Fachbereich Elektrotechnik und Informatik
Fachhochschule Münster

Untersuchung der IT-Sicherheit moderner ICS-Systeme am Beispiel der Siemens SIMATIC S7-1200

Bachelorarbeit

Maik Brüggemann
Matrikel-Nummer 658904

Erstprüfer Prof. Dr. Michael Tüxen
Zweitprüfer Hendrik Schwartke

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Das Internet und moderne IT-Systeme haben die Art und Weise der Kommunikation revolutioniert. Schnell, effizient und kostengünstig werden Informationen um den ganzen Erdball ausgetauscht. Heute kommunizieren Menschen meistens untereinander z.B. über Mobilfunk oder soziale Netzwerke. In Zukunft werden nicht nur Menschen vernetzt, sondern auch Maschinen bilden Netzwerke und kommunizieren untereinander. Dadurch können viele Abläufe automatisiert werden.

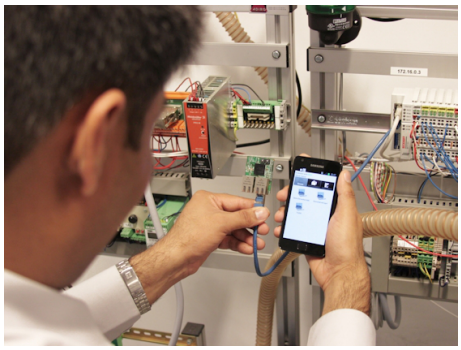


Abbildung 1.1: Vom Sensor direkt ins Internet.

In der Industrie ist der Trend bereits heute erkennbar. In einer modernen Fabrik tauschen Fertigungsanlagen ständig Informationen über den aktuellen Zustand und Fortschritt der Produktion aus. Die Vernetzung ermöglicht eine zentrale Überwachung aller Komponenten. Optimierungen am Produktionsprozess können durchgeführt werden, indem alle anfallenden Daten archiviert und anschließend ausgewertet werden. Die Vision der deutschen Wirtschaft ist die sogenannte Industrie 4.0. Der gesamte Prozess von der individuellen Bestellung des Kunden, über eine flexible Produktion bis zur Auslieferung soll automatisiert werden. Dies kann nur mit einem hohen Grad an Vernetzung erreicht werden.

Auch in anderen Bereichen zeichnet sich eine stärkere Vernetzung ab. Ein Beispiel ist die Energieversorgung. Der Wechsel von fossilen Brennstoffen zu erneuerbaren Energien, von wenigen großen Kraftwerken zu vielen kleinen, verändert die Anforderungen an das Stromnetz. Wird in Norddeutschland keine Windenergie produziert, muss z.B. ein Wasserkraftwerk aus Süddeutschland die fehlende Energie zur Verfügung stellen. Am besten voll automatisiert. Das automatisierte Stromnetz, auch Smart Grid genannt, kann das Wasserkraftwerk veranlassen seine Schleusen zu öffnen, um die fehlende Energie zu kompensieren.

Gemeinsam ist den verschiedenen Bereichen, dass sogenannte Industrial Control Systems eingesetzt werden. Diese Systeme enthalten die eigentliche Intelligenz der Industrie 4.0 oder des Smart

Grids. Sie sind im Wesentlichen vernetzte Computer, die alle wichtigen Entscheidungen treffen, um einen reibungslosen Ablauf der Prozesse zu ermöglichen.

Die weitgehende Vernetzung und die damit verbundene Öffnung der Industrieanlagen birgt aber auch Gefahren. Hacker können in die Netzwerke eindringen und sich Zugang zu den Steuerungssystemen verschaffen. In diesem Fall droht nicht nur ein Datenverlust, sondern es ist möglich, dass ein hoher materieller Schaden entsteht und sogar Menschenleben bedroht werden.

Es stellt sich die Frage, ob die aktuellen Industrial Control Systems die Anforderungen, die an sie in puncto Sicherheit gestellt werden, erfüllen können.

Diese Bachelorarbeit wird Schwächen in dem Umfeld der Industrial Control Systems aufzeigen und an einem Beispiel belegen. Weiterhin wird ein Konzept entworfen, mit dem einige der Schwächen kompensiert werden können.

2 Industrial Control System

2.1 Definition

Unter dem Begriff Industrial Control System (ICS) werden Computersysteme sowie die dazugehörigen Netzwerke verstanden, die physikalische Prozesse in der Industrie und in kritischen Infrastrukturen steuern. Diese werden beispielsweise in Kraftwerken, Wasserwerken, in der Gebäudeautomatisierung und in Fertigungsprozessen einer Fabrik eingesetzt.

2.2 Geräteklassen

Der folgende Abschnitt gibt einen Überblick über die verschiedenen Geräte, die typischerweise in einem ICS eingesetzt werden.

2.2.1 Intelligent Electronic Device (IED)

Unter einem Intelligent Electronic Device (IED) versteht man unterschiedliche Maschinen, die typischerweise im Umfeld von ICS genutzt werden. Wie z.B. ein Motor, ein Ventil, eine Pumpe oder ein Sensor. Sie besitzen einen Mikroprozessor, der ihnen eine digitale Kommunikation mit anderen Geräten ermöglicht. Ein IED wird normalerweise von einem Programmable Logic Controller (PLC) gesteuert.

2.2.2 Remote Terminal Unit (RTU)

Eine Remote Terminal Unit (RTU) ist ein regeltechnisches Instrument, das meistens in einer Außenstation oder an einer dezentralen Stelle eingesetzt wird. Dort werden Feldparameter von Maschinen aufgezeichnet oder es werden Steuerungsbefehle übermittelt. Wenn nötig, werden die Daten erst digitalisiert und dann zu einer zentralen Stelle übermittelt (z.B. einem PLC). Eine RTU

besitzt zu diesem Zweck typischerweise eine Kommunikationsschnittstelle wie z.B Ethernet, GSM oder DSL.

2.2.3 Programmable Logic Controller (PLC)

Ein PLC ist ein Computer, der in einem ICS eingesetzt wird, um Abläufe zu automatisieren. Im Gegensatz zu normalen Computern wird ein PLC an seine Einsatzumgebung angepasst, indem er besonders vor Staub oder elektromagnetischen Einflüssen geschützt wird. Es besteht die Möglichkeit, Sensoren und Aktoren anzuschließen. Das Gerät ist programmierbar, was es ihm erlaubt, komplexe Prozesse zu steuern.

Ein typisches PLC-Programm durchläuft eine Schleife, in der zunächst die Eingänge eingelesen werden. Das Programm des PLC wertet im nächsten Schritt die Eingaben aus und berechnet entsprechend seiner Programmierung Ausgabewerte. Im letzten Schritt werden die Ausgabewerte auf einer physikalischen Schnittstelle ausgegeben. Danach beginnt die Schleife erneut.

Der Steuerungsprozess ist oft zeitkritisch. Deswegen ist ein PLC für einen Echtzeitbetrieb optimiert. Darüber hinaus besitzt ein PLC Kommunikationsschnittstellen, um mit anderen Komponenten des ICS zu kommunizieren.

2.2.4 Human Machine Interface (HMI)

Ein Human Machine Interface (HMI) bildet eine Schnittstelle zwischen Mensch und ICS. Es ersetzt physikalische Schalter oder Hebel durch eine digitale Darstellung. Ein ganzer Produktionsprozess kann in einem grafischen Interface abgebildet werden. Das erlaubt einem Operator den Prozess zu starten und zu stoppen oder Feineinstellungen vorzunehmen. Der Operator kann sich dadurch auf den Gesamtprozess fokussieren und muss sich nicht um die Details des ICS kümmern. Ein HMI kann durch einen handelsüblichen Arbeitsplatzrechner realisiert werden. Viele Hersteller bieten auch spezielle Embedded-Geräte zu diesem Zweck an.

2.2.5 Data Historian

Ein Data Historian ist das Langzeitgedächtnis eines ICS. Es sammelt Daten und speichert sie strukturiert in einer Datenbank ab. Durch Analyseprogramme können z.B. Diagramme von Sensordaten erstellt werden. Diese Informationen ermöglichen die Optimierungen von Prozessen oder die Generierung von Berichten für das Management.

2.3 Struktur eines ICS

Die vorgestellten Geräte werden untereinander vernetzt, wodurch sie ein ICS bilden. Durch die Vernetzung entsteht typischerweise eine Struktur in der drei Ebenen erkennbar sind. Während die unterste Ebene sehr maschienenorientiert ist, liegt der Schwerpunkt in den höheren Ebenen auf der Koordination bzw. der Planung des ICS. Dieser Abschnitt erläutert die verschiedenen Ebenen in einem ICS beginnend mit der untersten.

2.3.1 Process Control System (PCS) bzw. Distributed Control System (DCS)

Ein Process Control System (PCS) zeichnet sich dadurch aus, dass es einen konkreten physikalischen Prozess steuert. Informationen der Feldgeräte laufen in einem zentralen Punkt zusammen (beispielsweise in einem PLC). Umgekehrt werden Steuerbefehle generiert und zu den Feldgeräten gesendet. Die Informationen in diesem Teilsystem sind sehr maschinennah, wie die Drehgeschwindigkeit oder Drehrichtung von Motoren. Ein PCS ist häufig zustandsgetrieben. Das System wechselt dazu beispielsweise von einem Zustand *Flüssigkeit aufwärmen* (Temperatursensor lesen, Heizung steuern) in den nächsten Zustand *Flüssigkeit umrühren* (Umdrehungszahl messen, Motor steuern).

Ein Distributed Control System (DCS) ist dem PCS sehr ähnlich. Jedoch werden Informationen nicht an einem zentralen Punkt gesammelt, sondern sind über das ganze System verteilt. Die einzelnen Bestandteile kommunizieren möglicherweise über größere Distanzen miteinander. Genau wie bei einem PCS gilt, es wird ein konkreter physikalischer Prozess gesteuert.

2.3.2 Supervisory Control and Data Acquisition (SCADA)

Im Gegensatz zu dem untersten Bereich ist die Aufgabe von Supervisory Control and Data Acquisition (SCADA) nicht die Steuerung der einzelnen Prozesse, sondern viel mehr die Koordinierung der Gleichen. Spricht man von SCADA ist damit ein zentraler Punkt gemeint, an dem Daten gesammelt und ausgewertet werden. Jedoch können die Stellen, an denen die Daten erhoben werden über große Distanzen verteilt sein. Ein Beispiel dafür ist eine Pipeline. Viele Sensoren sind über die ganze Strecke an der Pipeline angebracht und messen den Druck im Inneren des Rohres. Ein SCADA-System sammelt alle Sensordaten an einem Punkt und wertet sie aus. Bei bestimmten Ereignissen reagiert das SCADA-System. Denkbar wäre eine Beschädigung der Pipeline und ein damit verbundener Druckabfall. Als Gegenreaktionen stoppt das SCADA-System den Transport von weiteren Rohstoffen. Dazu sendet es Befehle an das PCS, das den konkreten physikalischen Prozess steuern.

2.3.3 Office Network

Unter einem Office Network versteht man ein Netzwerk zur Verbindung von Arbeitsplatzcomputern, wie es typischerweise in Büros vorhanden ist. Es hat häufig einen Zugang zum Internet, um den Mitarbeitern Dienste wie E-Mail bereitzustellen. Ein Office Network ist nicht direkt an der Steuerung physikalischer Prozesse beteiligt. Vielmehr wird es für die Planung und Auswertung der Prozesse benötigt. Der Zeitpunkt, wann ein Mitarbeiter neues Material für die Produktion bestellen muss, lässt sich möglicherweise einfach durch aufgezeichnete Daten bestimmen. Ein Data Historian kann genaue Informationen über alle gefertigten Produkte liefern. Diese können dann z.B. mit einem normalen Tabellenkalkulationsprogramm ausgewertet werden.

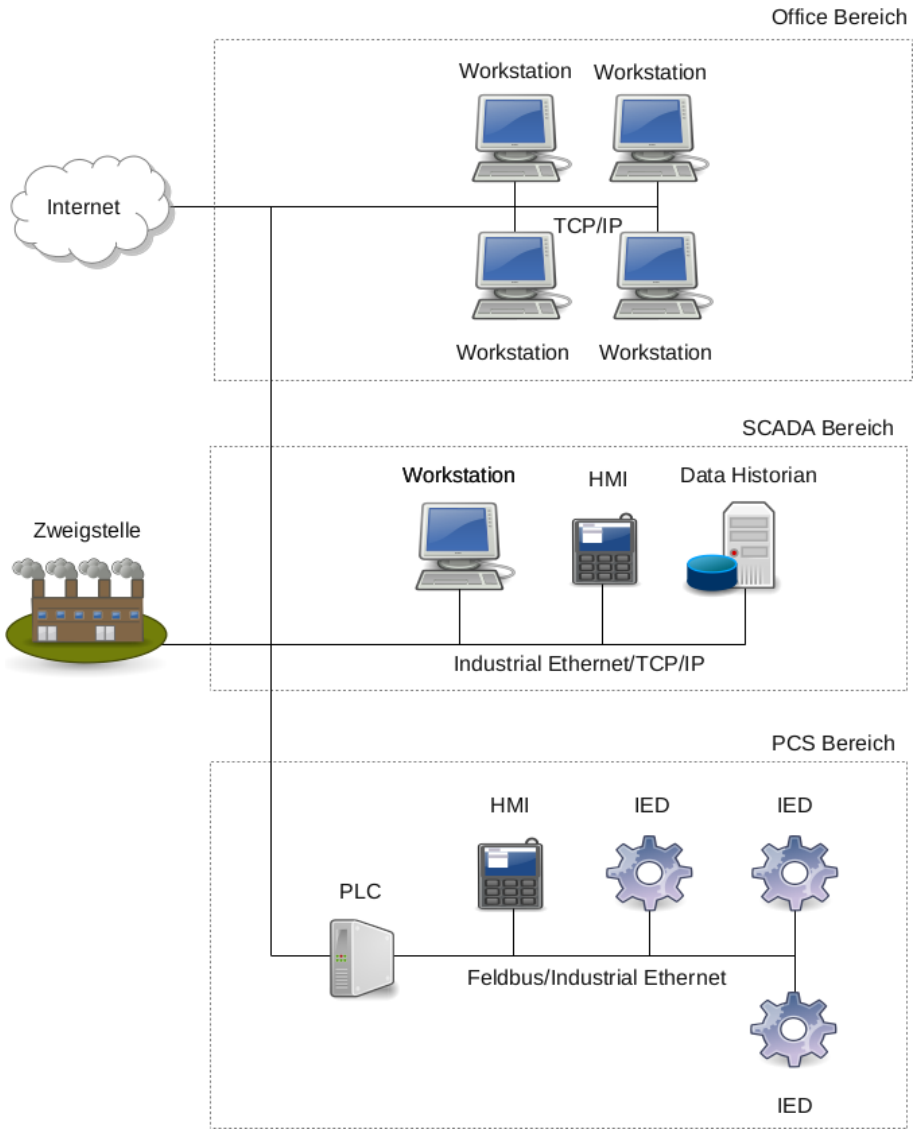


Abbildung 2.1: ICS Bereiche

2.4 Anforderungen & Vernetzung

Jeder der gerade vorgestellten Teile hat eine andere Aufgabe. Der PCS-Teil steuert die Maschinen, der SCADA-Teil koordiniert die gesamte Anlage und das Office Network befasst sich mit Planung und Auswertung. Dabei sind die verschiedenen Teile auf Informationen aus den jeweils anderen angewiesen. Um einen effizienten Informationsaustausch zu ermöglichen, werden alle Bereiche vernetzt.

