Methods for Identification and Classification of Industrial Control Systems in IP Networks

Tuomas Järekallio

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 3.8.2016

Thesis supervisor:

Prof. Jukka Manner

Thesis advisor:

M.Sc. (Tech.) Timo Kiravuo

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Tuomas Järekallio

Title: Methods for Identification and Classification of Industrial Control Systems in IP Networks

Date: 3.8.2016 Language: English Number of pages: 7+55

Department of Communications and Networking

Professorship: Network Technology

Supervisor: Prof. Jukka Manner

Advisor: M.Sc. (Tech.) Timo Kiravuo

Industrial Control Systems (ICS) are an essential part of the critical infrastructure of society and becoming increasingly vulnerable to cyber attacks performed over computer networks. The introduction of remote access connections combined with mistakes in automation system configurations expose ICSs to attacks coming from public Internet. Insufficient IT security policies and weaknesses in security features of automation systems increase the risk of a successful cyber attack considerably. In recent years the amount of observed cyber attacks has been on constant rise, signaling the need of new methods for finding and protecting vulnerable automation systems. So far, search engines for Internet connected devices, such as Shodan, have been a great asset in mapping the scale of the problem.

In this theses methods are presented to identify and classify industrial control systems over IP based networking protocols. A great portion of protocols used in automation networks contain specific diagnostic requests for pulling identification information from a device. Port scanning methods combined with more elaborate service scan probes can be used to extract identifying data fields from an automation device. Also, a model for automated finding and reporting of vulnerable ICS devices is presented. A prototype software was created and tested with real ICS devices to demonstrate the viability of the model. The target set was gathered from Finnish devices directly connected to the public Internet. Initial results were promising as devices or systems were identified at 99% success ratio. A specially crafted identification ruleset and detection database was compiled to work with the prototype. However, a more comprehensive detection library of ICS device types is needed before the prototype is ready to be used in different environments. Also, other features which help to further assess the device purpose and system criticality would be some key improvements for the future versions of the prototype.

Keywords: cyber security, industrial control systems, identification, classification, protection

AALTO-YLIOPISTO
SÄHKÖTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Tuomas Järekallio

Työn nimi: Teollisuusautomaatiojärjestelmien tunnistus ja luokittelu IP-verkoissa

Päivämäärä: 3.8.2016 Kieli: Englanti Sivumäärä: 7+55

Tietoliikenne- ja tietoverkkotekniikan laitos

Professuuri: Tietoverkkotekniikka

Työn valvoja: Prof. Jukka Manner

Työn ohjaaja: DI Timo Kiravuo

Yhteiskunnan kriittiseen infrastruktuuriin kuuluvat teollisuusautomaatiojärjestelmät ovat yhä enemmissä määrin alttiita tietoverkkojen kautta tapahtuville kyberhyökkäyksille. Etähallintayhteyksien yleistyminen ja virheet järjestelmien konfiguraatioissa mahdollistavat hyökkäykset jopa suoraa Internetistä käsin. Puutteelliset tietoturvakäytännöt ja teollisuusautomaatiojärjestelmien heikot suojaukset lisäävät onnistuneen kyberhyökkäyksen riskiä huomattavasti. Viime vuosina kyberhyökkäysten määrä maailmalla on ollut jatkuvassa kasvussa ja siksi tarve uusille menetelmille haavoittuvaisten järjestelmien löytämiseksi ja suojaamiseksi on olemassa. Internetiin kytkeytyneiden laitteiden hakukoneet, kuten Shodan, ovat olleet suurena apuna ongelman laajuuden kartoittamisessa.

Tässä työssä esitellään menetelmiä teollisuusautomaatiojärjestelmien tunnistamiseksi ja luokittelemiseksi käyttäen IP-pohjaisia tietoliikenneprotokollia. Suuri osa automaatioverkoissa käytetyistä protokollista sisältää erityisiä diagnostiikkakutsuja laitteen tunnistetietojen selvittämiseksi. Porttiskannauksella ja tarkemmalla palvelukohtaisella skannauksella laitteesta voidaan saada yksilöivää tunnistetietoa. Työssä esitellään myös malli automaattiselle haavoittuvaisten teollisuusautomaatiojärjestelmien löytämiselle ja raportoimiselle. Mallin tueksi esitellään ohjelmistoprototyyppi, jolla mallin toimivuutta testattiin käyttäen testijoukkona oikeita Suomesta löytyviä, julkiseen Internetiin kytkeytyneitä teollisuusautomaatiolaitteita. Prototyypin alustavat tulokset olivat lupaavia: laitteille tai järjestelmille kyettiin antamaan jokin tunniste 99% tapauksista käyttäen luokittelussa apuna prototyypille luotua tunnistekirjastoa. Ohjelmiston yleisempi käyttö vaatii kuitenkin kattavamman automaatiolaitteiden tunnistekirjaston luomista sekä prototyypin jatkokehitystä: tehokkaampi tunnistaminen edellyttää automaatiojärjestelmien toimintaympäristön ja kriittisyyden tarkempaa analysointia.

Avainsanat: tietoturva, teollisuusautomaatio, tunnistaminen, luokittelu, suojautuminen

Preface

This thesis work was created as a part of SAICS (Situation Awareness in Informatics and Cyber Security) national research project in the quest for protecting critical infrastructure and improving national cyber security. I appreciate the opportunity to work for Comnet as a part of this research endeavor. I would like to thank my instructor Timo Kiravuo for his advice and support and my co-workers for support and inspiration during my time in the Comnet. Special thanks to Jukka Manner for great supervision and advice which helped to guide me through the thesis writing process. Especially, I would like to thank all my family and friends for supporting me and believing in me throughout my studies.

Otaniemi, 3.8.2016

Tuomas Järekallio

Contents

Abbreviations

ADU	Application Data Unit
BBMD	BACnet Broadcast Management Device
BDT	Broadcast Distribution Table
CA	Certificate Authority
CERT	Cyber Emergency Response Team
CIP	Common Industrial Protocol
CPU	Central Processing Unit
COTP	Connection-Oriented Transport Protocol
DNS	Domain Name System
FDT	Foreign Device Table
FTP	File Transfer Protocol
ICS	Industrial Control System
IoT	Internet of Things
IP	Internet Protocol
ISP	Internet Service Provider
ISO-TSAP	ISO Transport Service Access Point
HMI	Human-Machine Interface
HTTP	Hyper Text Transfer Protocol
MIB	Management Information Base
NIST	National Institute for Standards and Technologies
NSE	Nmap Scripting Engine
OID	Object Identifier
OS	Operating System
PDU	Protocol Data Unit
PKI	Public Key Infrastructure
PLC	Programmable Logic Controller
RIR	Regional Internet Registry
RTU	Remote Terminal Unit
SCADA	Supervisory Control And Data Acquisition
SNMP	Simple Network Management Protocol
SSH	Secure Shell
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VPN	Virtual Private Network

1 Introduction

Modern highly developed societies depend heavily on information technology and automation systems for their every day functions. These functions range from industrial manufacturing all the way to civil infrastructure containing power plants and water treatment facilities. Information systems improve productivity and enable the use of flexible remote access into management systems. However, continually evolving networked information systems make societies more vulnerable to potential cyber attacks.

Cyber attacks against networked automation systems has in recent years raised the public awareness of the need for cyber security. This has been one of the key topics while forming national security policies. In 2013, the Finnish government formed a national cyber strategy [1] as an overall vision for protecting nationally vital key functions against cyber threats. It specifies guidelines for seamless co-operation between companies, governmental agencies and researchers to form situational awareness in information networks and to detect, and counteract, cyber attacks.

Globally networked world enables new ways to remotely access automation systems but it also makes them more vulnerable to malicious actors. These actors each have their own agenda to support their interest which may be financial gain by criminal activity, terrorism fueled by ideology or national security agencies practicing cyber espionage. Major sections of nationally critical infrastructure rely on the stable operation of automation systems. A wide scale cyber attack against critical equipment, e.g. power grid, could cause widespread chaos and paralyze society's ability to function. Therefore it is crucial to improve protection for this infrastructure to the highest level.

Industrial Control System (ICS) is a general term evolved to cover all types of control systems used in industrial automation installations. This includes Supervisory Control And Data Acquisition systems (SCADA) and process control systems which utilize Programmable Logic Controls (PLC) and Remote Terminal Units (RTU). ICSs can expand over multiple operational sites by using Distributed Control Systems (DCS). DCS provides a multi-level control architecture containing a supervisory control level which manages multiple sub-level supervisory and process control functions.

SCADA systems monitor field sites by getting process status information and controlling automation systems in the field level. Control is typically achieved by managing feed-back or feed-forward control loops which maintain process conditions around predefined standards. On the field level, PLCs are used to control the physical industrial process by regulating actuators, e.g. valves, arms or pumps. PLCs are usually complemented with RTUs which are used to gather process status information from sensors and pass it back to SCADA system for process monitoring. Human interaction is made possible with Human-Machine Interface (HMI) input-output devices which provide process data to human operators and enable operators to control the process. HMI systems typically rely on SCADA system databases and software to present detailed diagnostic and management views over the industrial process. This enables operators to fine tune the industrial process wherever needed.

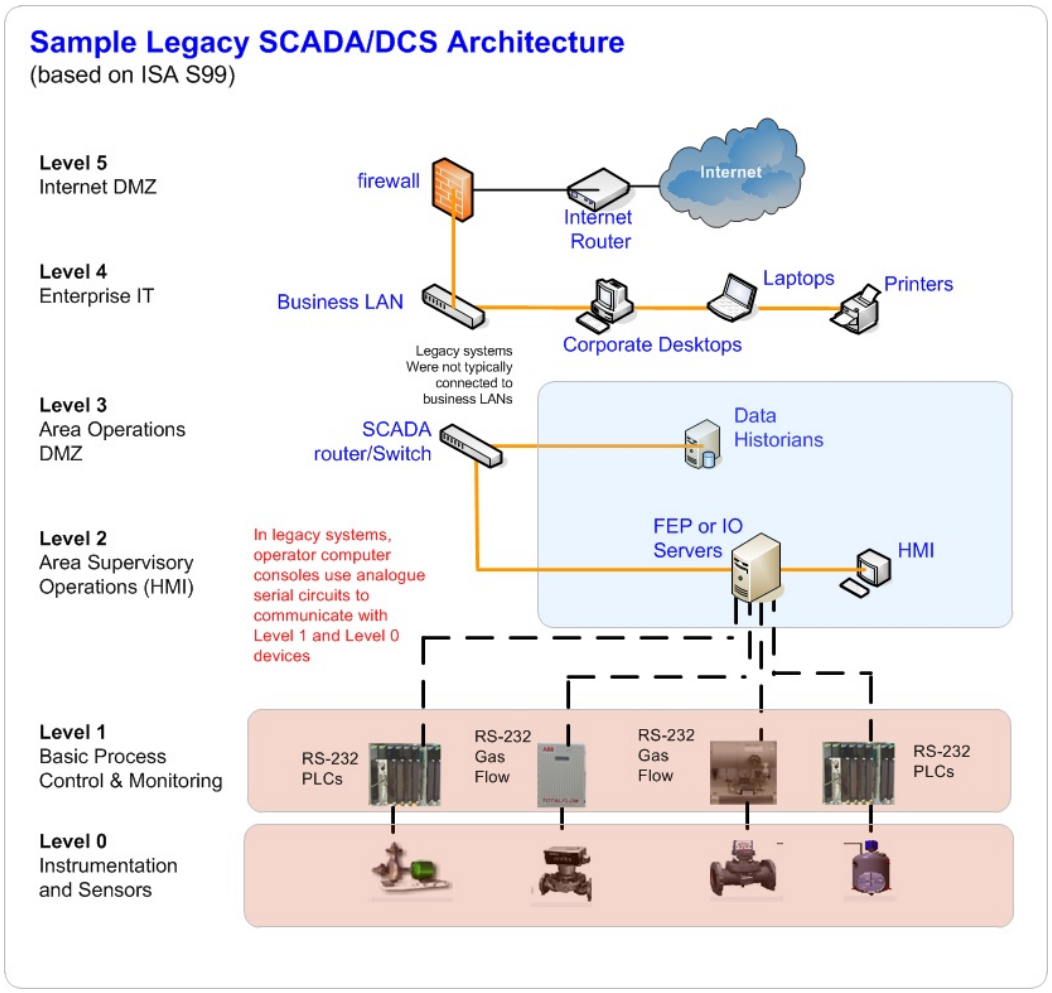

Figure 1: A traditional industrial control system architecture [2]

Figure 1 illustrates a traditional ICS architecture following ISA S99 security model [2]. This sample system contains only the essential parts of a SCADA/DCS architecture and is isolated from corporate IT network and Internet with firewalls. System architecture is divided into several separated security levels. Level 0 contains the physical instrumentation, e.g. valves and pumps, and sensors of industrial process. They operate the physical machinery needed in industrial process and are directly connected to field-level controllers and monitoring equipment, e.g. PLCs and RTUs, which reside in level 1.

PLCs and RTUs are managed through a network connection using a Front End Processor (FEP) as a gateway. Modern PLCs have TCP/IP stacks implemented but legacy automation systems still use serial connections to communicate. Therefore FEP could be needed to establish connectivity between packet based IP networks and serial connected devices. Together with HMI terminals they provide area supervisory operations for process control network in local control room environments (level 2).

Remote monitor and control functions are established in Area Operations DMZ (level 3) which is connected to the process control networks through a firewall.

SCADA systems provide high-level supervisory control over several process control networks in multiple geographical regions. Database and historian servers are used to collect and store monitored process data. SCADA software is typically also used to automatically flow data to business applications which access area operations network from corporate IT network (level 4). Corporate intranet can in most cases be remotely accessed through a VPN connection from public Internet (level 5). Appropriate security measures, e.g. firewalls, routers and security software, must be in place to ensure sufficient isolation between corporate IT network and area operations network.

This thesis studies mainstream ICS IP network protocols in the cyber security point-of-view and presents how they could be used to gather identity information from exposed ICS devices. The feasibility of an automated system for gathering identification fingerprints from ICS devices connected to IP networks is researched. Moreover, this thesis also discusses whether these fingerprints are enough to provide a sufficient dataset for reliable ICS device identification. A proof of concept network scanner and device classifier was built to demonstrate the overall concept in practice. 41 search terms were used to gather exposed ICS devices from Shodan search engine. Only Finnish IP addresses were collected for scan targets and safe scanning techniques were utilized to ensure compliance with the Finnish legislation. The prototype collected successfully usable identification data using multiple TCP and UDP protocols. Moreover, less than a hundred identification rules were enough to classify target devices with 99% accuracy when compared to Shodan database. More aggressive device enumeration could improve accuracy even more; however, the legality of such methods remain unclear as intrusive examination of computer systems is strictly forbidden without the consent of the owner.

In the next chapter, the most common security threats to ICSs are being explored. Reasons for common ICS security weaknesses are being analyzed while highlighting a few possible scenarios how malicious actors can utilize them. Chapter 3 focuses on methods for gathering identifying fingerprints from ICS devices. Several mainstream ICS network protocols are being analyzed for their usage for device identification. Also, the feasibility of IP traffic fingerprinting in device detection is considered. Chapter 4 describes a reference model for ICS device classification based on previously gathered device fingerprints. This thesis focuses on rule-based classification but machine learning is also briefly viewed. Chapter 5 presents a proof of concept prototype for scanning IP based networks and finding exposed ICS devices. Initial testing was carried out against ICS simulator honeypots and then with devices in real network environment. Finally, Chapter 6 presents conclusions and ideas for future work.

2 Cyber security issues in ICS

Industrial Control Systems (ICS) have been for decades an invaluable part of modern society, managing a large portion of core functions in critical infrastructure. ICSs can be found in most civil infrastructure, e.g. power plants, power grid, water, oil and gas distribution, building automation as well as industrial manufacturing systems. Historically, ICS architectures were not designed with security in mind because of physical isolation from public networks and the use of proprietary software with vendor specific serial protocols. Physical isolation and “security by obscurity" design pattern were enough to keep ICSs protected from cyber attacks. However, ICSs have not seen significant changes in architecture and protocols until very recently when commercial off-the-shelf ICS solutions have become more commonplace in industrial installations and IP based packet networks have been embedded within these previously isolated systems to simplify distributed remote control. Improved networking capabilities have also raised concern for cyber security.

