W32.Blaster [12] nutzte im Jahr 2003 eine Schwachstelle in der RPC-Schnittstelle von Microsoft Windows aus. Für die Verbreitung startete er einen TFTP-Server.

Über diesen konnte während des Infektionsprozesses eine Kopie auf das Ziel-System heruntergeladen werden.

Durch die Übertragung und die Zielerkennung erzeugt der Wurm Netzwerkverkehr, der eine Identifizierung des Wurms erlaubt. Um eine Identifizierung zu erschweren, kann die Übertragung in legitimen Netzwerkverkehr eingebettet werden (vlg. [13]). Ein Wurm könnte sich z.B. über das HTTP-Protokoll verbreiten, indem er eine Schwachstelle in einem Browser und einem Webserver ausnutzt. Sobald ein Webbrowser eine Webseite von einem infiziertem Webserver abruft, wird der Webbrowser infiziert. Seinerseits beginnt Webbrowser alle Webserver zu infizieren, von denen er Webseiten abruft.

2.3 Ausführungsfunktion

Nachdem eine Kopie des Wurms übertragen wurde, muss die Ausführung auf dem Zielsystem gestartet werden. Die Ausführungsfunktion ist neben der Zielerkennung entscheidend für die Verbreitungsgeschwindigkeit.

Ein Computerwurm kann manuell durch einen Benutzer gestartet werden. Der Wurm verwendet dafür Methoden des Social Engineerings. Im Jahr 2000 verbreitete sich der Wurm Loveletter. Er bewirkte, dass Microsoft Outlook eine E-Mail mit der Betreffzeile *ILOVEYOU* an alle Einträge im Adressbuch versendetet. Als Anlage wurde der Wurm angefügt. Die Betreffzeile weckte bei vielen Benutzern Interesse. Sie öffneten die E-Mail-Anlage und starteten so den Wurm. Diese manuelle Art der Aktivierung ist auf eine Benutzerinteraktion angewiesen und begrenzt damit die Verbreitungsgeschwindigkeit.

In einem anderen Verfahren macht sich der Wurm zunutze, dass bestimmte wiederkehrende Aufgaben auf einem Computersystem automatisiert werden. Der Wurm ersetzt eine bestimmte Datei durch sich selbst und wartet darauf, dass er gestartet wird. Hierzu eignen sich z.B. Backup-Skripte oder der Update-Mechanismus eines Programms.

Die höchste Ausbreitungsgeschwindigkeit wird erreicht, wenn sich der Wurm selbst aktivieren kann. Im Jahr 2004 verbreitete sich der Wurm Sasser [14] über eine Schwachstelle in der Microsoft Implementierung des SMB Protokolls. Die Schwachstelle erlaubte dem Wurm, unmittelbar eigenen Code auszuführen. Dieser Code übertrug zunächst eine Kopie des Wurms vom Ausgangspunkt der Infektion und begann danach selbst mit der Suche nach neuen Zielen.

2.4 Schadfunktion

Computerwürmer verursachen häufig einen Schaden auf den betroffenen Systemen. Die eingesetzten Schadfunktionen sind sehr verschieden und von der Motivation des Wurm-Autors abhängig.

In einigen Fällen entsteht dieser Schaden alleine durch die Verbreitung. Der Morris Wurm beanspruchte auf manchen Computer erhebliche Teile der Rechenleistung, wodurch sie ihre eigentliche Aufgabe nicht mehr ausführen konnten. Der Wurm SQL-Slammer überlastete durch die hohe Ausbreitungsgeschwindigkeit insbesondere Netzwerkkomponenten. Die Wurm-Autoren sind in diesen Fällen häufig an der technischen Umsetzung interessiert oder wollen ihren Bekanntheitsgrad steigern.

In anderen Fällen möchte der Wurm-Autor gezielt einen Schaden verursachen. Code Red startet einen Distributed-Denial-of-Service-Angriff auf die Webseite des Weißen Hauses, was eine politische Motivation nahelegt. Der Wurm W32.Blaster führte ebenfalls einen DDoS-Angriff durch. Dieser zielte auf die Microsoft Webseite. Beide Angriffe liefen ins Leere, weil die Betreiber rechtzeitig Gegenmaßnahmen ergreifen konnten.

Andere Wurm-Autoren verfolgen finanzielle Ziele, wie z.B. der Storm Worm [15] belegt. Er verbreitete sich über E-Mail und fügte infizierte Computer einem Botnet hinzu, welches den Versand von Spam-Mails erlaubte.

Geheimdienste entwickeln ebenfalls Computerwürmer. Sie verfolgen damit Spionageziele, wie der Flame Wurm [16] zeigt. Auch zum Zweck der Sabotage lassen sich Computerwürmer einsetzen. Der Wurm Stuxnet [3] zerstörte erfolgreich eine Urananreicherungsanlage im Iran.

3 Industrienorm: IEC 61131

Unter einer speicherprogrammierbare Steuerung (SPS) wird ein digitales System verstanden, welches für den Einsatz im industriellen Umfeld entwickelt wurde. Anweisungen zur Steuerung von Maschinen werden auf einem internen Speicher hinterlegt. Durch digitale oder analoge Ein- und Ausgänge werden die zu steuernden Maschinen mit der SPS verbunden.

Die Norm IEC 61131 definiert die Eigenschaften einer SPS. Insgesamt besteht die Norm aus neun Teilen, welche sich mit Hardware und Software befassen.

3.1 Teil 1 - Allgemeine Informationen

Eine SPS ist ein modulares System. Der erste Teil [17] der Norm beschreibt die grundsätzlichen Eigenschaften der Module und deren Zusammenspiel.

3.1.1 Das CPU-Modul

Das zentrale Modul wird als CPU bezeichnet. Seine Aufgabe besteht darin, das Betriebssystem der SPS auszuführen. Durch das Betriebssystem werden die einzelnen Module koordiniert, der Speicher verwaltet und das Benutzerprogramm ausgeführt. Das Benutzerprogramm enthält die Anweisungen zur Steuerung von Maschinen. Es wird von einem Ingenieur für den konkreten Einsatzfall der SPS entwickelt.

Die CPU verwaltet das sogenannte Prozessabbild. Hierbei handelt es sich um eine Datenstruktur, in der die Zustände aller Ein- und Ausgänge hinterlegt sind. Ein Benutzerprogramm arbeitet ausschließlich auf diesem Abbild. Das Prozessabbild wird zyklisch von der CPU aktualisiert. Abbildung 3.1 zeigt einen Zyklus. Nachdem die Ein- und Ausgänge mit dem Prozessabbild abgeglichen sind, wird das Benutzerprogramm ausgeführt. Zum Ende führt die CPU weitere Aktionen, wie z.B. die Kommunikation mit anderen Geräten aus.

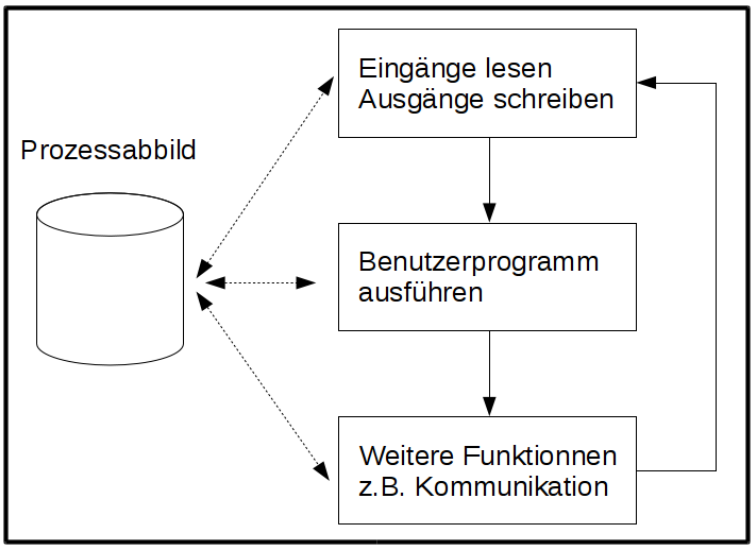

Abbildung 3.1: Zyklus einer SPS

3.1.2 Weitere Module

Ein Modul für die Programmierung erlaubt einem Ingenieur, das Benutzerprogramm auf die SPS zu übertragen. Das Modul vereinfacht das Schreiben und Testen des Benutzerprogramms, indem es interne Zustände der SPS anzeigt, das Programm Schritt für Schritt ausführt oder Eingaben simuliert.

Eine Mensch-Maschine-Schnittstelle ermöglicht dem Endanwender die Industrieanlage zu bedienen und zu überwachen.

Das Kommunikationsmodul erlaubt mehrere SPS untereinander zu vernetzen. Darüber hinaus benutzen die Mensch-Maschine-Schnittstelle und das Modul für die Programmierung die Kommunikationsfunktion, um eine Verbindung mit externen Geräten aufzubauen.

Eine Schnittstellen für Sensoren und Aktoren ermöglicht es, digitale oder analoge Signale zu erfassen und auszugeben. Aufgrund der großen Bandbreite an Sensoren und Aktoren existieren viele verschiedene Schnittellen.

Die Energieversorgung stellt allen Funktionseinheiten eine geeignete Versorgungsspannung zur Verfügung und isoliert diese von der Hauptversorgung.

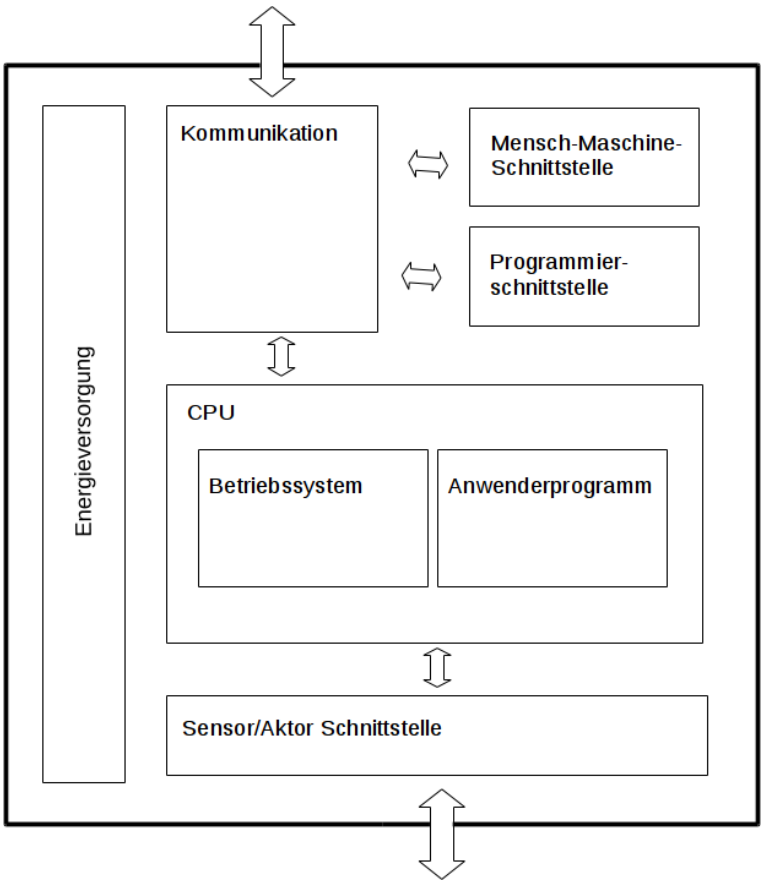

Abbildung 3.2: Aufbau einer SPS nach IEC 61131

3.2 Teil 2 - Anforderungen an die Hardware

Eine SPS ist für den Betrieb im industriellen Umfeld ausgelegt. Die eingesetzte Hardware muss daher robust gegenüber Vibrationen, elektromagnetische Störungen und hohen bzw. tiefen Temperaturen sein. Gleichzeitig darf von der SPS-Hardware keine Gefahr ausgehen. So müssen z.B. maximale Spannungspegel eingehalten werden. Alle Anforderungen an die Hardware einer SPS werden im zweiten Teil [18] der Norm beschrieben.

3.3 Teil 3 - Softwaremodell & Programmiersprachen

Der dritte [19] Abschnitt der Norm IEC 61131 beschreibt den abstrakten Aufbau eines Benutzerprogramms, indem es ein Softwaremodell einführt.

Komplexe Industrieanlagen werden nicht von einer zentralen Instanz gesteuert, sondern von einem Verbund aus SPS-Geräten. Jedes Gerät steuert einen spezifischen Teil der gesamten Anlage. Das Softwaremodell fasst einen Verbund unter dem Begriff Konfiguration zusammen. Die einzelnen SPS werden als Ressource bezeichnet. Das Softwaremodell erlaubt es, globale Variablen und Zugriffspfade zu definieren, über die Informationen innerhalb einer Konfiguration ausgetauscht werden können. Jede SPS führt ein individuelles Benutzerprogramm aus, welches mehrere Ausführungspfade besitzen kann.

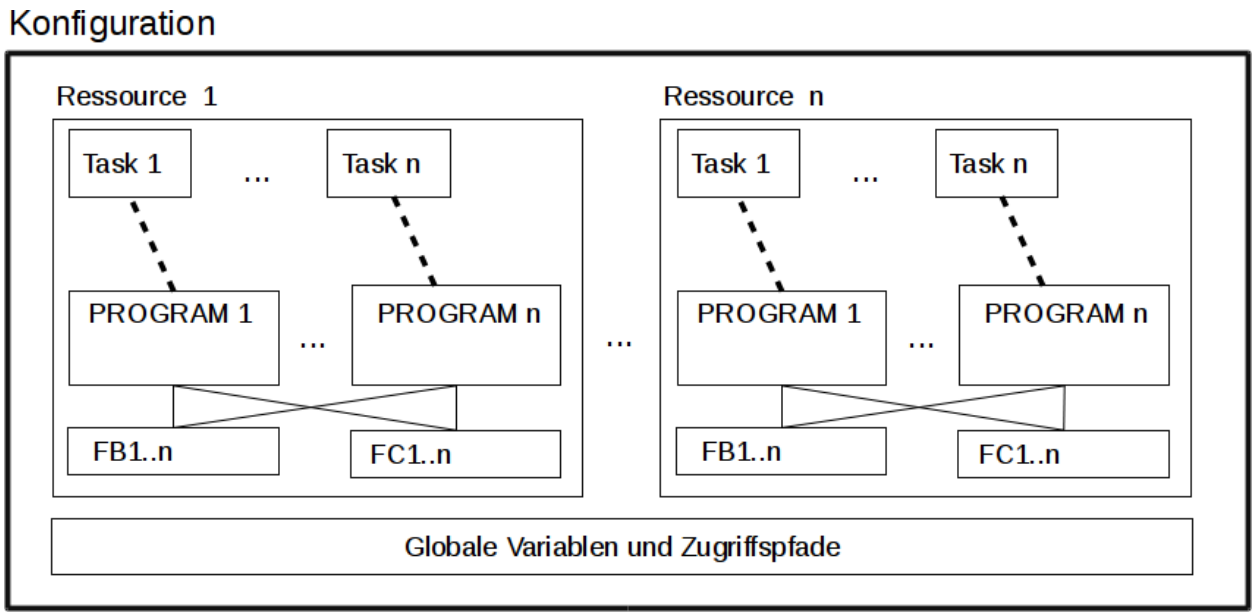

Abbildung 3.3: Das Softwaremodell einer SPS

3.3.1 Tasks

Für jeden Ausführungspfad wird eine Task definiert. Der Prozess-Scheduler des SPS-Betriebssystems kann die zeitliche Ausführung nach drei verschiedenen Verfahren durchführen. Das erste Verfahren führt einen Task unterbrechungsfrei bis zum Ende aus (Non Präemptiv). Das zweite Verfahren unterbricht jeden Task nach einer kurzen Zeitspanne und setzt die Ausführung mit einem anderen Task fort (Präemptiv). Als dritte Möglichkeit kann ein Task ausgeführt werden, wenn ein bestimmtes Ereignis eintritt (Ereignisorientiert). Abbildung 3.4 verdeutlicht die verschiedenen Verfahren. Die Synchronisation der Tasks wird vollständig von dem Betriebssystem übernommen. Jedem Task wird ein Zeitintervall zugeordnet, welches Zykluszeit genannt wird. Nachdem die vorgesehene Zeitspanne verstrichen ist, wird der Task automatisch neu gestartet. Die Programmausführung

einer SPS findet daher in Zyklen statt. Ist ein Task innerhalb eines Zyklus nicht vollständig abgearbeitet, wird ein Fehler ausgelöst. Jedem Task wird eine Programm-Organisationseinheit zugeordnet, dessen Ausführung er kontrolliert.