Durch die verschieden Aufgaben ergeben sich aber auch verschiedene Anforderungen. Während im PCS-Netz Verfügbarkeit und Echtzeit eine besondere Rolle spielen, sind in einem Office Network eine hohe Datenrate und geringe Kosten wichtig. Die folgende Tabelle zeigt die Anforderungen sowie die typischerweise eingesetzten Protokolle.

Teilnetz	Verfügbarkeit	Datenrate	Echtzeit	Protokolle
Office	wichtig	hoch	unkritisch	Ethernet, TCP/IP
SCADA	wichtig	mittel	weich	Feldbus, Industrial Ethernet, TCP/IP
Process Control	sehr wichtig	gering	hart	Feldbus, Industrial Ethernet

Tabelle 2.1: Typische Anforderungen & Protokolle in einem ICS

Es sei erwähnt, dass nicht zwingend alle Teile in einem ICS vorhanden sein müssen. Eine einfache Steuerung für eine Maschine kann vielleicht auch ohne SCADA und ohne ein Office Network funktionieren.

2.5 Anmerkung

Die hier vorgestellten Begriffe lassen sich in der Praxis nicht immer genau unterscheiden. Durch den Einsatz von immer besseren Mikroprozessoren können einfache Geräte Steuerungsaufgaben übernehmen. Deshalb lassen sich besonders IED, RTU und PLC nicht immer scharf voneinander trennen. Weiterhin werden die Begriffe PCS, DCS und SCADA häufig synonym verwendet. In der folgenden Arbeit werden die Begriffe jedoch in der hier erläuterten Weise verwendet.

3 IT-Sicherheit im ICS-Umfeld

3.1 Schutzbedürftigkeit

3.1.1 Einsatzumgebung

ICSs werden häufig in Elektrizitätswerken, in der Öl- und Gas-Produktion, in der Chemiebranche, im Transportwesen, in der Wasserwirtschaft oder in der industriellen Fertigung verwendet. Viele dieser Bereiche produzieren Güter oder erbringen Dienstleistungen, die für einen Industriestaat elementar sind. Das Bundesamt für Sicherheit in der Informationstechnik (BSI) bezeichnet viele dieser Einrichtungen daher als kritische Infrastrukturen (KRITIS). Denn bei einem Ausfall oder Beeinträchtigung würden "[...] nachhaltig wirkende Versorgungsengpässe, erhebliche Störungen der öffentlichen Sicherheit oder andere dramatische Folgen eintreten [...]" [1].

Auch wenn der Ausfall einer Fertigungsstraße in einer Fabrik nicht zwingend ein Problem für die öffentlichen Sicherheit darstellt, so kann er doch hohe finanzielle Schäden durch die Zerstörung von Maschinen oder durch Produktionsausfälle bedeuten. Werden Systeme beeinflusst, die für die Betriebssicherheit der Anlagen sorgen, können sogar Menschen gefährdet werden. Hieraus folgt eine besondere Schutzbedürftigkeit dieser Systeme.

3.1.2 Ein historisch gewachsenes Problem

Neben der Einsatzumgebung ergibt sich eine besondere Schutzbedürftigkeit aus dem geringen IT-Sicherheitsniveau dieser Systeme. Eine Analyse der historischen Entwicklung zeigt die Gründe für diesen Sachverhalt auf.

Früher

Die Entwicklung von ICS beginnt in den späten sechziger Jahren des 20. Jahrhunderts. Im Auftrag von General Motors entwickelte Dick Morley ein Gerät mit dem Namen Modicon (MOdular DIgital

CONtroller). Das 1968 erschienene Gerät gilt als der erste PLC. Im Gegensatz zu früheren Produkten, die in der Prozesskontrolle eingesetzt wurden, ist in diesem Gerät die Logik nicht mehr hart verdrahtet, sondern in Software implementiert. In den folgenden Jahren kommen weitere Firmen mit ähnlichen Produkten auf den Markt (z.B. 1973 Siemens Simatic S3).

Die Entwicklung im Bereich der Mikroprozessoren schreitet voran und die Vernetzung der Systeme nimmt zu. Im Jahre 1979 wird das Protokoll Modbus entwickelt. Es wird für die Kommunikation zwischen PLCs und IEDs auf der Feldebene eingesetzt. In den folgenden Jahren werden weitere Protokolle dieser Art entwickelt und besonders in den neunziger Jahren standardisiert. Sicherheitsfunktionen waren in diesen Protokollen nicht vorgesehen. Dazu gab es auch keinen Grund. Die Systeme sind untereinander inkompatibel. Eine Vernetzung über die Feldebene hinaus fand daher kaum statt. Durch die physikalische Isolation sind die Systeme sicher.

Heute

Ab dem Jahr 2000 ändert sich die Situation jedoch. Die neuen Feldbusse basieren auf Ethernet und verwenden TCP/IP. Man bezeichnet diese Feldbusse daher auch als Industrial Ethernet. Durch die Einführung von Ethernet ergeben sich eine Reihe von Vorteilen:

- Hohe Kompatibilität
- Integration in andere Netzwerke möglich
- Zukunftssichere Technologie
- Hohe Bandbreite
- Herstellerunabhängigkeit

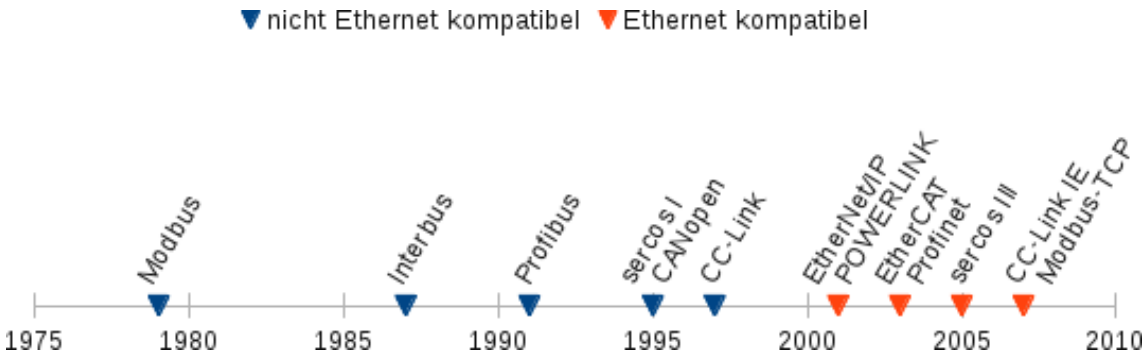


Abbildung 3.1: Verbreitete Feldbusse chronologisch nach Erscheinungsjahr

Durch diese Vorteile ist davon auszugehen, dass Industrial Ethernet in Zukunft weiter an Bedeutung gewinnt und ältere Feldbusse überflüssig machen wird.

Wenngleich die Einführung von Ethernet eine in vieler Hinsicht tiefgreifende Veränderung im Umfeld von ICS mit sich bringt, so sind die Modifikationen an den genutzten Protokollen typischerweise eher gering. Ein Beispiel hierfür ist Modbus. Wurde 1979 die Datenübertragung über die serielle Schnittstelle abgewickelt, wird heute bei Modbus-TCP das TCP-Protokoll verwendet. Das eigentliche Modbus-Protokoll hat sich kaum verändert. Insbesondere wurden keine neuen Sicherheitsfunktionen eingebaut, obwohl das Protokoll nun internetfähig ist. Auch bei anderen neuen Feldbussen wie Profinet, POWERLINK, SERCOS III liegt der Fokus nicht auf Sicherheit, sondern weiterhin auf Robustheit und Echtzeitfähigkeit.

Das Sicherheit im ICS-Umfeld eine untergeordnete Rolle gespielt hat, zeigt die Statistik aus Abb. 3.2, die bekannt gewordene ICS-spezifische Sicherheitslücken von 2000 bis 2012 zeigt. Man erkennt einen plötzlichen Anstieg im vierten Quartal 2011, in welchem die Schadsoftware Stuxnet bekannt wird.

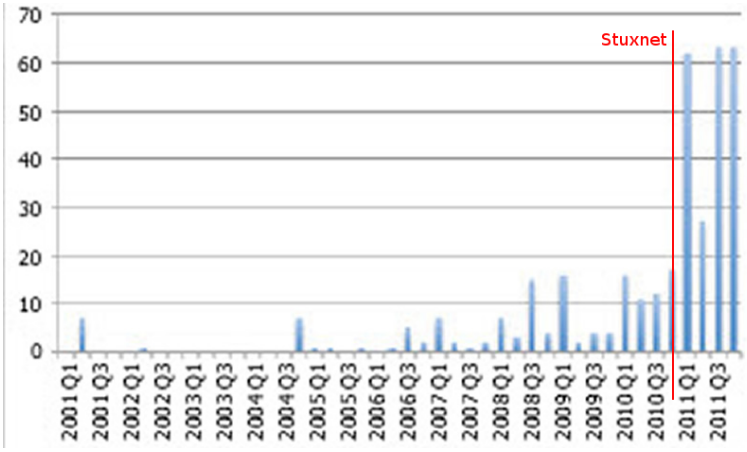


Abbildung 3.2: ICS-spezifische Sicherheitslücken von 2000 bis 2012 [2, Folie 25]

Es ist zu vermuten, dass der plötzliche Anstieg gefundener Sicherheitslücken auf eine erhöhte Sensibilität zurück zu führen ist, die durch Stuxnet ausgelöst wurde. Vorher wurde das Thema ICS-Sicherheit nicht ausreichend berücksichtigt. Viele Experten wandten sich nach Stuxnet selbst dem Thema zu und fanden zahlreiche Sicherheitslücken. Auch die Kunden der Systeme wurden über die Medienberichterstattung auf das Thema aufmerksam und hinterfragen jetzt die IT-Sicherheit der Produkte und ihrer Installationen. Insgesamt hat der Vorfall ein viel breiteres Bewusstsein für IT-Sicherheit im ICS-Umfeld geschaffen.

Zukunft

Auch wenn jetzt ein Umdenken zu mehr IT-Sicherheit stattfindet, wird es noch einige Zeit dauern bis neue Produkte auf dem Markt verfügbar sind. Schließlich müssen die neuen Produkte zunächst

entwickelt werden und sich anschließend bewähren. Weiterhin betragen die Lebenszyklen von ICS-Produkten nicht selten 20 Jahre oder mehr. Wird heute eine neue Anlage geplant, ist sie noch über viele Jahre in Betrieb. Die Vernetzung wird wahrscheinlich zunehmen und die Angriffe durch verfügbare Tools vereinfacht.

Solange Produkte eingesetzt werden, bei deren Entwicklung IT-Sicherheit nur eine untergeordnete Rolle gespielt hat, besteht für die Systeme eine besondere Schutzbedürftigkeit.

3.2 Bedrohungslage

3.2.1 Bekannte Sicherheitsvorfälle

Maroochy Water Services

Das Maroochy Shire Council in Queensland, Australien ist mit der Abwasseraufbereitung in der Region beauftragt. Die Firma Hunter Watertech realisiert zu diesem Zweck ein ICS für die Steuerung der Abwassersysteme. Ein ehemaliger Mitarbeiter leitet mittels gestohlenem Equipment große Mengen Abwasser ab. Dies führt, neben einem großen finanziellen Schaden auf Grund von Reinigungs- und Sanierungsmaßnahmen und einem beträchlichem Imageverlust des Unternehmens, auch zu einem massivem Fischsterben. Dieses Beispiel zeigt die Gefahr, die von Innentätern ausgehen kann [3].

Slammer Worm infiziert Davis-Besse Nuklearkaftwerk

Im Jahr 2003 infiziert der Slammer Wurm das Office-Network des Davis-Besse Nuklearkaftwerkes in dem US-Bundesstaat Ohio. Er breitet sich in dem Netzwerk aus und erreicht den SCADA-Bereich. Dort erzeugt er so viel Netzwerkverkehr, dass zwei Systeme ausfallen, die wichtige Funktionen des Kraftwerks überwachen. Dieser Vorfall zeigt, dass ein ICS-Netzwerk als Kollateralschaden eines eigentlich ganz anderen Angriffes betroffen sein kann [4].

Projekt Aurora

Im März 2007 führt das Department of Homeland Security der USA das „Projekt Aurora“ durch. Das Ziel ist es, einen großen Generator mit einem Angriff auf das ICS zu zerstören. In einem von CNN veröffentlichen Video ist der Erfolg des Projektes dokumentiert. Der Generator kommt zunächst aus dem Takt und fängt dann an zu qualmen. Nach kurzer Zeit ist er vollkommen zerstört.

Es wird deutlich, dass elektrische Systeme nicht nur gestoppt oder verlangsamt werden können, sondern durch einen IT-Angriff auch vollkommen zerstört werden können [5].

Ferngesteuerte Straßenbahn

Im Januar 2008 baut ein 14 Jähriger Junge aus einer alten Fernbedienung ein Gerät, mit dem er Weichen einer Straßenbahnen in Lodz, Polen verstellen kann. Insgesamt vier Straßenbahn entgleisen und 14 Menschen wurden verletzt. Dieses Beispiel zeigt, dass hinter einem Angriff nicht immer kriminelle Energie oder Organisationen mit großen Mitteln stehen müssen [6].