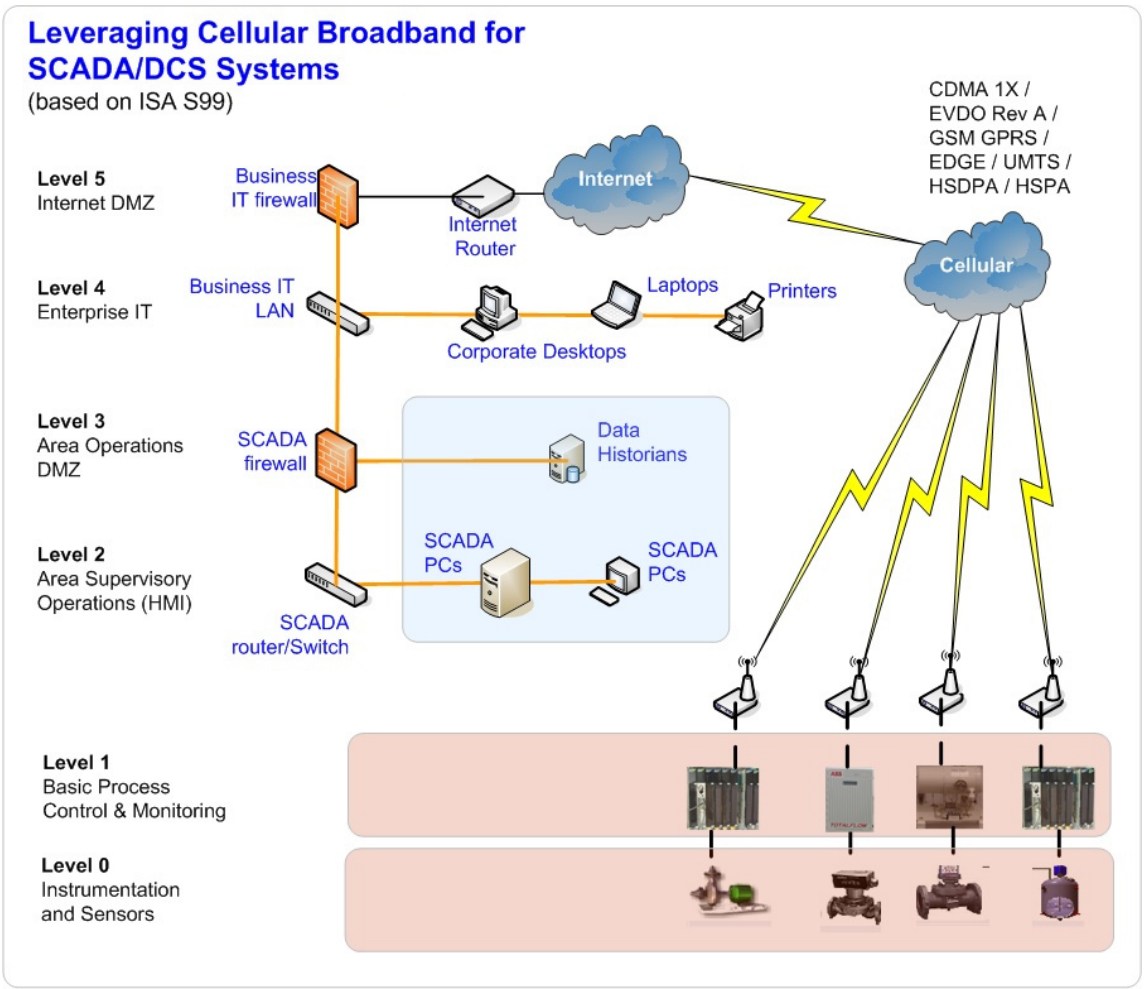

Figure 2: An industrial control system over IP based mobile broadband service [2]

Figure 2 illustrates a modern cellular ICS architecture where connections between

SCADA systems and field site are established over mobile IP network. While bringing flexibility and ease of deployment, routable TCP/IP communication exposes previously isolated field sites to cyber threats. With a traditional serial connection, physical access is required to gain control over the circuit. With IP networks this risk is increased because packets can be routed from other networks without physical access to the field site network. Process control networks still use fieldbus communication protocols; however, remote access is provided by gateways carrying data over IP network. Modern PLC, RTU and HMI devices have nowadays native IP protocol support allowing direct remote management over e.g. Telnet or SSH protocols. Improved accessibility makes these devices vulnerable to cyber threats if they are not properly protected. Firewalls can be helpful in securing field site from direct attacks from the Internet. However, more advanced cyber attacks can be indirect and the point of entry can be e.g. from inside corporate network. Therefore it is crucial to secure all interfaces where remote control or monitoring connection can be established.

This chapter explores the common cyber security issues in industrial control systems. A selection of reported severe ICS incidents are presented to illustrate the magnitude of destruction a well-placed cyber attack can cause. Systematic problems, both technical and organizational, in networked automation systems are discussed and the chapter is concluded by a look into the current exposure of ICS devices on the Internet.

2.1 ICS incidents

Security incidents in industrial control systems are a significant problem. Companies tend to suppress incident reports in fear of repercussions from governmental regulators and businesses considered insecure can also lose a significant portion of their customer base. A recent security study of 291 utility and energy companies in the U.S.A. [3] revealed that 76% of the companies had suffered at least one data breach during the previous 12 months. 21% of the companies had two or more security incidents. Moreover, 55% of respondents valued minimizing system downtime to be the top IT security objective whereas only 14% considered prevention of a cyber attack to be a high priority. This is natural because automation system processes are expected to be reliable and therefore availability is valued over data confidentiality. Meeting regulatory requirements while protecting physical security, including personnel and equipment, are considered more important than keeping up with data security. According to the study the two top core systems compromised were databases (according to 56% of respondents) and endpoints (according to 52% of respondents) whereas only 5% of respondents said SCADA networks were compromised as a results of a cyber attack. Cyber attacks are not in most cases crafted against SCADA systems but the potential of a serious disaster is real and should be taken into consideration in IT security policies.

In the past, SCADA system malfunctions have been a cause for large scale damages and even deaths. One of the most well-known ICS incidents, the Chernobyl nuclear power plant core meltdown in 1986 [4], has long-lasting effects for centuries.

Less than 50 deaths has been directly attributed to the radiation from the explosion but continuous radioactive fallout can be responsible for up to 4000 deaths. While the accident was caused by operator errors and unsafe testing procedures of the nuclear reactor, similar accidents are possible as a result of a cyber attack. There has been several ICS incident reports where nuclear power plants have been infected with computer viruses. In 2013, a malicious virus caused a shutdown of a U.S. power plant [5] and in 2016 a German nuclear power plant [6] was hit by several viruses. In both cases, careless handling of compromised USB drives caused computer systems to become infected. Luckily, physical isolation from the Internet prevented data theft and systems directly in control of the reactors were not compromised.

To this day, there are only a few publicly known examples of cyber attacks in which a malicious actor has caused large physical damage to industrial control systems:

- One of the most well-known and sophisticated attack was the Stuxnet computer worm sabotaging an Iranian uranium enrichment facility at Natanz. It exploited multiple zero-day vulnerabilities in Windows and Siemens Step7 software and was specifically targeting PLCs and SCADA systems. When introduced to the target system, usually via an infected USB drive, it installed rootkits to PLC and Step7 control software and used modified code to inject unexpected and harmful commands to the process network through PLC interfaces. In Iran it succeeded in destroying one-fifth of all Iran's nuclear centrifuges. The origins of Stuxnet has not been traced but several intelligence leaks implicate United States and Israel in developing it as a tool for cyber warfare purposes. Later diplomatic cables obtained by Wikileaks confirmed the intentions of targeting Iran's nuclear program through sophisticated covert operations. [7, 8]

- In December 2014, the German federal office of information security released a report stating that a malicious actor had gained access to control systems of a steel mill, impacting multiple process components in ICS network resulting them to become unregulated and cause massive physical damage in the plant. No detailed information about the attack has been presented but according to the report the adversary infiltrated the corporate network by sending an infected file containing malicious code injections to an on-site employee. Infected office computer was then used to proceed with additional attack stages into the plant network leading to the damages in manufacturing site. Multiple breakdowns of control components disabled partially or fully the control system preventing the furnace to shutdown in a controlled manner. Further details about the precise attack vectors have not been disclosed. [9]

Modern networked computer systems are under constant cyber threat of various levels. Cyber attacks have wide range of motivations behind them and they are performed by actors with different levels of expertise and resources. Low-threat probing is usually done by active hobbyists trying to find single vulnerable systems without any intention to break in. Cyber criminals are usually motivated by financial gain or ideology. However, national intelligence agencies have vast financial support,

manpower and technical capabilities to plan and execute advanced large-scale attacks. Governmental supported attacks are used for espionage but also for directly disrupting or sabotaging enemy systems. In recent years many superpower nation has added defensive and offensive usage of cyber weapons as a part of their strategic doctrine. USA, UK, Russia, China and Israel all have a strong presence in cyberspace operations. Well-planned cyber attacks leave very few traces to the origin of actors, sometimes there is no forensic evidence that an attack ever happened. Even if traces lead somewhere, connection routing, the use of proxies or other spoofing methods can conceal the real source of the attack. Without concrete evidence the actors can rely on plausible deniability.

Cyber attacks are effective in a supporting role of modern warfare. One of the recent usages was in the prolonged Ukrainian crisis. On December 23rd 2015, Ukrainian power grid suffered from large scale outages affecting approximately 225 000 customers. Similar failures have also been reported in several other Ukrainian critical infrastructure services. A large Ukrainian mining company and a railway company were attacked in a similar fashion. During the cyber attacks, malicious remote connections were discovered from outside network taking control over previously infected ICS systems. Later system analysis found several planted malware, e.g. KillDisk and BlackEnergy. [10] Remote connections have been traced back to a Russian Internet service provider and hackers also made phone calls from Russia. However, during the investigation direct Russian involvement could not be pointed as attribution for cyber attacks is very difficult without solid proof. [11, 12]

2.2 Problems in networked automation systems

ICSs are nowadays more and more networked with outside systems. Previously ICS had an isolated network between field site and control systems but external network connections provide flexibility in operation and maintenance tasks. Connections to corporate network helps with data flow from SCADA to office workers. Remote access through Virtual Private Network (VPN) connection enables operation around the world over any Internet connection. Centralized SCADA systems can collect data flows from unmanned remote field sites over network connection which results in significant operating cost reductions. Additionally, centralization helps with data management and efficient process control over several locations.

Designing reliable and secure network access requires IT engineers who have expertise in information security. However, most ICSs are designed by engineers with industrial systems design as their field of knowledge. It takes both disciplines to design an ICS that meets the safety and performance requirements and is also protected from cyber threats. Mistakes in design can easily happen when both teams are not working seamlessly for the same goal. Problems can also arise in installation and operation phase when proper security policy is not followed intentionally or due to ignorance.

Cyber security issues are caused by weaknesses in both organizational and technological policies. Cyber attacks, both internal and external, can take place when attacker is able to penetrate all layers of security measures set in place. Modern

ICSs, built with off-the-shelf products, benefit from standardized system components as shorter deployment times and lower purchase costs but leave the duty of proper configuration fully to the customer. Vendors typically ship units with well known default passwords, out-of-date software and non-restrictive security configuration. Oversights in installation may leave systems vulnerable for years without anyone noticing. National agencies, e.g. National Institute for Standards and Technologies (NIST) in the United States, have provided guidelines for ICS security [13], addressing possible vulnerabilities in hardware, software and configuration policies. Legacy ICSs have been designed and built before modern IT environments existed, making it hard to bring them up-to-date with current data security standards.

2.3 Issues in organizational and technological policies

Old ICSs were not designed with security as their primary objective. System availability, safety concerns and meeting regulatory requirements were the main design points while physical security was also considered important. Emphasis in these design aspects has also affected companies' security policies. To this day, shortcomings in security auditing, documentation and employee training are major contributors to system vulnerability.

Malicious actors prefer social engineering for finding initial attack vectors into an ICS system by exploiting the employees' willingness to help: access to the office network can be gained by sending an e-mail infected by malware to a company secretary, or the attacker can gain physical access to the plant site disguised as maintenance personnel. However, most organizational issues can be corrected by implementing common security principles. [13]

Most technical security issues can be fixed by implementing proper IT security policies. Policies for configuration management and auditing, password strength and service management are well-known and accepted in the industry. However, ICSs have their own unique needs which makes fixing them harder and more expensive. Automation systems run time-constricted process controls with high availability requirements. Equipment have slow processors and a small amount of memory which makes encrypted networking protocols and intrusion detection systems hard to implement as control systems are expected to respond to commands in real-time. [14]

Production downtime can cause major financial losses, therefore maintenance windows are kept narrow and scheduled to occur rarely. Complete process equipment overhaul is not usually possible, therefore devices are patched in smaller batches without interrupting the control process. Systems are usually operating for their full life-cycle which could postpone larger security upgrades for decades. Software systems are vendor-specific and delicate in nature which makes vulnerability patching difficult. It is also worth noting that equipment vendors prefer long-term support contracts which prevents switching from one equipment manufacturer to another without additional costs. A common practice in ICS maintenance is to operate the system with minimal upgrades and try to isolate the system as well as possible. [3]

The NIST guide for ICS security proposes multiple business oriented approaches for improving ICS security. It presents the benefits of well-documented and audited

security management policies while explains the repercussions of leaving security vulnerabilities in the system. It also proposes ways to improve the security of existing ICS installations by reinforcing proper isolation of network architecture and placement of end-point firewalls. [13]

2.4 Weaknesses in ICS network protocols

Common ICS protocols originate from an era of devices with direct serial connections. Serial connections are physically isolated from other networks which reduces exposure to the outside world. Cyber security was not a major concern until modern TCP/IP networking was implemented into control system devices. In order to maintain compatibility with old systems, protocol structure could not be changed and therefore security features have to be provided by transport layer protocols or by isolation from the Internet via firewalls but devices may still be subject to illicit cyber threats originating from inside network. Command injection, data injection and denial of service attacks leverage the lack of authentication in common control system communication protocols. Lacking cryptographic authentication prevents devices from validating the origin of commands and responses reliably thus making devices inherently vulnerable to data injection attacks. Next, security issues in two widely used ICS protocols, Modbus and Siemens S7comm, are presented as a case study.

2.4.1 Modbus

Modbus is a serial communications protocol originally developed and published by Modicon in 1979. It is an open and simple protocol for connecting PLCs and other industrial control system devices without any vendor specific restrictions. Modbus is commonly used to connect a supervisory workstation into a RTU in a SCADA system. There are binary (Modbus RTU) and text based (Modbus ASCII) serial versions of the protocol; however, a variant for IP networks (Modbus TCP) is becoming increasingly popular. Each device has a unique address assigned as Modbus's 8 bit address space provides addresses from 1 to 247. Commands can be sent out by any device but only commands coming from devices assigned as master are executed. In Modbus, devices executing commands are called slaves. Communication between devices happens in pairs on packets: every query from master device is responded by slave device.

Modbus frames are divided into Application Data Unit (ADU) and Protocol Data Unit (PDU). PDU contains message function code and function data payload. ADU contains PDU as payload alongside an address field and an error check field. Error check is only used to detect transmission errors, no authentication or encryption features are provided by Modbus protocol. The basic commands can instruct a slave device to change or return a value of a register (16-bit value), or control the status of an I/O coil (1-bit value). Modbus has additional functions for diagnostic purposes; Read Device Identification (function code 43) is very useful for querying device identification. [15]

Modbus vulnerabilities are well researched and understood in the industry and

scientific community. Most attacks focus on man-in-the-middle scenarios but fully remote attacks are also possible due to bugs in device firmwares. Modbus is commonly used to read process measurement data from PLCs in a feedback loop and make adjusting control commands in a control loop. Due to lack of any authentication, a compromised HMI or RTU terminal exposes PLCs to several kinds of attacks. One possible taxonomy for attacks groups them into four main categories [16]:

- Reconnaissance attacks are used to map network structure and identify devices. Address scan walks through every valid address and creates a list of assigned addresses by sending valid commands and waiting for responses from devices. Similar scan is possible for finding implemented function codes in control devices. Additionally, device identification scan can be used to gather specific information, e.g. device make and model, serial number and firmware version. More advanced attacks can map device data block addresses or even request a full memory dump.

- Response injection attacks intercept responses coming from an PLC and inject modified data into them. This can be very harmful to the process control loop as the monitoring HMI receives altered measurements and could instruct incorrect adjustments to the PLC. Alternatively, the attacker can craft and inject responses to the network spoofing as the responding PLC.

- Command injection attacks are used to inject malicious commands into control loop. Commands can have different effects based on the intentions of the attacker. An attacker can change the field site process state from safe to abnormal or even critical, usually by injecting commands that change the states of system actuators, e.g. pumps, valves and switches, controlled by PLCs. A repeated change of actuator states can cause physical and permanent damage to system components. Instead of sending harmful commands, the attacker can alter parameters or set point values (register values) used by process control routines on a PLC and cause the system to behave incorrectly.

- Denial of service attacks try to break communication between management system and field site preventing monitoring and control of the system. An attacker can try to overload slave devices by flooding a large amount of malformed packets to a PLC causing it to slow down or even crash. Radio links with carrier sense can be jammed by continuous radio transmission on the channel. PLC devices have limited computing resources and therefore an attacker may be able render them inoperable by injecting a constant heavy stream of request packets.

These attack vectors pose a significant risk for ICSs. Researchers have presented and confirmed 28 possible attacks against Modbus control systems. [16] These attacks exploited architectural weaknesses in Modbus protocol and the suggested ways of cyber attack mitigation are sufficient system isolation and implementation of proper intrusion detection system to reinforce digital forensic capabilities of the company.