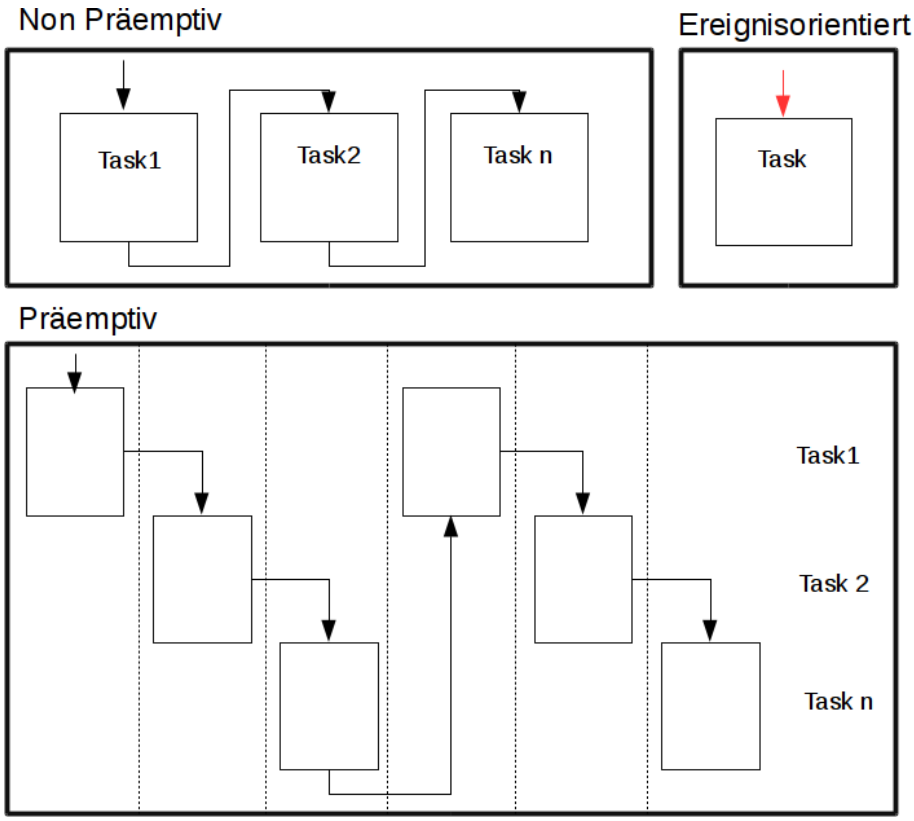

Abbildung 3.4: Verfahren zur zeitlichen Ausführung von Tasks

3.3.2 Programm-Organisationseinheiten

Benutzerprogramme werden durch Programm-Organisationseinheiten (POE) strukturiert. Sie enthalten alle Anweisungen, die erforderlich sind, um eine Maschine zu steuern und werden von einem Ingenieur programmiert. Es wird zwischen drei verschiedenen Typen von POE unterschieden:

- PROGRAM
- FUNCTION BLOCK (FB)
- FUNCTION (FC)

Ein PROGRAM wird von seinem assoziierten Tasks aufgerufen, wodurch es den Eintrittspunkt in das Benutzerprogramm darstellt. Nachdem alle enthaltenen Anweisungen abgearbeitet wurden, wird der Task beendet.

Ein FB bzw. eine FC stellt ein Unterprogramm dar. Beide können Über- und Rückgabeparameter besitzen und verfügen über lokalen Speicher. Sie sind vergleichbar mit Unterprogrammen aus klassischen Programmiersprachen wie C. Der lokale Speicher eines FB bleibt über mehrere Aufrufe erhalten. Identische Übergabeparameter können daher zu unterschiedlichen Rückgabewerten führen.

3.3.3 Programmiersprachen

Die POE können in verschiedenen Programmiersprachen implementiert werden. Insgesamt werden fünf verschiedene Sprachen von der Norm definiert:

- Instruction List (IL) – Vergleichbar mit Assembler
- Structured Text (ST) – Angelehnt an die Hochsprache Pascal
- Ladder Diagram (LD) – Vergleichbar mit einem Elektro-Schaltplan
- Function Block Diagram (FBD) – Vergleichbar mit Logik-Schaltungen
- Sequential Function Chart (SFC) – Ähnelt einem Zustandsdiagramm

Die Programmiersprachen sind entweder graphisch (LD, FBD, SFC) oder textbasiert (IL, ST). Abbildung 3.5 zeigt ein Beispielprogramm in den verschiedenen Sprachen. In allen Sprachen können POE verwendet werden, die in einer anderen Sprache implementiert sind. Je nach Entwicklungsumgebung kann eine Sprache automatisch in eine andere überführt werden. Alle Sprachen bieten somit einen vergleichbaren Funktionsumfang.

3.4 Teil 4 - Benutzerrichtlinien

Der vierte Teil [20] stellt eine Zusammenfassung des gesamten Standards dar. Er richtet sich an den End-Benutzer und schlägt Kriterien vor, die bei der Auswahl und Spezifikation von SPS helfen.

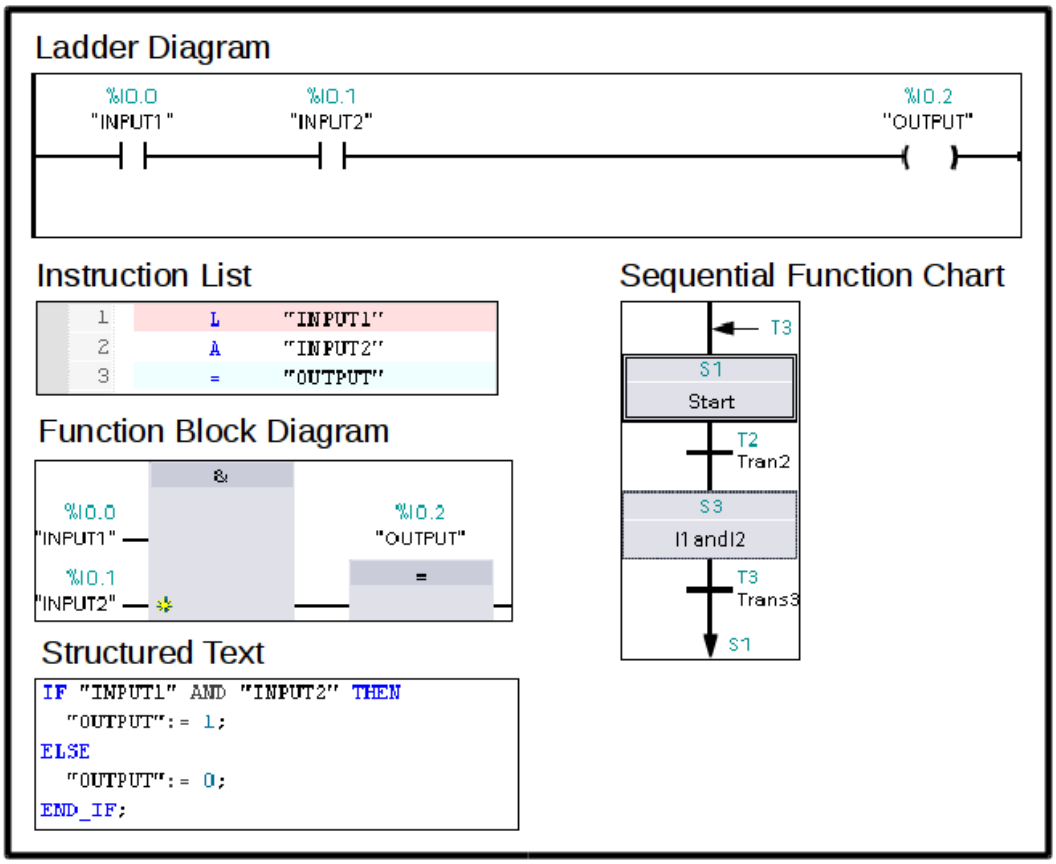

Abbildung 3.5: Beispielprogramme in IL, ST, LD, FBD und SFC

3.5 Teil 5 - Kommunikation

Der fünfte Teil [21] beschreibt, welche Dienste das Kommunikationsmodul anbietet. Die SPS kann je nach Anwendungsfall als Server oder Client auftreten. Die meisten Kommunikationsdienste können aus dem Benutzerprogramm verwendet werden. Dafür sind bestimmte POE vorgesehen.

3.5.1 Kommunikationsdienste

Verbindungsverwaltung

Die Verbindungsverwaltung öffnet, verwaltet und schließt Verbindungen mit Kommunikationspartnern. Dieser Dienst steht dem Benutzerprogramm zur Verfügung. Über die POE mit dem Namen

CONNECT kann das Benutzerprogramm eine Verbindung mit einem entfernten Kommunikationspartner initiieren oder eine Verbindungsanfrage akzeptieren. Eine bestehende Verbindung wird von der Verbindungsverwaltung überwacht und gegebenenfalls automatisch neu aufgebaut. Aktiven Verbindungen wird eine Identifikationsnummer zugeordnet. Andere Kommunikationsdienste können sich mit dieser Nummer auf die Verbindung beziehen.

Geräteprüfung

Die Geräteprüfung gibt Auskunft, ob eine SPS betriebsbereit ist. Die POE *STATUS* ermöglicht dem Benutzerprogramm, den Gesundheitszustand einer entfernten SPS abzufragen. Das Beantworten dieser Anfragen ist ebenfalls durch eine weitere POE möglich.

Alarmmeldungen

Im Falle eines Fehlers kann die Kommunikationsfunktion genutzt werden, um einen Alarm auszusenden. Das Benutzerprogramm generiert eine Alarm-Meldung und richtet sie z.B. an eine andere SPS.

Datenerfassung

Daten werden innerhalb einer SPS durch Prozessvariablen repräsentiert. Durch *USEND*/*URCV* können Variablen zwischen zwei SPS ausgetauscht werden. Einen Datenpuffer von einer beliebigen Länge kann durch die POE *BSEND*/*BRCV* übertragen werden. Die Korrektheit und Vollständigkeit der übertragenen Daten wird von dem Betriebssystem der SPS sichergestellt.

Steuerung und Synchronisierung

Mit Hilfe der POE *SEND* kann eine Anfrage an eine entfernte SPS gerichtet werden. Erst wenn diese durch die *RCV* POE empfangen und beantwortet wurde, ist der *SEND*-Aufruf vollständig. Diese POE dient daher sowohl dem Datenaustausch als auch mehrerer Synchronisierung der SPS untereinander.

Benutzerprogramm Transfer

Das Kommunikationsmodul erlaubt den Transfer des Benutzerprogramms zwischen der Entwicklungssoftware und der SPS. Dem Benutzerprogramm selbst werden keine POE bereitgestellt, um in diesen Prozess einzugreifen. Insgesamt erfüllt dieser Dienst vier Aufgaben:

- Transfer des Benutzerprogramms in die SPS zur initialen Programmierung
- Transfer des Benutzerprogramms in die SPS zur Herstellung eines definierten Zustands
- Transfer des Benutzerprogramms in die Entwicklungsumgebung zur Verifikation
- Transfer des Benutzerprogramms in die Entwicklungsumgebung zur Archivierung/Wiederherstellung des Projektes in der Entwicklungsumgebung

3.6 Teil 6 - Funktionale Sicherheit

Im industriellen Umfeld kann ein Fehler in einer SPS ein Risiko für Mensch, Ausrüstung oder Umwelt bedeuten. Es ist wichtig, auftretende Fehler zu erkennen und das gesamte System in einen sicheren Zustand zu überführen. SPS können mit speziellen Hardware- oder Softwarefunktionen ausgestattet sein, um die funktionale Sicherheit herzustellen. Diese Funktionen werden im sechsten Teil der Norm [22] beschrieben. Der Begriff der funktionalen Sicherheit (eng. Safety) ist von dem Begriff der IT-Sicherheit (eng. Security) abzugrenzen. Während funktionale Sicherheit dem Schutz der Umwelt vor der Maschine dient, bedeutet IT-Sicherheit, dass die Maschine vor der Umwelt geschützt wird. IT-Sicherheit ist kein Bestandteil von IEC 61131.

3.7 Teil 7 - Fuzzylogik

Fuzzylogik ist ein mathematisches Modell, welches eine Verallgemeinerung der Booleschen Logik beschreibt. Neben den Werten „Wahr“ oder „Falsch“ werden Mittelwerte zugelassen. Ziel des siebten Abschnittes [23] ist es, ein gemeinsames Verständnis der Fuzzylogik zwischen Herstellern und Anwendern zu schaffen und eine Integration in die Anwendungsprogramme zu ermöglichen.

3.8 Teil 8 - Leitlinien für Anwendung und Umsetzung von Programmiersprachen

Dieser Teil [24] des Standards präzisiert den dritten Teil, indem er Richtlinien für die Implementierungen der verschiedenen Programmiersprachen vorschreibt. Für Anwendungsentwickler werden Programmierratschläge formuliert.

3.9 Teil 9: IO-Link

IO-Link ermöglicht eine Punkt-zu-Punkt Kommunikation zwischen einer SPS und Sensoren oder Aktoren. Der Standard definiert die elektrischen Eigenschaften dieser Schnittstelle und legt ein einheitliches Kommunikationsprotokoll fest [25].

3.10 Relevanz & Einordnung

Die Norm wurde 1992 erarbeitet, um die verschiedenen Geräte und Programmierkonzepte im industriellen Umfeld zu vereinheitlichen. Die Organisation PLCopen wurde direkt nach der Einführung der Norm gegründet, um die Akzeptanz zu fördern. PLCopen gehören große SPS-Hersteller wie Siemens, ABB, Schneider Electric, Mitsubishi, General Electric und Rockwell Automation an [26]. Die Norm IEC 61131 ist der einzige weltweite akzeptierte Standard für speicherprogrammierbare Steuerungen [27]. Die in diesem Kapitel vorgetragenen Konzepte sind somit für eine Vielzahl von Geräten gültig.

Obwohl die Norm eine Vereinheitlichung der Geräte zum Ziel hat, unterscheiden sich die Implementierungen der Hersteller erheblich. Der Grund dafür wird an den Kommunikationsfunktionen besonders deutlich. Die Norm beschreibt, welche Kommunikationsdienste eine SPS anbieten muss. Ein Protokoll, welches diese Dienste ermöglicht, wird nicht standardisiert. Die Hersteller verwenden daher proprietäre Protokolle, welche sich stark unterscheiden. Darüber hinaus bieten die Hersteller verschiedene Funktionserweiterungen an. Im weiteren Verlauf der Arbeit wird zunächst geprüft, inwieweit SPS nach IEC 61131 allgemein anfällig für Computerwürmer sind. In einem zweiten Schritt wird ein Hersteller ausgewählt und eine Implementierung eines Computerwurms unternommen. Dieses Beispiel belegt, dass die theoretischen Annahmen zutreffend sind.

4 Computerwürmer im industriellen Umfeld

Um den Sicherheitsrisiken für ein Industrienetzwerk zu begegnen, wird von vielen Institutionen eine Defense in Depth Strategie empfohlen [28] [29] [30] [31]. Kernelement dieser Strategie ist es, die verschiedenen Ebenen eines Industrienetzwerkes durch Sicherheitsmaßnahmen voneinander zu isolieren. Abbildung 4.1 zeigt die verschiedenen Ebenen eines Industrienetzwerkes [32]. Die drei obersten Ebenen sind mit der Organisation und Planung der Produktion befasst. Klassische IT-Systeme werden in diesen Bereichen eingesetzt. Durch die zunehmende Automatisierung werden Geräte immer häufiger vertikal zwischen den Ebenen vernetzt. Gelingt einem Angreifer der Einbruch in eine der obereren Ebenen, erschwert die Defense in Depth Strategie das weitere Vordringen in die unteren Ebenen. Dort laufen die kritischen Prozesse zur Steuerung von Maschinen ab. Die Defense in Depth Strategie setzt einen Angreifer voraus, der sich von außen Zugang zu dem Produktionsnetzwerk verschaffen möchte.