Stuxnet

Der im Jahr 2010 entdeckte Einsatz des Schadprogramms Stuxnet ist eines der komplexesten und zielgenausten Angriffe auf ICSs, die bekannt sind. Das Schadprogramm infizierte Computer, auf denen spezielle Siemens Software vorhanden war. Nachdem die Computer infiziert wurden, suchte die Schadsoftware nach bestimmten PLCs und modifizierte diese ebenfalls. Da die Schadsoftware nur sehr bestimmte PLCs zum Ziel hatte, konnte auf den Zweck des Schadprogramms geschlossen werden. Stuxnet sollte Frequenzumformer, die die Drehgeschwindigkeit von Elektromotoren regeln, beeinflussen. Diese Frequenzumformer finden sich unter anderem in Urananreicherungslagen. Experten gehen daher davon aus, dass gezielt das iranische Atomprogramm sabotiert werden sollte. Aufgrund der Komplexität und den notwendigen Insiderinformationen, die für einen derartig gezielten Angriff notwendig sind, wird davon ausgegangen, dass die Schadsoftware von staatlichen Organisationen stammt [7].

3.2.2 Ergebnisse weiterer Untersuchungen

Aktuelle Forschungsergebnisse zeigen, dass Angriffe auf schlecht geschützte ICS-Netzwerke stattfinden. Die Firma Trend Micro stellt diesen Sachverhalt in ihrem Bericht „Who's Really Attacking Your ICS Equipment?“ [8] dar. Mittels mehrerer Honeypots konnten sie zeigen, dass bereits nach 18 Stunden die ersten Anzeichen eines Angriffes erkennbar waren. Nach 28 Tagen wurden bereits 39 Angriffe aus 14 verschieden Ländern registriert. Trend Micro geht davon aus, dass die Angriffe in Zukunft verbreiteter und anspruchsvoller werden.

3.3 Schutzziele

In der Informationssicherheit wird zwischen den drei grundlegenden Schutzzielen Vertraulichkeit, Verfügbarkeit und Integrität unterschieden. Können alle drei Schutzziele ausreichend[1] realisiert werden, kann ein IT-System als sicher gelten. Dieser Abschnitt definiert die einzelnen Schutzziele kurz und bewertet sie hinsichtlich der Wichtigkeit in den verschieden Bereichen eines ICS.

3.3.1 Vertraulichkeit

"Vertraulichkeit ist der Schutz vor unbefugter Preisgabe von Informationen. Vertrauliche Daten und Informationen dürfen ausschließlich Befugten in der zulässigen Weise zugänglich sein." [9, Seite 49]

Vertraulichkeit spielt auf der Feldebene eines ICS eine untergeordnete Rolle. Die ausgetauschten Informationen sind selten geheim. Sensordaten oder ähnliches beziehen sich oft auf den aktuell ablaufenden Prozess und bieten wenig Angriffsfläche für Spionage. Auch auf SCADA-Ebene werden nur Informationen ausgetauscht, die für die Steuerung des Gesamtsystems interessant sind, weniger für Dritte. Ein Office-Network ist im Bezug auf Vertraulichkeit angreifbarer. Es übermittelt E-Mails oder VoIP-Gespräche die häufig private oder sensible Daten enthalten.

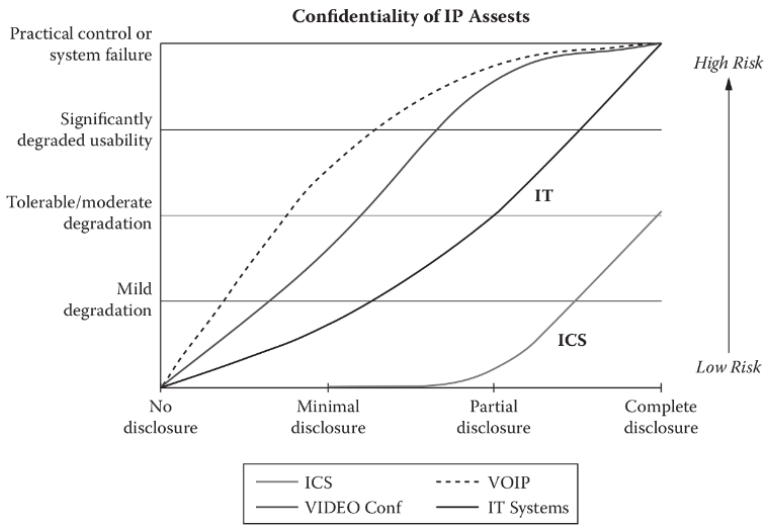


Abbildung 3.3: Anforderungen an die Vertraulichkeit: ICS vs IT [10, Seite 86]

1 Der Aufwand ein Schutzziel zu überwinden ist größer als der Nutzen der daraus entsteht.

3.3.2 Verfügbarkeit

“Die Verfügbarkeit von Dienstleistungen, Funktionen eines IT-Systems, IT-Anwendungen oder IT-Netzen oder auch von Informationen ist vorhanden, wenn diese von den Anwendern stets wie vorgesehen genutzt werden können.” [9, Seite 49]

Ein Verlust der Verfügbarkeit auf der PCS- oder der SCADA-Ebene ist für ein ICS kritisch. Die Echtzeitanforderungen erlauben keine Unterbrechungen. Im Bereich des Office-Netzwerkes ist der Verlust der Verfügbarkeit weniger kritisch. Beispielsweise stellt eine E-Mail, die einige Minuten verzögert übermittelt wird, kein erhebliches Problem dar.

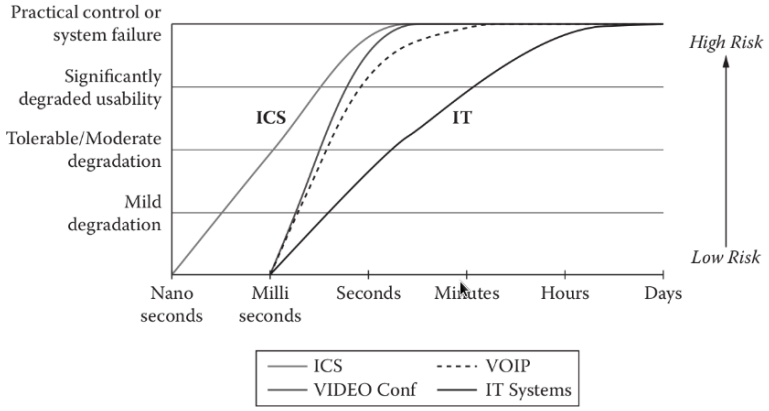


Abbildung 3.4: Anforderungen an die Verfügbarkeit: ICS vs IT [10, Seite 86]

3.3.3 Integrität

“Integrität bezeichnet die Sicherstellung der Korrektheit (Unversehrtheit) von Daten und der korrekten Funktionsweise von Systemen. [...]” [9, Seite 44]

Die Feldebene ist nicht nur besonders empfindlich gegen den Verlust der Verfügbarkeit, sondern auch gegen einen Verlust der Integrität. Eine kleine Veränderung der Sensordaten erzeugen in einem ICS ein völlig falsches Abbild der Realität. Auf der SCADA-Ebene kann sich eine Manipulation sogar auf das ganze System auswirken. Im Bereich des Office-Netzwerkes sind vor allen Dingen gezielte Angriffe auf die Integrität problematisch.

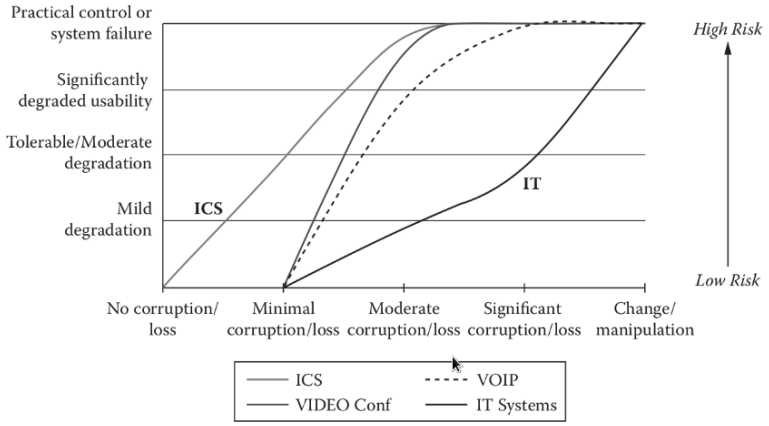


Abbildung 3.5: Anforderungen an die Integrität: ICS vs IT [10, Seite 87]

3.4 Verbreitete Schwachstellen

Nachdem im vorangegangen Abschnitt die grundlegenden Schutzziele erläutert worden sind, zeigt dieser Abschnitt, durch welche Schwachstellen Integrität, Verfügbarkeit und Vertraulichkeit gefährdet werden können. Als Grundlage für die Aufzählung verbreiteter Schwachstellen dient der Bericht „Common Cybersecurity Vulnerabilities in Industrial Control Systems“ [11] von der Homeland Security aus dem Jahr 2011.

3.4.1 Organisatorische Schwachstellen

Fehlendes Sicherheitskonzept und fehlende Dokumentation

Im ICS-Umfeld ist das Thema Sicherheit bislang noch sehr wenig berücksichtigt. Daher gibt es häufig kein Sicherheitskonzept, in dem eine übergreifende Strategie für den Schutz der Systeme definiert wird. Die einzelnen Geräte und deren Schutzbedürftigkeit werden nicht dokumentiert. Ein sicheres System ist auf dieser Grundlage nicht umzusetzen.

Kein Patch-Management

Technische Schwachstellen einer Software werden häufig durch Softwareupdates behoben. Auch bei einer Vielzahl von Geräten muss ein Überblick über die verfügbaren Updates vorhanden sein. Das Einspielen der Updates muss vorbereitet und durchgeführt werden. Das erfordert Organisation, die als Patch-Management bezeichnet wird. Häufig ist kein Patch-Management vorhanden, so dass nicht sichergestellt werden kann, dass alle Geräte mit einer aktuellen Softwareversion betrieben werden.

Kein Backup, keine Restore-Strategie

Nach einem technischen Fehler oder einem erfolgreichen Angriff ist ein schneller Übergang in den normalen Betrieb wünschenswert. Damit das gelingen kann, sollte bereits im Vorfeld eine Restore-Strategie entwickelt worden sein. Backups können hierbei sehr hilfreich sein. In der Praxis sind solche Strategien oft nicht vorhanden.

Keine Logging, keine Security Audits

Gescheiterte Login-Versuche oder ähnliches werden häufig nicht protokolliert. Um Sicherheitsverstöße erkennen zu können, sollte dies jedoch geschehen. Weiterhin wird meistens nicht aktiv in Form von Security Audits nach Schwachstellen in den Systemen gesucht.

3.4.2 Technische Schwachstellen

Fehlende Eingabevalidierung

Daten, die ein Programm einliest, sind nicht immer sinnvoll. Im Fall eines Angriffs können Eingabedaten sogar dazu verwendet werden, ein System zu kompromittieren. Daher ist die Validierung der Eingabedaten wichtig, um ein Programm robust und sicher zu machen. Viele ICS-Komponenten weisen bei der Eingabevalidierung jedoch Schwächen auf. Daraus ergeben sich eine ganze Reihe von Sicherheitslücken wie Buffer-Overflows, SQL Injection, XSS, Path Traversal oder Command Injecten. Die Folgen reichen von Informationsverlust über DoS bis zur kompletten Übername der Komponenten.

Mangelnde Zugriffskontrolle oder zu viele Zugriffsrechte

Zugriffskontrollen sind häufig gar nicht vorhanden oder mangelhaft umgesetzt. Teilweise ist es möglich, Programme auf ICS-Geräte aufzuspielen und zu starten ohne das eine Zugriffskontrolle stattfindet. Zugriffsrechte sind meist viel zu weit gefasst. Mangels eines rollenbasierten Zugriffssystems werden alle Prozesse mit den höchsten Rechten ausgeführt. Ein Benutzer hat häufig mehr Zugriffsrechte als für die Erledigung seiner Aufgabe notwendig wäre.

Unzureichende Authentifikation

Authentifikationsmechanismen sind häufig einfach zu umgehen. Die IP-Addresse oder andere einfach zu fälschende Informationen sind die einzigen Authentifikationsmerkmale. Kritische Systemfunktionen wie das Lesen oder das Schreiben von PLC-Variablen, führen teilweise überhaupt keine Authentifikation durch.

Unzureichende Verifikation von Daten

Daten können oft unbemerkt verändert werden, da die Integrität nur selten geprüft wird. Eine Verifikation der Daten mit Signaturen oder anderen kryptografischen Verfahren findet kaum statt. Gleiches gilt für Firmwareupdates die ohne Verifikation auf ein ICS-Gerät überspielt werden können.

Fehlende oder fehlerhafte Kryptographie

Sensible Informationen wie Passwörter werden häufig im Klartext gespeichert und übertragen. Wird Kryptographie implementiert, ist diese oft fehlerhaft oder riskant, weil der Algorithmus als unsicher gilt.

Falscher Umgang mit Passwörtern

Oft werden Standardpasswörter der Hersteller verwendet oder die Passwörter sind viel zu einfach gewählt. Eine weitere beliebte Schwachstelle sind fest einprogrammierte Passwörter.

Mangelnde Kontrolle von Speichermedien

Ein Angreifer kann nicht nur über das Netzwerk eindringen. Ebenso kommen Speichermedien wie USB-Sticks als Einfallstor infrage. Der Umgang mit Datenträgern ist jedoch häufig unbedacht.

3.4.3 Schwachstellen im Netzwerk

Keine Segmentierung der Netze

Die verschieden Bereiche eines ICS-Netzwerks werden häufig nicht untereinander getrennt. Wenn ein Bereich kompromittiert wird ist so der Zugriff auf weitere Teile des Netzwerks möglich.

Keine oder schlecht konfigurierte Firewall

Oft wird keine Firewall eingesetzt, um einzelne Teile oder Komponenten eines ICS-Netzwerkes zu schützen. Dadurch wird der Zugriff auf die Systeme unnötig erleichtert. Wird eine Firewall eingesetzt, ist die Konfiguration oft nicht optimal. Denn die Firewall lässt mehr Daten passieren, als für den Betrieb unbedingt erforderlich sind.