2.4.2 Siemens S7comm

S7 communication is a proprietary network protocol for S7 series PLCs developed and maintained by Siemens. It is used to program PLCs with Siemens Step7 software tools, exchange data between PLCs and access PLCs from SCADA systems for data request and diagnostic purposes. S7 series PLCs use PROFINET, a fieldbus protocol based on Ethernet, which supports multiple network layer protocols including TCP/IP. TCP connectivity is handled with ISO-TSAP compatibility protocol which handles encapsulation between transport layer and session layer protocols. ISO-TSAP packets are sent in plain text which makes reverse engineering and data injection easy for attacker. An attacker can capture and replicate several operator tasks, e.g. turning off the CPU, disabling memory protection or injecting PLC with malicious code.

A presentation made for Black Hat conference in 2011 introduced multiple possible attack vectors for exploiting Siemens S7 PLCs: [17]

- S7 protocol is vulnerable to packet capturing and replay attacks. An attacker can extract valuable data from TCP stream between engineering workstation and PLC, and reuse it to generate new commands authenticated as the user.
- User authentication is based on a password hash sent to PLC and compared to the one located on PLC's memory. A captured hash can be reused because connection sessions never expire between the client and the PLC. Additionally, Siemens PLCs lack checks for session validity which enables the attacker to reuse preexisting sessions on any other PLC of the same model. It is also possible to collect captured packets into a library for brute force password cracking.
- PLCs are vulnerable to specifically crafted packets forcing the device to raise an error condition that causes the CPU to go into STOP state. This will cause any equipment connected to the PLC to halt resulting in unpredictable and possibly dangerous situation.
- With special probe requests it is possible to read and rewrite logic on the PLC. Changes to the ladder logic and data block may cause serious damage to the process.
- Replay attacks can be used to directly read from and write to PLC memory blocks. Memory rewrites can be used to reprogram system configuration, inject malicious code or send payload data to execute a remote shell connection.

Since 2011 Siemens has patched numerous security vulnerabilities in S7 PLCs but the overall architecture has remained the same. The recommended approach on minimizing the risk of exploitation is to limit exposure to non-restricted networks for all control system devices.

2.5 Current exposure of ICS devices on the Internet

Modern Internet has evolved into a global fabric connecting billions of devices, Internet of Things (IoT). Connectivity between computers, smart phones, televisions and other home appliances helps to automate daily tasks. However, the same network contains also incorrectly configured industrial automation systems with open ports that present a great threat to national cyber security. Situational awareness in cyberspace is considered of great importance and techniques have been developed to help finding and securing critical automation system on the Internet.

An extensive research effort called Internet Census 2012 [18] scanned every public IPv4 address in the world between June and October 2012. The project was carried out by an anonymous researcher and all research data was released for free. The scan exploited vulnerable embedded devices to form a distributed scanning network of over 400 000 devices, named Carna botnet. Although the researcher gained access to the devices by using default credentials, the actions would be considered illegal in most countries. During the scanning period 420 million IP addresses responded to ICMP ping request at least twice and were targeted for more advanced scanning techniques. The results of the scan contained terabytes of data about open ports, used software, trace routes, reverse DNS queries and other valuable information for researchers around the world to analyze.

2.5.1 Shodan

Shodan is a search engine for finding specific types of devices connected to the Internet. Shodan scans the whole Internet in constant cycles and saves findings into a database where users can query data using a variety of filters. Shodan was launched in 2009 as a side project for John Matherly who was interested to determine how many embedded devices were connected to the Internet, initially scanning just four services in each device: FTP, SSH, Telnet and HTTP. Since then the scanning capabilities have improved significantly and Shodan is currently probing over 200 different services. In order to shorten scan times, Shodan uses a network of distributed scanning nodes. Scan database combines host information, geographical IP info, open ports and data from each service probe, operating system detection and other useful data for each scanned device. [19]

Shodan provides useful information for multiple different purposes. Researchers benefit from the comprehensive data set of devices currently connected to the Internet and it helps them to prove exposure of devices based on empirical data. Companies can use Shodan to ensure that their network domain does not contain any exposed devices. Software companies can get statistics about the usage of their software. The free availability of metadata enables the search engine to be used either as a tool for improving security or as a cyber weapon. Shodan can be used to perform intelligence gathering while planning a cyber attack against vulnerable targets but the benefits outweigh threats in terms of improving cyber security.

Cyber security researchers have utilized Shodan in efforts to find vulnerable devices of critical infrastructure. Project SHINE (Shodan Intelligence Extraction) was based on intelligence gathered from Shodan between April 2012 through January

2014. The main objective of the project was to define a reliable search term set to identify industrial control system devices of critical infrastructure. The set contains not just control systems but also the supporting infrastructure devices. The research team compiled a search term set on 927 items and was able to discover over 500 000 ICS devices from Shodan database. The team was able to conclude that most device exposures appeared accidental due to poorly configured network infrastructure. [20]

2.5.2 The exposure in Finland

Researchers from a cyber security company Nixu analyzed the Internet Census data set to map current Finnish cyber landscape. [21] 13 782 236 Finnish IP addresses were selected from data set and analyzed for statistics. It was discovered that almost 95% of scanned hosts had 4 or less open ports, an amount considered fairly healthy. A relatively high number of devices provided insecure protocol access, e.g. Telnet and FTP, making it possible to capture sensitive data or user credentials from the wire. A large amount of services handling sensitive information, 7650 databases, were found to be directly accessible from the Internet. Out of all available services, 326 high-, 577 medium- and 82 low-level unique vulnerabilities with CVE identifiers were discovered where old Apache web server and PHP versions were the most common vulnerable software. Regarding exposed web management interfaces, the Nixu report also pointed out that after a device is installed, usually no additional configuration or hardening is done, leaving the device with unwanted services listening on all possible network interfaces. In worst case scenario, the remote management interfaces are available with default credentials or no password at all. Over 50 000 web interfaces of embedded devices, printers, routers or surveillance systems were discovered while dissecting the data set. HTTP, HTTPS, SSH, Telnet, FTP and SMTP were revealed to be the protocols most commonly available in Finnish cyberspace. In detailed service probe analysis, the report presented possible SCADA related servers, illustrated in figure 3. Devices running Windows CE operating system were excluded and in total there were over 4100 SCADA related hits.

Another Finnish research project in Aalto University has mapped Finnish industrial automation systems exposed to the Internet. [22] The research conducted custom search queries on the Shodan search engine using 41 distinctive keywords to find devices manufactured by the most common vendors of industrial control systems, power management systems, building automation and remote access servers. Original research was conducted in early 2013 and it was revisited 8 months later to observe any changes. The first query revealed 2915 unique IP addresses for suspected ICS devices of which 1968 devices were confirmed to be on-line. The second query showed an increase of about 60% at 4695 unique IP addresses of which 3281 devices were found to be on-line. Most of the IP addresses were assigned to building automation and power management systems and the rest belonged to industrial automation systems. During the writing of this thesis, Shodan was queried again with the same set of keywords, 2 years after the original research, to investigate the current level of exposed automation system devices. Queries were made 4 times, from September to December in 2015, to observe monthly variation in devices on-line. Results were

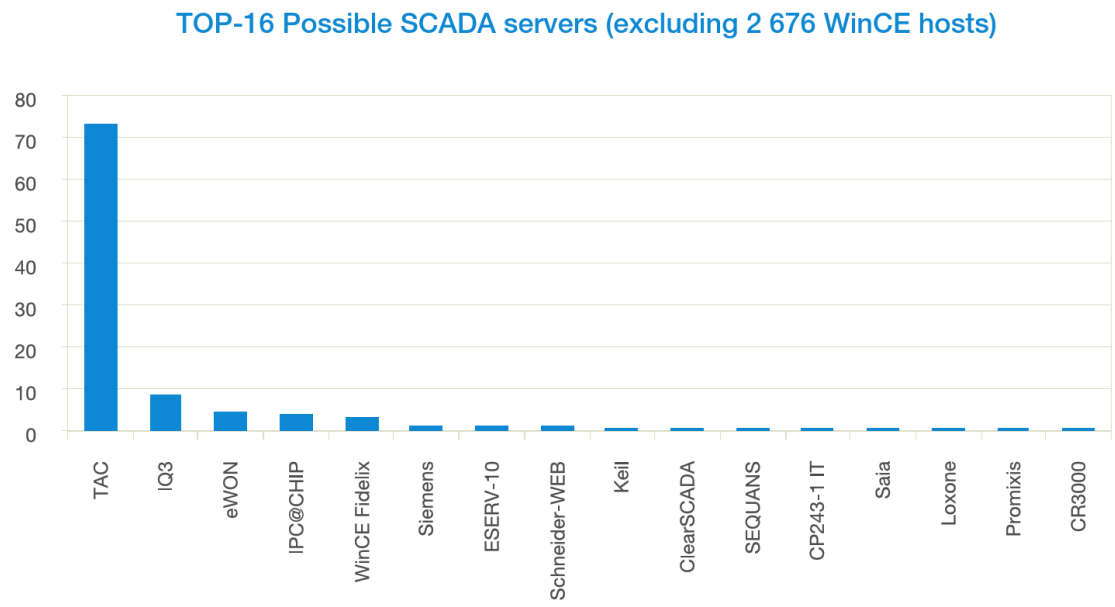

Figure 3: Internet Census 2012 - possible Finnish SCADA systems with web-access [21]

fairly unexpected as the amount of unique IP addresses had reduced down considerably. The amount of queried addresses fluctuated roughly between 1500 and 2000 month-to-month as can be seen in figure 4 below.

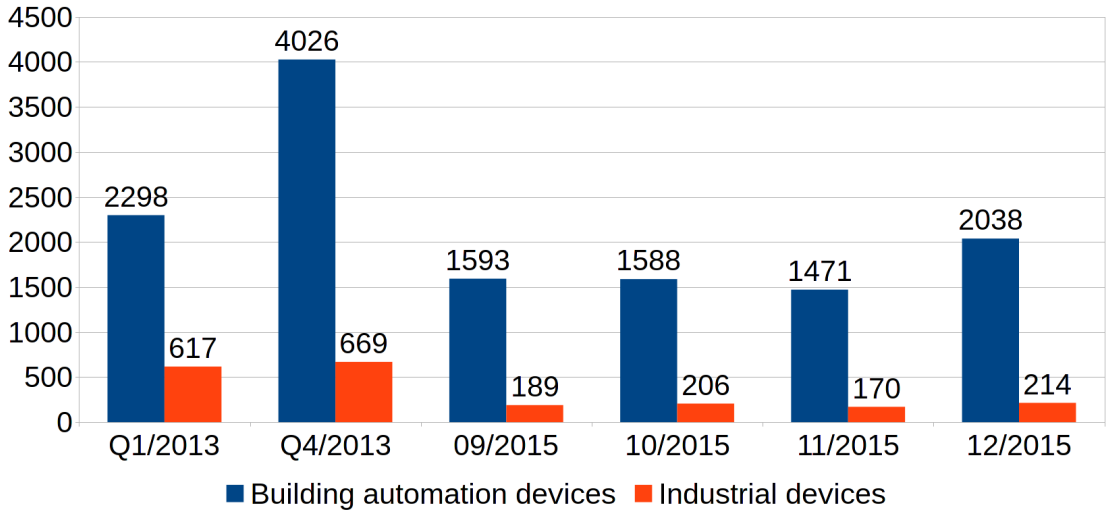

Figure 4: ICS devices found in Finland by Shodan in 2013 and 2015

One possible explanation for lower IP address count could be improvements in Shodan database refresh algorithms and purging of outdated database records. This theory is supported by the fact that about 20% to 30% of unique IP addresses were removed from Shodan query results between each month and new IP addresses were discovered. Most of these addresses were in dynamically allocated address pools and

hence could be discarded from the database after being reassigned to another host. New addresses belonged most likely to already discovered devices but this remained unconfirmed as scan data does not identify devices individually. It is also possible that these devices have been since taken off-line or protected by firewalls.

The research project also compared Finland with 8 other countries similar in the level of industrialization and Internet coverage. [22] Again, Shodan was queried with 53 carefully selected search terms to find an unbiased sample of ICS devices globally. A total of 132 775 unique IP addresses were found. After applying country filters, Finland held 2,48% of all queried devices based on the geographical locations of IP addresses. When the results were scaled by population of the country, Finland was discovered to have almost 0,6 exposed ICS devices per 1000 inhabitants, more than any other country. This query was revisited in 2015 to see if the landscape of cyber exposure had improved. With the same search terms, a total of 117 724 unique IP addresses were found. Finland held 2,45% of those addresses which translates into 0,53 devices per 1000 inhabitants. Figures 5 and 6 below illustrate the results. It can be concluded that the exposure has not decreased and new methods for finding and securing ICS devices are still much in demand.

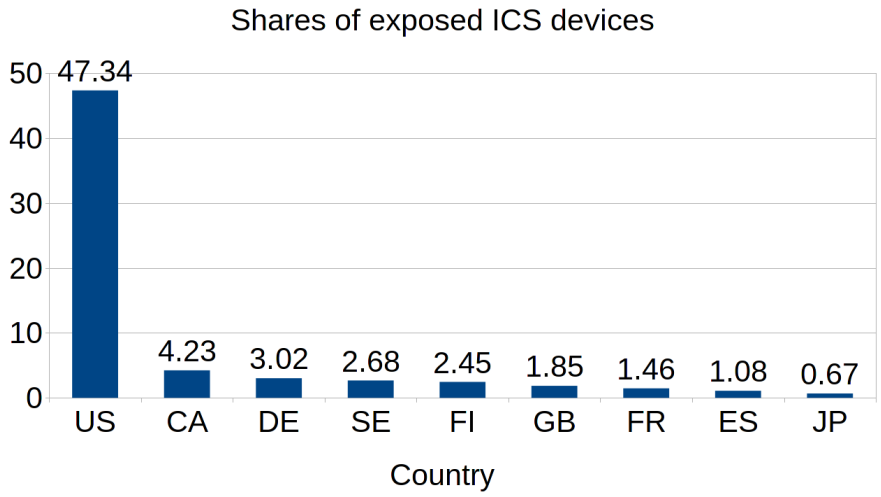

Figure 5: The distribution of exposed ICS devices

Concern about the sufficient protection of critical cyber infrastructure has been a catalyst for national cyber protection platforms. One of these is Havaro, a system for monitoring network traffic with early warning capabilities, alerting on incoming and outgoing suspicious intrusive activity. The system was developed and deployed by the Finnish Cyber Emergency Response Team (CERT-FI) in co-operation with companies and government agencies operating on critical infrastructure. Havaro consists of intrusion detection sensors placed in the border between public Internet and private networks. Sensors monitor traffic and report any suspicious activity to central computers for analysis. Data is further used to provide situational awareness in cyberspace for companies, Internet service providers and CERT-FI. [23]

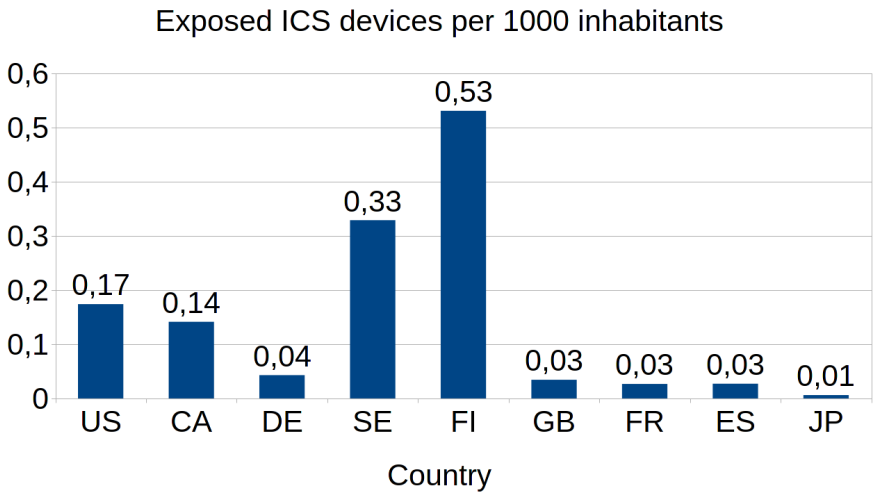

Figure 6: Exposed ICS devices per capita

2.6 Summary

Networked industrial control systems provide easy remote management and cost-efficiency but poorly designed network architectures combined with misconfigured remote access interfaces pose great threat for cyber security and could lead to horrific industrial accidents. Due to lacking security features in common ICS network protocols, firewalls and other means of network isolation should be carefully implemented whenever possible. However, Shodan and other scanning tools used today reveal a lot of exposed ICS devices on the Internet which indicates that companies don't value information security at the same level as physical security or personnel safety. As systems become more connected, the amount of exposed devices is on the rise as well. Luckily, research teams work relentlessly with national cyber emergency response teams to map and alert the owners of exposed and vulnerable ICS devices. In addition, national intrusion detection systems have been deployed to provide better situational awareness in cyberspace.