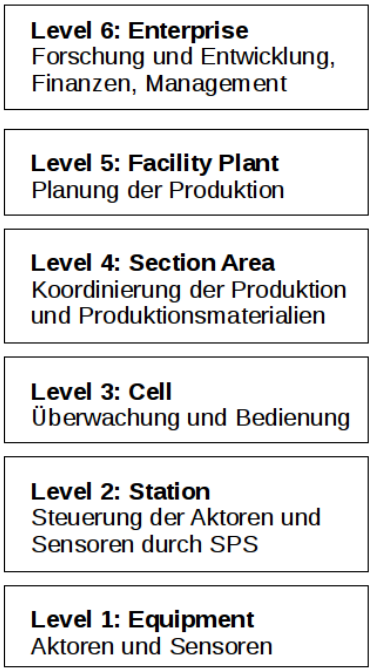

Abbildung 4.1: Fabic Automation Model nach dem Purdue Reference Model

Durch einen Computerwurm im industriellen Umfeld werden andere Angreifer-Modelle möglich. Eine SPS ist nicht Ziel eines Angriffes sondern der Ausgangspunkt. Eine SPS kann bereits vor der Installation im eigentlichen Industrienetzwerk mit einer Schadsoftware versehen werden. Dies kann z.B. durch den Zulieferer oder auf dem Transport zur Produktionsstätte passieren. Ein SPS-Wurm kann sich direkt und ohne Verbindung mit dem Internet in sensiblen Bereichen der Produktionsanlage ausbreiten. Er ist nicht auf die Unterstützung durch einen Computer angewiesen und kann daher nicht mit Standard-Tools, wie einem Virenscanner, identifiziert werden. Ein Anlagenbetreiber hat wenig Möglichkeiten, den

Programm-Code einer SPS auf Schadcode zu analysieren. Die Entwicklungsumgebungen zur Programmierung von SPS sind nicht für diesen Zweck ausgelegt und proprietäre Protokolle erschweren die Entwicklung von unabhängigen Werkzeugen. Einige SPS-Hersteller bieten zum Schutz des geistigen Eigentums Verfahren an, die eine Analyse des Programms unmöglich machen (z.B. Siemens Know-How Schutz [33]). Diese können auch durch einen Wurm genutzt werden.

Ein SPS-Wurm existiert in einem sensiblem Bereich der Produktion und ist einer Analyse nur schwer zugänglich. Durch einen SPS-Wurm entstehen verschiedene Gefährdungen für ein Industrienetzwerk, die im Folgenden erläutert werden.

4.1 Beispielhafte Angriffsvektoren

4.1.1 Denial of Service

Die Verfügbarkeit ist in einem Industrienetzwerk vor der Integrität und der Vertraulichkeit das wichtigste Schutzziel [34]. Häufig müssen die verschiedenen Geräte in Echtzeit untereinander kommunizieren. Ein SPS-Wurm kann bereits durch die Ausbreitung und den damit verbundenen Netzwerkverkehr Ausfälle verursachen. Zusätzlich kann gezielt die Produktion z.B. zu einem bestimmten Zeitpunkt durch den Wurm gestoppt werden. Der in dieser Arbeit implementierte Wurm realisiert eine derartige Schadfunktion (siehe Abschnitt 7.6.3).

4.1.2 Physischer Schaden

Eine SPS ist direkt für die Steuerung von Maschinen verantwortlich. Schadcode ist daher in der Lage, physischen Schaden anzurichten. Das Experiment Aurora [35] belegt diese Aussage. Es wurde gezeigt das ein 2.25 MW Dieselgenerator durch eine gezielte Manipulation zerstört werden kann. Stuxnet [3] richtete ebenfalls physischen Schaden an. Der Programm-Code einer Siemens SIMATIC S7-300 wurde um eine Schadroutine erweitert, um Zentrifugen zu zerstören.

4.1.3 Manipulation weiterer Geräte

Die Prozessvariablen einer SPS werden häufig von weiteren Systemen abgefragt und ausgewertet. Ein SPS-Wurm könnte Werte erzeugen, die ein falsches Lagebild einer Produktionsstätte erzeugen. Sind zudem Schwachstellen in der Verarbeitung dieser Daten vorhanden, können diese ausgenutzt werden. Mit Hilfe der vorhandenen Kommunikationsfunktionen sind auch aktive Angriffe denkbar. Bringt der Computerwurm zusätzlichen Schadcode mit, können weitere Systeme angegriffen werden. Diese Arbeit zeigt beispielhaft in Abschnitt 7.4.2 welche negativen Auswirkungen ein SPS-Wurm auf weitere Softwarekomponenten haben kann.

4.1.4 Überwindung einer Firewall

Eine infizierte SPS kann einen Brückenkopf in ein gesichertes Industrienetzwerk darstellen. Mit Hilfe der Kommunikationsfunktionen kann beispielsweise ein Proxy implementiert werden [36]. Ein Angreifer hat so die Möglichkeit, weitere Ziele in dem Industrienetzwerk anzugreifen. Diese Arbeit implementiert einen Proxy als Schadfunktion (siehe Abschnitt 7.6.2).

5 Ein IEC 61131 kompatibler Computerwurm

Die Norm IEC 61131 definiert verschiedene Programmiersprachen, mit denen komplexe Programme zur Steuerung von Maschinen entwickelt werden können. Damit der Wurm nicht auf weitere Hard- oder Softwarekomponenten angewiesen ist, soll der SPS-Wurm in einer dieser Sprachen implementiert werden. Es ist zu prüfen, inwieweit die verschiedenen Bestandteile eines Computerwurms durch ein IEC 61131 Programm umgesetzt werden können.

Die Infektion beginnt mit der Zielerkennung. Für die Implementierung dieser Funktionalität kann die Verbindungsverwaltung einer SPS genutzt werden. Es wird versucht, mit möglichen Kommunikationspartnern eine Verbindung aufzubauen. Sobald eine Verbindung erfolgreich etabliert ist, wurde ein neues Ziel identifiziert.

Der Übertragungsmechanismus kann ebenfalls realisiert werden. Das Kommunikationsmodul muss den Transfer von Benutzerprogrammen ermöglichen. Ein Benutzerprogramm kann diesen Vorgang nicht direkt steuern, jedoch stehen die Funktionen *BSEND* und *BRECV* zur Verfügung. Mit diesen Funktionen kann ein beliebiger Datenpuffer versendet und auf diese Weise das Protokoll für den Programmtransfer implementiert werden. Der Wurm gibt sich als Entwicklungsumgebung gegenüber einer zu infizierenden SPS aus.

Die Ausführungsfunktion ist implizit durch die SPS gegeben. Aus Sicht der infizierten SPS handelt es sich um einen normalen Programmtransfer. Die SPS wird den Wurm als Teil des Benutzerprogramms ausführen.

Wird der Wurm als normales Benutzerprogramm ausgeführt, können Ausgänge und globale Variablen verändert werden. Schadfunktionen können daher realisiert werden.

Die Norm IEC 61131 spezifiziert die notwendigen Funktionen, um einen SPS-Wurm zu implementieren. Eine generelle Aussage über die Machbarkeit ist damit noch nicht getroffen. Die *BSEND*/*BRECV* Funktionen sind möglicherweise nicht flexibel genug, um das Transferprotokoll

zu implementieren. Ressourcen wie z.B. die Speicherkapazitäten einer SPS sind limitiert und schränken eine Implementierung ein. Außerdem können Hersteller Schutzfunktionen einsetzen, um eine Ausbreitung eines Wurms zu verhindern. Die Machbarkeit wird im Folgenden an einem konkreten Beispiel belegt.

6 Auswahl einer speicherprogrammierbaren Steuerung

6.1 Anforderungen

Der Markt bietet eine Vielzahl verschiedener speicherprogrammierbarer Steuerungen. Aufgrund der wachsenden Bedeutung von Ethernet im industriellen Umfeld werden nur SPS betrachtet, welche diese Schnittstelle besitzen. Die folgenden Anforderungen werden an die SPS gestellt:

- Die SPS muss über eine Ethernet Schnittelle verfügen.
- Das Benutzerprogramm muss über diese Schnittstelle auf die SPS transferiert werden können.
- Das Benutzerprogramm muss beliebige TCP-/UDP-Nachrichten versenden können.

6.2 Marktübersicht

International gehören Mitsubishi Electric, Rockwell Automation, Schneider Electric und Siemens zu den bedeutendsten Herstellern [37]. Das Produktportfolio dieser Hersteller wurde nach den oben genannten Anforderungen untersucht. Das Ergebnis ist in Tabelle 6.1 aufgeführt. Demnach besitzen 96% der untersuchten SPS eine Ethernet-Schnittstelle und erlauben, das Benutzerprogramm über diese Schnittstelle zu transferieren. 54% der untersuchten SPS besitzen zusätzlich die Möglichkeit, TCP oder UDP Nachrichten aus dem Benutzerprogramm zu versenden.

Hersteller	Produkt	Ethernet	Programm-Upload über TCP/UDP	Programmierbare TCP/IP Funktionen
Siemens	S7-300	Ja	Ja	Ja
Siemens	S7-400	Ja	Ja	Ja
Siemens	S7-1200	Ja	Ja	Ja
Siemens	S7-1500	Ja	Ja	Ja
Mitsubishi Electric	MELSEC iQ-R	Ja	Ja	Ja
Mitsubishi Electric	MELSEC iQ-F	Ja	Ja	Ja
Mitsubishi Electric	MELSEC-QS/WS	Ja	Ja	Nein
Mitsubishi Electric	MELSEC-Q	Ja	Ja	Ja
Mitsubishi Electric	MELSEC-L	Ja	Ja	Ja
Mitsubishi Electric	MELSEC-F	Ja	Ja	Nein
Schneider Electric	Modicon Easy M	Nein	Nein	Nein
Schneider Electric	Modicon M	Ja	Ja	Nein
Schneider Electric	Modicon LM	Ja	Ja	Nein
Schneider Electric	Modicon Premium	Ja	Ja	Nein
Schneider Electric	Modicon Quantum	Ja	Ja	Nein
Schneider Electric	Preventa XPS	Ja	Ja	Nein
Rockwell Automation	ControlLogix	Ja	Ja	Ja
Rockwell Automation	CompactLogix	Ja	Ja	Ja
Rockwell Automation	MicroLogix	Ja	Ja	Ja
Rockwell Automation	SmartGuard 600	Ja	Ja	Nein
Rockwell Automation	SLC 500	Ja	Ja	Ja
Rockwell Automation	PLC-5	Ja	Ja	Ja
Rockwell Automation	GuardPLC	Ja	Ja	Nein
Rockwell Automation	Micro800	Ja	Ja	Nein

Tabelle 6.1: Marktübersicht von SPS-Modellen

6.3 Siemens SIMATIC S7-1200

Unter den 13 verbliebenen SPS mit Ethernet und TCP/UDP Funktionen wurde die Siemens SIMATIC S7-1200 aus wirtschaftlichen Gründen ausgewählt. Sie eignet sich für Automatisierungsaufgaben von kleinem Projektumfang und ist die günstigste SPS. Insgesamt existieren vier verschiedene Varianten der Steuerung. Diese unterscheiden sich in den Größen der vorhandenen Speicher (Arbeitsspeicher, Ladespeicher). Als Testgerät wurde die kleinste Variante S7-1211 gewählt. Die Tabelle 6.2 zeigt die technischen Daten der Steuerung.

Abbildung 6.1: Abbildung einer Siemens SIMATIC S7-1200

Kenngröße	**Wert**
Modell	S7-1211
Firmwareversion	3.0.2
Größe des Prozessabbilds	1024 Byte (E) 1024 Byte (A)
Merker (M)	4096 Byte
Arbeitsspeicher	50 KByte
Ladespeicher	1 MByte
Remanenterspeicher	10 KByte
Temporärer (lokaler) Speicher	16 KByte
Bausteine Typ	OB, FB, FC, DB
Baustein Maximalgröße	30 KByte
Bausteine Anzahl	max 1024
Ethernet	10/100MBit
Kommunikationsprotokolle	UDP, TCP, ISO on TCP, PROFINET, S7-Kommunikation, Modbus TCP, HTTP, HTTPS, SNMP, LLDP, NTP, DCP
Programmiersoftware	Totally Integrated Automation Portal V11 SP2 Update 5

Tabelle 6.2: Technische Daten der SIMATIC S71211

Totally Integrated Automation Portal

Das TIA-Portal stellt die Entwicklungsumgehung von Siemens zur Programmierung von SPS dar. Es unterstützt alle in Abschnitt 3.3.3 vorgestellten Programmiersprachen und erlaubt die Konfiguration und Programmierung von Siemens SIMATIC S7 Geräten. Das TIA-Portal ermöglicht außerdem, das Benutzerprogramm auf die SPS zu transferieren. In dieser Arbeit wird das TIA-Portal in Version 11 Update 5 betrachtet.

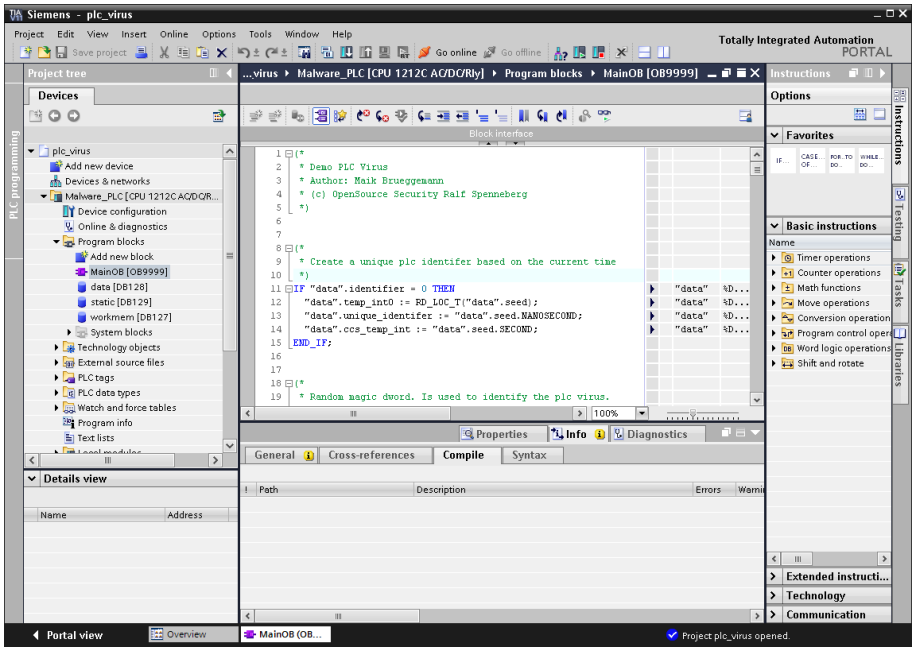

Abbildung 6.2: Screenshot des TIA-Portals

Wichtige Unterschiede zur IEC 61131 Norm

Die Norm IEC 61131 definiert drei verschiedene POE (PROGRAM, FUNCTION BLOCK, FUNCTION). Das TIA-Portal führt einen weiteren Block ein. Der DATA BLOCK (DB) stellt einen globalen Speicher von einer Größe bis zu 64KB dar. Außerdem wird ein PROGRAM als ORGANISATION BLOCK (OB) bezeichnet.

7 Implementierung eines Computerwurms für die S7-1200

7.1 Teststellung

Für die Implementierung des Wurms ist eine Teststellung entworfen worden. Diese besteht aus vier S7-1211 Modellen, welche über einen Switch verbunden sind. Die Geräte können über das Internet-Protokoll (IP) untereinander kommunizieren. Ein Angreifer ist ebenfalls mit dem Switch verbunden und befindet sich in dem gleichen IP-Netzwerk. Er wird eine SPS initial über die Netzwerkschnittstelle infiziert. Der Wurm verbreitet sich von dort auf die weiteren SPS. Die Implementierung gilt als erfolgreich, wenn der Wurm sich von einer SPS (welche nicht die initiale ist) auf eine weitere ausgebreitet hat. Der Versuch wird mit der S7-1211 Firmware 3.0.2 durchgeführt. Die Kompatibilität zu anderen Firmwareversionen wird nicht untersucht.

