



Norwegian University of  
Science and Technology

# Security in process control systems

Rafal Szostak

Master of Science in Communication Technology

Submission date: June 2009

Supervisor: Svein Johan Knapskog, ITEM



# Problem Description

Traditionally, process control systems in automation and power industries have resided on closed networks, and ICT security has not been an issue. With the current trend of using COTS technology and integration with other networks, this is now changing. Specifically, connecting equipment via the Internet results in exposure to a different, potentially much more diverse set of threats. The intent of this project is to analyze this threat situation, identify potential vulnerabilities of process control systems in such an environment, and by performing realistic experiments to verify the validity of the assumed vulnerabilities.

Assignment given: 20. January 2009

Supervisor: Svein Johan Knapskog, ITEM



Master Thesis  
Security in process control systems

Rafal Szostak  
*Department of Telematics*  
*Faculty of Information Technology, Mathematics and Electrical Engineering*  
*Norwegian University of Science and Technology*

June 15, 2009

# Contents

<b>List of Figures</b>	<b>3</b>
<b>List of Tables</b>	<b>5</b>
<b>1 Introduction</b>	<b>10</b>
1.1 Research questions . . . . .	11
1.2 Terminology . . . . .	11
1.3 Method . . . . .	11
1.4 Structure . . . . .	12
<b>2 Background</b>	<b>14</b>
2.1 Development and threats . . . . .	14
2.2 Importance of the topic . . . . .	15
2.3 Tools used by others . . . . .	18
<b>3 Equipment</b>	<b>21</b>
3.1 Laboratory equipment . . . . .	22
3.2 Security testing tools used . . . . .	24
<b>4 System scan</b>	<b>26</b>
4.1 Nessus scan . . . . .	26
<b>5 Attacks</b>	<b>34</b>
5.1 Attacks from the outside . . . . .	34
5.1.1 Successful outsider attack: Gain control over the OS .	35
5.1.2 Successful outsider attack: Denial of Service (DoS) on OS . . . . .	37
5.1.3 Unsuccessful outsider attack: Unsuccessful Metasploit attempts . . . . .	40
5.2 Attacks from the inside . . . . .	44
5.2.1 Successful insider attack: VNC server access for the OS	44
5.2.2 Successful insider attack: Packet replay through a switch	46
5.2.3 Unsuccessful insider attacks: Force OS forwarding . .	47
5.3 Summary . . . . .	50

<b>6</b>	<b>Discussion</b>	<b>52</b>
6.1	Answers to research questions . . . . .	52
6.1.1	Is it possible to get useful information about the system that can be used in attacks later on? . . . . .	52
6.1.2	Can the system be penetrated from the Internet and in that case what damage can an attacker do? . . . . .	53
6.1.3	What kind of security mechanisms can be found looking at the system from an outside attacker using the Internet? . . . . .	55
6.2	Reflections after the attacks . . . . .	56
6.2.1	Unanswered questions . . . . .	56
6.2.2	What could have been done to find out more? . . . . .	58
6.2.3	What could have been done differently by the industry? . . . . .	59
6.2.4	Further work . . . . .	61
<b>7</b>	<b>Conclusion</b>	<b>62</b>
<b>A</b>	<b>Lab problems</b>	<b>63</b>
<b>B</b>	<b>Details about tools used</b>	<b>65</b>
B.1	Nessus scan details . . . . .	65
B.2	Cain & Abel details . . . . .	90

# List of Figures

1.1	ICS relations . . . . .	12
3.1	System overview . . . . .	21
3.2	System overview with details . . . . .	23
4.1	Options set in Nessus . . . . .	27
4.2	Nessus scan overview . . . . .	28
4.3	Information collected through a NetBIOS request . . . . .	31
4.4	Account log in possibilities . . . . .	31
4.5	HTTP information . . . . .	33
5.1	Remaining chapter division . . . . .	35
5.2	Directory displayed through Metasploit attack . . . . .	37
5.3	System info displayed through Metasploit . . . . .	38
5.4	DoS attack setup . . . . .	39
5.5	DoS attack exchange . . . . .	40
5.6	DoS attack traffic . . . . .	41
5.7	CPU usage on the OS . . . . .	43
5.8	Remote control of OS using VNC . . . . .	45
5.9	Insider attack . . . . .	46
5.10	Replay packet fields edited . . . . .	49
6.1	System remake to make forwarding between networks possible	58
B.1	Nessus scan options . . . . .	66
B.2	Nessus scan plugins used . . . . .	67
B.3	Nessus network options . . . . .	68
B.4	Advanced Nessus options . . . . .	69
B.5	Nessus scan overview . . . . .	70
B.6	Syn and Fin can be set to bypass firewall rules . . . . .	71
B.7	VNC server . . . . .	72
B.8	TFTPD is running on Port 69 . . . . .	72
B.9	NTP server is listening on port 123 . . . . .	73
B.10	Windows RPC service is in use . . . . .	74
B.11	NetBIOS in use . . . . .	75