In the next chapter, fingerprinting methods for reliable ICS device identification are discussed.

3 Fingerprinting ICS devices in IP networks

Modern Internet ecosystem connects billions of devices together which makes it difficult to find a specific device among others without knowing the correct IP address. A wide range of scanning tools and other software have been developed to help finding and identifying devices. Tools, e.g. port scanners, can be invaluable for national cyber security in finding exposed and vulnerable devices that control critical automation infrastructure. In this chapter, port scanning over IP protocol is viewed as a method of extracting device identification fingerprints for later analysis. Both pros and cons of port scanning are discussed and a set of common ICS protocols as well as OS fingerprinting and geolocation are proposed as possible data sources for ICS fingerprinting. To conclude the chapter, legality issues of device probing are pondered and scanning methods conforming with Finnish legislation are proposed.

3.1 Port scan as a fingerprinting method

Port scan is a process where scanner sends client requests to a target computer in order to gather information, e.g. open ports or available network services. In IP networks a scanner can target TCP and UDP ports to find responsive services. If a service responds in an open port, scanner can proceed with more advanced scanning techniques to identify the service. The legality of a port scan depends on motivations tied to it. Neutral port scan without aggressive service probing can be seen as a way to audit cyber security of the network and therefore improve overall security. On the other hand, a scan done with intent of malicious service exploitation can be classified as a cyber attack and considered illegal activity.

Scanning IP addresses for open services is a straightforward process. The difficulty lies in reliable device identification based on responses from the device. Therefore, the scanner needs to know which ports to scan and how to probe available services to collect the most accurate and condense set of identifying information. The formed data set has two quality requirements: no two devices can have the same set of identifying data and the set has to remain stable over time. Probing the device and compiling the data set is known as device fingerprinting. Unfortunately, fingerprinting is never 100% accurate process and insufficient amount of fingerprints in decision making can lead to both missed identifications and false positive identifications.

Scanner has to be consistent in both scanning methods and data collection. Target devices on the other hand can be configured to provide misleading information or deny probes for the service. Scanner has to filter responses for relevant data fields and collect only valuable data for device identification process. Calculated device fingerprints are saved and forwarded to a classifier which compares them to the previously known responses from similar devices and tries to reliably identify the device. This is known as classification and is discussed more in chapter 4.

3.2 Issues in device scanning

Scanner may run into several obstacles during the scan process. The most common problem is a lack of response from the device or blocked service ports, usually due to a firewall. Even if the service responds on the port, it may block all scanning efforts or accept connections only with encrypted, proprietary protocols. This prevents the scanner from gathering any service information and therefore hinders the possibility of a successful identification.

Depending on the network configuration, the ICS device can be placed behind routers or gateways. Packets sent to a port of an IP address can be forwarded to a different device which means that one public IP address can represent multiple devices on the inside local area network. Therefore, the scanner cannot assume that one target IP address is dedicated to a single device.

ICS devices are known to have limitations in processing capacity. They are designed to have real-time responsiveness in the control loop while management interfaces and network stack run on lower priority. Users typically experience this as slow response times from the device. Scanning software has to take this slowness into account by adjusting connection timeout values. A long timeout increases scan times but a timeout too short closes the connection before client has a chance to respond, resulting in loss of possibly valuable response data.

3.3 Data sources for fingerprinting

To ensure efficient fingerprinting process, the scanner should only query data sources that provide useful identification data. In IP protocol networking this means scanning a specific selection of ports used in networking between ICS devices. In addition, analysis of TCP flows can be used to fingerprint operating system running on the target device and GeoIP databases are useful for locating the target device geographically. The combination of all available data sources ensure the best possible identification match for the device.

In Internet protocols there are 65535 possible ports for services connecting through TCP or UDP sockets. Scanning every port on every target host would be time-consuming and provides diminishing returns compared to enumerating a selection of ports used by popular services in ICS devices. Previous research efforts in project SHINE have mapped the most common services used in ICS networks. [20] Devices typically have multiple services open, many of which are considered general Internet protocols while a few of them are vendor specific automation protocols. Next, a few popular services are explored and their properties for device fingerprinting are evaluated. Sample scan results are presented for most protocols, detected with a prototype scanner which is discussed further in chapter 5.

3.3.1 Common network protocols

A few common protocols used in ICS devices are HTTP(S), FTP and SNMP. They are not ICS specific and have more generic use cases. However, they leak information in server responses which is useful in fingerprinting the device.

HTTP

Hyper Text Transfer Protocol (HTTP) is an request-response application protocol originally designed to transfer hypermedia content. [24] The default port for HTTP server is TCP port 80 although vendor specific web servers are known to be bound on numerous other ports as well. Most new ICS devices have an HTTP server implemented to provide a remotely accessible web management system. In most cases, it is sufficient to send a single GET request to pull the web server front page and gather descriptive fingerprints for device type. HTTP responses include useful descriptive system information in server headers, e.g. server software version and descriptions of access-protected objects in www-authenticate header fields as the defined realm name is usually preconfigured by the device manufacturer. Response payload (the HTML document) may contain valuable information, the most valuable being document title but other tags may also be worth a keyword search.

Listing 1: Captured HTTP server header

```
status : HTTP/1.1 401 Unauthorized
request type : GET
title : Digi One SP&amp;nbsp;Configuration and Management
transfer-encoding : chunked
server : Allegro-Software-RomPager/3.12
connection : close
location : /
content-type : text/html
www-authenticate : Basic realm="Digi One SP"
```

HTTPS

Hyper Text Transfer Protocol Secure (HTTPS) improves the security of HTTP by enabling transmission over encrypted SSL/TLS layer and is usually bound to TCP port 443. A crucial part of this encryption is connection signing with a digital certificate. The certificate can be self-signed or signed by a trusted certificate authority (CA) and contains detailed information about the certificate subject, following the X.509 cryptography standard. [25] If the certificate is properly filled and signed by a trustworthy CA, the certificate can reveal information about the organization or company subject to the certificate including name and physical location.

FTP

File Transfer Protocol (FTP) has been used to transfer files between devices for decades and is considered one of the base protocols of the Internet. The protocol utilizes two-port connectivity: separate ports are used for control channel and data channel. Port 21 on the server-side is used to establish control channel after which data channel is established to client port 20. If the client requests passive connection mode, data channel is established originating from client-side to some other port on server-side. FTP includes a clear-text protocol for authentication but anonymous login is also possible if the server allows it. Identifying information can be gathered in several stages: After connection has been opened, the server sends a welcome greeting (a login banner) which usually contains at least server software name and version

but can also contain additional text, e.g. device or system name. If anonymous access is allowed, any user may login without credentials by giving an e-mail address as a password. Then, more information about the system can be recorded by, e.g. listing available files in the home directory. The range of user actions is limited by the server configuration; however, there are legal limitations on how the scanner can proceed without infringing privacy. [26]

Listing 2: Scanned FTP service

```
ftp-anon : No (FTP code 530)
banner : VxWorks FTP server (VxWorks 5.4) ready.
```

SNMP

Simple Network Management Protocol (SNMP) is a standard communications protocol for collecting, organizing and modifying information about managed devices on IP networks. Managed devices include routers, switches, bridges and other network elements. By default the SNMP agent software listens on UDP port 161. SNMP presents management data in the form of a hierarchical data tree where each available management data object is accessed with a unique object identifier (OID). Metadata for each OID is described by Management Information Bases (MIB). Network management stations, the clients, can send queries to devices running SNMP agent software which handles requests and manages data on the device. Most devices have SNMP configured to provide detailed system information for public access. This includes system description, system name and possibly even location of the device. [27]

Listing 3: System description queried with SNMP

```
1.3.6.1.2.1.1.1.0 : TAC Xenta911 5.1.6-11. Copyright 2006 by TAC AB, Malmo,
    Sweden
```

3.3.2 Terminal connection protocols

Direct terminal connections are still needed for managing networked devices over command line interface. They have low resource requirements and client access is possible from any computer with a keyboard attached. Telnet is a client-server protocol that provides widely supported bidirectional virtual terminal facilities. Due to not having encryption, Telnet has been replaced by Secure Shell (SSH) protocol in some modern systems.

Telnet

Standard server port for Telnet is TCP port 23; however, multiple serial connection gateways prefer other ports. In Telnet connection, after negotiating terminal options a full-duplex line-buffered virtual terminal device is activated and the user is presented with a prompt. At this stage, the server typically sends a greeting banner which may reveal system and software information. Depending on the configuration, the server may then require login credentials or directly open a terminal session. [28]

Listing 4: Telnet banner captured after successful connection handshake

```
banner : Welcome to the Windows CE Telnet Service on TPC-650H\r\n\r\nlogin:
```

SSH

SSH has its own handshake protocol in connection initialization. After an SSH connection has been established, usually into server TCP port 22, both server and client are required to send a line-terminated identification string. The string contains requested protocol version (1 or 2) and the software version. This string is useful because it will reveal the server software version and whether it will accept vulnerable SSH version 1 connections. Key exchange begins immediately after this. SSH uses Public Key Infrastructure (PKI) for verifying the identity of the server which makes the fingerprint of server's public key an unique identifier that can be used in device identification process. [29]

Listing 5: Detected OpenSSH server

```
sshv1 : No
version : SSH-2.0-OpenSSH_7.0
fingerprint : 1024 97:d5:93:3e:db:e9:e4:3a:db:13:9f:c7:00:04:1c:bd (DSA)
```

3.3.3 ICS specific protocols

Most vendor supplied ICS management tools and PLC programmers utilize protocols specifically designed for the needs of ICS environments. These protocols have built-in dedicated functions for querying device information. A selection of common ICS protocols, and how they can be used to gather device information, are presented next.

BACnet

BACnet is a network protocol used specifically in building automation and control networks for applications, e.g. heating, ventilation and air-conditioning units. Building automation systems are widely adopted in corporate networks which makes them interesting targets. The protocol enables information exchange between automation devices of different use classes and utilizes UDP port 47808 for connectivity. Interoperability is provided by using object-oriented model, where every object has a set of standardized properties and they provide services to other objects. Every device has identifying property fields that can be queried by other devices. These fields define object parameters (identifier, name, type), vendor specific parameters (vendor name, device model, firmware, application version) and optional fields revealing location and descriptive data. In addition to querying individual devices, BACnet enables device discovery by connecting each subnet through a gateway known as BACnet Broadcast Management Device (BBMD). Each BBMD in BACnet network has a synchronized Broadcast Distribution Table (BDT) which contains all BBMDs in the network. By querying BDT from a gateway the scanner can learn every subnet containing devices in the BACnet network. Additional registered BACnet devices

outside these subnets can be found by querying a Foreign Device Table (FDT) from the gateway. [30]

Listing 6: Detected air-conditioning unit with BACnet

```
Application Software : 2010-01-28
Firmware : 1.1.28s
Location : Street
Model Name : City
Vendor Name : IVprodukt
Vendor ID : Siemens Schweiz AG (Formerly: Landis & Staefa Division Europe) (7)
Object-identifier : 98
Description : City
```

S7Comm

Siemens S7 PLCs communicate over S7Comm proprietary protocol which can be used to query information from devices. S7comm establishes IP connections with ISO-TSAP compatibility layer protocol and utilizes TCP as its transport protocol in port 102. An independent research group called SCADA Strangelove, focused on ICS cyber security, has researched extensively vulnerabilities of popular devices, e.g. Siemens PLCs. [31] The group has developed and released multiple tools for scanning and fingerprinting of ICS devices. For Siemens S7 series PLCs they have developed methods for gathering device identification by sending specifically crafted S7Comm requests over Connection-Oriented Transport Protocol (COTP) connection transport protocol. By parsing the response packets the scanner is able to extract device information, e.g. device name, model, firmware information and plant location. [32]

Listing 7: Detected Siemens Simatic PLC

```
Version : 3.2.10
Copyright : Original Siemens Equipment
Module Type : IM151-8 PN/DP CPU
Module : 6ES7 151-8AB01-0AB0
Serial Number : S C-E4TC76412014
System Name : ET200S_PN_01
Plant Identification : None
Basic Hardware : 6ES7 151-8AB01-0AB0
```

Modbus

Modbus is an open serial communications protocol developed to connect PLCs and other ICS devices without vendor specific restrictions. It has a variant used for connecting devices over TCP/IP networks by using TCP port 502. The protocol specification defines multiple useful function codes that can be used to identify Schneider Electric / Modicon devices and other PLCs supporting the protocol. Two useful function codes for querying information are 43 and 90. However, some vendors have not implemented support for these functions in their hardware. Function 43 (Read Device Identification) is used to test if the target device supports Modbus. If the device supports function 43, it will return identification data containing vendor name, module model, CPU type and firmware info. Even if the device returns an error code, we still know that it supports Modbus protocol. The second proprietary

function code, number 90, is reserved and used for uploading and downloading ladder logic to the Schneider Electric PLC over insecure, unauthenticated connection. It can be used to pull and extract project file information from the PLC. Malicious actor may also use this function to overwrite existing ladder logic program on the PLC and severely harm the controller and the process it is managing. [15]

Listing 8: Sample Modbus device [33]

```
Vendor Name : Schneider Electric
Network Module : BMX NOE 0100
CPU Module : BMX P34 2000
Firmware : V2.60
Memory Card : BMXRMS008MP
Project Information : Project -  V4.0
Project File Name : Project.STU
Project Revision : 0.0.9
Project Last Modified : 7/11/2013 5:55:33
```

Niagara Fox

Niagara Fox, develop by Tridium, is a proprietary network protocol used in Tridium building automation systems worldwide. It handles both station-to-station and workbench-to-station connectivity over TCP port 1911. By using a specifically crafted query packet, device identification data can be pulled from any active station. Response contains a wide range of descriptive data, e.g. host name and address, application versions, Java virtual machine details, vendor name and running operating system of the device. Additionally, the device can return location name if configured. [33]

Listing 9: Detected Vykon brand building automation system

```
VM Name : Java HotSpot(TM) Server VM
Brand ID : vykon
Host Address : 192.168.196.201
Application Version : 3.7.106.4
Fox Version : 1.0.1
VM Version : 23.7-b01
Application Name : Station
Host ID : Win-6F71-0080-FFFC-BB77
OS Name : Windows 7
VM UUID : 8d586c11-523c-43f0-9748-4c1552bd70e9
Station name : <REMOVED>
Timezone : Europe/Helsinki
OS Version : 6.1
ID : 379279
Host Name : <REMOVED>
```

PC Worx

PC Worx protocol is used to program and communicate with Phoenix Contact ILC PLCs. The protocol is proprietary and uses TCP port 1962 for connections between devices. After connection has been established, a session ID is formed for the remaining of the connection. After that the scanner can pull device identification data with a single request. Device response contains PLC type, model number and firmware info. [33]

Listing 10: Sample PC Worx system [33]

```
PLC Type : ILC 330 ETH
Model Number : 2737193
Firmware Version : 3.95T
Firmware Date : Mar  2 2012
Firmware Time : 09:39:02
```

OMRON FINS

OMRON Global makes industrial and manufacturing machinery, and uses proprietary FINS protocol for remote access over network interfaces. The protocol has variants for both TCP and UDP connectivity and uses port 9600 for traffic. Device information can be pulled with Read Controller Status request. Response from the device contains vendor model and version number as well as current device state info. [33]

Listing 11: Sample OMRON system [33]

```
Controller Model : CJ2M-CPU32 02.01
Controller Version : 02.01
For System Use :
Program Area Size : 20
IOM size : 23
No. DM Words : 32768
Timer/Counter : 8
Expansion DM Size : 1
No. of steps/transitions : 0
Kind of Memory Card : 0
Memory Card Size : 0
```

ProConOS

ProConOS is a PLC runtime engine for embedded and PC based control applications. A proprietary network protocol is used to exchange data between PLCs and to program and administer control systems from MULTIPROG workstation. PLCs are accessible through TCP port 20547 and only one request packet is needed to get identification data from the device. The response contains ladder logic runtime status, PLC type, firmware version and the name of the running project file. [33]

Listing 12: Sample ProConOS system [33]

```
Ladder Logic Runtime : ProConOS V3.0.1040 Oct 29 2002
PLC Type : ADAM5510KW 1.24 Build 005
Project Name : 510-projec
Boot Project : 510-projec
Project Source Code : Exist
```

Ethernet/IP

Ethernet/IP is an open standard for industrial Ethernet developed by Rockwell Automation. Together with Common Industrial Protocol (CIP) it provides an object-oriented framework for real-time control applications and other automation functions. Devices using this protocol can be queried by sending an request function List Identity to TCP port 44818. Response from the device contains information, e.g. vendor

ID, product name and serial number, device type, product code, device revision and internal configured IP address. [34, 33]

Listing 13: Sample Ethernet/IP system [33]

```
Vendor : Rockwell Automation/Allen-Bradley (1)
Product Name : 1769-L32E Ethernet Port
Serial Number : 0x000000
Device Type : Communications Adapter (12)
Product Code : 158
Revision : 3.7
Device IP : 192.168.1.1
```

3.3.4 Remote operating system fingerprinting

Modern IP networking establishes interconnectivity between a plethora of different devices. Due to vendor specific software, subtle differences in network stack implementations enables operating system (OS) fingerprinting and identification of the precise version running on a target host. Advanced probing methods combined with pre-calculated heuristic database can be used to directly identify thousands of different device models as well as operating systems they run. OS detection can provide coarse-grained, e.g. Linux vs. Windows, as well as fine-grained, e.g. detecting the version of the Linux kernel, differentiation.