Unterlaufen einer Firewall

Eine Firewall kann in machen Fällen komplett unterlaufen werden. Dies kann z.B. durch einen Fernwartungszugang geschehen, der über eine eigene Modem-Verbindung realisiert wird.

4 Machbarkeitsuntersuchung von Angriffen an einem realen Beispiel

4.1 Zielsetzung

Während sich das vorangegangen Kapitel allgemein mit der IT-Sicherheit von ICS-Infrastrukturen beschäftigt hat, werden in diesem Kapitel reale Produkte untersucht. Der Fokus liegt hierbei auf den Netzwerkprotokollen. Es soll untersucht werden, wie einfach die Kommunikation zwischen verschiedenen Komponenten gestört oder manipuliert werden kann. Es wird außerdem aufgezeigt, wie einfach in den Programmablauf eines PLC eingegriffen werden kann.

4.2 Eingesetzte Produkte

4.2.1 Simatic S7-1200

Für den Versuch wird ein PLC der Firma Siemens verwendet. Es handelt sich um das Gerät Simatic S7-1212C. Der PLC besitzt verschiedene digitale Ein- und Ausgänge. Als Kommunikationsschnittstelle ist ein Ethernet-Port vorhanden. Er wird mit der aktuellen Firmware Version 03.00.02 betrieben. Durch das Modul SM 1223 DC/DC wird der PLC um 16 digitale Eingänge sowie 16 weitere Ausgänge erweitert. Da Siemens in Deutschland Marktführer ist und sich die S7-1200 PLCs im unteren Preissegment befinden, bietet sich dieser PLC für die Bachelorarbeit an.

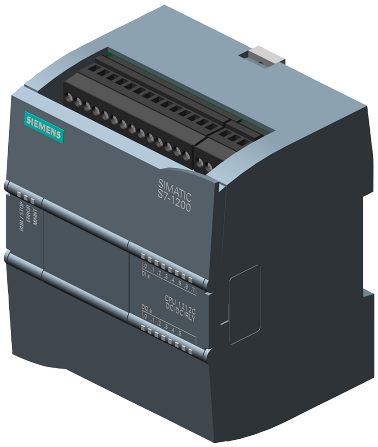


Abbildung 4.1: Simatic S7-1212C

4.2.2 KTP400 Basic Color PN

Als Kommunikationspartner für den PLC wird das HMI KTP400 Basic Color PN von Siemens eingesetzt. Dieses Gerät kann Eingaben über einen Touchscreen sowie vier Tasten entgegennehmen. Für die Kommunikation steht ebenfalls ein Ethernet-Port zur Verfügung. Auch das HMI wird mit der aktuellen Firmware-Version betrieben.

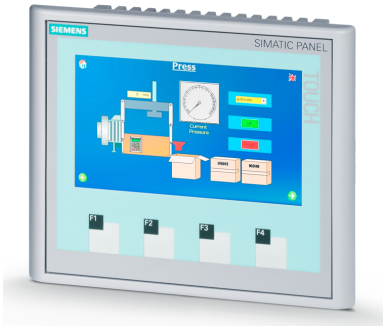


Abbildung 4.2: KTP400 Basic Color PN

4.3 Versuchsaufbau

4.3.1 Physikalischer Aufbau der Ampel

Für den Versuch wird eine einfache Ampel simuliert, die von dem PLC gesteuert wird. Die Ampel besteht aus 12 LED-Leuchten. Diese sind über einen elektrischen Widerstand mit den digitalen Ausgängen des PLC verbunden. Zusätzlich simulieren vier Fotowiderstände die Induktionsschleifen einer realen Ampel. Eine elektrische Schaltung wandelt die analogen Signale der Fotowiderstände in digitale Signale um und gibt sie an den PLC weiter.

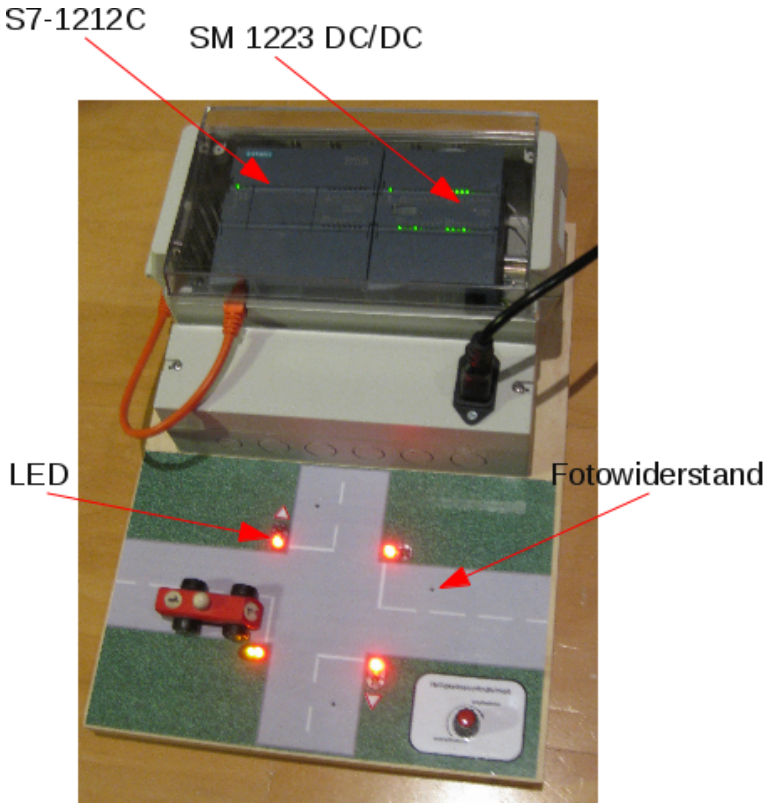


Abbildung 4.3: Foto der Modell-Ampel

Das HMI visualisiert den aktuellen Zustand der Ampel. Über einen Ethernet-Switch wird die Verbindung zwischen dem PLC und dem HMI hergestellt. Zusätzlich ist ein Computer direkt mit dem Switch verbunden, von dem Angriffe durchgeführt werden können.

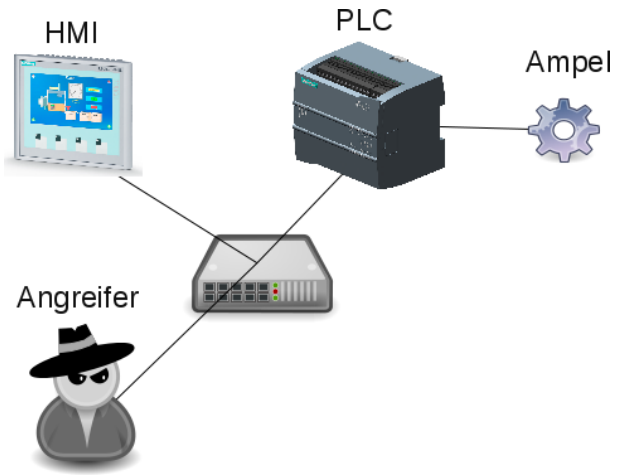


Abbildung 4.4: Schematische Darstellung des Versuchsaufbaus

4.3.2 Programmierung der Anwendungssoftware

Die Anwendungssoftware für den PLC und das HMI wird mit der Entwicklungsumgebung *TIA Portal v11* programmiert. Die Anwendungssoftware wird mit der Programmiersprache *SCL* erstellt. Das Programm für das HMI wird graphisch erstellt.

Durch das *TIA Portal v11* wird eine Verbindung eingerichtet, mit der PLC und HMI Daten austauschen können.

4.3.3 Sicherheitsmaßnahmen der CPU

Das Handbuch der Simatic S7-1200 Baureihe beschreibt, dass die CPU drei Sicherheitsstufen bietet, um den Zugang zu bestimmten Funktionen einzuschränken. Die Siemens S7-1212C wird mit der höchsten Schutzstufe *Lese-/Schreibschutz* konfiguriert:

> Ein Passwort ist zum Lesen der Daten in der CPU, für Änderungen (Schreiben) in der CPU und für den Wechsel des Betriebszustands der CPU (RUN/STOP) erforderlich [12, Seite 86]

Neben diesen drei Sicherheitsstufen existieren noch zwei weitere Sicherheitsmechanismen. Der erste bezieht sich auf den Gebrauch von SD-Karten. Der zweite auf den Download von Anwenderprogrammen aus dem PLC. Beide werden in diesem Versuch nicht berücksichtigt.

4.4 Ziel des Versuchs

Ziel des Versuchs ist die Beeinflussung des PLC (z.B. alle Ampeln auf grün). Gleichzeitig soll die Kommunikation zwischen dem PLC und dem HMI manipuliert werden. Dadurch soll der Zugriff auf den PLC verschleiert werden. Hierzu müssen die folgenden Schritte durchgeführt werden:

- Die Netzwerkprokolle werden auf Schwachstellen untersucht
- Es wird eine Anwendung entwickelt, die einen Angriff ermöglicht
- Der Angriff wird an dem Ampelmodell simuliert

4.5 Analyse der Netzwerkprotokolle

Zu Beginn der Analyse wird der reguläre Netzwerkverkehr zwischen dem PLC und dem HMI sowie dem PLC und der Entwicklungsumgebung betrachtet. Es können zwei Protokolle identifiziert werden über die die gesamte Kommunikation abgewickelt wird:

- Profinet - Discovery and basic Configuration Protocol
- S7-Communication

Beide Protokolle werden im Folgenden analysiert und bezüglich der vorhandenen Schwachstellen bewertet.

4.5.1 Profinet - Discovery and basic Configuration Protocol

Die Simatic S7-1212C implementiert den Industrial Ethernet Standard Profinet. Dieser Standard ist in [13] spezifiziert. Die Aufgabe von Profinet ist die Realisierung von schnellen Datenverbindungen zwischen verschiedenen ICS-Komponenten. Dazu verwendet es eine Reihe von verbreiteten Standards wie Ethernet oder TCP/IP. Neben dem Austausch von Daten definiert Profinet durch das DCP eine Möglichkeit, Geräte über das Netzwerk zu konfigurieren. Das DCP ist obligatorischer Bestandteil von Profinet und wird von der S7-1212C unterstützt. Neben der S7-1212C unterstützt das Netzwerkanalyseprogramm *Wireshark* DCP ebenfalls. *Wireshark* wird verwendet um das Netzwerkprotokoll zu analysieren. Die Spezifikation muss Kostenpflichtig erworben werden und wird daher nicht verwendet.

Funktionsumfang

DCP besitzt zwei Funktionen:

- Finden von Profinet Geräten im Subnetz
- Lesen und Setzen von Konfigurationswerten (insbesondere IP-Adresse)

Funktionsweise

Damit DCP in der Lage ist, Geräte zu konfigurieren, die noch keine IP-Konfiguration besitzen, setzt es direkt auf der Sicherungsschicht (OSI Schicht 2) auf. Es ist ein binäres nachrichtenorientiertes Protokoll.

DCP definiert eine Basis-Nachricht, die in Abbildung 4.5 dargestellt ist. Die Basis-Nachricht bestimmt im Wesentlichen, welche Aktion ausgeführt wird (lesen, schreiben von Konfigurationswerten oder Identifizierung). Weiterhin bestimmt die Basis-Nachricht, ob es sich um eine Anfrage oder eine Antwort handelt. Das Datenfeld der Nachricht überträgt sogenannte Blöcke. Für jeden Konfigurationswert ist ein bestimmter Block definiert.

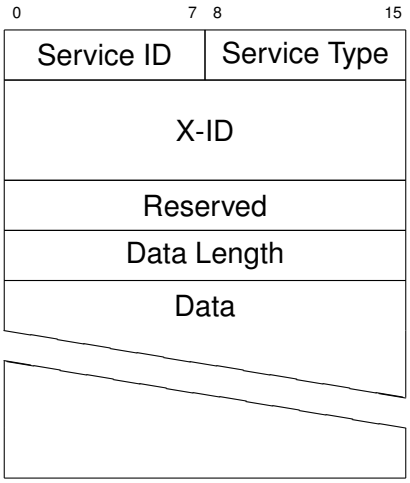


Abbildung 4.5: DCP Basis-Nachricht

Feldname	Beschreibung
Service ID	Bestimmt die durchzuführende Aktion (GET, SET, IDENTIFY).
Service Type	Definiert, ob es sich um eine Anfrage, eine Antwort handelt.
X-ID	Identifikation der Transaktion
Data Length	Die Länge der folgenden Daten
Data	Enthält einen oder mehrere sogenannter Blöcke

Tabelle 4.1: Beschreibung der DCP Basis-Nachricht

Die erste Funktion des DCP identifiziert alle Geräte in einem Subnetz. Um diese Funktion auszuführen wird eine Basis-Nachricht an die Multicast-Adresse *01:0e:cf:00:00:00* gesendet. Die Nachricht bestimmt mit ihrem Aktionsfeld, dass eine Identifizierung durchgeführt wird. Alle Geräte, die diese Nachricht erhalten, antworten mit der gleichen Basis-Nachricht. Der Datenteil der Antwort beschreibt das Gerät durch die enthaltenen Blöcke. Dazu zählen: IP-Konfiguration, Name der Station, Typ der Station und einige weitere.

Die zweite Funktion ändert Konfigurationsparameter in Profinet-Geräten. Zu diesem Zweck wird eine Nachricht an die MAC-Adresse des Zielsystems gesendet. Die Basis-Nachricht definiert, dass es sich um eine Anfrage zum Setzen von Konfigurationswerten handelt. Der Datenteil enthält Blöcke mit neuen Konfigurationswerten.

Für die IP Konfiguration ist ein Datenblock definiert, der in Abb. 4.6 dargestellt ist. Die ersten drei Felder sind in jedem Block identisch. Die restlichen Felder sind spezifisch für die IP-Konfiguration.