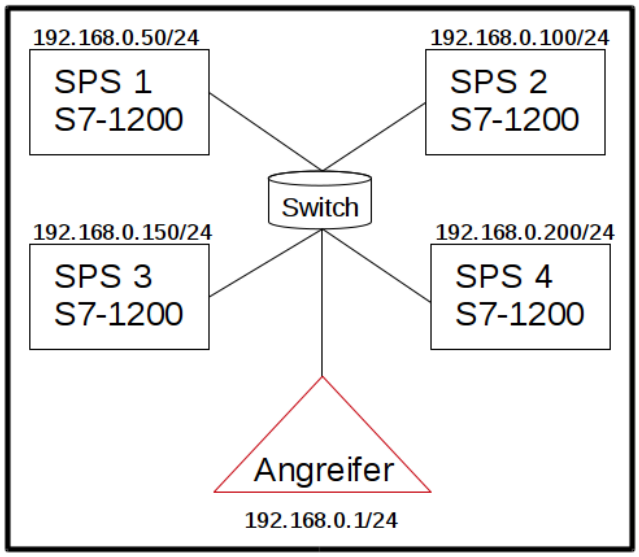

Abbildung 7.1: Skizze der Teststellung

7.2 Architektur

Der Wurm wird als reguläres SPS-Programm entworfen. Er wird in der Programmiersprache SCL (siehe Abschnitt 3.3.3) geschrieben und mit dem TIA-Portal implementiert. Während der Entwicklung müssen die besonderen Anforderungen an ein SPS-Programm berücksichtigt werden. Insbesondere darf die Zykluszeit nicht überschritten werden. Der Wurm muss aus diesem Grund nach wenigen Millisekunden seine Ausführung unterbrechen. In einem folgendem Zyklus kann die Programmausführung fortgesetzt werden. Zu diesem Zweck wird der gesamte Wurm als Zustands-Maschine entworfen. Der aktuelle Zustand wird in einer globalen Variable gespeichert. Zu Beginn jedes Zyklus wird an die Stelle im Programm gesprungen, welche den aktuellen Zustand repräsentiert. Nachdem alle Anweisungen eines Zustands abgearbeitet wurden, findet ein Sprung zum Programmende statt und die Programmausführung wird beendet. Das Programm wird auf diese Weise in kleine Abschnitte unterteilt, wodurch eine Überschreitung der Zykluszeit vermieden wird.

Einen Überblick über den Programmablauf gibt die Abbildung 7.2. Die Infektion beginnt mit dem Aufbau einer Verbindung zu einem möglichen Ziel. Ist diese Verbindung erfolgreich etabliert worden, wird zunächst geprüft, ob das Ziel bereits infiziert ist. Falls keine Infektion vorliegt, wird die Ausführung des Benutzerprogramms auf der angegriffenen SPS gestoppt, um den Programmtransfer zu ermöglichen. Der Wurm überträgt eine Kopie auf das Ziel. Danach wird die SPS gestartet und der Wurm beginnt mit einer anderen IP-Adresse von vorne.

Parallel zum Infektionsvorgang werden die Schadfunktionen ausgeführt. Realisiert werden sie ebenfalls über eine Zustands-Maschine. Eine infizierte SPS versucht eine Verbindung mit einem Command & Controll Server aufzubauen. Sobald das gelungen ist, warte sie auf Kommandos, welche die Schadfunktionen steuern.

Die folgenden Abschnitte erläutert im Detail die Implementierung der einzelnen Komponenten.

7.3 Zielerkennung

Für die Zielerkennung kommen nach Abschnitt 2.1 verschiedene Verfahren in Frage (zufälliges Probieren, Ziel-Informationen aus infiziertem Gerät extrahieren, Ziel-Liste). Für diese Implementierung werden Ziele durch Probieren ermittelt. Siemens SIMATIC SPS können an dem offenen

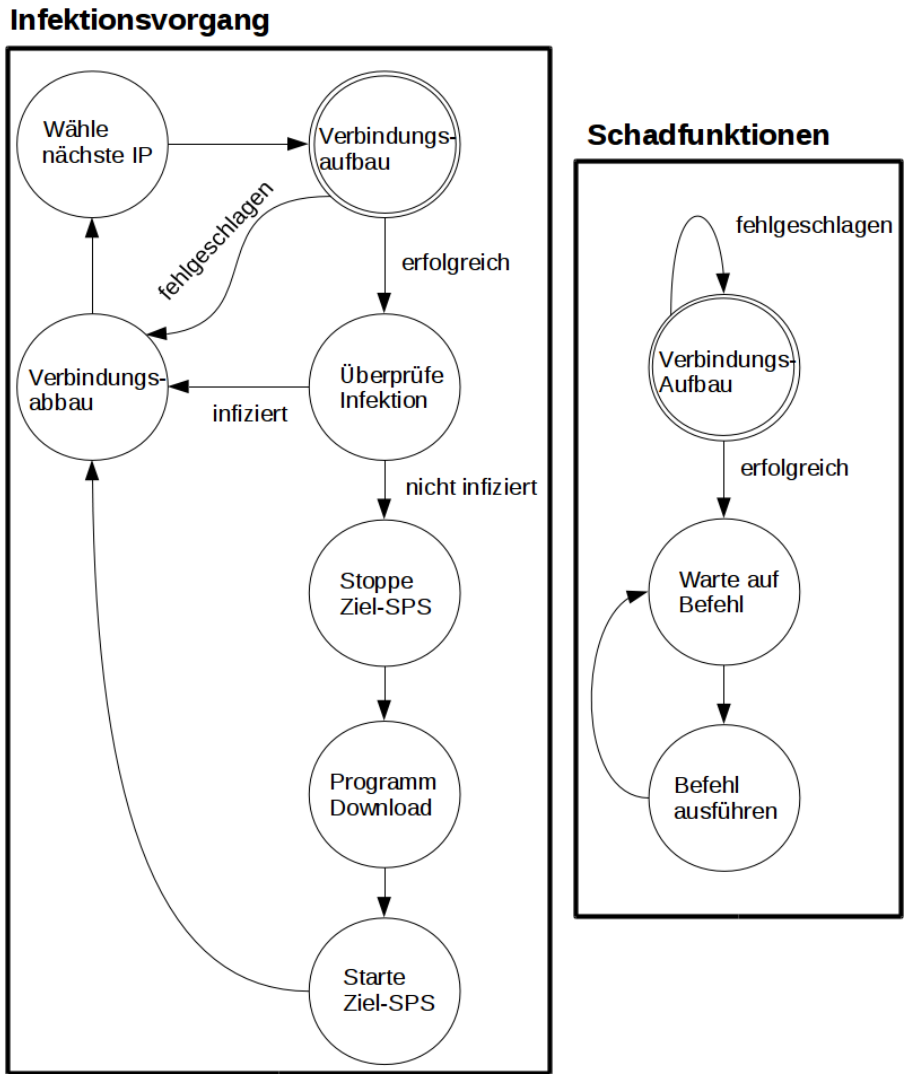

Abbildung 7.2: Programmablauf des SPS-Wurms

TCP Port 102 identifiziert werden. Dieser Port kann durch den Anwender nicht geschlossen werden. Darüber hinaus existiert kein anderer weit verbreiteter Netzwerkdienst, der diesen Port benutzt.

Die Verbindungsverwaltung der S7-1200 wird über die POE *TCON* gesteuert. Der Aufruf dieser POE ist in Listing 7.3 Zeile 3 dargestellt. Eine frei wählbare IP-Adresse und eine Portnummer werden als Argument an die POE übergeben (Zeile 9). Nach dem Aufruf wird das Betriebssystem der S7-1200 versuchen, die Verbindung herzustellen. Die Funktion ist asynchron und wird während der weiteren Programmausführung abgearbeitet. In einem weiteren Zyklus kann mit einem wiederholten Aufruf der Status abgefragt werden. Der Rückgabewert *DONE* (Zeile 5) gibt Auskunft, ob

```
1  IF "data".con_state = 10 THEN
2
3    "TCON_DB"(REQ:="data".action,
4              ID:=1,
5              DONE=>"data".con_done,
6              BUSY=>"data".con_busy,
7              ERROR=>"data".con_error,
8              STATUS=>"data".con_status,
9              CONNECT:="data".con_param);
10
11   IF "data".con_done = True THEN
12     // connection open
13     "data".con_state := 20;
14   ELSE
15     // connection not open
16     "data".con_timeout := "data".con_timeout + 1;
17     // connection timeout?
18     IF "data".con_timeout > 200 THEN
19       "data".con_state := 0;
20     END_IF;
21   END_IF;
22
23   GOTO CYCLE_END;
24 END_IF;
```

Abbildung 7.3: Implementierung der Zielerkennung in SCL (Verbindungsaufbau)

eine Verbindung aufgebaut werden konnte. Die Infektion wird fortgesetzt, wenn der Wert der Variable wahr ist. Ist die IP-Adresse oder der Port nicht erreichbar, signalisiert *TCON* keinen Timeout. Der Timeout muss manuell implementiert werden. In jedem durchlaufenden Zyklus wird ein Zähler inkrementiert. Erreicht der Zähler einen bestimmten Wert wird der Zustand gewechselt und die Programmausführung im nächsten Zyklus an anderer Stelle fortgesetzt.

Konnte nach 200 Zyklen keine Verbindung aufgebaut werden, wird das Programm in Listing 7.4 fortgesetzt. Obwohl keine Verbindung etabliert werden konnte, muss die POE *TDISCON* aufgerufen werden. Der Aufruf wird mit einem Fehler beendet, welcher signalisiert, dass die Verbindung nicht abgebaut werden konnte. Der Aufruf führt dazu, dass die Verbindungsverwaltung die Ressourcen für einen weiteren Verbindungsversuch zu einer anderen IP-Adresse frei gibt. In Zeile 13 wird die nächste IP-Adresse bestimmt. Das letzte Byte der Adresse wird inkrementiert. Die Zielerkennung

```
1  IF "data".con_state = 0 THEN
2
3    "TDISCON_DB"(REQ:="data".action,
4                 ID:=1,
5                 DONE=>"data".con_done,
6                 BUSY=>"data".con_busy,
7                 ERROR=>"data".con_error,
8                 STATUS=>"data".con_status);
9
10   IF "data".con_error = True OR
11     "data".con_done = True
12   THEN
13     "data".con_param.REM_STADDR[4] := \
14       ("data".con_param.REM_STADDR[4] + 1) MOD 255;
15     "data".con_timeout := 0;
16     "data".con_state := 10;
17   END_IF;
18
19   GOTO CYCLE_END;
20 END_IF;
```

Abbildung 7.4: Implementierung der Zielerkennung in SCL (Verbindungsabbau)

durchsucht ein /24 Subnetz nach einem offenen TCP Port 102. Weitere Optimierungen aus Kapitel 2.1 sind denkbar. Um den Wurm besser kontrollieren zu können und eine ungewollte Ausbreitung zu verhindern, wird auf diese Optimierungen verzichtet.

7.4 Übertragungsmechanismus

Mit dem Übertragungsmechanismus wird eine Kopie des Wurms auf das Zielsystem transferiert. Normalerweise findet der Transfer eines Benutzerprogramms zwischen dem TIA-Portal und der S7-1200 über das S7CommPlus Protokoll statt. Der SPS-Wurm realisiert den Übertragungsmechanismus, indem er dieses Protokoll selbst implementiert. S7CommPlus ist ein proprietäres Protokoll. Siemens stellt keinerlei Informationen über die Funktionsweise bereit. Parallel zu dieser Arbeit wurde ein Wireshark-Plugin von Thomas Wiens [38] entwickelt, welches in der Lage ist, das Protokoll zu analysieren. Für die Implementierung des Übertragungsmechanismus wird im Folgenden das Protokoll analysiert. Die Arbeit stützt sich dabei nicht auf die Erkenntnisse aus dem Wireshark-Plugin.

7.4.1 Protokollanalyse

S7CommPlus

S7CommPlus ist ein binäres Protokoll, welches über die Transportprotokolle TPKT [39] und ISO8073 [40] übertragen wird. Typischerweise werden diese beiden Protokolle über TCP/IP auf dem Port 102 transportiert. In Abbildung 7.5 ist der Protokollstack dargestellt.

Die wesentlichen Funktionen von S7CommPlus sind:

- Konfiguration
- Starten und Stoppen
- Lesen und Verändern von Prozessvariablen
- Programm Transfer (Up-/Download)
- Debugging
- Bereitstellen von Diagnoseinformationen
- Alarmmeldungen

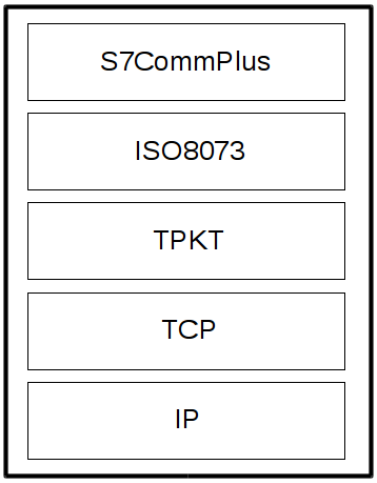

Abbildung 7.5: Protokollstack des S7CommPlus Protokolls

Aufbau einer Nachricht

Jede Nachricht besitzt einen gleichbleibenden Aufbau. In Abbildung 7.6 ist die erste Nachricht dargestellt. Sie wird von dem TIA-Portal an die S7-1200 gesendet und für den Verbindungsaufbau genutzt. Der generelle Aufbau wird an diesem Beispiel erläutert. Die ersten beiden Felder stellen das TPKT und ISO8073 Protokoll dar. Die Bedeutung wird in den oben genannten Quellen erläutert. Es folgt das Byte `0x72`, welches den Beginn der S7CommPlus Nachricht markiert. Eine Versionsnummer unterscheidet die verschiedenen Varianten des Protokolls. Das Längenfeld bestimmt die Anzahl der folgenden Bytes. Nicht in die Länge wird die Frame-Begrenzung einbezogen. Fehlt die Frame-Begrenzung am Ende ist das Paket fragmentiert und in der nächsten Nachricht wird ein weiterer Teil übertragen. Nach der Länge wird der Typ (Anfrage, Antwort, Fehler) der Nachricht angegeben. Es folgen zwei Byte, die konstant null sind. Der Subtyp gibt Auskunft über den weiteren Aufbau der Nachricht. Eine Sequenznummer nummeriert die versendeten Nachrichten. Es folgen mehrere Bytes deren genaue Bedeutung nicht ermittelt werden konnte. Weitere Informationen werden in Attributblöcken übertragen.

```
00000023  03 00 00 e1 02 f0 80 72  01 00 d2 31 00 00 04 ca  .......r ...1....
00000033  00 00 00 02 00 00 01 20  36 00 00 01 1d 00 04 00  ....... 6.......
00000043  00 00 00 00 a1 00 00 00  d3 82 1f 00 00 a3 81 69  ........ .......i
00000053  00 15 16 53 65 72 76 65  72 53 65 73 73 69 6f 6e  ...Serve rSession
00000063  5f 36 42 36 31 38 32 46  31 a3 82 21 00 15 2c 31  _6B6182F 1..!..,1
00000073  3a 3a 3a 36 2e 30 3a 3a  54 43 50 2f 49 50 20 2d  :::6.0:: TCP/IP -
00000083  3e 20 49 6e 74 65 6c 28  52 29 20 50 52 4f 2f 31  > Intel( R) PRO/1
00000093  30 30 30 20 4d 54 20 44  2e 2e 2e a3 82 28 00 15  000 MT D .....(..
000000A3  00 a3 82 29 00 15 00 a3  82 2a 00 15 11 4d 41 49  ...)....  .*...MAI
000000B3  4b 2d 50 43 5f 33 32 39  31 38 39 35 31 35 a3 82  K-PC_329 189515..
000000C3  2b 00 04 01 a3 82 2c 00  12 00 2d c6 c0 a3 82 2d  +.....,.  ..-....-
000000D3  00 15 00 a1 00 00 00 d3  81 7f 00 00 a3 81 69 00  ........ ......i.
000000E3  15 15 53 75 62 73 63 72  69 70 74 69 6f 6e 43 6f  ..Subscr iptionCo
000000F3  6e 74 61 69 6e 65 72 a2  a2 00 00 00 00 72 01 00  ntainer. .....r..
00000103  00
```

TPKT, ISO8073, Magic-Byte, Version, Länge, Typ, Reserviert, Sub-Typ, Sequenznummer, Attributblöcke, Frame-Begrenzung, Unbekannt

Abbildung 7.6: Aufbau einer S7CommPlus Nachricht

Attributblöcke

Innerhalb einer Nachricht werden die zu übertragenen Werte durch Attributblöcke strukturiert. Genau wie eine Nachricht selbst besitzen sie einen bestimmten Aufbau. In Abbildung 7.7 ist der erste Attributblock aus dem vorangegangen Beispiel abgebildet. Jeder Attributblock wird durch das Byte 0xA3 eingeleitet. Es folgt eine Attribut-ID, welche den zu übertragenden Wert eindeutig bestimmt. Der Datentyp des zu übertragenen Wertes wird angegeben. In diesem Beispiel handelt es sich um einen String. Der Datentyp String beginnt mit einer Angabe der Länge gefolgt von dem eigentlichen Wert.

```
                                                  a3 81 69  ........ .......i
00000053  00 15 16 53 65 72 76 65  72 53 65 73 73 69 6f 6e  ...Serve rSession
00000063  5f 36 42 36 31 38 32 46  31                       _6B6182F 1
```

Attributblock Start, Attribut-ID, Format?, Datentyp, Länge, Wert,

Abbildung 7.7: Aufbau eines Attributblocks

Zahlenkodierung in den Attributblöcken

Für das Verständnis der Attributblöcke ist die Kodierung von Zahlen innerhalb der Blöcke wichtig. Zahlen besitzen eine variable Länge. Das erste Bit jeder Zahl gibt an, ob ein weiteres Byte folgt. Abbildung 7.8 zeigt die Umrechnung der Attribut-ID und des Längenfelds aus dem vorangegangenen Beispiel. Werden Zahlen als Wert eines Attributblocks übertragen, findet diese Kodierung keine Anwendung.