There are two main approaches to the classification: detection with automatic machine learning algorithms or the use of database containing previously collected entries of known operating systems.

Machine learning algorithms try to automatically generate suitable OS fingerprints by using large training data pool of different OS instances. A research paper was released in 2010 discussing the possibilities and limitations of using machine learning in OS detection. The premise of using automatic algorithms seemed feasible. However, during their research the group concluded that this method faces significant technical challenges in generating reliable and accurate OS detection fingerprints. First, target probing may encounter random, non-OS related variation in traffic flow and header field options that are difficult to automatically exclude from training data set. This leads to overfitting of a classification model and limited accuracy in host detection. Second, generated fingerprints tend to be biased towards the distribution of operating system hosts in training set which may not represent real world environment and thus hinders the accuracy of fingerprints. Third, fine-grained classification granularity becomes difficult as the small observed differences in packets may be caused by external factors instead of being actual implementation differences. Fourth, machine learning algorithms have limited understanding of relation between network protocols and packet fields. The researchers found that it takes human expertise to fully make use of all little intrigued semantics of a network protocol in the effort for reliable OS fingerprint generation. [35]

Another classification model is based on comparing probing results to a previously compiled database of known OS fingerprints. One example of fingerprinting tools using this model is called Nmap. Nmap works by sending out 16 specifically crafted TCP, UDP and ICMP probe packets to known open and closed ports of the target host. A

set of tests are run for response packets and results are condensed into host fingerprint. Host fingerprint is then matched against the database and the closest matching OS entries are returned. Closeness is determined by fine-tuned heuristic algorithms that emphasize certain fingerprint features over others. Listing 14 illustrates the compiled host fingerprint and possible closest matches from Nmap's database. Nmap is efficient but keeping the classification database up-to-date is cumbersome and requires manual tweaking. New OS versions may require a completely new set of probing packets for feature extraction and reliable differentiation. [36]

Listing 14: Nmap OS detection

```
Host fingerprint:

SCAN(V=6.40%E=4%D=12/4%OT=23%CT=21%CU=161%PV=N%DS=7%DC=I%G=N%TM=56618DD0%P=x86_64-
    unknown-linux-gnu)
SEQ(CI=RD%TS=U)
OPS(O1=M218%O2=%O3=%O4=%O5=%O6=)
WIN(W1=2000%W2=0%W3=0%W4=0%W5=0%W6=0)
ECN(R=Y%DF=N%T=3E%W=2000%O=M218%CC=N%Q=)
T1(R=Y%DF=N%T=3E%S=O%A=S+%F=AS%RD=0%Q=)
T2(R=Y%DF=Y%T=2A%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)
T3(R=N)
T4(R=Y%DF=Y%T=1E%W=0%S=A%A=O%F=R%O=%RD=0%Q=)
T5(R=Y%DF=N%T=2D%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
T6(R=Y%DF=Y%T=25%W=0%S=A%A=O%F=R%O=%RD=0%Q=)
T7(R=Y%DF=N%T=22%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
U1(R=Y%DF=Y%T=3E%IPL=8E%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)
IE(R=Y%DFI=N%T=3E%CD=S)

OS match: Digi PortServer TS serial-to-Ethernet bridge, propability 94%
OS match: Digi PortServer TS 4 serial-to-Ethernet bridge, propability 90%
OS match: Digi PortServer TS 16 Rack serial-to-Ethernet bridge, propability 86%
OS match: Digi One SP serial-to-Ethernet bridge, propability 86%
OS match: Digi PortServer TS 4 H MEI serial-to-Ethernet bridge, propability 85%
```

3.3.5 Geolocation

If direct device probing does not reveal the location of the device, subsequent method is to try geolocate its IP address. Regional Internet Registries (RIR) are organizations that allocate and register IP address blocks for Internet Service Providers (ISP) under their jurisdiction. RIRs register information of every assigned address block into a who-is database. Likewise, ISPs can add information of their customers, e.g. customer name, e-mail, location or contact info, when assigning IP addresses. However, most customers do not register static address blocks but use dynamic address allocation for their networking needs. Therefore, who-is database usually has ISP information associated with their allocated address blocks. [37]

Most geolocation databases are based on RIR who-is entries. Database accuracy can be improved by collecting customer location data from 3rd party sources. Only relying on RIR database results in major inaccuracy as IP address blocks are not fixed to any geographical location and they may get reassigned when needed. Also, dynamically leased addresses for customer connections come from a larger address pool and does not represent the actual location of the device. Therefore, geolocation based on IP address can be very inaccurate and should not be relied on for precise location

data. However, geolocation has its place in various applications, e.g. visualizing purposes over larger geographical areas.

MaxMind is one of the companies producing and providing multiple constantly updated geolocation databases for their customers. It claims to reach 99,8% accuracy in detecting the correct country of the IP address. For another database containing city level geolocations, in the USA, 90% accuracy is claimed for recognizing the correct state and 84% accuracy for the correct city within 50 km distance of the true location. The accuracy varies depending on the country and may be below the minimum required level for any practical purpose. In Finland, MaxMind claims only 53% accuracy in detecting the correct city of the IP address within 50 km distance of the actual location. [38]

3.4 Legality concerns

Active network scanning methods may face serious legality concerns, depending on the legislation of each country. Even if the scan was done with good intentions, it may be considered illegal according to the law. Researchers and security auditors have to be very careful to stay inside the legal boundaries set by local legislation. Even though law enforcement officers have more rights for surveillance when investigating crimes, that does not cover actions for crime prevention and the improvement of overall cyber security.

In this section, a brief look is taken into Finnish criminal law related to communications offenses and how scanning should be conducted while staying inside legal boundaries.

3.4.1 Finnish legislation regarding communications

In the Finnish criminal law, communications offenses are covered in chapter 38. In the chapter 38 the relevant parts for network scanning are the parts described in sections §5 to §8. Additionally, in chapter 34 section §9 is relevant by disallowing the endangerment of any computer system.

- Sections §5 and §6 criminalize the disturbing of communication transmissions. In normal cases a security scanner does not do anything to prevent the normal operation of the network. However, if the scanning process is too time-efficient and sends a huge amount of probing packets in a small time frame, it risks overloading the transmission link capacity or blocking the resource-limited network stacks of ICS devices. Therefore, scanning process should always implement some rate-limiting mechanism in order to avoid the possible interruption of normal target system operation.

- Section §7 disallows disturbing or interrupting the normal operation of a computer system by inputting, transferring, damaging, altering or removing any data related to it. To comply with this requirement, the scanner cannot do anything that may change the state of the target system. Scanning software

should always avoid the use of any hacking or exploiting tools available and only use safe probing methods that are within protocol specifications.

– Section §8 forbids breaking in a computer system by using illegally acquired credentials or through a security breach by making use of a system vulnerability. Section §8 also indicates that login attempts with factory default credentials or brute forcing are not allowed by the law. It is also forbidden to gain access to the data inside the system by circumventing the security parameters without breaking into the system.

– Chapter 34 section §9 criminalize, with a malicious intent, to import, acquire, make, sell or distribute devices or computer software designed to harm the operation of a computer system, communications system or a protection of a computing system. It is also illegal to possess or distribute illegally obtained password or other credential. This section would question the legality of available exploitation tools. However, as the law states, the legality of such tools depends on the intent of the user. Using them for security auditing, penetration testing or any other way to improve cyber security with a permission from target system owner, is legal.

3.4.2 Legal and illegal scanning methods

As the cyber security sections of criminal law are worded, intentions behind the actions have a great role in defining legality. Scanning a system for the purpose of finding open or vulnerable services to exploit is clearly illegal. However, scanning a system for the purpose of security research may be considered legal as long as the scan does not harm or interrupt the target system, and all data acquired during the scan is kept confidential and disclosed, if needed, only with the authorities. Moreover, the scan may not use any methods that would extract private data from inside the system boundaries.

Querying a selected set of devices and probing a small set of ports to acquire identification data directly available, without using any exploitive measures or unsafe scanning methods, and used only for security research purposes, can be considered legal. The scanner may also login to the system without using any credentials, e.g. by utilizing anonymous FTP access, or query HTTP server for public content. However, further intrusive scanning for these services may be illegal including gathering any user data protected by privacy laws. Intrusive scanning inside private networks is also forbidden which limits the scanner to finding only devices that are directly connected to the Internet.

If possible, the scanned target owners should be informed about the scan beforehand. Disclosing scanning methods and transparent communication between all parties reduces the risk of unnecessary investigations of a possible hacking attempt. Larger scanning events, up to nation wide, should be coordinated with authorities in advance in order to keep everyone involved up-to-date and to ensure smooth information exchange.

3.5 Summary

Successful identification of an ICS device requires sending precise probing requests to available network services for identifying information, combined with other data sources, e.g. operating system detection and geolocation. Most common ICS protocols have integrated methods for extracting system identifiers from the device but the scanner should be careful not to overburden the device or network with probing packets. Also, only safe probing methods should be used to avoid interrupting the normal operation of the target system. Scanning procedures have to respect the regional laws set to protect the safety and privacy of computer systems and thus adjust scanning methods accordingly.

After collecting a sufficient amount of identifying fingerprints, the device is ready to be identified and classified. For that, a reliable decision making model and a classifier is needed. In the next chapter, a model for ICS device assessment and classification is proposed.

4 Model for ICS device identification and classification

In this chapter, a model for ICS device identification and classification is presented. Major system components are defined and methods for data source analysis are proposed. A system following this model is able to scan a set of ICS target hosts and analyze device fingerprints to assess device identification. Additionally, the system combines a list of available services, detected software vulnerabilities and device location to assess device threat level. However, this model focuses on device assessment and does not define additional supplementary components needed to integrate with other cyber security systems.

First, the target device is scanned for service fingerprinting and operating system detection. After successful device fingerprinting the collected data is filtered, condensed and refined into a unified structure. The scanner saves results into a report which will then be processed by a classifier. Classification process evaluates the data based on predefined rulesets and makes an overall assessment of device model and system threat level. For devices assessed to be serious cyber security threats, additional actions will be taken.

4.1 System components

The base system consists of two main components: a scalable pool of scanning nodes and a classifier. The nodes can be distributed in different networks and they can process allocated scan jobs individually in parallel. For each assigned job a report file is compiled which contains all scan results with additional scan metadata. Figure 7 illustrates the system architecture and data flow in higher level.

4.1.1 Scanner

Each scanning node has a copy of the scanning software. They get scan assignments as jobs which contains a set of target hosts or subnets to be scanned and scan parameters which determine the ports and services to be scanned on each target host. Nodes can scan multiple target hosts in parallel to save time. After determining which ports are open, the scanner proceeds with service scan in selected target ports and extracts fingerprints from services. Service scanning modules handle sending appropriate probing packets to the device, receiving responses, extracting fingerprinting markers from response data and formatting service scan results into a unified data structure containing fingerprints as key-value pairs, as illustrated in chapter 3. After each target host has been scanned, a report file is compiled and sent to the classifier for analysis.

There are several popular port scanner software that could be used to scan ICS devices. In this section, three popular scanners are compared: ZMap, Scanrand and Nmap.

ZMap is a powerful open source port scanner which utilizes efficient asynchronous scanning methods bypassing kernel packet interfaces and directly writing frames

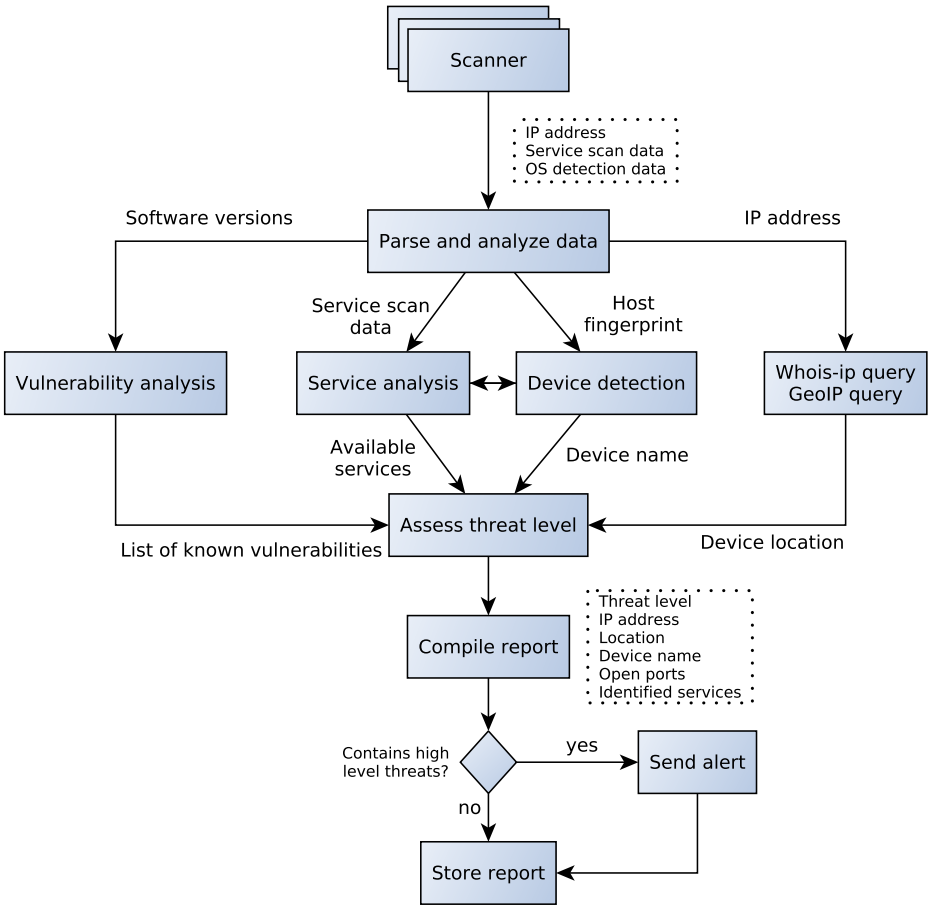

Figure 7: A process for ICS device assessment

to the network hardware. ZMap tries to maximize network link utilization by not using advanced traffic flow features, e.g. connection tracking, retransmissions or rate limiting. Instead it sends out probes as fast as it can. According to a paper published in 2013, ZMap is capable of scanning the entire IPv4 address space in under 45 minutes by using an ordinary gigabit Ethernet connection without specialized hardware or kernel modules. That is 1300 times faster than Nmap with default settings while achieving similar accuracy. [39]

Another asynchronous scanner called Scanrand is, according to their paper [40], able to reach speeds of six times faster than Nmap without losing accuracy. Scanrand, just like ZMap, utilizes multi-threaded architecture where packet sending and receiving are handled by separate threads running concurrently.

It is worth noting that the maximum scanning speed of ZMap assumed only one port to be scanned per target host. [39] Increasing the number of target ports will slow the scanner down while it has to wait for server responses. Most embedded hardware have prioritized responses to external interfaces lower to save resources for internal real-time process control functions. Therefore it is safe to assume that the scanner has to wait for response from an embedded HTTP server or ICS protocol

connection for up to several seconds. Without extensive multi-threading, the internal worker pool of the scanner will be exhausted, effectively slowing the overall scan down. With UDP connections, the wait can be even longer and require retransmissions of lost packets.

Nmap, while being the slowest and synchronous scanner, has several features missing in ZMap and Scanrand. With default settings, Nmap uses polite rate limiting algorithms that do not disrupt sensitive hardware, e.g. PLCs, with fast-paced packet probes. The scanning speed adjusts itself based on connection stability and response times from servers. Therefore, fast responding servers increase the overall probing speed and scan parallelism. Likewise, encountered packet loss and slow response times in combination with retransmissions slow the probing down. Additionally, Nmap has a feature-rich scripting engine based on Lua programming language which enables the scanner to perform complex service scan routines in both text-based and binary protocols. Nmap also has a large library of helper functions to ease probing of several common protocols, e.g. HTTP and FTP. The scripting feature is essential for scanning ICS devices because most identification data has to be queried by using proprietary protocols which are not supported in scanner software themselves. Therefore, Nmap was selected to be used in the prototype scanner featured in chapter 5 and multiple ICS specific scanning scripts were made to utilize its internal scripting engine. [41]

4.1.2 Classifier

A classification model is needed to solve the problem of identifying which of the categories the scanned device belongs to. In the case on device identification, the observed device can be assigned into a multitude of categories based on device type, manufacturer, model or purpose. On the other hand, device also represents a possible cyber security threat and therefore it should be assigned into the right threat level category based on device location, exposed vulnerabilities or available network services.