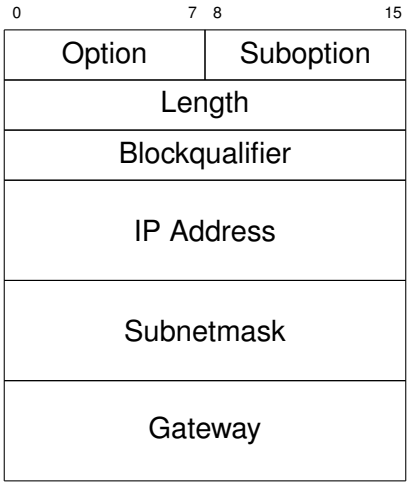


Abbildung 4.6: Datenblock IP-Konfiguration

Feldname	Beschreibung
Option	Identifiziert den Typ des Blockes
Suboption Type	Unterteilt den Typ in weitere Subtypen
Length	Die Länge der nachfolgenden Blockfelder
Blockqualifier	Gibt an, ob die IP-Konfiguration permanent oder temporär gesetzt werden soll
IP Address	Die IP-Adresse
Subnetmask	Die Subnetzmaske
Gateway	Der Gateway

Tabelle 4.2: Beschreibung des IP-Blockes

Wird eine Nachricht zum Setzen einer neuen IP-Konfiguration versendet, antwortet die Gegenseite mit einer Bestätigung. Der Datenteil der Antwort beinhaltet einen Kontrollblock, der angibt, ob die Aktion erfolgreich war. Der Kontrollblock ist in Abb. 4.7 dargestellt.

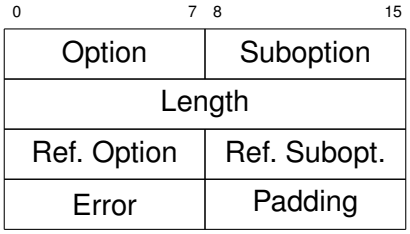


Abbildung 4.7: DCP Kontrollblock

Feldname	Beschreibung
Option	Identifiziert den Typ des Blockes
Suboption Type	Unterteilt den Typ in weitere Subtypen
Length	Die Länge der nachfolgenden Blockfelder an
Reference Option	Gibt an, auf welche Option sich das Error-Feld bezieht
Reference Suboption	Gibt an, auf welche Suboption sich das Error-Feld bezieht
Error	Zeigt einen Fehler an
Padding	Datenblöcke bestehen immer aus einer geraden Anzahl Bytes

Tabelle 4.3: Beschreibung des Kontrollblockes

Alle weiteren Konfigurationswerte können analog gesetzt werden. Für jeden Konfigurationswert ist ein spezifischer Blocke definiert.

Schwachstellenanalyse

Durch die Identifizierungsfunktion existiert eine effiziente Möglichkeit, eine Beschreibung aller Geräte abzurufen. Netzkonfiguration, der Type des Gerätes oder Angaben über den Hersteller kön-

nen für die Planung eines Angriffs sehr hilfreich sein.

Weiterhin ist durch die Analyse deutlich geworden, dass das Protokoll keine Felder definiert, die eine Authentifizierung ermöglichen. Ein potenzieller Angreifer hat somit die Möglichkeit, ungehindert in die Netzwerkkonfiguration einzugreifen und die Kommunikation zu stören bzw. gezielt umzuleiten.

4.5.2 S7-Communication

Alle SIMATIC S7 Geräte implementieren das Protokoll S7-Communication. Dieses Protokoll ist kein Bestandteil von Profinet, sondern eine Entwicklung von Siemens, über die keine offizielle Dokumentation vorhanden ist[1]. Daher muss das Protokoll durch Reserve Engineering analysiert werden. Für dieses Protokoll existiert eine *Wireshark*-Erweiterung [14] die bei diesem Prozess hilft. Das Protokoll ist unabhängig von dem unterliegenden Kommunikationssystem. Im Rahmen dieser Arbeit wird auf die Übertragung per Ethernet und TCP/IP eingegangen.

Funktionsumfang

Mit dem S7-Communication-Protokoll können die meisten Funktionen eines SIMATIC S7-Gerätes gesteuert werden. Dazu zählen:

- Einspielen von Konfigurationen
- Upload von Benutzerprogrammen
- Lesen von Diagnoseinformationen
- Schreiben und Lesen von Prozessvariablen
- Starten und Stoppen von Geräten

Funktionsweise

S7-Communication ist ein binäres nachrichtenorientiertes Protokoll. Die Struktur jeder Nachricht ist identisch und ist in Abb. 4.8 dargestellt. Jede Nachricht besteht aus drei Teilen.

Der erste Teil enthält allgemeine Informationen wie eine Versionsangabe oder eine Sequenznummer. Weiterhin bestimmt der erste Teil durch ein Typen-Feld, ob es sich um eine Anfrage oder eine Antwort handelt.

Der zweite Teil legt die Funktion der Nachricht fest und enthält die dazu notwendigen Parameter. Wenn z.B. Prozessvariablen geschrieben werden sollen, enthält der zweite Teil die Adressen der zu schreibenden Prozessvariablen.

Der dritte Teil ist der Datenteil. Im Falle eines Schreibvorganges von Prozessvariablen sind hier die eigentlichen Werte der Variablen enthalten.

1 Aufgrund dieser Tatsache kann nicht sicher gestellt werden, dass die folgenden Angaben vollständig sind oder für alle SIMATIC S7 Geräte zutreffend sind.

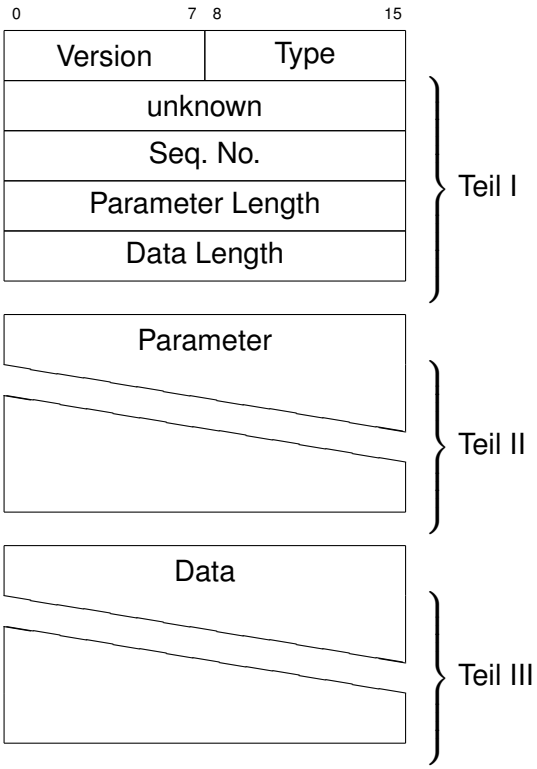


Abbildung 4.8: Struktur einer S7-Communication Nachricht

Feldname	Beschreibung
Version	Protokollversion
Type	Bestimmt, ob die Nachricht eine Anfrage oder Antwort ist
Unknown	Unbekannt
Seq. No.	Eine Sequenznummer
Parameter Length	Länge des zweiten Teils
Data Length	Länge des dritten Teils

Tabelle 4.4: Beschreibung einer S7-Communication Nachricht

Zu Beginn jeder Kommunikation findet ein Verbindungsaufbau statt. Dazu werden Verbindungsparameter durch eine Nachricht übermittelt. Der Parameterteil dieser Nachricht ist in Abb. 4.9 dargestellt. Ein Datenteil ist während des Verbindungsaufbaues nicht vorhanden. Die Gegenseite antwortet mit der gleichen Nachricht, wenn sie die Parameter akzeptiert oder mit neuen Werten, wenn sie andere Parameter wünscht. Danach ist die Verbindung aufgebaut.

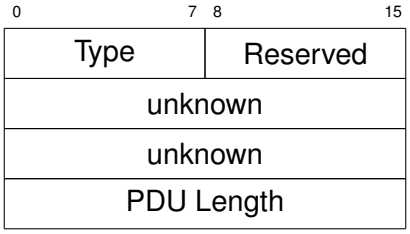


Abbildung 4.9: Parameterteil einer S7-Communication Nachricht mit Verbindungsparametern

Feldname	Beschreibung
Version	Protokollversion
Type	Bestimmt, ob die Nachricht eine Anfrage oder Antwort ist
Reserved	Reserved
unknown	Unbekannt
unknown	Unbekannt
PDU Length	Maximale Länge einer Nachricht

Tabelle 4.5: Beschreibung einer S7-Communication Nachricht mit Verbindungsparametern

Um Prozessvariablen in den PLC schreiben zu können, ist die Nachricht aus Abbildung 4.10 definiert. Ein Typenfeld bestimmt, dass ein Schreibzugriff stattfinden soll. Es folgt die Anzahl der zu schreibenden Variablen sowie detaillierte Informationen über jede einzelne Variable.

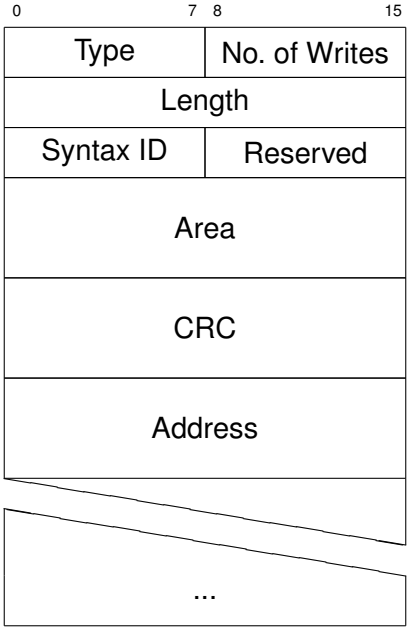


Abbildung 4.10: Parameterteil zum Schreiben von Variablen

Feldname	Beschreibung
Type	Identifiziert, die Funktion die ausgeführt werden soll (hier Variablen schreiben)
Number of Writes	Anzahl der Variablen
Syntax ID	Angabe über den Aufbau der folgenden Felder (hier nur S7-1200)
Reserved	Reserved
Area	Bestimmt den Bereich, in dem die Variable definiert ist (z.B. Eingänge, Ausgänge, Speicher)
CRC	Prüfsumme
Address	Adresse der Variable

Tabelle 4.6: Beschreibung eines Parameterteiles zum Schreiben von Variablen

Die eigentlichen zu schreibenden Werte folgen im Datenteil. Für jeden Wert sind Metainformationen vorhanden, die den Datentyp sowie die Länge des Wertes beschreiben. Die Abb. 4.11 zeigt diesen Datenteil für einen Wert.

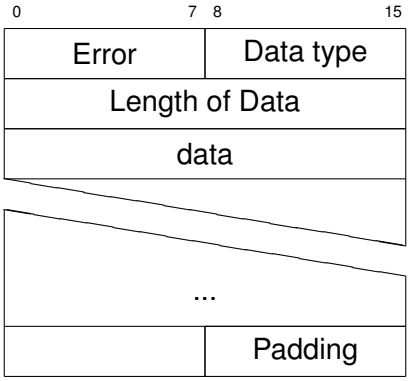


Abbildung 4.11: Datenteil für eine Variable

Feldname	Beschreibung
Error	Möglichkeit Fehler anzuzeigen (nur bei Antworten relevant)
Data type	Gibt den Datentyp der Variablen an (z.B. int, char, usw.)
Length of Data	Länge des folgenden Feldes
data	Der eigentliche Wert der Variablen
Padding	Ggf. ein Padding-Byte

Tabelle 4.7: Beschreibung des Datenteiles für eine Variable

Die Gegenseite bestätigt den Schreibvorgang mit einer Nachricht. Das Fehlerfeld im Datenteil der Bestätigung gibt Auskunft, ob der Schreibvorgang der Variablen erfolgreich war.

Das Lesen von Prozessvariablen läuft analog ab. Nur das Typenfeld wird verändert, um einen Lesezugriff anzuzeigen. Weiterhin ist in der Leseanfrage kein Datenteil vorhanden.

Das S7-Communication-Protokoll wird nicht direkt per TCP übertragen. Es wird in den Protokollen ISO8072-Class0 und TPKT gekapselt. Die beiden Protokolle ermöglichen es, das nachrichtenorientierte S7-Communication in dem stromorientierten TCP zu übermitteln. Diese Protokolle sind im ISO-Standard 8073 [15] und RFC1006 [16] genauer beschrieben.

4.5.3 Schwachstellenanalyse

Bei der Betrachtung des Protokolls fällt auf, dass in den einzelnen Nachrichten keine Felder vorhanden sind, die eine Authentifizierung oder eine Integritätsprüfung ermöglichen. Dadurch kann ein Angreifer eine Verbindung zu dem PLC aufbauen und beliebige Prozessvariablen schreiben. Dies ist besonders kritisch da so direkt in den Programmfluss eingegriffen werden kann. Durch die

fehlende Integritätsprüfung können die Daten ebenso durch einen Angreifer manipuliert werden. Diese zwei grundlegenden Probleme zählen zu den verbreiteten Schwachstellen in ICS-Geräten und sind unter 3.4.2 aufgeführt.

4.6 Entwicklung einer Anwendung zur Versuchsdurchführung

4.6.1 Anforderungsanalyse

Der Funktionsumfang der Anwendung ergibt sich im Wesentlichen aus den Zielen des Versuches:

- Identifizieren der Ziel-Geräte im Netzwerk
- Schreiben von Prozessevaribalen über das S7-Communication-Protokoll
- Umleiten der S7-Kommunikation zum Angreifer
- Aufzeichnen der S7-Kommunikation
- Wiedergabe der aufgezeichneten S7-Kommunikation

Allgemeine Anforderungen

Die Anwendung soll auf aktuellen Linux-Distributionen lauffähig sein. Um dieses Ziel zu erreichen und eine möglichst einfache und schnelle Programmierung zu ermöglichen, wird Python als Entwicklungssprache festgelegt. Die Bedienung erfolgt über die Kommandozeile. Auf eine graphische Oberfläche wird verzichtet.

Identifizieren der Ziel-Geräte im Netzwerk

Um die Ziele im Netzwerk identifizieren zu können, wird das DCP verwendet werden. Die entsprechende Identifizierungsfunktion muss implementiert werden.

Schreiben von Prozessevariablen über das S7-Communication-Protokoll

Damit Prozessvariablen geschrieben werden können, müssen die notwendigen Protokolle implementiert werden. Ethernet und TCP/IP sind bereits im dem Linux Kernel vorhanden und können über die Socket-API genutzt werden. Die weiteren Protokolle müssen selbst implementiert werden. Das TPKT- und das ISO8073 Class0-Protokoll sollen vollständig implementiert werden. Mit Hilfe des S7-Communication-Protokolles sollen nur Prozessvariablen gelesen und geschrieben werden können.