B.12 SMB in use . . . . .	76
B.13 Port 445 overview . . . . .	77
B.14 SMB log in . . . . .	78
B.15 SMB null session log in . . . . .	79
B.16 Windows Terminal Services running . . . . .	80
B.17 VNC server running . . . . .	81
B.18 VNC on port 5900 . . . . .	82
B.19 Sentinel web server running . . . . .	83
B.20 Protocol in use on the Sentinel server . . . . .	84
B.21 Information on the last three open ports . . . . .	85
B.22 ICMP protocol . . . . .	86
B.23 TCP information . . . . .	87
B.24 Additional TCP info . . . . .	88
B.25 UDP trace route . . . . .	89
B.26 3DES reverse decryption . . . . .	90

# List of Tables

2.1	Threats to ICS . . . . .	17
2.2	Survey tools used by others . . . . .	18
2.3	Attack tools used by others . . . . .	19
4.1	Open ports on OS . . . . .	32
5.1	A summary of outsider attacks . . . . .	50
5.2	A summary of insider attacks . . . . .	51

# Acknowledgements

I would like to thank professor Svein Knapskog who not only served as my supervisor but also inspired everyone in my class through our academic programs. Thanks is also in order for Martin Gilje Jaatun from SINTEF for giving me good ideas on attacks I could try.

My good friends Torjus and Øivind also deserve praise for keeping my spirit up through the long hours of work.

Last but not least I would like to thank my family for their endless support.

# Abstract

In this thesis we have considered the security of a process control system delivered by Kongsberg Maritime. The main focus has been to look at the threats these type of systems face from the Internet. Hence, identification of security vulnerabilities of the system was made. The vulnerabilities found were then attempted exploited in attacks. Possible mitigation paths to remove these vulnerabilities are proposed as well.

# Acronyms

**AES:** Advanced Encryption Standard.

**ARP:** Address Resolution Protocol.

**COTS:** Commercial off-the-shelf.

**CPU:** Central Processing Unit.

**DCS:** Distributed Control Systems.

**DES:** Data Encryption Standard.

**DDoS:** Distributed Denial of Service.

**DoS:** Denial of Service.

**FIN:** Finished.

**HTTP:** Hypertext Transfer Protocol.

**ICMP:** Internet Control Message Protocol.

**ICS:** Industrial Control Systems.

**ICT:** Information and Communication Technologies.

**IDS:** Intrusion Detection Systems.

**IP:** Internet Protocol.

**IPv6:** Internet Protocol version 6.

**ISP:** Internet Service Provider.

**I/O:** Input/Output.

**LAN:** Local Area Network.

**MAC:** Media Access Control.

**NAT:** Network Address Translator.

**NetBIOS:** Network Basic Input/Output System.

**NTP:** Network Time Protocol.

**OS:** Operator Station

**PC:** Personal Computer.

**PCS:** Process Control Systems.

**PLC:** Programmable Logic Controller.

**PS:** Process Station.

**RPC:** Remote Procedure Call.

**SCADA:** Supervisory Control and Data Acquisition.

**SMB:** Server Message Block.

**SYN:** Synchronizing.

**TCP:** Transmission Control Protocol.

**TFTPD:** Trivial File Transfer Protocol.

**TP:** Twisted Pair.

**TTL:** Time To Live.

**UDP:** User Datagram Protocol.

**VNC:** Virtual Network Computing.

**VoIP:** Voice over Internet Protocol.

# Chapter 1

## Introduction

Process Control Systems (PCS) are designed to be efficient and provide time-critical data. Traditionally security has not been a strong design factor as performance, reliability, safety and flexibility has always been the priority. Traditionally PCS were isolated physically and based on proprietary software and hardware, with own communication channels. As the systems are increasingly introduced to the Internet new possibilities opens up, as well as new threats. Merging PCS with Information and Communication Technologies (ICT) has several consequences. Some of the consequences are introducing risks formally only known in ICT systems to PCS [19].