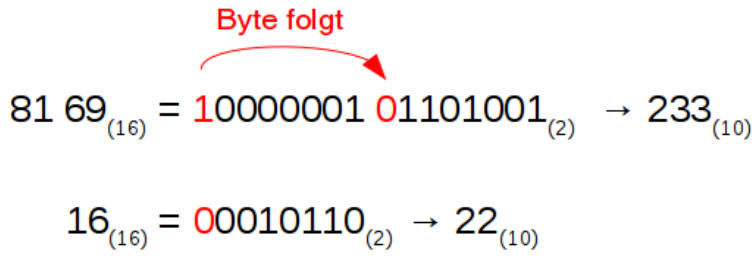

Abbildung 7.8: Codierung von Zahlen im S7CommPlus Protokoll

Anti-Replay-Mechanismus

Das S7CommPlus Protokoll ist vor Replay-Angriffen geschützt. Der Anti-Replay-Mechanismus funktioniert wie folgt. Die in der Abbildung 7.9 dargestellte Nachricht ist Teil des Verbindungsaufbaus und folgt der Nachricht aus dem vorangegangenen Beispiel. Sie wird als Antwort von der S7-1200 an das TIA-Portal gesendet. Das fünfundzwanzigste Byte nimmt bei jedem Verbindungsaufbau einen zufälligen Wert zwischen `0x06` und `0x7f` an. Es stellt eine Anti-Replay-Challenge dar.

Um den Verbindungsaufbau abzuschließen, muss das TIA-Portal diese Nachricht beantworten. In der Antwort müssen das vierundzwanzigste und neunundzwanzigste Byte in Abhängigkeit der Anti-Replay-Challenge berechnet werden. Folgende Formel wird für die Berechnung verwendet:

```
Anti-Replay-Byte = Challenge + 0x80
```

```
00000023  03 00 00 89 02 f0 80 72  01 00 7a 32 00 00 04 ca  .......r ..z2....
00000033  00 00 00 02 36 11 02 87  22 87 3d a1 00 00 01 20  ....6... ".=.... 
00000043  82 1f 00 00 a3 81 69 00  15 00 a3 82 32 00 17 00  ......i. ....2...
00000053  00 01 3a 82 3b 00 04 82  00 82 3c 00 04 81 40 82  ..:.;... ..<...@.
00000063  3d 00 04 84 80 c0 40 82  3e 00 04 84 80 c0 40 82  =.....@. >.....@.
00000073  3f 00 15 1b 31 3b 36 45  53 37 20 32 31 32 2d 31  ?...1;6E S7 212-1
00000083  42 45 33 31 2d 30 58 42  30 20 3b 56 33 2e 30 82  BE31-0XB 0 ;V3.0.
00000093  40 00 15 05 32 3b 35 34  34 82 41 00 03 00 03 00  @...2;54 4.A.....
000000A3  a2 00 00 00 00 72 01 00  00
```

Challange, Attributblock

Abbildung 7.9: Anti-Replay Mechanimus Nachricht 1

In allen weiteren Nachrichten, die vom TIA-Portal ausgehen und an die S7-1200 gerichtet sind, muss das vierundzwanzigste Byte dem berechnetem Wert entsprechen. Darüber hinaus muss der in grau abgebildete Attributblock während des Verbindungsaufbaus gespiegelt werden.

```
0000010B  03 00 00 8c 02 f0 80 72  02 00 7d 31 00 00 05 42  .......r ..}1...B
0000011B  00 00 00 03 00 00 03 a2  34 00 00 03 a2 01 01 82  ........ 4.......
0000012B  32 01 00 17 00 00 01 3a  82 3b 00 04 82 00 82 3c  2......: .;.....<
0000013B  00 04 81 40 82 3d 00 04  00 82 3e 00 04 84 80 c0  ...@.=.. ..>.....
0000014B  40 82 3f 00 15 00 82 40  00 15 1a 31 3b 36 45 53  @.?....@ ...1;6ES
0000015B  37 20 32 31 32 2d 31 42  45 33 31 2d 30 58 42 30  7 212-1B E31-0XB0
0000016B  3b 56 33 2e 30 82 41 00  03 00 00 00 00 00 00 04  ;V3.0.A. ........
0000017B  e8 89 69 00 12 00 00 00  00 89 6a 00 13 00 89 6b  ..i..... ..j....k
0000018B  00 04 00 00 00 00 00 00  72 02 00 00
```

Anti-Replay-Byte, Attributblock

Abbildung 7.10: Anti-Replay Mechanimus Nachricht 2

Programmtransfer Nachricht

Für den Programmtransfer existiert eine Protokollnachricht, welche in Abbildung 7.11 ausschnittsweise dargestellt ist. Pro Nachricht kann eine POE übertragen werden. Der POE-Typ unterscheidet zwischen den verschiedenen POE-Varianten. Die Blocknummer gibt den Speicherplatz an, auf dem die POE auf der S7-1200 abgelegt werden soll.

Nach dem Nachrichtenkopf folgen eine Reihe von Attributblöcken. Neben dem Byte-Code, der von der S7-1200 ausgeführt wird, werden Metainformationen gespeichert. Hierzu zählen der

```
00000901  03 00 04 00 02 f0 00 72  02 05 a9 31 00 00 04 ca .......r ...1....
00000911  00 00 00 1d 00 00 03 a2  34 00 00 00 03 00 04 00 ........ 4.......
00000921  00 00 00 00 a1 8a 32 00  01 94 57 20 00 a3 81 69 ......2. ..W ...i
00000931  00 15 04 4d 61 69 6e
          ...
```

POE-Typ, Blocknummer, Attributblock

Abbildung 7.11: Programmtransfer Nachricht

benötigte Programmspeicher, der Zeitpunkt der Erstellung, die Blocknummer, die verwendete Programmiersprache, der Quelltext oder Informationen über aktivierte Schutzfunktionen (siehe Kapitel 9). Das TIA-Portal kann alle Informationen auslesen, bearbeiten und das Benutzerprogramm verifizieren.

7.4.2 Sicherheitsbetrachtung des Programmtransfers

Bearbeiten/Entfernen von Attributblöcken

Attributblöcke können beliebig verändert werden. Es findet keine Integritätsprüfung statt. Ein Wurm-Autor kann daher Attributblöcke in seinem Interesse manipulieren. Wird z.B. der benötigte Programmspeicher auf Null gesetzt, kann der Wurm nicht mehr über eine höhere Speicherplatznutzung identifiziert werden. Eine Plausibilitätsüberprüfung der übertragenen Werte findet nicht statt.

Das TIA-Portal erlaubt eine Verifikation des SPS-Programms, um eine Veränderung feststellen zu können. Diese Verifizierung findet über den Attributblock statt, welcher den Zeitpunkt der Erstellung angibt. Ein Angreifer kann alle Informationen bis auf den Erstellungszeitpunkt verändern, ohne das die Manipulation auffällt.

Bestimmte Attributblöcke können ganz entfernt werden, ohne dass die Programmausführung auf der SPS beeinträchtigt wird. Ein Attributblock speichert z.B. den gesamten Quelltext. Ein Wurm-Autor kann diesen Quelltext entfernen, um die Analyse zu erschweren.

Redundante Attributblöcke

Neben dem ausgeführten Byte-Code wird der Quelltext im Speicher der S7-1200 abgelegt. Der Quelltext liegt in einem XML-Format vor. Abbildung 7.12 zeigt den Quelltext und Abbildung 7.13 die zweite Repräsentation als Byte-Code. Ein Angreifer kann die Attributblöcke unabhängig voneinander verändern. Die tatsächliche Funktion eines Programms kann verschleiert werden, indem ein Quellcode erzeugt wird, der nicht mit dem Byte-Code übereinstimmt. Findet eine Analyse des Programms mit dem TIA-Portal statt, wird nur der Quelltext ausgewertet.

```
...
<BC>
 <Fold UId="23">
  <NL UId="24"/>
  <BCL TE=" * This is a comment."/>
  <NL UId="21"/>
  <BCL TE=" "/>
  <BCE/>
 </Fold>
</BC>
<NL UId="42"/>
<NL UId="38"/>
<Statement TE="IF" UId="59" SI="IF">
...
```

Abbildung 7.12: Ausschnitt des auf der S7-1200 abgelegtem Quelltextes in XML

```
0000   02 4C 00 00 E0 02 4C 04  00 E0 02 4C 08 00 E0 02   .L....L....L....
0010   4C 0C 00 E0 02 4C 10 00  E0 02 4C 14 00 F8 18 58   L....L....L....X
0020   02 F8 18 58 06 18 40 01  F8 70 00 04 01 02 1A 40   ...X..@..p.....@
0030   05 6F 00 2C 7C 00 01 6C  01 68 00 68 01 14 40 01   .o.,|..l.h.h..@.
```

Abbildung 7.13: Byte-Code welcher von der S7-1200 ausgeführt wird

Redundante Informationen zwischen Nachrichtenkopf und Attributblöcken

Sowohl im Nachrichtenkopf der Programmtransfer-Nachricht als auch in einem Attributblock wird die POE Speicherplatznummer angegeben. Diese Datenredundanz kann ausgenutzt werden um eine POE vor dem TIA-Portal zu verstecken. Ein Angreifer wählt eine freie POE-Nummer

im Nachrichtenkopf aus und überträgt die POE auf die S7-1200. Diese wird als Bestandteil des normalen Benutzerprogramms ausgeführt. Als POE-Nummer innerhalb des Attributblocks wird eine bereits vergebene Nummer durch den Angreifer gewählt. Findet eine Programmverifikation durch das TIA-Portal statt, existieren aus Sicht des TIA-Portal zwei POE mit einer identischen POE-Nummer. Pro POE-Nummer kann jedoch nur eine POE gespeichert werden. Das TIA-Portal löst diesen Widerspruch auf, indem es nur einen Block anzeigt. Schadsoftware kann auf diese Weise versteckt werden. Dies gilt nicht für einen DB.

Fehler in der Verarbeitung von Attributblöcken

Attributblöcke können so gewählt werden, dass bei der Verarbeitung durch das TIA-Portal Fehler auftreten. Abbildung 7.14 zeigt einen Attributblock, der den Namen einer POE speichert. Wird die Länge des Strings auf Null reduziert (Abbildung 7.15), beendet das TIA-Portal die Ausführung, sobald es den Attributblock auswertet. Ein Angreifer kann so die Analyse des auf der SPS ausgeführten Programms erschweren. Abbildung 7.16 zeigt den Absturz des TIA-Portals

```
0000    A3 81 69 00 15 0E 4D 6F  74 6F 72 73 74 65 75 65    ..i...Motorsteue
0010    72 75 6E 67                                         rung
```

Attributblock-Start, Attribut-ID (Name), Format, Datentyp (String), Länge, Wert

Abbildung 7.14: Normaler Attributblock mit dem Namen einer POE

```
0000    A3 81 69 00 15 00                                   ..i...
```

Attributblock-Start, Attribut-ID (Name), Format, Datentyp (String), Länge

Abbildung 7.15: Attributblock mit einem Namen der Länge Null

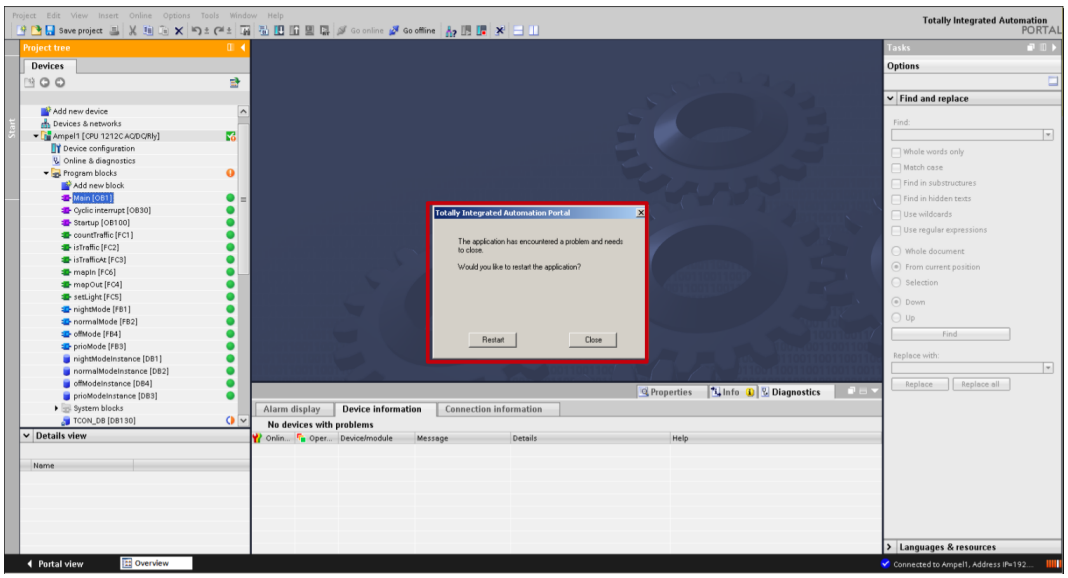

Abbildung 7.16: Verarbeitung von Attributblöcken beendet das TIA-Portal

7.4.3 Erforderliche Nachrichten ermitteln

Wenn das TIA-Portal eine Verbindung zu der S7-1200 aufbaut, um ein Programm zu transferieren, werden Nachrichten übertragen, die für den Programmtransfer nicht relevant sind. Irrelevante Nachrichten würden den Speicherplatzverbrauch des Wurms erhöhen und haben möglicherweise ungewollte Nebeneffekte. Um zu ermitteln, welche Nachrichten für den Programmtransfer erforderlich sind, wird die Kommunikation zwischen dem TIA-Portal und der S7-1200 aufgezeichnet. Die gesammelten Nachrichten werden ohne das TIA-Portal abgespielt und ein Programmtransfer auf diese Weise wiederholt. Sukzessive werden aus der Wiederholung immer mehr Nachrichten entfernt, bis nur noch relevante Nachrichten übrig bleiben.

Abbildung 7.17 zeigt den erforderlichen Nachrichtenaustausch für eine Infektion. Die Kommunikation beginnt mit dem Verbindungsaufbau des ISO8073-Protokolls. Danach wird die S7CommPlus-Verbindung aufgebaut. Um eine mehrmalige Infektion zu verhindern, wird eine Download-Anfrage bezüglich des Wurm-POE gestellt. Wird diese negativ beantwortet, kann die Infektion fortgesetzt werden. Eine Programmerweiterung ist nur möglich, wenn die S7-1200 gestoppt ist. Die erforderliche Nachricht wird daher übertragen bevor der eigentliche Programmtransfer beginnt. Mehrere DB müssen neben der Wurm-POE auf das Ziel-SPS übertragen werden. Zum Schluss wird die S7-1200 durch eine Nachricht gestartet.