Umleiten der S7-Communication Kommunikation zum Angreifer

Die einfachste Möglichkeit die Kommunikation umzuleiten, besteht darin, dem PLC eine neue IP-Adresse zu geben und dem Angreifer die alte IP-Adresse des PLC. Auf diese Weise verbinden sich alle Geräte automatisch mit dem Angreifer. Zu diesem Zweck soll die DCP-Funktion implementiert werden, die das Setzen einer neuen IP-Adresse ermöglicht. Durch dieses Vorgehen werden die bestehenden TCP-Verbindungen des PLCs unterbrochen. Da die Komponenten aber die Verbindung selbständig wieder aufbauen, wird diese kurze Unterbrechung zugelassen.

Aufzeichnen und Wiedergabe der S7-Communication Kommunikation

Eine Möglichkeit, den Angriff vor dem HMI zu verschleiern besteht darin, zunächst reguläre Kommunikation aufzuzeichnen. Zum Zeitpunkt des Angriffes wird die Aufzeichnung einfach wieder abgespielt, um den HMI möglichst echt aussehende Daten zu präsentieren. Zu diesem Zweck wird ein Proxy entwickelt. Der Proxy nimmt die umgeleiteten Verbindungen an und leitet sie an den PLC weiter. Dabei sollen alle Lese- und Schreibzugriffe aufgezeichnet werden. Nach einer angegebenen Zeitspanne leitet der Proxy die Anfragen nicht mehr weiter, sondern beantwortet sie aus der Aufzeichnung.

4.6.2 Entwurf

Gliederung der Anwendung

Die Anwendung wird in mehrere kleinere Programme gegliedert. Dadurch wird die gesamte Anwendung flexibler und die Komplexität wird auf ein überschaubares Maß reduziert. Jedes der folgenden Programme realisiert eine der Anforderungen:

- s7-setvar: Schreiben von Prozessvariablen
- s7-proxy: Aufzeichnen und Wiedergeben der S7-Kommunikation
- dcp-setip: IP-Adresse per DCP setzen
- dcp-discovery: Profinet Geräte auflisten

Identifizieren von gemeinsamen Komponenten

Um zu vermeiden, dass durch die Gliederung der Anwendung bestimmte Programmteile doppelt programmiert werden müssen, werden gemeinsam genutzte Komponenten ausgelagert. Die Abb. 4.12 zeigt die gemeinsam genutzten Komponenten und deren Beziehungen zu den Programmen.

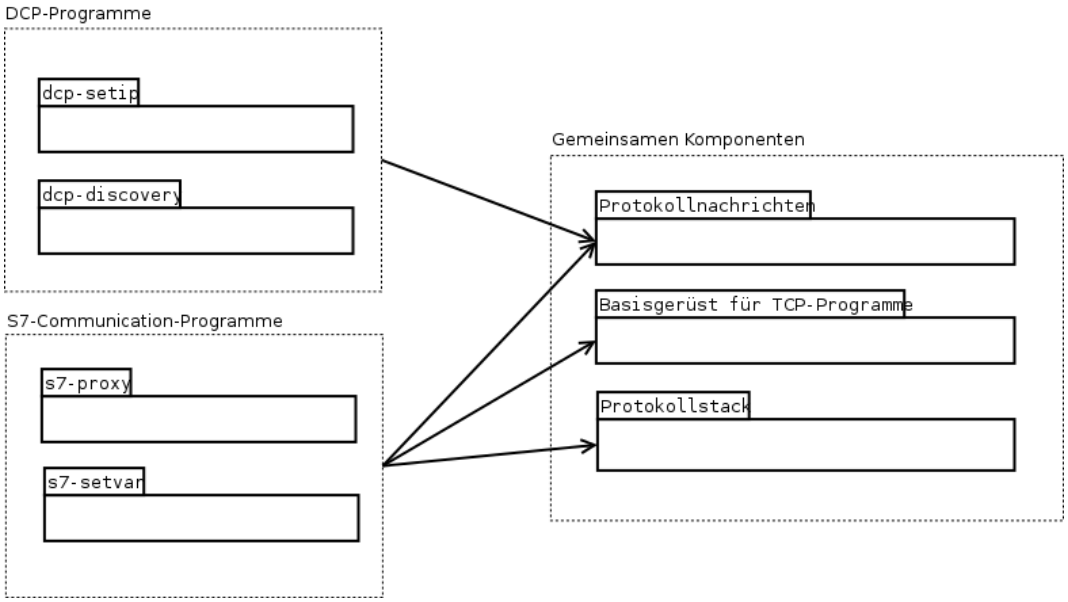


Abbildung 4.12: Gemeinsamen Komponenten

Basisgerüst für TCP-Programme

Die Ein- und Ausgabeverarbeitung soll asynchron erfolgen. Im Gegensatz zur synchronen Form wartet ein Prozess nicht auf die Beendigung einer Lese- oder Schreiboperation. Sind mehrere Ein- und Ausgabeströme vorhanden, werden die verschiedenen Ströme in einem Prozess gemultiplext. Das Linux-Betriebssystem realisiert dieses Konzept über den *select* Systemaufruf. Der Systemaufruf überwacht die einzelnen Kommunikationskanäle und signalisiert, wann gelesen oder geschrieben werden kann. Dadurch ist es möglich, ein Programm zu implementieren, das ereignisorientiert arbeitet. Diese Eigenschaft kann man sich besonders bei der Implementierung des S7-Proxys zunutze machen, da nicht vorhergesagt werden kann, wann einer der Kommunikationspartner eine neue Nachricht versendet.

In der Standardbibliothek der Programmiersprache Python ist das Modul *Asynchronous socket handler* enthalten. Dieses Modul ermöglicht ereignisorientierte Netzwerkprogrammierung in Python. Mit Hilfe des Moduls wird ein Grundgerüst für ein ereignisorientiertes Netzwerkprogramm entwickelt. Das UML des Entwurfs ist in Abb. 4.13 dargestellt und zeigt die Klasse *asyncore.dispatcher* aus dem Python Modul sowie die entwickelten Klassen *TcpCommunicationChannel*, *TcpServer*, *AppletFactory* und *NetworkApplet*.

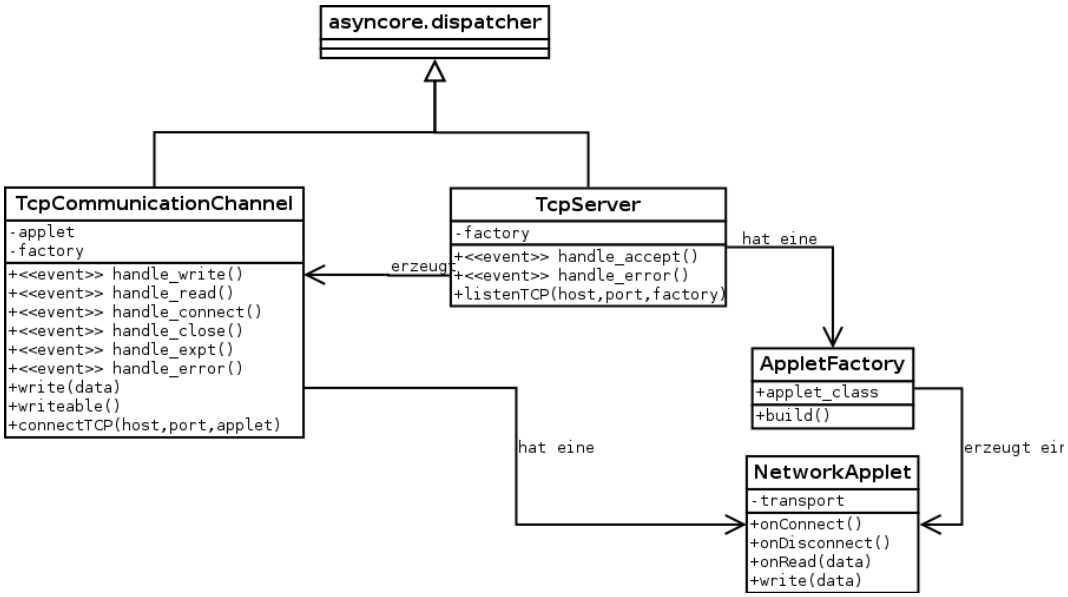


Abbildung 4.13: Basisgerüst für TCP-Programme in UML

asyncore.dispatcher stellt eine Erweiterung des normalen Socket Objektes in Python dar. Neben den normalen Socket-Funktionen (*send*, *recv*, *accept*, usw.) werden Callback-Funktionen definiert, die ausgeführt werden, sobald bestimmte Ereignisse eintreten (*handle_send*, *handle_recv*, *handle_accept*, usw.).

TcpCommunicationChannel implementiert die verschiedenen Callback-Funktionen. Für jede Verbindung existiert eine Instanz dieser Klasse.

TcpServer nimmt neue Verbindungen an und erzeugt eine neue Instanz der Klasse *TcpCommunicationChannel*.

AppletFactory stellt sicher, dass für jede neue Verbindung eine Instanz der Klasse *NetworkApplet* erzeugt wird.

NetworkApplet ist die Basisklasse für die konkreten Programme.

Protokollstack

Der Protokollstack fast die verschiedenen Protokolle zusammen. Es wird eine Schnittstelle definiert, mit der Nachrichten bearbeitet werden können.

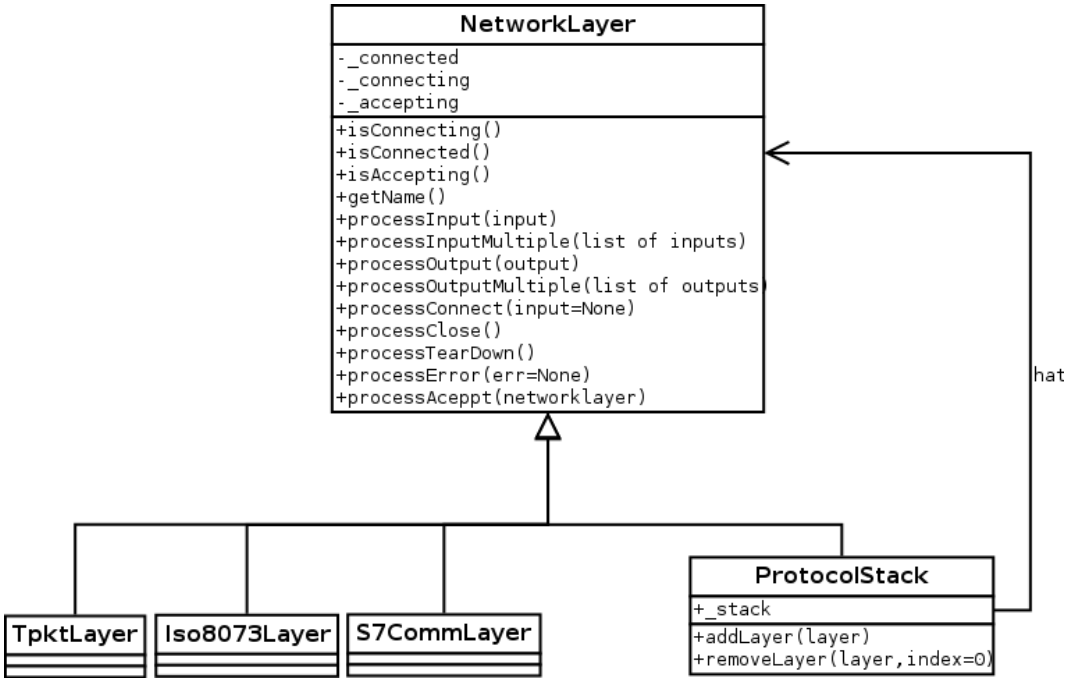


Abbildung 4.14: Protokollstack in UML

NetworkLayer definiert die einheitliche Schnittstelle.

ProtocolStack nimmt die einzelnen Protokolle auf und führt sie der Reihe nach aus.

TPKT, ISO8073, S7-Communication implementieren die Protokolle.

Protokollnachrichten

Die notwendigen Protokollnachrichten werden mit dem Python-Programm *Scapy* einheitlich beschrieben. Dieses Programm stellt ein Framework bereit, mit dem individuelle Nachrichten definiert werden können. Die Nachrichten können dann sehr leicht von Python-Objekten in einen Bytestrom verwandelt werden. Die Berechnung von ausgezeichneten Längenfeldern oder die Konvertierung zwischen Network Byte Order und Host Byte Order erledigt *Scapy* dabei automatisch. Genauso einfach kann ein Bytestrom wieder in ein Python-Objekt verwandelt werden. Neben der automatischen Berechnung können alle Werte auch manuell gesetzten werden. Dadurch besteht die Möglichkeit fehlerhafte Pakete zu erzeugen, mit denen andere Implementierungen auf Schwachstellen untersucht werden können.

```
1  class ParamPartItemDescription(ParamPart):
2
3      name = "ParamPartItemDescription"
4      fields_desc= [
5            XByteField("var_spec", 0x12),
6                      ByteField("length", 14),
7                      XByteField("syntax_id", 0xb2),
8                      XByteField("reserved", 0xFF),
9                      XIntField("area", 0),
10                     XIntField("crc",0),
11                     XIntField("lid", 0)
12                   ]
13
14 item = ParamPartItemDescription()
15 item.area = 0x12345
16 item.crc  = 0x12345
17 item.lid  = 0x12345
18
19 byte_stream = str(item)
20 python_object = ParamPartItemDescription(byte_stream)
```

Abbildung 4.15: Definition einer Nachricht mit Scapy & Beispiel für die Verwendung

s7-setvar

Das Programm *s7-setvar* verwendet den Protokollstack sowie das Basisgerüst für TCP-Programme, um eine Verbindung zu dem PLC aufbauen zu können. Nachdem die Verbindung aufgebaut ist, versendet das Programm eine Nachricht mit dem Befehl eine Variable zusetzen. Die neue Variable wird über die Kommandozeile angegeben.

s7-proxy

Der S7-Proxy muss so konzipiert werden, dass er sich auf der einen Seite wie ein PLC verhält. Dazu wartet er auf eingehende Verbindungen. Sobald ein Client eine neue S7-Communication Verbindung wünscht, baut der Proxy diese Verbindung mit dem Client auf. Auf der andern Seite muss sich der Proxy wie ein Client des PLC verhalten. Dazu baut er für jede eingebende Verbindung auch eine ausgehende Verbindung zum PLC auf. Aus diesem Grund besitzt der Proxy für jede Verbindung mit einem Client zwei Instanzen der Klasse *Protokollstack* sowie der Klasse *TcpCommunicationChannel*. Alle Anfragen und Antworten werden nun zur Gegenseite weitergeleitet.