Since PCS are used to control electric utilities, petroleum (oil and gas), water, waste, chemicals and pharmaceuticals they are part of the critical infrastructure in the society. The consequences of cyber attacks on PCS systems could hence have very serious consequences as they involve health and safety of human lives, as well as having a huge impact on national and global economy. An attack on a pharmaceutical company for example can lead to release of hazardous substances, put human lives in danger and have huge consequences for the environment. Therefore security of PCS is vital in preventing this from happening.

To address and deal with the issue of security in PCS there was a Presidential decision directive to establish the framework for protecting critical infrastructure in the USA. This was done in 2003, with the National Strategy to Secure Cyberspace and stated that PCS systems are a national priority[24]. This is an implication that the issue described in this thesis is of great importance and has to be taken seriously. Especially after the terrorist attacks on the two towers, September 11'th 2001, the awareness of how important the security of critical infrastructure is has increased significantly.

## 1.1 Research questions

This section will present some research questions we will try to answer throughout the thesis.

The questions are as following:

- Is it possible to get useful information about the system that can be used in attacks later on?
- Can the system be penetrated from the Internet and in that case what damage can an attacker do?
- What kind of security mechanisms can be found looking at the system from an outside attacker using the Internet?

## 1.2 Terminology

In this section we will present some terms used in the thesis, how they are connected together and used.

In the literature there are many terms used on the systems described and some of them overlap in certain areas. Industrial Control Systems (ICS) is a general term used in the industry for many types of control systems. ICS includes Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS) and other smaller systems [17]. Both DCS and SCADA systems are used in the industry to control distributed system from a master location. The main difference between the two is the location. SCADA systems are geographically spread over large areas while the DCS usually cover systems on a plant. That is why DCS systems usually communicate over a Local Area Network (LAN). Both SCADA and DCS systems is a collective name and also cover PCS. From now on we will use the term PCS instead of DCS or SCADA. The reason for this is that in our case we are working on a scaled down PCS system.

## 1.3 Method

There is a need to define the scientific methodology used in this thesis. The definition of a hypothesis along with carrying out tests to support the hypothesis does only partly apply to our technological specialization field.



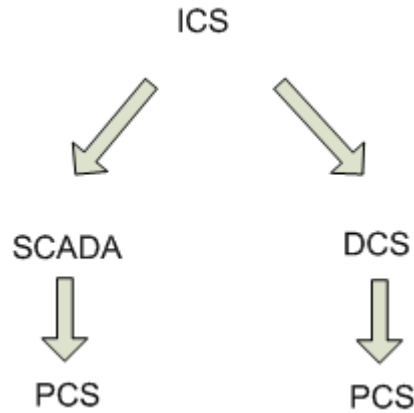


Figure 1.1: ICS relations

The research questions defined in Section 1.1 along with the task description are suited as a hypothesis that has to be tested. Hypothesis testing does not entirely cover the work done in this thesis. The Engineering method<sup>1</sup> described by Glass [22] is more suitable in our case. The reason for that the Engineering method is better suited in our case is that a series of tests will be performed on the equipment provided by Kongsberg Maritime, the results will be analyzed and improvements will be suggested. Hence not only hypothesis testing will be done.

In the thesis background details about the protocols, ports and services used will not be investigated in detailed depth, since the focus is on the testing of the equipment.

## 1.4 Structure

The remainder of the thesis is structured as follows:

**Chapter 2** provides background information about the system covered in this thesis as well as similar systems. The chapter describes the development of ICS and PCS systems regarding cyber security and the threats to the systems. Finally the chapter covers some tools used by others to attack ICS and PCS systems to get the scope of attack possibilities.

**Chapter 3** describes the laboratory equipment. Each of the components in the PCS system is described. In addition the tools used to carry out the attacks are described as well along with their area of application.

---

<sup>1</sup>Engineering method consists of: observing a solution, propose better solutions, build or develop, measure and analyze, repeat until no further improvements are possible.