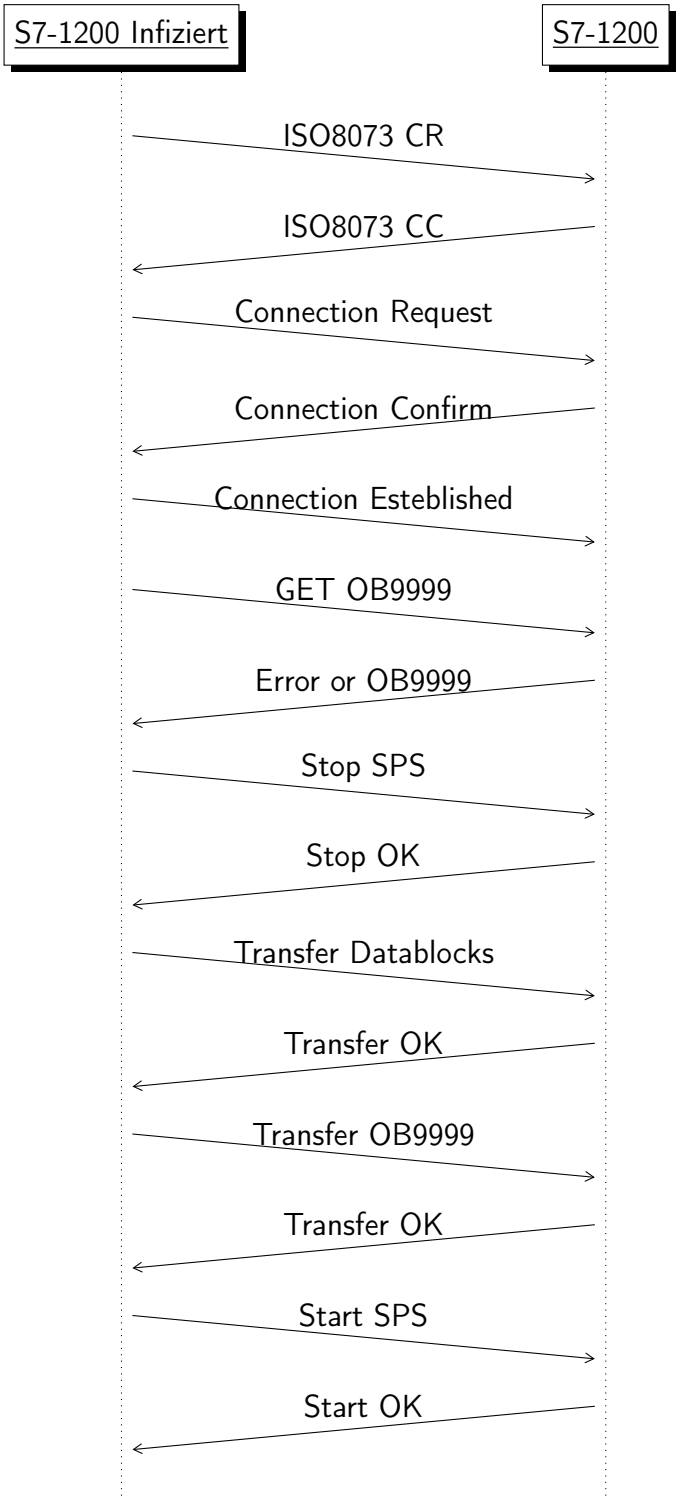

Abbildung 7.17: Nachrichtenaustausch während der Infektion

7.4.4 Implementierung

Mit der Erkenntnis aus der Protokollanalyse kann ein Programmtransfer aufgezeichnet, modifiziert und neu auf die S7-1200 übertragen werden. Alle relevanten Nachrichten sind bekannt. Der Übertragungsmechanismus wird implementiert, indem der vollständige Nachrichtenstrom unter Berücksichtigung des Anti-Replay-Mechanismus an ein Ziel übertragen wird. Ein DB wird für die Speicherung temporärer Variablen verwendet. Ein zweiter DB dient als Puffer für die aktuell zu sendende oder zu empfangende Nachricht. Ein dritter statischer DB wird alle zu sendenden Nachrichten enthalten. Aufgrund der Speichergröße kann er nicht in dem Arbeitsspeicher der S7-1200 vorgehalten werden, sondern wird auf den persistenten Speicher ausgelagert. Das Programm durchläuft die Zustands-Maschine und sendet die erforderlichen Nachrichten, nachdem sie abschnittsweise in den Arbeitsspeicher geladen wurden.

7.4.5 Manuelle Verschachtelung und initiale Infektion

Der dritte DB muss alle Nachrichten, die für den Programmtransfer erforderlich sind, enthalten. Der Block kann nicht mit dem TIA-Portal erstellt werden, weil der genaue Nachrichteninhalt erst bekannt ist, wenn das Programm kompiliert und auf die S7-1200 übertragen wurde. Der dritte Datenblock wird zwar durch das TIA-Portal erzeugt, die eigentlichen Daten müssen jedoch in einem manuellen Bearbeitungsschritt eingefügt werden. Dazu wird der Wurm von dem TIA-Portal auf die SPS übertragen. Der dritte DB ist zu diesem Zeitpunkt leer. Die übertragenen Nachrichten werden aufgezeichnet. Nachdem alle Nachrichten neu fragmentiert sind, um sie optimal versenden zu können, werden sie an einem freien Speicherplatz in dem dritten DB abgelegt. Damit das Wurm-Programm die einzelnen Nachrichten versenden kann, wird die genaue Nachrichtenlänge und der Offset innerhalb des DB für die einzelnen Nachrichten hinterlegt. Hierzu zählt auch der Nachrichtenkopf für den Transfer des dritten DB selbst. Mit einem eigens entwickeltem Programm werden alle Nachrichten, inklusive des modifiziertem dritten DB, auf eine S7-1200 übertragen. Dieser Vorgang stellt den initialen Infektionsvorgang dar. Der gesamte Vorgang wird in Abbildung 7.18 visualisiert.

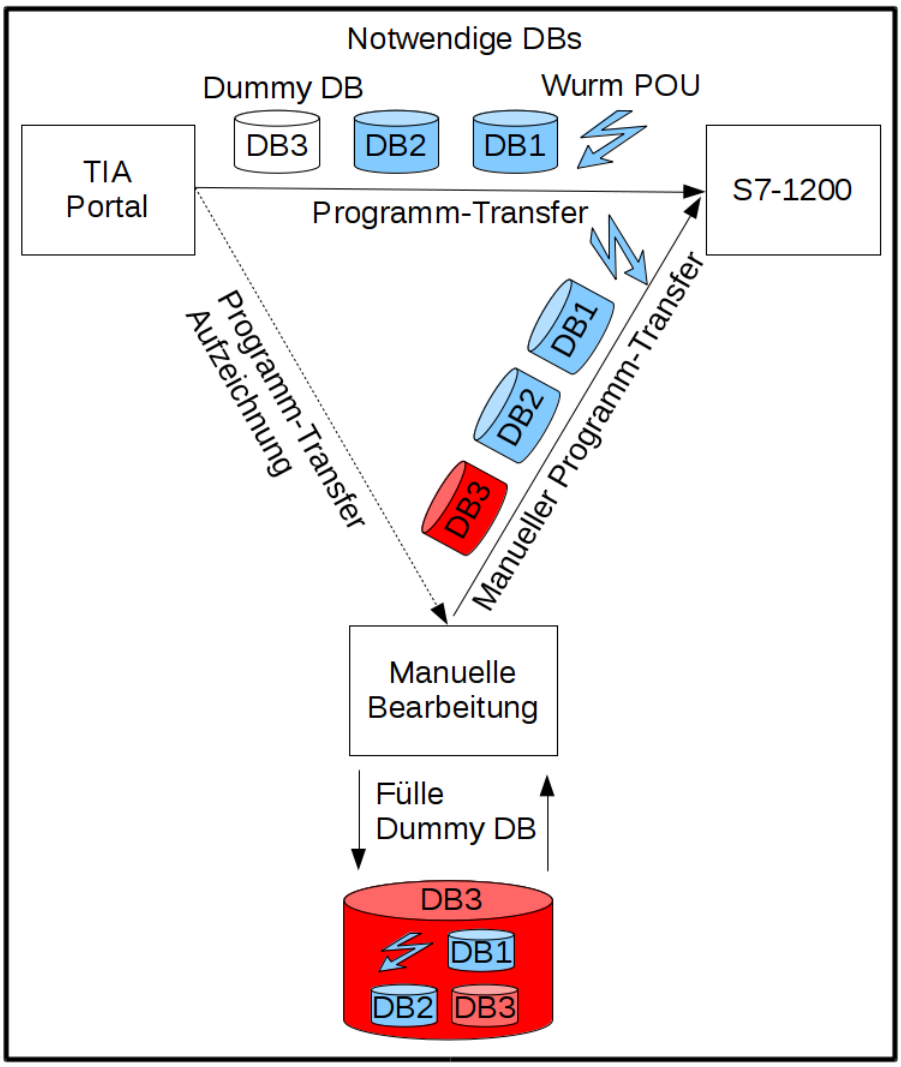

Abbildung 7.18: Manuelle Erzeugung des DBs für die Protokollnachrichten

7.5 Aktivierungsfunktion

Der Übertragungsmechanismus ergänzt das Benutzerprogramm der S7-1200. Es wird ein neuer OB und die notwendigen DB hinzugefügt. Das bereits auf der SPS vorhandene Programm bleibt unberührt. Wie in Kapitel 3.3.2 dargestellt, ist ein OB die Schnittstelle zwischen dem Betriebssystem und dem Benutzerprogramm. Die S7-1200 erkennt den neuen OB und führt diesen sequenziell nach dem bereits vorhandenen OB aus.

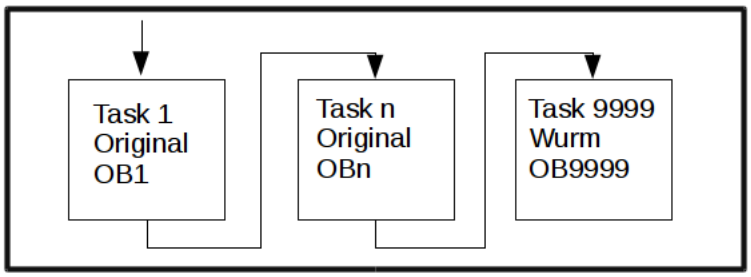

Abbildung 7.19: Der Wurm wird als regulärer Task ausgeführt

7.6 Schadfunktion

Um zu demonstrieren, welchen Schaden ein SPS-Wurm anrichten kann, wurden verschiedene Schadfunktionen implementiert.

7.6.1 Command & Control Server

Der SPS-Wurm baut nach der Infektion eine Verbindung zu einem Command & Control Server auf. Die Kommunikation wird über TCP realisiert. Die verschiedenen weiteren Schadfunktionen können von dem Betreiber des Command & Control Server angestoßen werden. Durch ein Byte wird die Schadfunktion ausgewählt. Weitere Bytes sind die Argumente für die Schadfunktionen.

7.6.2 Socks4-Proxy

Der Wurm implementiert einen Socks4-Proxy. Nachdem die SPS eine Verbindung zu dem Command & Control Server aufgebaut hat, kann dieser Verbindungen von beliebigen Clients akzeptieren. Über das Socks4-Protokoll wird der SPS signalisiert, zu welchem Ziel sie eine Verbindung aufbauen soll. Nachdem die Verbindung etabliert ist, kann der Proxy-Client beliebige Daten mit dem Ziel austauschen. Es sind beliebig viele aufeinander folgende Verbindungen zu unterschiedlichen Zielen möglich. Abbildung 7.20 zeigt die Implementierung.

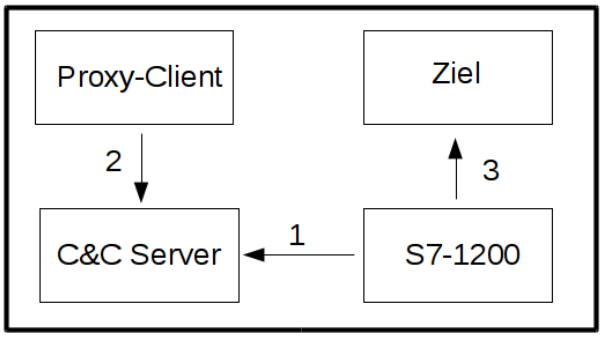

Abbildung 7.20: S7-1200 als Proxy-Server

7.6.3 Betriebsunterbrechung

Die Betriebsunterbrechung stoppt das Benutzerprogramm und den Wurm selbst. Die SPS kann durch das TIA-Portal neu gestartet werden. Diese Schadfunktion macht sich die Beschränkung der Zykluszeit zu nutze. Durch eine Endlosschleife wird die maximale Zykluszeit überschritten und die SPS wechselt in einen Fehlermodus.

7.6.4 Manipulation von Ausgängen

Um die Ausgänge einer SPS zu manipulieren muss das Prozessabbild verändert werden. Die POE *POKE* eignet sich für diesen Zweck. Die wesentlichen Argumente sind der Bereich, der geändert werden soll (z.B. Ausgänge, Eingänge, ...), ein Wert und ein Offset, an dem der Wert geschrieben werden soll. Die Argumente werden von dem Betreiber des Command & Control Server übersendet und erlauben auf diese Weise, das Prozessabbild beliebig zu verändern

8 Identifizierung, Persistenz, Ressourcenverbrauch & Optimierungspotenziale

8.1 Identifizierung

8.1.1 TIA-Portal

Das TIA-Portal kann das Benutzerprogramm auf der SPS verifizieren und veränderte oder neu hinzugefügte POE anzeigen. Abbildung 8.1 zeigt im rot umrandeten Bereich die vom Wurm genutzten POE. Wie bereits erläutert wurde (Abschnitt 7.4.2), kann die Änderungserkennung unterlaufen werden. Neue POE können mit Ausnahme der DB versteckt werden. Der von der SPS ausgeführte Byte-Code muss nicht dem XML-Source-Code entsprechen (Abschnitt 7.4.2). Durch einen Fehler in der Auswertung von Attributblöcken kann das TIA-Portal gezielt zum Absturz gebracht werden, bevor veränderte oder hinzugefügte Blöcke angezeigt werden (Abschnitt 7.4.2). Das TIA-Portal ist nicht geeignet, um zuverlässig Schadcode auf einer SPS analysieren oder diesen zu erkennen.

8.1.2 Stoppen der SPS

Die S7-1200 wird während der Infektion für 9,3 Sekunden gestoppt. Das originale Benutzerprogramm wird in dieser Zeit nicht ausgeführt. Diese Unterbrechung führt möglicherweise zu einem auffälligen Verhalten der gesteuerten Maschine und kann erkannt werden. Das Starten und Stoppen der SPS erzeugt außerdem einen Log-Eintrag im in der SPS (siehe Abbildung 8.2).

Abbildung 8.1: Das TIA-Portal zeigt die Wurm-POE an

2	12:11:17:276 am	01.01.1970	CPU info: Communication initiated request: WARM RESTART
3	12:11:17:276 am	01.01.1970	CPU info: New startup information
4	12:11:02:876 am	01.01.1970	CPU info: New startup information
5	12:11:01:761 am	01.01.1970	CPU info: New startup information
6	12:11:01:061 am	01.01.1970	CPU info: New startup information
7	12:11:00:961 am	01.01.1970	CPU info: Communication initiated request: STOP

Abbildung 8.2: Log-Eintrag erzeugt durch das Starten und Stoppen der S7-1200

8.1.3 Netzwerkverkehr

Der Wurm erzeugt durch die Zielerkennung viele ausgehende Verbindungsversuche zu unterschiedlichen IP-Adressen. Dieses Verhalten ist für eine SPS atypisch. Ebenfalls auffällig ist, dass ein Programmtransfer zwischen zwei SPS stattfindet und nicht zwischen einer SPS und dem TIA-Portal. Wird der Netzwerkverkehr auf diese Auffälligkeiten hin überwacht, kann der Wurm erkannt werden.

8.2 Persistenz

8.2.1 Restart/Reboot

Der SPS-Wurm ist im persistenten Speicher abgelegt. Er ist regulärer Bestandteil des SPS-Benutzerprogramms. Der Wurm bleibt auf der SPS erhalten, wenn sie neu gestartet wird oder von der Spannungsversorgung getrennt ist.

8.2.2 Factory Reset

Mit Hilfe des TIA-Portals kann ein Factory Reset ausgelöst werden. Alle Einstellungen werden auf Standardwerte zurückgesetzt und das Benutzerprogramm wird vollständig gelöscht. Der Wurm ist nach einem Factory-Reset von der SPS entfernt.

8.2.3 Programmtransfer

Der Wurm wird als OB9999 auf der SPS gespeichert. Wird diese POE durch das TIA-Portal überschrieben, ist der Wurm entfernt.