Classifier reads the scan report received from a scanning node and analyses scanned target hosts. The report is parsed and each scanned service is analyzed individually before combining results and assessing target threat level. There are two main classification categories to choose from: rule-based classification and statistical machine learning classification. Both approaches have their pros and cons which will be discussed next.

- Rule-based classification relies of manually formed rulesets that classify targets based on matched rules. For each identifiable device type a new ruleset entry has to be made. In most cases simple keyword matching rules are sufficient to identify a device as a certain make and model. As the ruleset grows, it becomes harder and more time-consuming to maintain but the amount of effort needed can be considered acceptable for a relatively small number of ICS device types. The real benefit of rule-based classification is the simplicity of classifier implementation: typically the classifier contains only a mechanism for fingerprint comparison and handlers for each rule type used.

– Machine learning algorithms classify by making predictions based on decision boundaries accumulated during training phase. Algorithms are able to make generalizations and learn relations between different variables. Machine learning algorithms are able to adapt to new situations or new device types whereas rule-based classification requires explicit new rules added by an expert. However, a certain amount of data science expertise is needed to apply machine learning in device detection whereas rule-based matching is relatively simple to set up. Machine learning algorithms can be one of two types: supervised learning or unsupervised learning. Supervised learning algorithm is adjusted by a training set of correctly observed and validated detections. On the other hand, unsupervised learning algorithm, also known as clustering, does not need a training set but it is more suitable for summarizing and explaining the differences between observed device categories than actually making correct identifications.

In this thesis, a simple rule-based classifier was selected for implemented prototype scanner because of the scarcity of available training data and the lack of experience with machine learning algorithms. Currently no complete database of available ICS devices does publicly exist which would make machine learning algorithms hard to properly configure without an appropriate training data set.

4.1.3 Additional components

In this section, additional components for the model are proposed. These components are not part of the base model but they add value to the overall system functionality. Moreover, some of the components presented do not comply with the current legislation of Finland; however, they are valid when executed with the consent of the target system owner.

– A dispatcher and control server is needed to manage the operation of distributed scanning nodes and handle the data exchange between scanners and the classifier node. An integrated management system is used to add new target addresses or subnets and to monitor the system operation.

– A checker for default credentials would be a useful part of more intrusive service scan mode. The scanner would try to brute force login attempts with a pre-compiled list of default user names and passwords for the particular device model. In case of a successful login, the scanner would also try to figure out the purpose of the device as part of system threat assessment. Vendors ship units configured with well-known default passwords which may not be changed during the installation process. Checking devices for weak passwords would be a useful way to find vulnerable systems. Unfortunately, such methods are currently considered illegal.

– In addition to the vulnerability assessment, the scanner could also test whether potentially detected vulnerabilities have been patched on the device. This

method would require exploiting a possible remote vulnerability on the device software. This method is also highly illegal but would complement the vulnerability analysis.

– The classifier could have a separate module producing statistics of exposed vulnerable automation control systems. The collected statistics would be automatically shared with authorities and therefore improve the national cyber situational awareness. The module could also include reporting functions for device owners.

4.2 Data sources for device assessment

The importance of an ICS device depends on more than just the type of the device. Two identical devices in different environments could have differing threat levels if, e.g. one of the devices controls manufacturing utilities in a workshop whereas the other is used to regulate power grid. Separating critical devices from non-critical ones is a constant challenge while being the most important part of detecting exposed vulnerable devices on the Internet.

Automatic threat assessment system is an integral part of device classification. Accurate assessments without human interaction is a difficult task: device importance is a balanced combination of device type, location and device vulnerability. An automated system featured in figure 7 relies on reliable analysis of each aspect. Next, possible analysis models for device importance are presented.

4.2.1 Location analysis

Device location can be extracted from the device itself by probing available services or by resolving the location data associated with device's IP address. As explained in chapter 3, several popular ICS protocols reveal the device location if it is configured on the device. A suitable keyword search could be able to flag important devices on specific location based on these descriptive protocol fields. Another approach is to resolve who-is database and GeoIP locations associated with the IP address. However, as stated previously in chapter 3, the IP address could be registered to an ISP that is hesitant to disclose customer identities without a request from the authorities.

In the model, resolved location info, e.g. country and city, is passed on to the classifier to help assess device criticality.

4.2.2 Service analysis and operating system detection

Most useful fingerprints come from direct service probing analysis. The scan report contains all the data extracted from available network services on target host. After being formatted into a unified form, typically key-value pairs, rules of different types can be targeted to the data. Most rules are of keyword based pattern matching type, targeting software versions, vendor names and device models but other rule

types, e.g. software version number comparison, are possible. By combining several matched keyword rules a positive identification can be made for the device.

Detected operating system name can be useful in device identification. Nmap OS detection database, presented in chapter 3, has entries with variable granularity and therefore the most probable match can be either an operating system name or an exact device model. By combining keyword search results between service analysis and operating system detection, a more precise identification result can be reached. Device name and the set of detected services are a great factor in assessing the criticality of the exposed device.

4.2.3 Vulnerability analysis

After extracting software versions from service scan data and detected operating system, a check for non-patched vulnerabilities could be made. Most announced software vulnerabilities have a Common Vulnerabilities and Exposures (CVE) number and a specific vulnerability description of affected software products and versions. For each installed software products a check against version number reveals relevant vulnerabilities. A list of compiled vulnerabilities can be passed on to the classifier which will elevate the overall threat level if needed.

4.3 Tasks after assessment

After devices have been identified and threat level assessed, a report is compiled containing every analyzed device. Each device entry combines the assessed threat level with device information, e.g. IP address, location, device name or operating system, open ports and identified services. The report is inspected for devices with high threat level. For these devices, an alert procedure is triggered. The model itself does not define how an alert should be made or what contents it has.

Unfortunately, in most cases, the owner or organization holding the device cannot be found by direct scanning methods. The device may not contain any information about the location and geolocating the IP address may be too inaccurate. ISPs hold records for IP address leases but they are under strict confidentiality and won't be disclosed unless requested by the authorities as a part of investigation. Even if the vulnerable device is located successfully, there is no guarantee that anything will happen to secure it. As pointed out in chapter 2, cyber security is not taken seriously enough in the industry, therefore it is likely that nothing will be done to improve the situation. If the system in case is operating reliably despite the cyber security issues, the owner is likely to ignore any warnings or will postpone the required upgrade indefinitely. It will require a national effort to raise awareness about the need of better cyber security policies. Until then, plenty of devices are and will be potential targets for malicious actors.

4.4 Problems with the model

The usefulness of this model depends on the ability to successfully scan and identify exposed vulnerable ICS devices. Successful scan requires that the right combination of open ports is detected and thorough service scan is performed on each port. If the device does not respond to the scanner probes as expected, fingerprint data will be lost and proper identification may not be possible or identification is only partial.

Successful identification needs properly made and maintained identification rulesets that trigger on right fingerprint markers. The system can only identify devices that have correct identification rules in the database. Effective operation of the identification process requires a large library of device rules for every vendor. As there is no concise descriptive collection of every ICS device available, it can be assumed that most ICS devices would remain undetected until the database is built up. Easy ruleset database management tools are essential for convenience in adding new device rules. Partial identifications are problematic because, in some cases, scan results do not contain enough fingerprints for full identification of the device. Resulting identification may contain only device vendor name or just software version the device is running. False positive identifications are not a problem as long as rulesets are strict enough and vague rules are removed from the database.

Because of the simplistic nature of scanning nodes, they do not perform any identification matching but only collect scan data and provide it to the classifier. This keeps the overall architecture clean but causes more load on the classifier host which does all the analysis work. On the other hand, scanning nodes are lightweight and easily deployable when needed.

4.5 Summary

An integrated model for ICS device identification and classification is a viable way to probe and detect exposed vulnerable ICS devices on the Internet. Device detection combined with location information creates a basis for finding critical exposed systems. The effectiveness of this proposed model is based on the robustness of service scan scripts and the vastness of identification rule database. Without a properly build device identification database, a lot of ICS devices go unnoticed. More research is needed for methods to automatically resolve accurate context, e.g. location combined with device purpose, for the target device. Otherwise, the resulting device assessment consists mostly of device identification without any information about its importance. This part of the model remains as a challenge for future research work.

In the next chapter, a proof of concept prototype scanner and classifier is presented. It implements the core parts of the model presented in this chapter and leaves more advanced features, e.g. vulnerability analysis, for future work.

5 Proof of concept

A prototype scanner and classifier software was made to test the viability of the model presented in previous chapter 4. This chapter presents a working prototype which implements the core model, leaving the more advanced features for future work. The operating principle and key features of the prototype are discussed. The prototype was tested with both simulated virtual devices and real devices directly connected to the Internet. The testing in real network environment was carried out by querying Shodan with 41 distinctive keywords to gather a sample of common ICS devices exposed to the Internet. Country filters were applied to restrict the test scans to Finland. The chapter is concluded with presentation of scan results and discussion of issues encountered during prototype testing.

5.1 Operating principle

The prototype has the core features of the model implemented, separated into two software modules: a scanner which does the target scanning and a classifier which analyses scan results. Figure 8 illustrates the operation of each component and data flow between them.

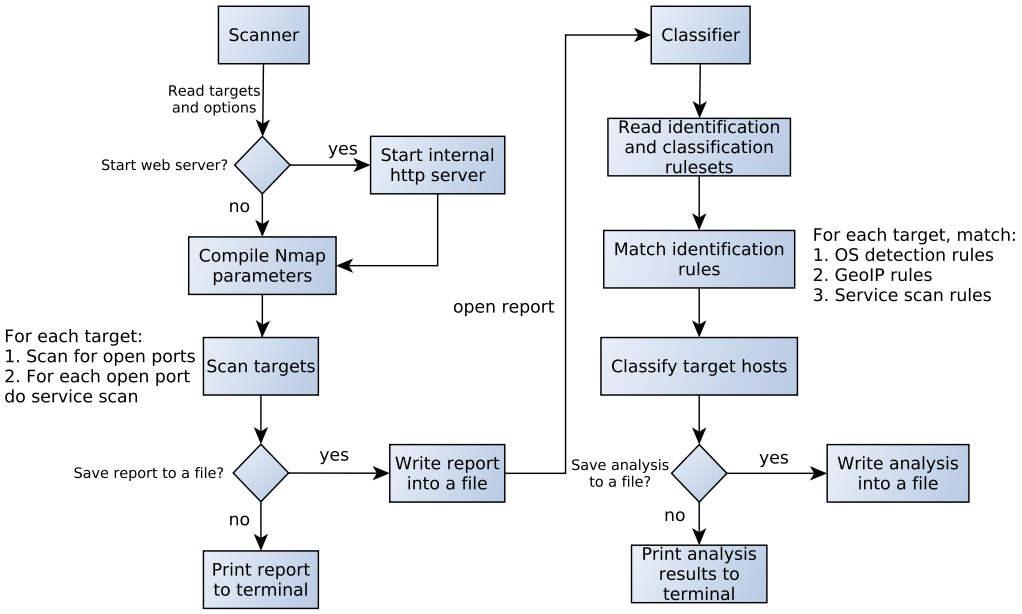

Figure 8: A proof of concept prototype

The scanner performs port scan on selected TCP and UDP ports generally used in ICS devices. Additionally, it can run a selection of more in-depth service scans to extract useful information from devices. The scanner is programmed in Python language and an external libnmap toolkit library is used to launch the Nmap port scanner process, retrieve results and compile an easy-to-parse report structure for later

analysis. Service probing scans are performed with a set of specifically crafted Nmap Scripting Engine (NSE) scripts programmed in Lua language. In this prototype, a set of NSE scripts specifically designed to scan ICS devices were acquired from Redpoint research project. These scripts were used as the base for service scan logic; however, they were modified to suit the needs of the scanner. [33]

The classifier takes a scan report file from the scanner as an input. After reading the report it loads identification rulesets and a detection database, and analyses scanned hosts one-by-one. Analysis results can be saved into an output file for further processing.

5.1.1 Scanning procedure

The scanner starts by reading input arguments from command line. Arguments include scan target selection, ports to be scanned, invocation of an optional HTTP server to inform about the scan in progress and the selection of service scan scripts. By default, no service scans are performed without explicitly requested which keeps the scanner operation non-intrusive by nature. Preset arguments are included to conveniently select a common set of ICS services with a single argument. The scanner is able to read scan targets from a file in order to save the effort of listing them one at the time as command line arguments.

After input arguments have been set, the informative HTTP server is started, if requested with a command line switch. Command line arguments are then compiled into a Nmap argument string and the scan is started in a background thread. In the port discovery stage, Nmap performs TCP SYN and UDP scan methods to probe open ports on each target host. Additionally, operating system discovery probes are sent at this stage. After open ports have been discovered, selected service scan scripts are launched based on port rule matching. Each script handles the use of network functions and data management individually: sending probe packets, receiving responses from target host, parsing response data, and compiling scan result table. Service scan scripts use a structured output table to organize service fingerprints which are stored as nested elements inside of XML output format.

The scanner has been preconfigured with presets for ICS specific port selection and script support. The presets select 16 TCP ports and 2 UDP ports to be scanned for service discovery. Table 1 shows the default selection of ICS ports for the scanner.

After each target host has been scanned, a report is compiled. Report is formed as a libnmap report object data structure and it contains scan header metadata with a list of host objects. Each host object contains the scan results for the target host. The libnmap library contains programming interfaces for easy parsing of the report object which is later utilized in the classifier module. If requested in command line arguments, the report is saved into a file. Otherwise, the report is parsed and the contents printed to terminal output.

5.1.2 Classification procedure

The classifier begins by reading identification rulesets and detection database from configuration files. After that it opens the specified scan report file and performs host

Service	Port
FTP	tcp/21
SSH	tcp/22
Telnet	tcp/23
HTTP	tcp/80/81/8000/8080
Siemens S7	tcp/102
SNMP	udp/161
HTTPS	tcp/443
Modbus/TCP	tcp/502
Niagara Fox	tcp/1911
Phoenix Contact ILC	tcp/1962
Omron FINS	tcp/9600
Lantronix	tcp/9999
ProConOS	tcp/20547
Ethernet/IP	tcp/44818
Bacnet	udp/47808

Table 1: A preset selection of ICS service ports

classifications. By default, every target host found in scan report will be classified unless a list of target hosts are provided as a command line argument. In that case, only selected target hosts are classified and the others are ignored.

In the first phase, the classifier iterates over each scanned host in the scan report and matches them against the identification rulesets. OS matching, GeoIP location matching and service scan matching are performed separately because the data fields reside in separate structures in the scan report. Service specific identification rules are grouped based on the port number of the service. Additionally, there is a generic ruleset for rules non-specific for any particular service. Every service scan fingerprint, presented as a key-value pair, is processed against service specific rules and generic rules: a rule can target specific key or it can be generic. If a rule matches against a fingerprint entry, the matched rule id is added to the rule list for the host, the host point counter is incremented by the cost of the rule, and the host threat level is elevated if the matched rule has higher level than the host currently has.

In the second phase, the matched rule list for each analyzed host is compared against the detection database. Each detection rule is a combinational logic (AND, OR) set of identification rule identifiers. If the detection entry matches against the matched rule list for the host, a positive detection has been made. Each host can have multiple positive detections based on detected device name, manufacturer name, available service or other detectable quality. This relaxed nature of detection process enables detectability for host features of several granularity.

After each host has been processed, analysis results can be written into a file for later usage or post-processing purposes. The prototype does not have any alerting capabilities; therefore, it is the responsibility of other software solutions to extract high risk targets from analysis results and proceed with appropriate action.

5.1.3 Identification rulesets and detection database

The classifier uses two libraries for device identification and classification: a set of identification rulesets and a detection database. The library formats are defined next.

Identification rulesets reside in text files and each ruleset for each service, GeoIP location or OS match is placed into a separate file. In the classifier, both GeoIP and OS detection are handled as services. Additionally, there is a generic ruleset for rules that can apply to any port or service. Each line in a ruleset defines one fingerprint rule. A rule is a set of rule type, rule target, rule item, rule cost, rule severity and rule id.

- Rule type defines how the rule is processed when matching. In the prototype classifier, keyword type is the most common while there are also rule types for version number comparison. However, the classifier does not put any restrictions on rule types and more types can be added by including additional rule type handlers into the classifier code.
- Rule target is the service scan fingerprint key which the rule applies to. If target is 'generic' then the rule is matched against any key.
- Rule item is the actual rule data (keyword, integer) which is compared against the fingerprint value.
- Rule cost is a predetermined cost associated with the rule. If the rule matches then host specific point counter is incremented accordingly. This value has no specific purpose in the prototype classifier but can be used to find interesting targets.
- Rule severity defines the threat level this rule represents. It can be either 'Unspecified', 'Low', 'Medium', 'High' or 'Critical'. When the rule is matched, the threat level of the host is elevated to correspond the highest matched identification rule.
- Rule ID is the unique UUID identifier associated with the rule. This identifier is used to separate rules from each other; therefore, it is crucial that UUID collisions do not happen in rulesets.