**Chapter 4** gives an overview of the system scan. The scan is later used as a basis for the attacks. It also defines the scope for what possibilities there are to attack the system.

**Chapter 5** covers the attacks. This chapter includes a scan that derives information about the system. This information is then used in the attacks described in this chapter. Both attacks done from a Internet perspective and attacks that can be done by an insider of the system are covered here.

**Chapter 6** is where the discussion is presented. The various questions presented during the previous chapters like the research questions, questions from the attacks carried out and unanswered questions are covered in this chapter.

**Chapter 7** sums up the thesis. A conclusion is drawn based on the experiences gained throughout the working process.

## Chapter 2

# Background

In this chapter we will present the development of ICS and introduce the threats that these systems face today. The importance of the topic described throughout the thesis will be presented, which involves giving examples of attacks done on PCS and similar types of systems. In the next section material about the tools used in attacks carried out by others are described. The purpose of this is to get an overview of the work done by others and use this information to try similar tests in our laboratory, if possible. The attacks done by others can also give ideas on approaches to try that has not been done before or adjust known methods to our purpose.

### 2.1 Development and threats

In the early days of SCADA and PCS serial communication was used. Because of the limitations of this technology, such as low channel capacity, only the most vital information used to keep control over the systems were exchanged and there were almost no attention to security of the information. The information was sent in clear text and the remote devices accepted it without any kind of authentication. This lack of security did not pose major problems because the ICS were isolated.

As the ICT developed and new technologies were introduced other possibilities became available. The new information exchange technologies were better in regards to the amounts of information exchanged. There were many products capable of the information exchange, yet few that actually were applicable to the rigorous field conditions or were suited for industrial application. The adaptation of the Ethernet technology became increasing for indoor industrial fields. As for the outdoor technologies other factors had to be taken into account such as temperature variations, power consumption and the range of the information exchange. For example, the information exchange had to be tested in temperature variations spanning from  $-30^{\circ}\text{C}$  to  $+60^{\circ}\text{C}$ . The power consumed had to be as low as possible, since most oil

and gas systems run on solar and battery powered systems [21].

The technology evolution lead to application of various technologies in ICS dependent on the industrial environment and needs. Today the information is exchanged through Ethernet, wireless, shared leased lines, and even the Internet [31]. These communication channels are inhomogeneous and less isolated than the original ICS systems were. The mixing and lack of isolation of communication channels poses a serious threat to ICS because forgery and control loss now pose a new threat.

The adaptation of previously isolated ICS to the new communication channels such as wireless, shared leased lines and the Internet, as well as adaptation of Commercial off-the-shelf (COTS) products such as Microsoft Windows has many advantages, but it has also created threats previously unknown to these systems. Now ICS has to face common cyber attacks which means that the focus has to be increased on security measures. In contrast to ordinary ICT systems such as personal computers (PC) the security breach in ICS does not only result in a compromise of the security on the individual computer, but it can result in loss of service to utility customers, financial loss to service providers due to damaged equipment and corruption of metering information, and finally environmental damage and potential loss of human lives [19]. In the next section this topic is continued and examples of security breaches along with their consequences are described.

## 2.2 Importance of the topic

Like mentioned in the first chapter the PCS are part of the critical infrastructure in the society. Protection of them is a very important issue. To stress this point we will show examples of things that can occur if security in PCS and similar systems is breached.

2003 was the worst year for viruses until then, according to the F-Secure website <sup>1</sup>. There were several major worms and viruses that had very big impact on the ICT industry and hence also on ICS. Amongst them two had especial impact on the power grid network.

The first one was the Slammer worm. On January 25th the worm began hitting computer networks around the world. It was the biggest worm attack on the Internet ever with the worm trying to hit all possible addresses on the Internet (theoretically 4 billion) in less than 15 minutes <sup>1</sup>. It exploited a vulnerability in the Microsoft SQL database. The computers that were not patched for the vulnerability got affected by the worm. Amongst many others the worm spread to a nuclear power plant in Ohio, USA. The breach did not pose a safety hazard since the plant was offline but just the fact

---

<sup>1</sup>The annual reports can be found on [http://www.f-secure.com/en\\_EMEA/security/security-lab/latest-threats/security-threat-summaries/2003.html](http://www.f-secure.com/en_EMEA/security/security-lab/latest-threats/security-threat-summaries/2003.html)

that the virus got into the system is a major security breach. The virus got into the plant through an unsecured corporate network connected to the plants network and infected at least one of the plants unpatched SQL servers. Administrators were not even aware that there was a patch, which Microsoft released six months before Slammer struck. The virus overloaded the plants network by continuously trying to spread leading to malfunction of the plants monitoring system controlling various parts of the plant, such as the coolant systems, core temperature sensors, and external radiation sensors. Many of those continue to require careful monitoring even while a plant is offline [30].