8.3 Ressourcenverbrauch

8.3.1 Zyklus-Zeit

Die Zyklus-Zeit stellt eine Obergrenze für die unterbrechungsfreie Ausführung des Benutzerprogramms dar. In der Standardeinstellung liegt dieser Wert bei 150ms. Damit kein Fehler ausgelöst wird, darf der SPS-Wurm diese Grenze in keinem Zyklus überschreiten. Durch das TIA-Portal wird der längste Zyklus während des Infektionsvorgangs gemessen. Dazu wird eine S7-1200 ohne Benutzerprogramm verwendet. Die Zykluszeit beträgt in diesem Zustand 0ms. Die untersuchte S7-1200 scannt die Teststellung und infiziert eine weitere S7-1200. Die gemessene maximale Zykluszeit

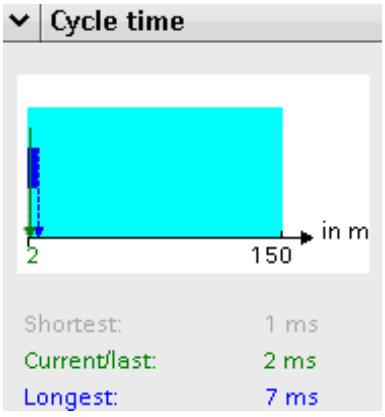

Abbildung 8.3: Zykluszeit während der Ausführung des Wurms gemessen mit dem TIA-Portal

beträgt 7ms, was 4,7% der maximal verfügbaren Zeit entspricht. Dieser geringe Wert entspricht dem erwartetem Ergebnis. Der Wurm verwendet hauptsächlich asynchrone Kommunikationsfunktionen.

8.3.2 Speicher

Der Wurm benötigt 38,5KB Arbeitsspeicher. Davon entfallen 9,0KB (23,3%) auf die Schadfunktionen. Neben dem Arbeitsspeicher werden 216,6KB persistenter Speicher benötigt. Die Tabelle 8.1 zeigt die Größe des Arbeitsspeichers nach S7-1200 Modell sowie den prozentualen Anteil des Wurms. Außerdem ist die prozentuale Auslastung des persistenten Speichers angegeben.

Modell	Arbeitsspeicher	Persistenter Speicher
S7-1211	50KB (77%)	1MB (21%)
S7-1212	75KB (51%)	1MB (21%)
S7-1214	100KB (38%)	4MB (5%)
S7-1215	125KB (30%)	4MB (5%)
S7-1217	150KB (25%)	4MB (5%)

Tabelle 8.1: Speicherverbrauch des SPS-Wurms nach S7-1200 Modell

8.4 Optimierungspotenziale

8.4.1 Unterbrechungsfreie Infizierung

Die Infizierung einer SPS mit dem Wurm führt zu einer 9,3 Sekunden langen Unterbrechung der Ausführung des Benutzerprogramms. Durch eine Optimierung wird die Unterbrechung unnötig. Die SPS bietet die Möglichkeit, ein Benutzerprogramm zur Laufzeit zu verändern. Dies ist jedoch nur mit bereits existierenden OBs möglich. Der Wurm kann dazu durch einen FB implementiert werden. Ein FB und die notwendigen DB können ohne Unterbrechung auf die S7-1200 übertragen werden. Damit der Wurm ausgeführt wird, muss ein existierender OB durch den Wurm um einen Aufruf des neuen FB erweitert werden. Dies ist zur Programmlaufzeit möglich. Dazu muss der Wurm

einen existierenden OB von einer SPS herunterladen, verändern und zurück zur SPS schicken (vlg. [36]).

8.4.2 Reduzierung des benötigen Speichers

Der benötigte Speicher kann reduziert werden. Die umfangreiche und allgemein gehaltene Schadfunktion kann im Falle eines gezielten Angriffes reduziert werden. Es müssen nur die für den Angriff erforderlichen Anweisungen enthalten sein. Zum Senden und Empfangen wird ein Pufferspeicher im Arbeitsspeicher vorgehalten. Dieser ist in diesem Proof-of-Conecpt großzügig berechnet und kann verkleinert werden. Auf die Schadfunktion entfallen 9,0KB, der Pufferspeicher kann von 15KB auf 1,5KB reduziert werden. Damit können etwa 22KB (58,5%) Arbeitsspeicher eingespart werden.

9 Schutzmaßnahmen der S7-1200

Die S7-1200 bietet drei verschiedene Schutzmaßnahmen. Dieser Abschnitt analysiert die technische Umsetzung der Schutzmaßnahmen und beurteilt die Auswirkungen auf den SPS-Wurm. Die Analyse bezieht sich auf das TIA-Portal V11 SP2 Update 5 und die S7-1200 Firmware 3.0.2.

9.1 Knowhow-Schutz

Der Knowhow-Schutz kann das Benutzerprogramm vor externem Zugriff schützen. Das Siemens Handbuch beschreibt diese Funktion wie folgt:

> Mit dem Knowhow-Schutz können Sie einen oder mehrere Codebausteine (OB, FB, FC oder DB) in Ihrem Programm vor unbefugtem Zugriff schützen. Sie können ein Passwort eingeben, um den Zugriff auf einen Codebaustein einzuschränken. Der Passwortschutz verhindert das unbefugte Lesen oder Ändern des Codebausteins. [33]

Jede POE wird mit einem eigenen Passwort versehen. Soll eine Knowhow geschützte POE bearbeitet werden, wird der Benutzer durch die Eingabe des Passwortes aufgefordert, diesen zu entsperren. Eine Löschung ist ohne Eingabe des Passwortes möglich.

9.1.1 Implementierung

Für den Knowhow-Schutz existiert ein Attributblock. Dieser wird bei einem Programm Up- oder Download zwischen dem TIA-Portal und der S7-1200 übertragen. Der Attributblock ist in Abbildung 9.3 dargestellt. Das Flag-Feld gibt an, ob der Knowhow-Schutz aktiviert ist. Der Passwort-Hash H wird nach folgender Vorschrift aus dem Klartext Passwort P gebildet:

```
H = sha-1(encode_utf-16le(P))
```

```
0000   A3 93 5C 00 17 00 00 0D 77 9A 78 00 0B 00 01 9A   .......w.x.....
0010   79 10 02 14 E8 F9 7F BA 91 04 D1 EA 50 47 94 8E   y...........PG..
0020   6D FB 67 FA CD 9F 5B 73 00                        m.g...[s.
```

Attributblock Start, Datentyp, Container-ID, Attribut-ID (Container Knowhow), Datentyp, Attribut-ID (Flag), Datentyp, Flag, Attribut-ID (Passwort-Hash), Datentyp, Länge, Passwort-Hash,

Abbildung 9.1: Aufbau des Attribut-Blocks: Knowhow-Schutz

Das TIA-Portal wertet diesen Attributblock aus. Ist das Flag gesetzt, verhindert es die Bearbeitung des Programm-Blockes. Gibt der Benutzer ein Passwort ein, wird die Korrektheit anhand des Passwort-Hashes überprüft.

Der in XML vorliegende Source-Code wird zusätzlich mittels AES128-CBC verschlüsselt. Hierdurch wird ein Einblick in den Source-Code verhindert. Die Abbildung 9.2 zeigt, wie der verschlüsselte Quelltext auf der SPS gespeichert wird.

```
<NetworkContainer>
 <Network Lang="SCL" ProgrammingContext="Plain" Mnemonic="International" RefID="1">
  <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element"
        xmlns="http://www.w3.org/2001/04/xmlenc#">
   <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc" />
    <CipherData>
     <CipherValue>
      AQIDAQIDAQIDAQIDAQIDAS/acA/s+L3CoQYaeAQMOqbwSzs53tzErj5UX8g ...
     </CipherValue>
    </CipherData>
   </EncryptedData>
   ...
```

Abbildung 9.2: Verschlüsseltes Benutzerprogramm im Source-Code

9.1.2 Schwachstellen

Keine Integritätsprüfung

Entgegen der Behauptung aus dem Siemens Handbuch kann ein Block trotz aktiviertem Knowhow-Schutz aufgrund einer fehlenden Integritätsprüfung beliebig verändert werden. Das TIA-Portal

implementiert den Manipulationsschutz nur clientseitig. Insbesondere kann der in dem Knowhow-Attributblock enthaltene Passwort-Hash bzw. das Flag verändert werden. Auf diese Weise kann der Knowhow Schutz abgeschaltet werden.

Byte-Code im Klartext gespeichert

Der Quelltext wird AES verschlüsselt auf der SPS gespeichert. Dies gilt jedoch nicht für den Byte-Code, welcher von der SPS ausgeführt wird. Mit Hilfe eines Disassemblers kann die Funktionsweise des Programms auch ohne den Quelltext nachvollzogen werden. Aufgrund der fehlenden Integritätsprüfung kann der Byte-Code beliebig verändert werden.

Download im Klartext

Durch einen Fehler in der Implementierung des TIA-Portals wird der Quelltext eines Knowhow geschützten Blockes unter bestimmten Umständen im Klartext auf der SPS gespeichert. Öffnet ein Benutzer einen Knowhow geschützten Block, fragt das TIA-Portal das Passwort ab. Wurde das Passwort korrekt eingegeben, wird der Quelltext in einem Editor angezeigt. Der Benutzer kann das Programm beliebig verändern. In einem weiteren Schritt kann der Benutzer das veränderte Programm auf die S7-1200 übertragen. Ist das Editor-Fenster bei diesem Vorgang noch geöffnet, wird der Quelltext trotz aktiviertem Knowhow-Schutzes unverschlüsselt auf der S7-1200 gespeichert.

AES-Schlüssel berechenbar

Für eine AES128 Verschlüsselung wird ein 128 Bit langer Schlüssel benötigt. Dieser Schlüssel wird von dem Passwort-Hash aus dem Knowhow-Attributbock abgeleitet. Der AES-Schüssel K wird nach folgender Vorschrift aus dem Passwort-Hash H berechnet:

```
K = truncate128Bit(H) XOR M
```

wobei für M gilt:

```
M = 0x28,0x6f,0x76,0x5c,0x6e,0x3b,0x1e,0x4c,
    0xd0,0x8e,0x42,0x31,0x43,0x7b,0x8e,0xbf
```

Weil ein Programm-Block mit aktiviertem Knowhow-Schutz aus der SPS ausgelesen werden kann, ist der Passwort-Hash bekannt. Der AES Schlüssel kann daher ohne Wissen des Passwortes berechnet werden. Zusammen mit der fehlenden Integritätsprüfung kann der Knowhow-Schutz vollständig entfernt werden.

9.1.3 Beurteilung der Schutzwirkung

Die Schutzfunktion bietet keinen Schutz gegen den SPS-Wurm da,

- der Schutz nur auf einzelne POE wirkt und der SPS-Wurm nur Blöcke hinzufügt,
- ein geschützter Block gelöscht werden kann und durch einen beliebigen (z.B. den Wurm-Block) ersetzt werden kann,
- der von der SPS ausgeführte Byte-Code nicht geschützt ist und von dem Wurm verändert werden kann und
- der gesamte Schutz ohne Wissen des Passwortes entfernt werden kann.

9.2 Kopierschutz

Der Kopierschutz schützt das Benutzer-Programm vor ungewollter Vervielfältigung. Das Siemens Handbuch beschreibt diese Schutzfunktion wie folgt:

> Eine weitere Sicherheitsfunktion ermöglicht Ihnen, das Programm oder die Codebausteine mit einer bestimmten Memory Card oder CPU zu verknüpfen. Diese Funktion ist vor allem zum Schutz geistigen Eigentums nützlich. Wird ein Programm oder ein Baustein mit einem bestimmten Gerät verknüpft, so ist die Verwendung dieses Programms oder dieses Bausteins nur in Verbindung mit einer bestimmten Memory Card oder CPU möglich. [33]

Der Kopierschutz wird für jede POE einzeln aktiviert. Die Seriennummer einer bestimmten SPS wird von dem Benutzer hinterlegt. Findet eine Übertragung auf eine SPS statt, überprüft das TIA-Portal, ob die hinterlegte Seriennummer mit der Ziel-SPS übereinstimmt. Ist das nicht der Fall, wird die Übertragung abgebrochen.

9.2.1 Implementierung

Für den Kopierschutz existiert ein Attributblock, welcher in Abbildung 9.3 dargestellt ist. Der Attributblock besteht im Wesentlichen aus einer Liste mit Seriennummern. In dem abgebildeten Beispiel ist eine Seriennummer eingetragen.

```
00000F83                                        a3 98 4f 00 ...r1... .3/...O.
00000F93 17 00 00 0d a0 9b 21 00  08 01 9b 22 00 15 10 53 ......!. ..."...S
00000FA3 5a 56 43 30 59 45 42 30  30 36 31 39 35 20 20 9b ZVC0YEB0 06195  .
00000FB3 23 00 08 00 00
```

Attributblock Start, Attribut-ID (Container), Datentyp, Container ID (Knowhow), Attribut-ID (Kopierschutz aktiv?), Datentyp, Flag, Attribut-ID (Seriennummer), Datentyp, Länge, Seriennummer, Attribut-ID (Unbekannt), Datentyp, Wert

Abbildung 9.3: Aufbau des Attribut-Blocks: Knowhow-Schutz

9.2.2 Schwachstellen

Keine Integritätsprüfung

Diese Schutzfunktion stellt die Integrität der POE nicht sicher. Die Liste der Seriennummern kann beliebig verändert bzw. entfernt werden. Weiterhin gleicht die S7-1200 diese Liste nicht mit ihrer eigenen Seriennummer ab, sondern akzeptiert eine beliebige Liste. Die Schutzfunktion ist aus diesem Grund lediglich clientseitig implementiert.

9.2.3 Beurteilung der Schutzwirkung

Die Schutzfunktion bietet keinen Schutz gegen den SPS-Wurm da:

- der Schutz nur auf einzelne Blöcke wirkt und der SPS-Wurm nur Blöcke hinzufügt und
- der Schutz nur clientseitig implementiert ist.

9.3 Zugriffsschutz

Der Zugriffsschutz schützt nicht das Benutzerprogramm, sondern erlaubt es, bestimmte Funktionen des S7CommPlus Protokolls erst nach einer Authentifizierung zu nutzen. Das Siemens Handbuch beschreibt diese Schutzfunktion wie folgt:

> Die CPU bietet drei Sicherheitsstufen, um den Zugang zu bestimmten Funktionen einzuschränken. Mit dem Einrichten der Schutzstufe und des Passworts für eine CPU schränken Sie die Funktionen und Speicherbereiche ein, die ohne Eingabe eines Passworts zugänglich sind. [33]

Je nach konfigurierter Schutzstufe sind bestimmte Funktionen durch eine Passwort-Authentifizierung geschützt. Die Tabelle 9.1 zeigt die verschiedenen Funkionen und die Einschränkungen durch den Zugriffsschutz.

Funktion	Kein Schutz	Schreib Schutz	Lese/Schreib Schutz
Start/Stop CPU	ja	nein	nein
Programm schreiben	ja	nein	nein
Programm auslesen	ja	ja	nein
Memory/Output verändern	ja	ja	ja
Identifiktion auslesen	ja	ja	ja
IP-Adresse setzen	ja	ja	ja
Datum setzen	ja	nein	nein
Factory-Reset	ja	nein	nein

Tabelle 9.1: Einschränkungen der S7CommPlus Funktionen mit aktiviertem Zugriffsschutz

9.3.1 Implementierung

Bei aktiviertem Zugriffsschutz muss sich ein Client gegenüber der SPS authentifizieren, bevor er eine eingeschränkte Funktion nutzen möchte. Der Client sendet eine Authentifizierungsanfrage an

die SPS, welche diese mit einem zufälligen 32 Bit Wert beantwortet. Der Client berechnet nun nach folgender Vorschrift eine Antwort und sendet sie an die SPS:

```
ANSWER = HMAC(PASSWORD, CHALLENGE)
```

Die HMAC-Funktion benutzt als Hash-Funktion SHA-1. Die SPS berechnet dieselbe Funktion. Stimmt die Antwort des Clients mit dem berechnetem Wert überein, ist die Authentifizierung erfolgreich [41].

Im Falle einer fehlgeschlagenen Authentifizierung kann ein weiterer Versuch erst nach fünf Sekunden vorgenommen werden.

9.3.2 Schwachstellen

Die S7-1200 besitzt einen Webserver. Der Zugriff auf den Webserver wird mit demselben Passwort wie das des Zugriffsschutzes gesichert. Authentifizierungsversuche können über den Webserver ohne zeitliche Verzögerung ausgeführt werden. Bei aktiviertem Webserver ist der Brute-Force-Schutz zwecklos. Außerdem kann das Passwort im Klartext abgefangen werden. Der Webserver ist in der Standardeinstellung nicht aktiviert.

9.3.3 Beurteilung der Schutzwirkung

Der Zugriffsschutz kann die Verbreitung des SPS-Wurms verhindern. Die Schutzstufe *Schreib-Schutz* verhindert, dass neue Programme auf die SPS geladen werden können. Das eingesetzte Challenge-Response-Verfahren wird als sicher angesehen. Ist dem Wurm-Autor das Passwort nicht bekannt, kann der Wurm diesen Schutz nicht umgehen. In der Standardeinstellung ist der Zugriffsschutz nicht aktiviert.