Detection database is located in a separate file. Detection database contains the predefined detections of a device or software. Each detection rule is a combinational set of identification rule identifiers. Each rule is a set of rule name, rule certainty and the list of identification rule UUIDs.

- Rule name provides the string representation of detection (i.e. device name or server software name). There can be several made detections with the same name so that the same device or software can be detected in many different ways.

- Rule certainty specifies the certainty of detection in a percentage value from 0 to 100. This enables the use of alternative detection rules with different detection levels, each with different subset of fingerprinting rules.
- List of rules defines the collection of identification rules used to make a detection match. Rules in the file format are separated either with a semicolon (AND) or a colon (OR), e.g. A;B:C states that detection requires rule A and either rule B or rule C to match. UUID identifiers are used to represent rules in this list.

5.1.4 Classifier output

After successful scan report analysis, the results can be written into a file in a CSV type format. Each line represents one analyzed host. A host is skipped if it is down or it has no open ports. Host entry contains following fields, separated with semicolons:

- Host alert level is the analyzed alert level of the host based on matched rules. It can be 'Unspecified', 'Low', 'Medium', 'High' or 'Critical'.
- The IP address of the analyzed host.
- The Domain Name System (DNS) host name associated with it's address. If there is no host name, the IP address is presented here.
- The GeoIP resolved city for the IP address. 'None' if city name cannot be resolved.
- The GeoIP resolved country for the IP address. 'None' if country name cannot be resolved.
- The identified OS running in host. Alternatively, a device name if Nmap OS detection is able to determine that. 'None' if there is no OS fingerprints available.
- A list of open ports (TCP or UDP) available in the host.
- A list of (detection, detection certainty) tuples gathered during host analysis.

Resulting file is easy to parse with common text interpreter tools. A list of hosts with high or critical threat level is simple to extract from the file and an alerting procedure can be invoked if needed.

5.2 Testing in virtual environment

Initial testing was conducted by using a honeypot software in virtual machine environment. This enabled agility in software development and continuous testing for feature validation and finding bugs. Honeypot systems simulate real industrial hardware or other attractive systems for attackers. Instead of controlling any industrial process or providing any actual service, they deceive attackers and record their actions in

a safe and controlled manner. Conpot is a popular ICS/SCADA honeypot that is able to simulate an ICS PLC using multiple IP network protocols. It supports S7comm, Modbus, BACnet, IPMI, HTTP and SNMP protocols thus providing an excellent target for scanner testing purposes. Listing 15 presents a starting Conpot honeypot system and service initialization. The honeypot has versatile configuration possibilities and supports the addition of new protocols as software modules. [42]

Listing 15: Conpot honeypot starting

```
# conpot --template default


                      _
 ___ ___ ___ ___ ___ _| |_
|  _| . |   | . | . |  _|
|___|___|_|_|  _|___|_|
            |_|

  Version 0.5.1
  MushMush Foundation

2015-11-08 11:24:02,150 Starting Conpot using template: /usr/local/lib/python2.7/
    dist-packages/Conpot-0.5.0-py2.7.egg/conpot/templates/default
2015-11-08 11:24:02,150 Starting Conpot using configuration found in: /usr/local/
    lib/python2.7/dist-packages/Conpot-0.5.0-py2.7.egg/conpot/conpot.cfg
2015-11-08 11:24:02,291 Fetched xxx.xxx.xxx.xxx as external ip.
2015-11-08 11:24:02,295 Found and enabled ('modbus', <class conpot.protocols.modbus
    .modbus_server.ModbusServer at 0x7fe0d70a27a0>) protocol.
2015-11-08 11:24:02,299 Conpot S7Comm initialized
2015-11-08 11:24:02,299 Found and enabled ('s7comm', <class 'conpot.protocols.
    s7comm.s7_server.S7Server'>) protocol.
2015-11-08 11:24:02,300 Found and enabled ('http', <class 'conpot.protocols.http.
    web_server.HTTPServer'>) protocol.
2015-11-08 11:24:02,301 Found and enabled ('snmp', <class 'conpot.protocols.snmp.
    snmp_server.SNMPServer'>) protocol.
2015-11-08 11:24:02,302 Conpot Bacnet initialized using the /usr/local/lib/python2
    .7/dist-packages/Conpot-0.5.0-py2.7.egg/conpot/templates/default/bacnet/bacnet.
    xml template.
2015-11-08 11:24:02,303 Found and enabled ('bacnet', <class 'conpot.protocols.
    bacnet.bacnet_server.BacnetServer'>) protocol.
2015-11-08 11:24:02,304 IPMI BMC initialized.
2015-11-08 11:24:02,305 Conpot IPMI initialized using /usr/local/lib/python2.7/dist
    -packages/Conpot-0.5.0-py2.7.egg/conpot/templates/default/ipmi/ipmi.xml
    template
2015-11-08 11:24:02,305 Found and enabled ('ipmi', <class 'conpot.protocols.ipmi.
    ipmi_server.IpmiServer'>) protocol.
2015-11-08 11:24:02,305 No proxy template found. Service will remain unconfigured/
    stopped.
2015-11-08 11:24:02,305 Modbus server started on: ('0.0.0.0', 502)
2015-11-08 11:24:02,306 S7Comm server started on: ('0.0.0.0', 102)
2015-11-08 11:24:02,306 HTTP server started on: ('0.0.0.0', 80)
2015-11-08 11:24:02,461 SNMP server started on: ('0.0.0.0', 161)
2015-11-08 11:24:02,462 Bacnet server started on: ('0.0.0.0', 47808)
2015-11-08 11:24:02,462 IPMI server started on: ('0.0.0.0', 623)
2015-11-08 11:24:07,307 Privileges dropped, running as "nobody:nobody"
```

To test the scanner, a Conpot honeypot was installed to a Debian Linux virtual machine and scanned. Listing 16 shows a successful scanning and detection of the honeypot masquerading as a Siemens Simatic S7-200 PLC device. In addition to testing with a single honeypot, a virtual network containing several concurrently running Conpot instances was created to test the parallel scanning capabilities of the prototype scanner.

Listing 16: Detecting a honeypot system

```
====== Analysing host 10.0.0.10 (debian-honeypot) ======
GeoIP information not available.
=== PORT 80 ===
-> Port 80 (http) is open. (+1 point)
Analyse service scan result:
  status : HTTP/1.1 200 OK
  content-length : 575
  request type : GET
  title : Overview - Siemens, SIMATIC, S7-200
-> Keyword match: 's7-200' (+20 points, severity: Medium)
-> Keyword match: 'simatic' (+20 points, severity: Medium)
  set-cookie : path=/
  last-modified : Tue, 19 May 1993 09:00:00 GMT
  location : /index.html
  date : Mon, 26 Oct 2015 15:12:59 GMT
  content-type : text/html
=== PORT 102 ===
-> Port 102 (iso-tsap) is open. (+1 point)
Analyse service scan result:
  Version : 0.0
  Copyright : Original Siemens Equipment
  Module Type : Siemens, SIMATIC, S7-200
-> Keyword match: 's7-200' (+20 points, severity: Medium)
-> Keyword match: 'simatic' (+20 points, severity: Medium)
  Plant Identification : Mouser Factory
  Serial Number : 88111222
  System Name : Technodrome
=== PORT 502 ===
-> Port 502 (modbus) is open. (+1 point)
Analyse service scan result:
  sid 0x1 : {'Device identification': 'Siemens SIMATIC S7-200', 'Error': 'ILLEGAL
      FUNCTION'}
-> Keyword match: 's7-200' (+20 points, severity: Medium)
-> Keyword match: 'simatic' (+20 points, severity: Medium)
=== PORT 161 ===
-> Port 161 (snmp) is open. (+1 point)
Analyse service scan result:
  1.3.6.1.2.1.1.1.0 : Siemens, SIMATIC, S7-200
-> Keyword match: 's7-200' (+20 points, severity: Medium)
-> Keyword match: 'simatic' (+20 points, severity: Medium)
-> Keyword match: 'simatic' (+5 points, severity: Medium)

Analysis result (1 hosts):

Host: 10.0.0.10 (debian-honeypot)
Open ports: [80, 102, 502, 161]
 Ruleset: 102, Points: 41
 Ruleset: 80, Points: 41
 Ruleset: 502, Points: 41
 Ruleset: 161, Points: 46
 Total points: 169
 Threat level: Medium
Possible identification: Siemens Simatic device (90% certainty)
Possible identification: Siemens Simatic S7-200 PLC (90% certainty)
```

5.3 Scanning in Finland

After successful testing phase in a virtual honeypot environment, a larger scan experiment was commenced to test the prototype in real environment. The scanner was modified to provide a static web page informing about the scans made in conjunction with a research project, providing contact information and a list of time intervals when scanning took place. This was done for the purpose of transparency

and to dispel any concerns of malicious intent.

5.3.1 Target selection

For the target selection, the same set of 41 distinctive ICS keywords were used to query Shodan for scan target acquisition that were used to find the most popular ICS devices in chapter 2. To limit the scan scope inside Finnish borders, a country filter was applied to Shodan search terms. After record extraction and removal of duplicate addresses, a set of targets were ready to be scanned. Shodan was queried 4 times in late 2015 with same keywords to refresh the target set monthly. Test scan was run several times during the testing period from early October to early December. Scan reports were saved from each scan for later analysis.

5.3.2 Scan results

The scan results are presented in table 2. It contains a metric called detection ratio, defined in equation 1. Detection ratio is a metric for classifier efficiency by comparing the amount of detected devices or services to the devices available to be detected. However, it does not take into account partial detections and therefore has limited usability. Additionally, the overall figure of detected devices contains also detected devices or computers that were not actual ICS devices. In most cases the non-ICS device was an ordinary PC, usually detected as a Windows machine, that had received the same IP address from ISP that was previously leased to an ICS device.

Date	**02.10.**	**16.10.**	**21.10.**	**11.11.**	**25.11.**	**04.12.**
Nr. of targets	1768	1783	1783	1631	1631	2238
Targets up	1545	1600	1564	1434	1410	1973
Targets with open ports	1447	1543	1504	1373	1240	1726
Targets identified overall	1012	1118	1365	1254	1217	1703
Targets identified as ICS	1012	693	1087	1135	1217	1703
Detection ratio	0,70	0,72	0,91	0,91	0,98	0,99

Table 2: Scan results for Finnish target hosts

$$\text{Detection ratio} = \frac{\text{Targets identified overall}}{\text{Targets with open ports}} \tag{1}$$

Observations on scan result statistics:

– The amount of target IP addresses received from Shodan varies from month to month. Shodan tries to keep the database fresh by constant scanning cycles on the Internet. There is a significant drop in targets with open ports on scan results from 16.10. to 21.10. (down 2,5%) and from 11.11 to 25.11. (down 9,4%) indicating that IP addresses were reallocated in between those dates. Queried IP addresses do seem to get out-of-date very fast due to dynamic

address allocation provided by ISPs. Therefore, it is suggested to refresh target addresses just before each scan.

– The efficiency of detection was greatly improved during the fall. This was because of constant development on scan scripts which resulted in more reliable service scan performance, increasing the quantity of available service scan fingerprints.

– During the scans 13% to 24% of targets were either down or didn't have any open port. However, in Shodan records the search engine found open services successfully in all of them. There are several possible explanations for this: Shodan has implemented less obvious scanning techniques, the devices become off-line occasionally and connect to the network only when transmitting data, or Shodan has a large quantity of out-of-date entries in its database.

Detected device types

The classifier was able to provide a detection for a device or service in 99% of the cases. If the device or computer provided multiple detectable services, each one of them could be detected individually. This means that the device can have multiple classifications based on detection rules available. Detection is also possible in various granularities. The same device can be detected as 'Siemens Simatic' device but can additionally have a more precise 'Siemens Simatic S7-200' classification because of the extra available fingerprint.

A total of 64 detection rules were made as a part of the prototype development to detect a selection of ICS devices and services. In most cases a reliable detection of a device needs only one good fingerprint entry. The same information is usually available through several network services, e.g. SNMP system description in addition to traditional ICS specific fingerprint. Table 3 provides a sample of what device types the classifier was able to detect.

Device type	Purpose	Amount
ATM/EZ-IP	Security gateway	95
Centraline	Building automation	12
Digi One SP	Serial-to-Ethernet gateway	64
Ouman EH-Net, WebSCADA	Building automation	31
Schneider TSX	Industrial automation	3
Siemens Simatic S7, HMI and NET	Industrial automation	9
TAC Xenta 511, 711, 911, 913	Building automation	342
Westermo EDW-100	Serial-to-Ethernet gateway	32
Windows CE Fidelix	Building automation	359
Wisepro	Building automation	3
Vykon QNX	Building automation	171

Table 3: A sample of detected device types

Identification rules used

Only a small amount of identification rules were needed to detect devices. In most cases it was beneficial to make a generic rule that can apply to any protocol available instead of a targeted protocol specific rule. That way the same keyword is searched in fingerprints generated from any protocol or service. This is especially useful feature for HTTP and Telnet fingerprint matching because the service can be in a multitude of different port numbers.Table 4 shows the amounts of identification rules used when detecting the devices against Shodan database records.

Port	**21**	**22**	**23**	**102**	**161**	**502**	**1911**	**9999**	**44818**	**Generic**
Amount	4	2	11	2	14	2	5	2	1	42

Table 4: Identification rules per protocol

Open ports

While the open port does not directly indicate the available service on host, standardized port numbers can implicate the types of available services. Combinations of certain port numbers can be used to evaluate the nature of the device and devices considered important can be selected for more advanced service scans. The distribution of open ports for December 2015 scan is presented in figure 9. A total of 2238 hosts were scanned in which 512 hosts did not have any ports open. This portion, 22,9% of all scanned hosts, were either protected by firewall or were not ICS devices at all. As stated before, Shodan database loses freshness quickly which means that a sizable amount of these devices were most likely some other devices that had acquired the dynamic IP address previously leased by an ICS device.

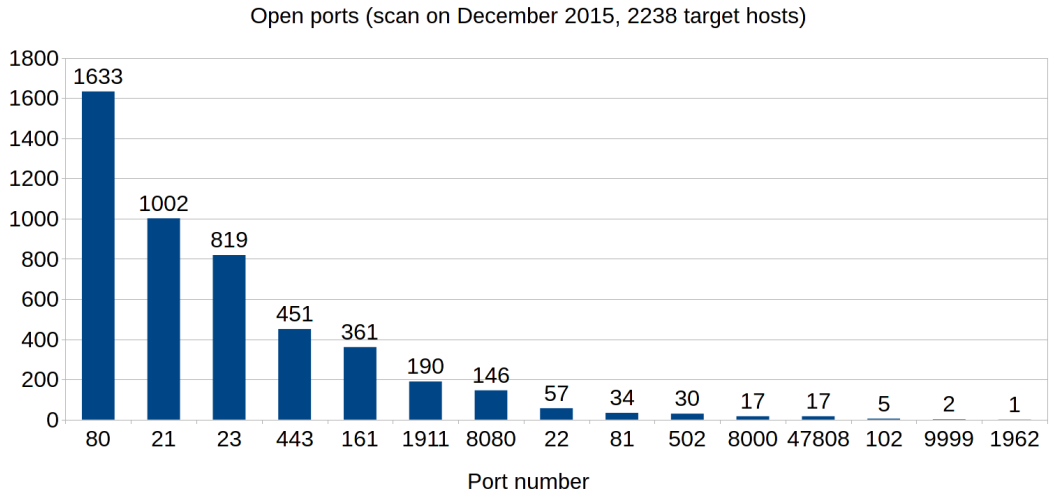

Figure 9: Distribution of open ports, December 2015

For those 1726 devices having at least one port open, almost 95% of them had port 80 open which suggests a responding HTTP server. Most modern devices have

a web management interface implemented which explains the high number. Other interesting and popular ports were port 21 (FTP, 1002 hosts), port 23 (Telnet, 819 hosts), port 443 (HTTPS, 451 hosts) and port 161 (SNMP, 361 hosts). FTP and Telnet are vulnerable, non-encrypted protocols that should be either replaced by modern alternatives or removed from use. Likewise, SNMP is a dangerous protocol that should be protected.

Ports 1911, 502, 47808 and 102 are directly linked to ICS related devices. A significant amount of Niagara Fox automation devices, 190 hosts, were responding to the public Internet. Similarly, Modbus devices (port 502), Ethernet/IP devices (port 47808) and S7comm devices (port 102) are possibly vulnerable and should be protected by firewalls. Remote access to these devices must be restricted to secure VPN connections only. Whether these protocols are accessible because of a configuration mistake or intentionally, having them available exposes the devices to malicious actors.