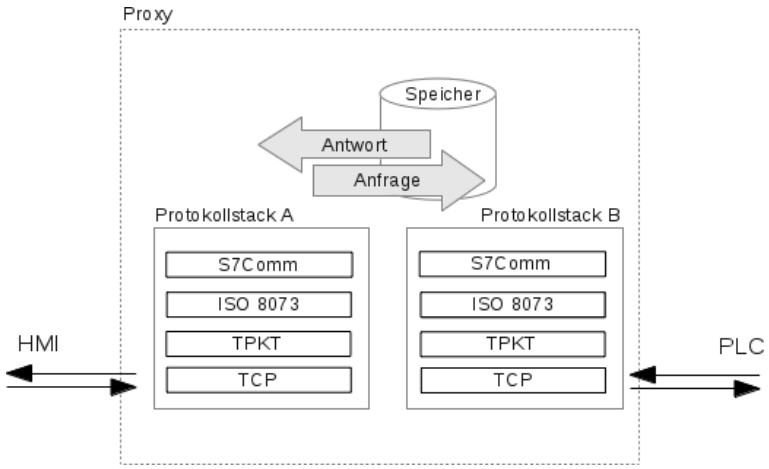


Abbildung 4.16: Schematische Darstellung des S7-Proxys

Sobald ein Lesezugriff auf eine Variable erkannt wird, speichert der Proxy den Namen sowie den Wert der Variablen ab. Zusätzlich wird der Zeitpunkt des Lesezugriffs gespeichert. Nach einer von dem Benutzer angegeben Zeit leitet der Proxy die Anfragen nicht mehr weiter, sondern beantwortet sie aus seinem Speicher. Ist die Variable nicht im Speicher enthalten, wird ein Zufallswert generiert.

dcp-discovery & dcp-setip

Die beiden Programme *dcp-discovery* und *dcp-setip* sind sich in ihrer Funktion sehr ähnlich. Beide senden ein DCP-Paket und werten die Antwort aus. Abbildung 4.17 zeigt den Ablaufplan der Programme.

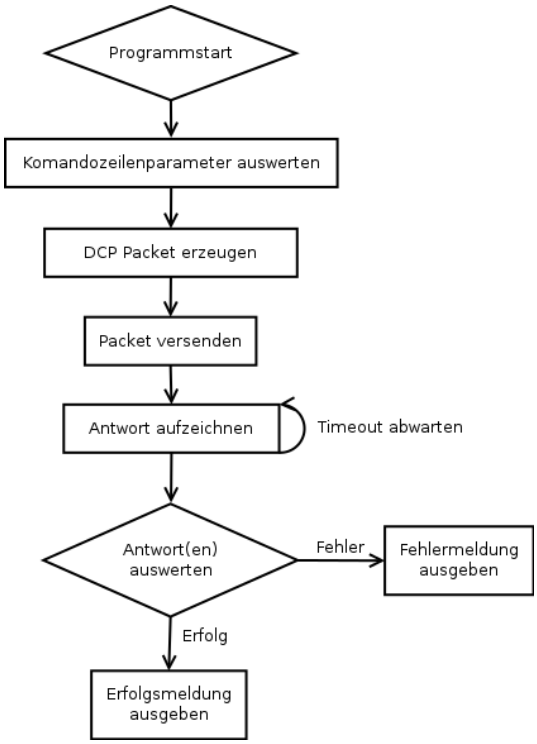


Abbildung 4.17: Protokollstack in UML

Das Programm *scapy* besitzt eine Funktion mit der Nachrichten direkt auf Schicht 2 gesendet und empfangen werden können. Von dieser Funktion machen die beiden Programme Gebrauch.

4.6.3 Test

Das Testen der Anwendung findet im Wechsel mit der Implementierung statt. Nachdem eine neue Unterfunktion implementiert ist, wird sie sofort getestet. Dieses geschieht durch einen Interoperabilitätstest mit den Zielsystemen. Die zu testende Unterfunktion wird ausgeführt. Verhält sich die Unterfunktion und das Zielsystem wie gewünscht, ist der Test erfolgreich abgeschlossen und die nächste Unterfunktion wird implementiert. Dadurch wird eine schnelle Entwicklung ermöglicht, weil auftretende Probleme frühzeitig erkannt und behoben werden können.

4.7 Versuchsdurchführung

Ziele identifizieren

Der simulierte Angriff beginnt damit, dass sich der Angreifer einen Überblick über alle Profinet-Geräte verschafft. Dazu wird das Programm *dcp-discovery* benutzt. Das Programm wird ausgeführt und liefert eine Liste mit zwei Geräten. Zu jedem Gerät werden eine Reihe von Details ausgegeben (siehe Abb. 4.18).

Die Details erlauben einen Rückschluss auf die verwendeten Geräte. Durch das Feld *Type of Station* kann durch eine Internet-Suche einfach festgestellt werden, dass es sich um ein HMI sowie einen PLC der Firma Siemens handelt.

Der Angreifer vermutet, dass das HMI und der PLC untereinander kommunizieren und versucht die Kommunikation zu analysieren.

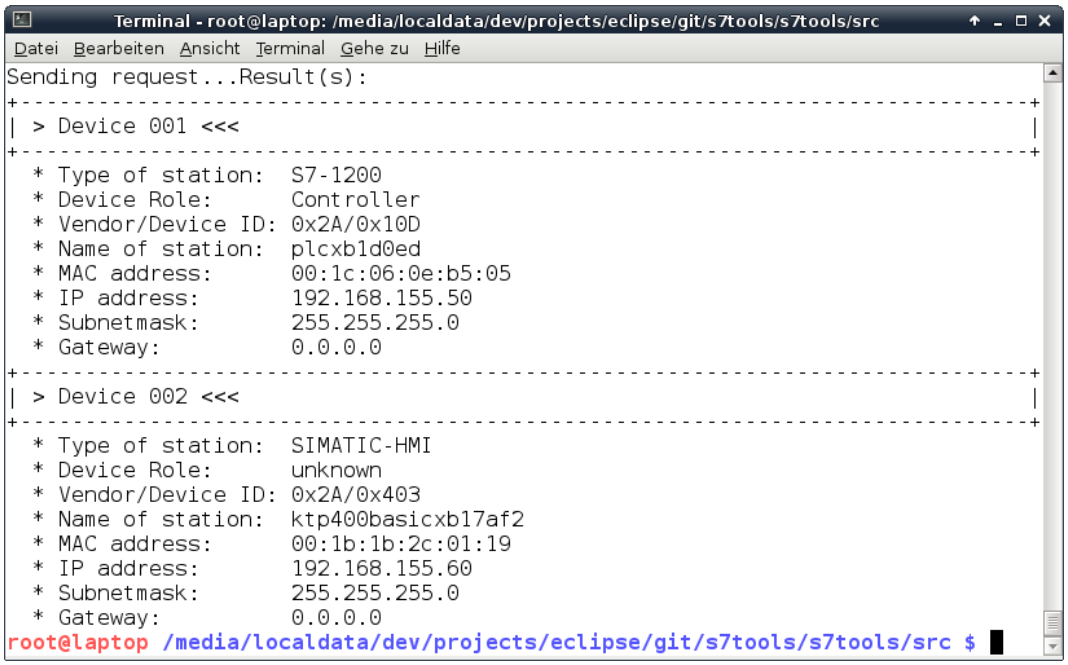


Abbildung 4.18: Ausgabe des Programmes *dcp-discovery*

Netzwerkverkehr umleiten

Damit der Angreifer den Netzwerkverkehr analysieren kann, versucht er die Kommunikation unter seine Kontrolle zu bringen. Um dieses Ziel zu erreichen wird dem PLC zunächst eine neue verfügbare IP-Adresse zugewiesen. Das Programm *dcp-setip* wird dafür benutzt. Es erwartet die MAC-Adresse des PLC und die neue IP-Adresse als Kommandozeilenparameter. Die MAC-Adresse des PLC ist bereits aus dem ersten Schritt bekannt. Eine neue freie IP-Adresse wird beliebig gewählt. Der PLC akzeptiert die neue IP-Konfiguration ohne vorherige Authentifizierung.

Der Angreifer kann nun seine Netzwerkkarte mit der alten IP-Adresse des PLC konfigurieren. Das HMI versucht daraufhin, die Verbindung mit dem Angreifer aufzubauen. Dieser nimmt die Verbindung mit Hilfe des Programmes *s7-proxy* an und baut seinerseits eine Verbindung zum PLC auf. Der Angreifer leitet alle Informationen zur Gegenseite weiter.

Netzwerkverkehr analysieren

Der *s7-proxy* zeichnet die ausgetauschten Variablen auf. Dadurch kann dem HMI im weiteren Verlauf nicht nur ein falsches Abbild der Realität vorgespielt werden, sondern der Angreifer kann die ausgetauschten Variablen analysieren. Er erhält Informationen über die Adresse und den Datentyp der Variablen. Gleichzeit lernt er typische Werte dieser Variable kennen. Die Informationen werden für den nächsten Schritt benötigt.

Nach einer bestimmten Zeit entschließt sich der Angreifer den PLC zu manipulieren. Dazu veranlasst der den *s7-proxy*, die Anfragen des HMI nicht mehr weiterzuleiten, sondern aus seinem Speicher zu beantworten. Dadurch versucht er seinen Angriff zu verschleiern.

Variablen verändern

Durch die Analyse des Netzwerkverkehrs zwischen PLC und HMI kennt der Angreifer eine Reihe von Variablen. Diese beeinflusst er mit dem Programm *s7-setvar*. Eine vorherige Authentifizierung ist nicht erforderlich. In diesem Beispiel ist für jedes Ampellicht eine Variable vorhanden. Gelingt es dem Angreifer die Variable den Lichtern zuzuordnen, kann er die Ampel gezielt steuern. In Abb. 4.19 ist dieser Fall dargestellt. Gleichzeitig zeigt das HMI ein normales Bild einer Ampel.

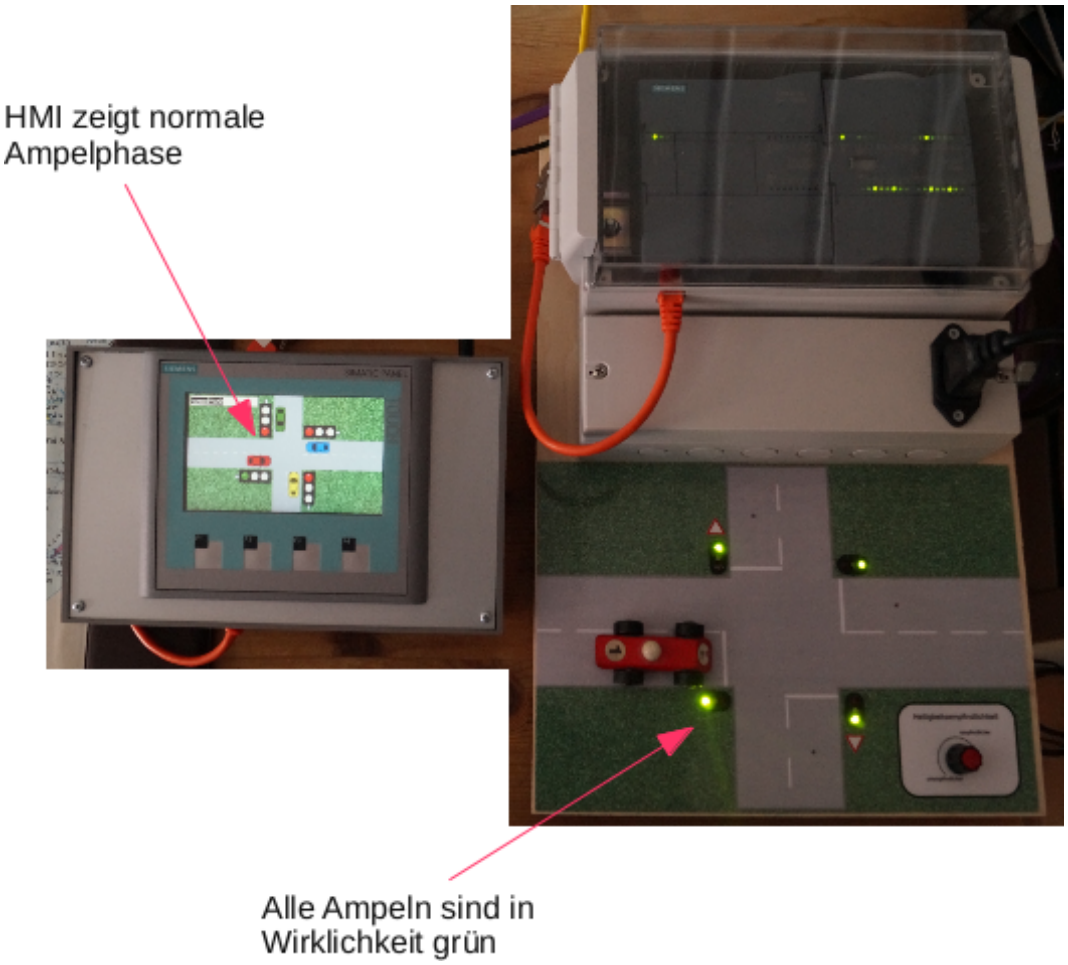


Abbildung 4.19: Ausgabe des Programmes *dcp-discovery*

4.7.1 Bewertung

Der erfolgreiche Versuch zeigt, wie einfach in den Ablauf der Beispielanwendung eingriffen werden kann. Es ist nicht nötig, Sicherheitsmaßnahmen zu überwinden. Das Setzen von Konfigurationswerten wie der IP-Adresse oder das Schreiben von Prozessvariablen ist ohne vorherige Authentifizierung möglich. In einer realen Anwendung könnte ein Angreifer auf diese Weise erheblichen Schaden anrichten. In klassischen IT-Systemen würden derartige Eingriffe als schwerwiegende Sicherheitslücke bezeichnet werden. Es wird deutlich, dass die Systeme nicht ohne weitere Sicherheitsmaßnahmen betrieben werden können.