10 Mögliche Gegenmaßnahmen durch den Hersteller

10.1 Schutzfunktion in der Grundeinstellung

Der Zugriffsschutz bietet die Möglichkeit das Benutzerprogramm der S7-1200 wirksam vor Veränderungen zu schützen. In der Grundeinstellung wird dieser Schutz nicht aktiviert. Die SPS könnte bereits mit einem individuellen Passwort ausgeliefert werden oder das TIA-Portal erzwingt das Setzen eines Passwortes bei der initialen Programmierung der SPS.

10.2 Keine freie Kommunikation auf Port 102

Der Wurm implementiert das Protokoll zum Transferieren des Benutzerprogramms. Hierfür muss eine Verbindung zu dem TCP-Port 102 aufgebaut werden. Wenn das SPS-Betriebssystem die Verbindung verhindert, ist die Ausbreitung des Wurms in seiner jetzigen Form unmöglich.

10.3 Prüfsumme/Integritätssicherung

Zusätzliche Sicherheitsfunktionen können die Verbreitung von SPS-Würmern erschweren. Trotzdem ist es möglich, dass Schwachstellen in diesen Funktionen durch einen Wurm ausgenutzt werden können und eine Manipulation des Benutzerprogramms stattfindet. Diese Arbeit konnte zeigen, dass diese Manipulationen durch den Wurm nicht durch das TIA-Portal erkannt werden können. Die Einführung einer kryptographisch sicheren Prüfsumme, welche über das gesamte Programm und die Konfiguration der SPS gebildet wird, erlaubt eine Erkennung dieser Manipulation zu erkennen.

11 Weitere Forschungsfragen

11.1 Analyse weiterer Modelle und Hersteller

Eine genaue Analyse weiterer SPS-Modelle verschiedener Hersteller scheint lohnenswert. Die Marktanalyse legt nahe, dass weitere SPS für einen Wurm anfällig sind. Zwölf weitere SPS verfügen über frei programmierbare TCP oder UDP Funktionen (siehe Abschnitt 6.2). Durch die weitere Analyse könnte nachgewiesen werden, dass es sich nicht um ein Siemens spezifisches Problem handelt. Möglicherweise begünstigen die von der Norm IEC 63113 vorgeschriebenen Funktionen (Programm Transfer, Kommunikation) die Verbreitung von Computerwürmern.

11.2 Feldbusse

Ethernet stellt eine vergleichbar junge Technologie zur Vernetzung von SPS dar. Vor der Einführung von Ethernet erfolgte die Kommunikation über Feldbusse. Die meisten der zur Zeit betriebenen Industrieanlagen nutzen einen Feldbus [42]. Die Systeme sind technologisch nicht direkt mit dem IT-Netzwerk verbunden und gelten daher als sicher. Genau wie über die Ethernet-Schnittstelle kann der Programmtransfer über einen Feldbus erfolgen (z.B. Profibus von Siemens [43]). Ob auch diese Systeme durch einen SPS-Wurm angegriffen werden können, sollte erforscht werden.

11.3 Verbesserung des Übertragungsmechanismus

Der Übertragungsmechanismus kann verbessert werden. In Abschnitt 2.2 wurde die Möglichkeit eines eingebetteten Übertragungsmechanismus vorgestellt. Übertragen auf diese Situation muss die S7-1200 das TIA-Portal bei einem Zugriff infizieren. Das infizierte TIA-Portal verteilt den Wurm

auf weitere S7-1200 Geräte. Dazu ist eine Schwachstelle in dem TIA-Portal notwendig. Es konnten bereits Fehler in der Bearbeitung von Attributblöcken nachgewiesen werden (Abschnitt 7.4.2). Der persistente Speicher der S7-1200 bietet ausreichend Platz, um auch komplexe Schadsoftware für das TIA-Portal zu transportieren. Das eingebettete Verfahren besitzt mehrere Vorteile. Der Zugriffsschutz für die S7-1200 wird unterwandert. Findet ein legitimer Programmtransfer statt, wird das Kennwort von dem Ingenieur eingegeben. Der Wurm erweitert den Programmtransfer um seine eigene POE. Weiterhin wird eine Identifikation erschwert. Der legitime Programmtransfer muss detailliert nach Schadsoftware untersucht werden. Netzwerkverkehr zum Auffinden von Zielen kann vollständig entfallen. Wird zudem ein mobiles Gerät für die Programmierung der SPS genutzt, kann sich der Wurm möglicherweise über die Grenzen von physikalisch getrennten Netzwerken ausbreiten. Die Möglichkeit, diesen Übertragungsmechanismus zu implementieren, sollte erforscht werden.

12 Fazit

Diese Arbeit konnte am Beispiel der Siemens SIMATIC S7-1200 zeigen, dass ein Wurm entwickelt werden kann, der ausschließlich speicherprogrammierbare Steuerungen für seine Verbreitung nutzt. Der Wurm ist regulärer Bestandteil des SPS-Programms und bleibt aktiv, bis er durch ein anderes Benutzerprogramm überschrieben wird oder ein Factory Reset das gesamte SPS-Programm löscht. Der Ressourcenverbrauch ist gering genug, um das originale Programm der SPS nicht negativ zu beeinflussen. Wird der Wurm optimiert und die unterbrechungsfreie Infizierung implementiert, ist die Schadsoftware von außen nicht erkennbar. Mit Hilfe des TIA-Portals ist eine Analyse des Wurms nicht möglich, denn der ebenfalls abgelegte Quelltext kann manipuliert oder entfernt werden und der ausgeführte Byte-Code wird von dem TIA-Portal nicht berücksichtigt. Durch Manipulation einiger Attributblöcke kann das TIA-Portal zusätzlich zum Absturz gebracht werden. Über das TIA-Portal hinaus sind keine Programme bekannt, mit denen eine Überprüfung des Benutzerprogramms einer S7-1200 durchgeführt werden kann. Die Schadfunktionen demonstrieren, dass die Möglichkeit besteht, physikalische Ein- und Ausgänge zu manipulieren. Durch das Aurora-Experiment oder Stuxnet ist bekannt, dass eine solche Manipulation physischen Schaden in den gesteuerten Maschinen verursachen kann.

Der Zugriffsschutz verhindert die Ausbreitung des SPS-Wurms, indem er eine Programmveränderung erst nach einer Authentifizierung erlaubt. Der Knowhow-Schutz und der Kopierschutz sind darauf ausgerichtet, das bestehende Programm zu schützen und können eine Infektion durch den Wurm-Code nicht verhindern. Wenn die gefundenen Schwachstellen in dem Knowhow-Schutz behoben werden, eignet sich dieser sogar dazu, den Schadcode effektiv vor einer Analyse zu schützen. Der Hersteller sollte den Zugriffsschutz bereits vor der Auslieferung aktivieren. Es sollte außerdem die Möglichkeit geschaffen werden, das Programm einer SPS auf Veränderung überprüfen zu können. Dazu eignet sich eine kryptographische Prüfsumme.

Die Marktanalyse legt nahe, dass SPS-Würmer nicht nur mit der S7-1200 implementiert werden können. 54 % der untersuchten SPS verfügen über frei programmierbare TCP- oder UDP-

Funktionen. Ob ein Wurm für diese Modelle realisiert werden kann, muss im Einzelfall überprüft werden.

Die Norm IEC 63113 begünstigt die Möglichkeit eines SPS-Wurms. Durch das Kommunikationsmodul muss der Programmtransfer möglich sein. Die Folgen für die IT-Sicherheit werden jedoch nicht berücksichtigt. In Zukunft sollte die Norm um einen Abschnitt für IT-Sicherheitsprobleme erweitert werden und geeignete Gegenmaßnahmen vorgeschlagen werden.

Für die Implementierung des Wurms wurde keine Schwachstelle ausgenutzt. Die S7-1200 ist in ihrem Auslieferungszustand anfällig für SPS-Würmer. Die IT-Sicherheit dieser Geräte muss verbessert werden, damit die von der Wirtschaft und Politik gewünschte Industrie 4.0 realisiert werden kann.

Literaturverzeichnis

[1] Deutsche Messe AG. Industrie 4.0 in der Praxis. Abgerufen von http://www.hannovermesse.de/de/messe/leitmessen/digital-factory/ am 25.02.2016, 2016.

[2] Presse und Informationsamt der Bundesregierung. Merkel: Standards für 'Industrie 4.0' entwickeln. abgerufen von http://www.bundesregierung.de/Webs/Breg/DE/Mediathek/Einstieg/mediathek_einstieg_podcasts_node.html?id=1323442 am 03.02.2015, 2015.

[3] Eric Chien Nicolas Falliere, Liam O Murchu. W32.Stuxnet Dossier. Abgerufen von https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf am 06.02.2016, 2011.

[4] J. Reynolds. The Helminthiasis of the Internet. Abgerufen von https://tools.ietf.org/html/rfc1135 am 05.02.2016, 1989.

[5] pressetext.deutschland. Malware-Jubiläum: 20 Jahre Internet-Würmer. Abgerufen von http://www.pressetext.com/news/20081101001 am 06.02.2016, 2008.

[6] Nicholas Weaver u.a. A Taxonomy of Computer Worms. Abgerufen von https://www1.icsi.berkeley.edu/~nweaver/papers/2003-taxonomy.pdf am 05.02.2016, 2003.

[7] k claffy David Moore, Colleen Shannon. Code-Red: a case study on the spread and victims of an Internet worm. Abgerufen von https://www.eecis.udel.edu/~mills/teaching/eleg867b/dos/p273-moore.pdf am 06.02.2016, 2002.

[8] Edward Ray. MS-SQL Slammer. Abgerufen von https://www.sans.org/security-resources/malwarefaq/ms-sql-exploit.php am 06.02.2016, 2003.

[9] Internet Software Consortium. ISC Domain Survey: Number of Internet Hosts. Abgerufen von http://www.isc.org/ds/host-count-history.html über web.archive.org am 06.02.2016, 2008.

[10] Eugene H. Spafford. The Internet Worm Program: An Analysis. Abgerufen von `http://spaf.cerias.purdue.edu/tech-reps/823.pdf` am 06.02.2016, 1988.

[11] Matt Goldencrown. MyDoom is Your Doom: An Analysis of the MyDoom Virus. Abgerufen von `https://www.giac.org/paper/gcih/568/mydoom-dom-anlysis-mydoom-virus/106069` am 06.02.2016, 2004.

[12] Jason V. Miller u.a. Microsoft DCOM RPC Worm Alert. Abgerufen von `http://www.cs.put.poznan.pl/mmilostan/od/030811-Alert-DCOMworm.pdf` am 06.02.2016, 2003.

[13] Nicholas Weaver Stuart Staniford, Vern Paxson. How to 0wn the Internet in Your Spare Time. Abgerufen von `https://www.usenix.org/legacy/event/sec02/full_papers/staniford/staniford.pdf` am 06.02.2016, 2002.

[14] Symantec. W32.Sasser.Worm. Abgerufen von `https://www.symantec.com/security_response/writeup.jsp?docid=2004-050116-1831-99&tabid=2` am 06.02.2016, 2014.

[15] Bruce Schneier. The Storm Worm. Abgerufen von `https://www.schneier.com/blog/archives/2007/10/the_storm_worm.html` am 06.02.2016, 2007.

[16] Kaspersky. Kaspersky Lab and ITU Research Reveals New Advanced Cyber Threat. Abgerufen von `http://www.kaspersky.com/about/news/virus/2012/Kaspersky_Lab_and_ITU_Research_Reveals_New_Advanced_Cyber_Threat` am 06.02.2016, 2012.

[17] International Electrotechnical Commission. Programmable controllers - part 1: General information. *IEC Std. 61131*, 2003.

[18] International Electrotechnical Commission. Programmable controllers - part 2: Equipment requirements and tests. *IEC Std. 61131*, 2007.

[19] International Electrotechnical Commission. Programmable controllers - part 3: Programming language. *IEC Std. 61131*, 2013.

[20] International Electrotechnical Commission. Programmable controllers - part 1: General information). *IEC Std. 61131*, 2003.

[21] International Electrotechnical Commission. Programmable controllers - part 5: Communications. *IEC Std. 61131*, 2000.

[22] International Electrotechnical Commission. Programmable controllers - part 6: Functional safety. *IEC Std. 61131*, 2012.

[23] International Electrotechnical Commission. Programmable controllers - part 7: Fuzzy control programming. *IEC Std. 61131*, 2000.

[24] International Electrotechnical Commission. Programmable controllers - part 8: Guidelines for the application and implementation of programming languages. *IEC Std. 61131*, 2003.

[25] International Electrotechnical Commission. Programmable controllers - part 9: Single-drop digital communication interface for small sensors and actuators (sdci). *IEC Std. 61131*, 2013.

[26] PLCopen. Voting Members. Abgerufen von `http://www.plcopen.org/pages/organization/members/voting_members/` am 12.02.2016.

[27] Karl Heinz John and Michael Tiegelkamp. *SPS-Programmierung mit IEC 61131-3*. Springer, 2009.

[28] Industrial Control Systems Cyber Emergency Response Team (USA). Recommended Practice: Improving Industrial Control Systems Cybersecurity with Defense-In-Depth Strategies. Abgerufen von `https://ics-cert.us-cert.gov/sites/default/files/recommended_practices/Defense_in_Depth_Oct09.pdf` am 15.02.2016, 2009.

[29] International Electrotechnical Commission. Industrial communication networks – network and system security. *IEC 62443*, 2008.

[30] National Institute of Standards and Technology. Guide to Industrial Control Systems (ICS) Security. Abgerufen von `http://csrc.nist.gov/publications/nistpubs/800-82/SP800-82-final.pdf` am 15.02.2016, 2011.

[31] U.S. NUCLEAR REGULATORY COMMISSION. CYBER SECURITY PROGRAMS FOR NUCLEAR FACILITIES. Abgerufen von `http://csrc.nist.gov/publications/nistpubs/800-82/SP800-82-final.pdf` am 15.02.2016, 2010.

[32] Theodore J. Williams. *A Reference Model For Computer Integrated Manufacturing (CIM)*. Instrument Society of America, 1991.

[33] Siemens AG. SIMATIC S7-1200 Automatisierungssystem Systemhandbuch. Abgerufen von `https://support.industry.siemens.com/cs/attachments/36932465/s71200_system_manual_de-DE_de-DE.pdf?download=true` am 08.02.2016, 2012.

[34] Bryan Sing Tyson Macaulay. *Cybersecurity for Industrial Control Systems*. CRC Press, 2011.

[35] US Department of Homeland Security. FOIA response documents. Abgerufen von http://s3.documentcloud.org/documents/1212530/14f00304-documents.pdf am 15.02.2016, 2014.

[36] Johannes Klick u.a. Internet-facing PLCs - A New Back Orifice. Abgerufen von https://www.blackhat.com/docs/us-15/materials/us-15-Klick-Internet-Facing-PLCs-A-New-Back-Orifice-wp.pdf am 15.02.2016, 2015.

[37] Technavio. Global PLC Market 2014-2018. Abgerufen von http://www.technavio.com/report/global-plc-market-2014-2018 am 15.02.2016, 2014.

[38] thomas_v2. S7comm Wireshark dissector plugin. Abgerufen von http://sourceforge.net/projects/s7commwireshark/files/ am 25.02.2016.

[39] Dwight E. Cass Marshall T. Rose. ISO Transport Service on top of the TCP. Abgerufen von https://tools.ietf.org/html/rfc1006 am 21.02.2016, 1987.

[40] International Organization for Standardization. Connection oriented transport protocol. *ISO Std. 8073*, 1988.

[41] SCADAStrangeLove. S4x13 Releases: S7 password offline bruteforce tool. Abgerufen von http://scadastrangelove.blogspot.de/2013/01/s7brut.html am 28.02.2016.

[42] HMS Industrial Networks. Marktanteile industrieller Netzwerke weltweit im Jahr 2015 . Abgerufen von http://de.statista.com/statistik/daten/studie/457627/umfrage/marktanteile-industrieller-netzwerke-weltweit/ am 18.02.2016, 2015.

[43] Siemens. Produktkatalog: Datenkommunikation. Abgerufen von https://mall.industry.siemens.com/mall/de/WW/Catalog/Products/9300198?tree=CatalogTree am 18.02.2016, 2015.

Anhang

Als Anhang ist eine CD beigefügt. Diese CD beinhaltet den vollständigen SPS-Wurm im Quelltext und ein Video, dass die Ausbreitung in einem Netzwerk mit vier SPS zeigt.