Open port combinations

Table 5 represents the most popular open port combinations during the December 2015 scan. Most devices run a standardized set of software with little or no control over what services run on the device. Therefore, devices of similar type or model tend to provide a similar set of network services and hence have same combinations of open ports. While open port combinations do not provide additional identification information as such, it can be used in cluster analysis when evaluating device classification.

Port combination	Amount
21, 23, 80	536
80	193
21, 80, 443, 161	181
80, 1911	150
21, 80, 443	81
23, 80	76
80, 443, 161	69
21, 23, 80, 161	52
21, 23, 80, 8080	41
80, 8080	39

Table 5: 10 most popular open port combinations, December 2015

Scanning times

Scanning took significantly more time than expected in the beginning of prototype development. Scanning times for each scan attempt are presented in table 6. Most of the slowness is due to Nmap's synchronous scanning methods which progress in a fixed sequence. The use of UDP scanning forces Nmap to wait for timeouts on

non-responsive hosts. Additionally, operating system detection adds up to several seconds of scan time on each target host and service scan scripts also take their own time to process. Overall, the scanner setup is not optimized for speed but to get the most precise scan results possible. The scanning speed varied between scanning attempts from 3,5 seconds per host to over 5 seconds per host. There seems to be no correlation between the amount of targets and the overall scan time which could mean that Nmap parallelization is sub-optimal and the overall performance depends on external factors, e.g. network condition. Optimizing Nmap scanning options and reducing timeout periods would help the scanner to perform faster in the expense of not finding as many vulnerable hosts. The best way to improve overall scanning capacity is to distribute targets among a larger pool of scanning nodes, each operating individually in parallel and thus bringing the overall scan time down.

Date	**02.10.**	**16.10.**	**21.10.**	**11.11.**	**25.11.**	**04.12.**
Scan time (s)	7396,55	9625,76	6246,02	5856,79	7439,31	11250,25
Scanned hosts	1768	1783	1783	1631	1631	2238
Time per host (s)	4,18	5,40	3,50	3,59	4,56	5,03

Table 6: Scan times

5.3.3 Encountered issues and other notes

Occasionally, for the reasons unknown, service scans failed on a few target hosts even though the corresponding ports were open. Because of sufficiently large timeout values used in the scanning scripts, the failure of the scan cannot be explained by a slowly responding device. This phenomena was also studied with packet captures and it was observed that both request probe and response were sent and received correctly in network layer. However, for some reason, Nmap was never able to receive the response from server. It can be assumed that compatibility between Nmap and some server software is not 100% guaranteed and in some rare cases a response from server is lost.

More aggressive scanning methods were disabled in service scan scripts in order to keep the scanning safe for every target device and to respect the privacy laws to their fullest extent. BACnet gateway can reveal additional targets when querying BDTs and FDTs but they would most likely be inside private networks and, therefore, out of reach for legal scanning methods. Likewise, the full address space in Modbus networks can be probed for device discovery but hammering the network with probes may disturb some delicate controller. Additional information is not worth risking the reliable functionality of any automation system.

Connection timeout tweaking was a bit problematic because of several slowly responding devices. A safe timeout for SNMP scanning was set to 15 seconds wheres another service suffering from slow response times, Telnet, had a timeout value set to 10 seconds. These were evaluated to be long enough without risking unintentional connection loss from a slow device. Nmap performance options were also experimented with; By restricting the operating system detection retries, a few

second speedup was achieved without losing detection accuracy. Also, by increasing the minimum host group for parallel scanning, some speedup was gained for the overall scan job duration.

5.4 Summary

The prototype was implemented successfully based on the model presented in chapter 4. The functionality was split into two modules, a scanner and a classifier. The scanner was configured to have ICS scanning presets, targeting common network services in ICS devices and performing an advanced service scan when possible. 16 TCP and 2 UDP ports were selected as a suitable target port set. The prototype was tested in a virtual machine environment by scanning Conpot honeypot software instances. After the scan results were positive, the prototype scanner was tested in real environment. The testing was performed targeting less than 2000 Finnish IP addresses that were queried from Shodan using a carefully selected set of ICS related keywords. The scan was performed multiple times during late 2015 while still improving the service scan scripts and building fingerprinting rulesets and a detection database for the classifier. In December 2015, the classifier was able to detect scanned devices up to 99% success rate. This was possible thanks to fine tuned detection rulesets which were able to perform efficient matching on service scan fingerprint entries. Long scan times was found to be a possible culprit for larger scale scans. However, it was determined that UDP scanning combined with operating system detection probes and additional service scans were the possible cause of slowdown. As a solution, a distributed scanning with concurrently running scan nodes was advised.

6 Conclusions

The importance of securing national critical infrastructure has been recognized worldwide in recent years. Critical facilities, e.g. power plants and factories operate on industrial control systems that need to be protected from malicious actors. In the past, these systems were isolated from outside networks thanks to the use of direct serial connections between devices. Companies valued system availability and the physical safety of equipment and personnel over information security. However, the recent introduction of networked system architectures and remote management connections improves cost-efficiency but exposes these automation systems to a completely new set of cyber security threats which in turn creates pressure to bring information security policies up-to-date. Insufficient isolation between office networks and field automation networks enables attackers to create clever remotely executed cyber attacks even if field networks are not directly connected to the Internet. Attacks to high-risk targets have been rare so far but the attacks against Iranian nuclear centrifuges [7] and a German steel mill [9] have demonstrated the potential of a well-planned cyber attack. Shodan [19], a search engine for Internet connected devices, has been a great asset in finding exposed ICS devices, and has helped cyber security researchers and governmental agencies to track down and alert companies about potential security risks. However, research based on Shodan scan databases has revealed that the exposure of ICS devices is still a valid concern and new methods for finding and securing vulnerable systems are in demand. The increasing popularity of IoT is expected to make the cyber security situation even worse by bringing millions of new devices accessible on the Internet. [20, 21]

Issues in automation cyber security are enabled by weaknesses in organizational and technological security policies but the root cause is the inherent lack of security features in industrial automation systems. Common ICS network protocols have been retroactively modified to work over IP based networks but are missing proper authentication and cryptography features in order to maintain compatibility with old hardware. Several studies have demonstrated numerous exploits against architectural weaknesses in these protocols. Suggested ways for cyber attack mitigation are sufficient network isolation by firewalls and VPNs together with implementation of proper intrusion detection systems. [17, 16]

The challenge in protecting critical automation infrastructure lies in finding the exposed devices among billions of others connected to the Internet. Port scanning is an effective way to fingerprint available devices based on open network services they provide. ICS devices tend to be lightly configured and therefore provide device identification information when probed with general and ICS specific network protocols. Vendor specific automation protocols have in most cases built-in diagnostic instructions which enables the scanner to pull device identification and other operational information from a device. However, port scan procedures should always comply with regional legislation to ensure the safety and privacy of computer systems even if it means reducing the efficiency of device probing methods. In addition to port scanning, operating system detection and geolocation data has great value in device fingerprint.

In this thesis, a model for ICS device assessment and classification is proposed. This model takes in scan reports containing device probing fingerprints and analyses multiple device features, e.g. software vulnerabilities, location and available services, combining all results into assessed threat level and positive device identification. The model scales well into several network sizes by adjusting the pool of scanning nodes. Also, new analyzable features can be added to the process to increase the accuracy of device assessment.

To validate the feasibility of the proposed model, a proof of concept prototype was created. The prototype implements a set of key features of the model, being able to perform elaborate service scans on target hosts and classify them based on predefined rulesets. During the prototype development, a total of 64 detection rules for different device types were made. After the prototype was tested with virtual honeypot environment, a target set for testing in real environment was compiled by searching Shodan with 41 distinctive ICS keywords. The target set was scanned 6 times in late 2015 and the successful device detection ratio improved from 70% up to 99% during the prototype development cycle. Even though the prototype was slow, taking from 3,5 to 5 seconds per scanned host, it was fairly reliable with different types of devices.

Future work

The prototype software was designed to be an extensible framework where new features are easy to add. Scanner can be improved by additional service scan scripts for new protocols and classifier functionality can be improved by adding new identification rule types. The more advanced features of the model presented in chapter 4 but not part in the initial prototype version, e.g. software vulnerability analysis, can be implemented into the scanner by utilizing Nmap's scripting engine. Also, other features which help to assess the device purpose would be key improvements for the future versions of the prototype. Security auditing by checking default credentials is currently implemented for selected protocols in the prototype but the feature is disabled by default because of its intrusive nature.

The classifier has only 64 device detection rules in its ruleset library in the initial version. Usefulness in device detection relies on comprehensiveness of the detection library and therefore a large set of identification rules for a wide range of ICS device types should be compiled before using the prototype in different environments.

Another useful feature addition would be the introduction of distributed scanning nodes as the initial version is configured to be used as a single node. Scanning nodes can be lightweight and contain only the minimal software needed to do the scanning jobs whereas the classifier node can operate a central server in charge of scan target allocation between scanning nodes.

Finally, more research is needed to create mechanisms for integration between the prototype and other security systems. An automatic system for target scanning, classification, reporting and alerting is needed for auditing national cyberspace and protecting critical infrastructure.

References

[1] Puolustusministeriö, *Suomen kyberturvallisuusstrategia: Valtioneuvoston periaatepäätös 24.1.2013*. Puolustusministeriö, 2013.

[2] Centre of the Protection of National Infrastructure, "Securing the move to IP-based SCADA/PLC networks." Online: `http://www.cpni.gov.uk/documents/publications/2011/2011034-scada-securing_the_move_to_ipbased_scada_plc_networks_gpg.pdf`, November 2011. Retrieved Aug 4, 2015.

[3] Ponemon Institute, "State of IT Security: Study of Utilities and Energy Companies." Online: `http://www.ponemon.org/local/upload/file/Q1_Labs%20_WP_FINAL_3.pdf`, April 2011. Retrieved Dec 25, 2015.

[4] WHO, "Chernobyl: the true scale of the accident." Online: `http://www.who.int/mediacentre/news/releases/2005/pr38/en/`, September 2005. Retrieved Nov 15, 2015.

[5] J. Finkle, "Malicious virus shuttered power plant: DHS." Online: `http://www.reuters.com/article/us-cybersecurity-powerplants-idUSBRE90F1F720130116`, January 2013. Retrieved Jan 28, 2015.

[6] C. Steitz and E. Auchard, "German nuclear plant infected with computer viruses, operator says." Online: `http://www.reuters.com/article/us-nuclearpower-cyber-germany-idUSKCN0XN2OS`, April 2016. Retrieved Apr 30, 2016.

[7] D. Kushner, "The Real Story of Stuxnet." Online: `http://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet/`, February 2013. Retrieved Dec 28, 2015.

[8] J. Halliday, "Wikileaks: US advised to sabotage Iran nuclear sites by German thinktank." Online: `http://www.theguardian.com/world/2011/jan/18/wikileaks-us-embassy-cable-iran-nuclear`, January 2011. Retrieved Dec 29, 2015.

[9] R. Lee, M. Assente, and T. Conway, "German steel mill cyber attack." Online: `https://ics.sans.org/media/ICS-CPPE-case-Study-2-German-Steelworks_Facility.pdf`, December 2014. Retrieved Jan 15, 2016.

[10] F-Secure Labs, "Blackenergy and Quedagh: The convergence of crimeware and APT attacks." Online: `https://www.f-secure.com/documents/996508/1030745/blackenergy_whitepaper.pdf`. Retrieved Feb 21, 2016.

[11] K. Wilhoit, "Killdisk and BlackEnergy Are Not Just Energy Sector Threats." Online: http://blog.trendmicro.com/trendlabs-security-intelligence/killdisk-and-blackenergy-are-not-just-energy-sector-threats/, February 2016. Retrieved Mar 20, 2016.

[12] P. Polityuk, "Ukraine sees Russian hand in cyber attacks on power grid." Online: http://www.reuters.com/article/us-ukraine-cybersecurity-idUSKCN0VL18E, February 2016. Retrieved Mar 21, 2016.

[13] K. Stouffer, S. Lightman, V. Pillitteri, M. Abrams, and A. Hahn, "Guide to industrial control systems (ICS) security," *Recommendations of the National Institute of Standards and Technology, Special Publication 800-82 revision 2*, 2015. Retrieved Nov 21, 2015.

[14] V. Igure, S. Laughter, and R. Williams, "Security issues in SCADA networks," *Computers & Security*, vol. 25, no. 7, pp. 498–506, 2006.

[15] Modbus Organization, "Modbus application protocol specification." Online: http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf. Retrieved Mar 15, 2016.

[16] W. Gao and T. H. Morris, "On cyber attacks and signature based intrusion detection for Modbus based industrial control systems," *Journal of Digital Forensics, Security and Law*, vol. 9, no. 1, pp. 37–56, 2014.

[17] D. Beresford, "Exploiting Siemens Simatic S7 PLCs." Online: https://media.blackhat.com/bh-us-11/Beresford/BH_US11_Beresford_S7_PLCs_WP.pdf, July 2011. Retrieved Aug 21, 2015.

[18] Anonymous author, "Internet Census 2012." Online: http://internetcensus2012.bitbucket.org/paper.html, January 2013. Retrieved Feb 10, 2016.

[19] J. Matherly, "Shodan." Online: http://www.shodan.io. Retrieved Jul 25, 2015.

[20] Infracritical, "Project SHINE - Findings Report." Online: http://www.slideshare.net/BobRadvanovsky/project-shine-findings-report-dated-1oct2014, 10 2014. Retrieved Oct 20, 2015.

[21] Nixu, "Analysis of the Internet Census data - The Finnish Cyber Landscape." Online: https://www.nixu.com/en/insights/analysis-internet-census-data-finnish-cyber-landscape, 10 2013. Retrieved Mar 12, 2016.

[22] S. Tiilikainen, "Improving the National Cyber-security by Finding Vulnerable Industrial Control Systems from the Internet," Master's thesis, Aalto University, February 2014.

[23] Huoltovarmuuskeskus, "Havaro turvaa yhteiskunnan huoltovarmuuskriittisiä toimintoja." Online: http://www.varmuudenvuoksi.fi/aihe/huoltovarmuuden_toteutuksia/106/havaro_turvaa_yhteiskunnan_huoltovarmuuskriittisia_toimintoja. Retrieved Nov 18, 2015.

[24] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," RFC 2616, Internet Engineering Task Force, June 1999.

[25] E. Rescorla, "HTTP Over TLS," RFC 2818, Internet Engineering Task Force, May 2000.

[26] J. Postel and J. Reynolds, "File Transfer Protocol," RFC 959, Internet Engineering Task Force, October 1985.

[27] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "Simple Network Management Protocol (SNMP)," RFC 1157, Internet Engineering Task Force, May 1990.

[28] J. Postel and J. Reynolds, "Telnet Protocol Specification," RFC 854, Internet Engineering Task Force, May 1983.

[29] T. Ylonen and C. Lonvick, "The Secure Shell (SSH) Transport Layer Protocol," RFC 4253, Internet Engineering Task Force, January 2006.

[30] BACnet Advocacy Group, "BACnet addenda and companion standards." Online: http://www.bacnet.org/Addenda/. Retrieved Sep 10, 2015.

[31] "SCADA strangelove." Online: http://scadastrangelove.org. Retrieved Sep 5, 2015.

[32] Digital Bond, "Plcscan." Online: http://www.digitalbond.com/tools/plcscan/. Retrieved Sep 6, 2015.

[33] Digital Bond, "Digital Bond's ICS Enumeration Tools." Online: https://github.com/digitalbond/Redpoint. Retrieved Sep 6, 2015.

[34] Rockwell Automation, "EtherNet/IP: Industrial Protocol White Paper." Online: http://literature.rockwellautomation.com/idc/groups/literature/documents/wp/enet-wp001_-en-p.pdf. Retrieved Sep 6, 2015.

[35] D. Richardson, S. Gribble, and T. Kohno, "The limits of automatic OS fingerprint generation," *ACM workshop on artificial intelligence and security (AISec)*, pp. 24–34, 2010.

[36] "Nmap - Remote OS Detection." Online: https://nmap.org/book/osdetect.html. Retrieved Sep 10, 2015.

[37] RIPE Network Coordination Centre, "Ripe database documentation." Online: https://www.ripe.net/manage-ips-and-asns/db/support/documentation/ripe-database-documentation. Retrieved Mar 23, 2016.

[38] MaxMind, "GeoIP2 City Accuracy." Online: https://www.maxmind.com/en/geoip2-city-database-accuracy. Retrieved Nov 12, 2015.

[39] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast Internet-wide scanning and its security applications," in *Proceedings of the 22nd USENIX Security Symposium*, Aug. 2013.

[40] SANS, "Intrusion detection FAQ: What is Scanrand?." Online: http://www.sans.org/security-resources/idfaq/scanrand.php. Retrieved Oct 15, 2015.

[41] "Nmap - Nmap Scripting Engine." Online: https://nmap.org/book/nse.html. Retrieved Sep 10, 2015.

[42] "Conpot - ICS/SCADA Honeypot." Online: http://conpot.org/. Retrieved Aug 2, 2015.