5 Entwurf eines Basissicherheitskonzeptes

5.1 Segmentierung des Netzwerkes

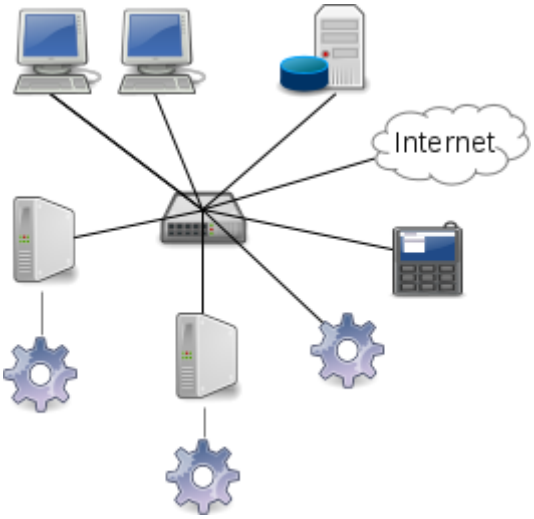
Die Grundidee des Sicherheitskonzeptes ist es, das Netzwerk zu segmentieren. Hierbei erfolgt eine horizontale Trennung entlang der verschiedenen Bereiche eines ICS (Office SCADA, PCS). Aber auch vertikal innerhalb eines Bereiches. Durch die Trennung können klare Grenzen festgelegt werden, die den Zugriff beschränken können. Gelingt es einem Angreifer einzudringen, ist nicht das ganze Netzwerk exponiert, sondern im Optimalfall nur ein kleiner unkritischer Bereich.

5.1.1 Identifizierung von funktionalen Gruppen

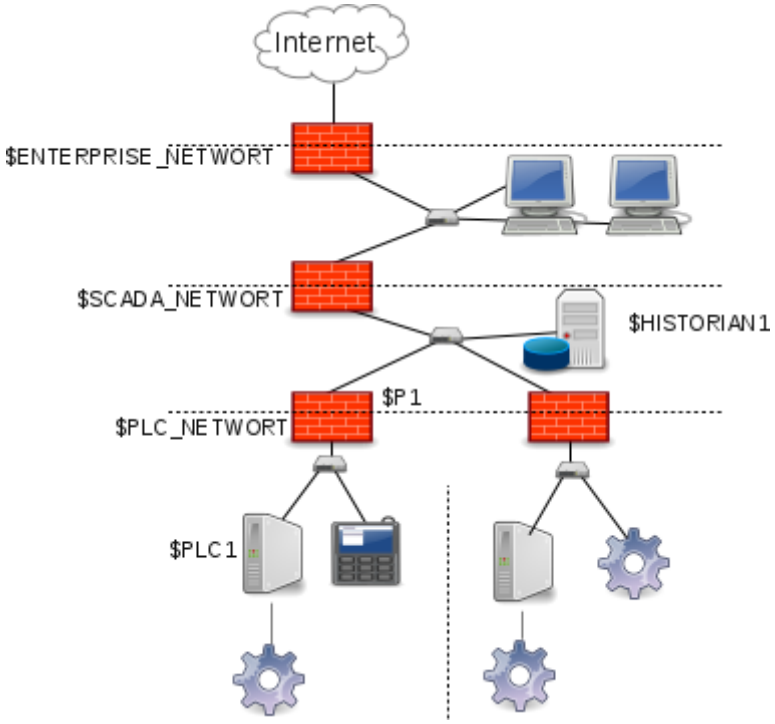
Um ein Netzwerk zu segmentieren, müssen zunächst alle Komponenten erfasst werden. Danach können sie in funktionale Gruppen eingeteilt werden. Unter einer funktionalen Gruppe versteht man einen Verbund von Komponenten, auf die ein bestimmtes Kriterium zutrifft. Dies können alle Komponenten sein, die über ein bestimmtes Protokoll kommunizieren. Weiterhin können es auch Komponenten sein, auf die ein bestimmter Benutzer Zugriff hat. Welche Kriterien genau verwendet werden, kann sehr unterschiedlich sein. Je nach Netzwerk können sich so eine ganze Reihe von Kriterien ergeben. Im Ergebnis erhält man eine bestimmte Anzahl von funktionalen Gruppen, die sich auch teilweise überlappen können. Ist das Netzwerk sehr komplex, kann es erforderlich sein, die funktionalen Gruppen sinnvoll zusammenzufassen. Für jede Gruppe können wir bestimmte Sicherheitsregeln ableiten. Die funktionalen Gruppen werden genau dokumentiert und bilden die Grundlage für das weitere Vorgehen [17, Seite 148ff.].

5.1.2 Netzwerkstruktur anpassen

Nachdem das Netzwerk mit Hilfe der funktionalen Gruppen logisch segmentiert wurde, kann im nächsten Schritt die physikalische Netzwerkstruktur angepasst werden. Das Ziel hierbei ist es, klare Perimeter definieren zu können. Erst durch diesen Schritt wird ein effektiver Schutz der Komponenten ermöglicht. Es liegt auf der Hand, dass nicht alle funktionalen Gruppen gleichzeitig in eine physikalische Netzwerkstruktur überführt werden können. Daher bietet es sich an, die Komponenten, die viel miteinander kommunizieren, physikalisch direkt zu vernetzen. An den Übergängen der verschiedenen Ebenen eines ICS werden die Perimeter installiert. Die Abb. 5.1 verdeutlicht diesen Vorgang. Auf der linken Seite der Grafik ist die Netzwerkstruktur denkbar einfach. Auf der rechten Seite ist jedoch eine Segmentierung erkennbar. Die verschiedenen Ebenen sind getrennt. Auch innerhalb einer Ebene wird eine Trennung vorgenommen. In diesem Beispiel müssen nicht alle Komponenten der untersten Ebene direkt mit einander kommunizieren und sind daher nicht direkt physikalisch vernetzt [17, Seite 161ff.].



(a) Keine Segmentierung



(b) Segmentierung mit Perimetern

Abbildung 5.1: Netzwerk vor und nach der Segmentierung.

5.1.3 Perimeter

Die Aufgabe der eingeführten Perimeter ist es nun, die verschiedenen Sicherheitsregeln, die sich aus den funktionalen Gruppen ableiten, bestmöglich umzusetzen. Dazu wird der gesamte Datenverkehr, der durch einen Perimeter läuft analysiert und entweder verworfen oder zugelassen.

Perimeter mit OpenSource

Ein Perimeter kann mit Hilfe von OpenSource-Software realisiert werden. Programme wie *Netfilter* oder *Pf* eignen sich für diese Aufgabe. Der „Good Practice Guide on Firewall Deployment for SCADA and Process Control Networks“ [18] gibt einige grundsätzliche Regeln an, die folgenden zusammengefasst werden:

Regel	Beschreibung
1	Grundsätzlich werden alle Pakete verworfen. Dieses schließt insbesondere Protokolle kleiner OSI Schicht drei ein.
2	Alle Firewall-Regeln müssen sich auf spezifische Ports oder IP-Adressen beziehen
3	Protokolle die zwischen Office-Bereich und dem SCADA-Bereich erlaubt sind, werden expliziert zwischen dem SCADA-Bereich und dem PCS-Bereich verboten (und umgekehrt).
4	Neben eingehenden Netzwerkverkehr wird auch der ausgehende Netzwerkverkehr nur erlaubt wenn er benötigt wird.
5	Der Management-Netzwerkverkehr für die Perimeter muss in einem separaten Netz übermittelt werden

Tabelle 5.1: Grundsätzliche Firewall-Regel für einen Perimeter

Grenzen von OpenSource

Netfilter und *Pf* arbeiten auf der Vermittlungsschicht sowie auf der Transportschicht (OSI-Layer 3 und 4). Damit ist es nicht möglich die höheren Sichten zu analysieren. OpenSource Firewall-Systeme, die auf der Anwendungsschicht arbeiten und typische ICS-Protokolle filtern können, sind zur Zeit nicht vorhanden. Das zeigt die Grenzen von OpenSource als Perimeterfirewall auf. Die Kommunikation kann eingeschränkt werden. Es ist aber nicht möglich, detailliert auf die Inhalte der Kommunikation einzugehen. Sinnvoll wäre, wenn man Lesezugriffe auf PLCs ermöglichen kann, Schreibzugriffe jedoch am Perimeter abzufangen wären.

6 Fazit

Sicherheit im ICS-Umfeld ist den letzten Jahren wichtiger geworden. Dieser Trend ist positiv. Trotzdem sind ICS-Systeme zur Zeit nicht auf einem ausreichenden Sicherheitsniveau.

Die Machbarkeitsuntersuchung am Beispiel der S7-1212C zeigt deutlich wie ohne vorherige Authentifizierung die physikalischen Ausgänge des PLC verändert werden können. In einer realen Umgebung kann auf diese Weise eine Maschine direkt beeinflusst werden. Die Kommunikation zwischen dem PLC und HMI kann gezielt umgeleitet und verändert werden. Für keine dieser Eingriffe muss eine Sicherheitsmaßnahme überwunden werden. Ein sicherer Einsatz dieser Geräte ist dadurch nur schwierig möglich.

Diese Probleme können nur die Hersteller der ICS-Systeme lösen. Sie sind in der Verantwortung, Produkte zu entwickeln, die sicher sind. Besonders wenn sie in kritischen Infrastrukturen eingesetzt werden. Bei der Entwicklung neuer Geräte sollte dieses Thema von Anfang an einen hohen Stellenwert besitzen. Geräte dürfen in Zukunft keine fest einprogrammierten Passwörter mehr verwenden oder Schwächen bei der Eingabevalidierung aufweisen. Auch die Kommunikationsprotokolle bedürfen einer grundlegenden Überarbeitung. Es sollten Funktionen eingebaut werden, die die Authentizität und die Integrität der Kommunikation sicherstellen können.

Aber auch die Betreiber können mehr für die Sicherheit ihrer Systeme leisten. Sie können Druck auf die Hersteller ausüben und sichere Systeme einfordern. Gleichzeitig müssen sie dann aber auch die neuen Komponenten in ihre Infrastruktur integrieren und Sicherheitsprozesse etablieren. Auch die richtige Vernetzung der Komponenten liegt in der Verantwortung der Betreiber. ICS-Komponenten sollten in Zukunft nicht mehr direkt mit dem Internet verbunden sein. Eine Segmentierung der Netzwerke sollte immer erfolgen.

Zuletzt können Sicherheitsforscher daran arbeiten, weitere Sicherheitsprobleme aufzudecken. Dies sollte auf eine verantwortungsvolle Art und Weise geschehen. Full Disclosure ist sicherlich nicht der richtige Weg. Der potentielle Schaden, der angerichtet werden kann, ist groß. Dennoch sollten die Sicherheitsforscher den Druck auf die Hersteller aufrecht erhalten.

Die IT-Sicherheit von ICSs kann erheblich verbessert werden. Die Angreifer werden jedoch nicht auf diesen Prozess warten, sondern die Systeme kompromittieren. Daher müssen alle Beteiligten sich jetzt ihrer Verantwortung stellen.

Literatur

[1] Bundesamt für Sicherheit in der Informationstechnik. *Schutz Kritischer Infrastrukturen*. https://www.bsi.bund.de/DE/Themen/KritischeInfrastrukturen/kritischeinfrastrukturen_node.htm.

[2] Sean McBride. *Documenting the Lost Decade. An Empirical Analysis of publicly disclosed ICS vulnerabilities since 2001*. http://vimeo.com/user10193115/review/35801119/2ed0598ff4.

[3] National Institute of Standards und Technology. *Maroochy-Water-Services case study report*. http://csrc.nist.gov/groups/SMA/fisma/ics/documents/Maroochy-Water-Services-Case-Study_report.pdf.

[4] Kevin Poulsen. *Slammer worm crashed Ohio nuke plant net*. http://www.theregister.co.uk/2003/08/20/slammer_worm_crashed_ohio_nuke.

[5] Jeanne Meserve. *Sources: Staged cyber attack reveals vulnerability in power grid*. http://edition.cnn.com/2007/US/09/26/power.at.risk.

[6] Graeme Bakere. *Schoolboy hacks into city's tram system*. http://www.telegraph.co.uk/news/worldnews/1575293/Schoolboy-hacks-into-citys-tram-system.html.

[7] Frank Rieger. *Der digitale Erstschlag ist erfolgt*. http://www.faz.net/aktuell/feuilleton/debatten/digitales-denken/trojaner-stuxnet-der-digitale-erstschlag-ist-erfolgt-1578889.html.

[8] Kyle Wilhoit. *Who's Really Attacking Your ICS Equipment?* http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-whos-really-attacking-your-ics-equipment.pdf.

[9] Bundesamt für Sicherheit in der Informationstechnik. *BSI IT-Grundschutz-Kataloge 12. EL*. 2011. ISBN: 3-88784-915-9.

[10] Tyson Macaulay; Bryan Singer. *Cybersecurity for Industrial Control Systems*. 2012. ISBN: 978-1-4398-0196-3.

[11] Trent Nelso; May Chaffin. *Common Cybersecurity Vulnerabilities in Industrial Control Systems*. http://ics-cert.us-cert.gov/sites/default/files/documents/DHS_Common_Cybersecurity_Vulnerabilities_ICS_2010.pdf.

[12] Siemens AG. *Siemens Easy Book*. Odernumber: A5E02486775-05. 2012.

[13] International Electrotechnical Commission. *IEC61158-6-10; IEC61158-5-10; IEC61784-2*. http://webstore.iec.ch.

[14] Thomas W. *S7comm Wireshark dissector*. http://sourceforge.net/projects/s7commwireshark/.

[15] ISO. *International Standard 8073*. http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=24077. 1997.

[16] M. Rose; D. Cass. *ISO Transport Service on top of the TCP*. http://tools.ietf.org/html/rfc1006. 1987.

[17] Eric D. Knapp. *Industrial Network Security*. 2011. ISBN: 978-1-59749-645-2.

[18] Eric Byres; John Karsch; Joel Carter. *FIREWALL DEPLOYMENT FOR SCADA AND PROCESS CONTROL NETWORKS*. http://energy.gov/sites/prod/files/Good%20Practices%20Guide%20for%20Firewall%20Deployment.pdf.

A Anhang

Akronyme

BSI Bundesamt für Sicherheit in der Informationstechnik

DCP Discovery and basic Configuration Protocol

DCS Distributed Control System

HMI Human Machine Interface

ICS Industiral Control System

IED Intelligent Electronic Device

KRITIS kritische Infrastrukturen

PCS Process Control System

PLC Programmable Logic Controller

RTU Remote Terminal Unit

SCADA Supervisory Control and Data Acquisition

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.

Ort, Datum Unterschrift