The second worm that had major impact on the power grid in 2003 was the Blaster worm. Blaster utilized known vulnerabilities in the Microsoft Remote Procedure Call (RPC) protocol [14] and slowed down the communication links on the ICS. This prevented real-time data from reaching the control centers and they did not know about a generator and multiple line failures. The lack of control lead to a cascade of blackouts along the north east coast of the USA and Canada [25].

These two incidents along with many similar ones caused by the Slammer and Blaster worms are examples of how vulnerable parts of the critical infrastructure such as the power grid can be for these types of malicious attacks.

Not only worms are affecting the ICS. There are also security breaches done by foreign agencies. On the 8th of April 2009 an article was posted in the Wall Street Journal describing espionage attempted by Chinese and Russian spies on the power grid of the USA [23]. According to the article the attackers tried to map the power grid and left software that can be activated during times of crisis or war to disrupt the function of the grid.

Disgruntled ex-employees pose a major threat as well. In 1992 a disgruntled former employee hacked into Chevron Corporation's emergency alert network and disabled the alert system. This was not noticed until an accidental chemical release 10 hours later in a Chevron refinery in Richmond. During this time the emergency alert system in 22 states was not functioning and thousands of lives were potentially in danger if an emergency had occurred [16]. According to FBI and the Computer Security Institute on Cybercrime, released in 2000 over 71% of the security breaches carried out on ICS were done by insiders [13]

Table 2.1 shows an overview of various threats to the ICS with a short description of each one.

Threat source:	Description:
Attackers	Attackers break into the systems for the thrill of the challenge or to brag about it in hacker communities. As tools have become more available to the public and more sophisticated these types of attacks are becoming more frequent. Isolated attacks may have serious consequences.
Bot-network operators	Bot-network operators are attackers that have control of several systems and can attack in a coordinated fashion. Sometimes the services of compromised systems are available for purchase in underground communities (like denial of service attacks, spam or fishing attacks)
Criminal groups	Criminal groups seek profit from attacking systems (usually in the form of money). Especially organized crime groups use various approaches (like spam, phishing, spyware/malware) to achieve their goal of profit.
Foreign intelligence	Foreign intelligence use various tools for information gathering. This can be later be used to disrupt vital parts of the critical infrastructure in case of military actions or political confrontations.
Insiders	Insiders such as disgruntled employees or former employees have unique system knowledge. Through their experience and access to the inner parts of the system they can cause damage to the system or outsource information. Intentional impact from insiders are the most common threat source.
Terrorists	Terrorists aim to destroy or incapacitate critical infrastructure such as ICS. The goal is usually to threaten national security, cause massive casualties and weaken the public morale.

Table 2.1: Threats to ICS

## 2.3 Tools used by others

We will divide this section in two and look at tools that are used to make a survey of a system and tools that are used after the information is gathered by the survey tools and attack a system using that information.

There are various tools used to make a survey of a system. Table 2.2 shows some of the survey tools used by others, a description of each tool and a reference to the material this is based upon.

Regarding the attack tools used by others, there are very many to choose

Tool name:	Description:	Reference:
Nmap	Nmap is a free and open source tool for network exploration or security auditing. It can do port scans on large networks as well as single hosts. A port scan is used to specify services running on a host. Packet filter and firewalls can also be detected.	[9] [28] [33]
IP Stack Integrity Checker	The integrity checker can exercise the stability of an IP Stack and its component stacks such as TCP, UDP, ICMP etc. It generates piles of pseudo random packets of the target protocol and the packets can be configured to penetrate the firewall rules or find bugs in the IP stack.	[5] [33]
Ethereal	Ethereal is a open source network protocol analyzer that allows monitoring of communications between components. The communication can be analyzed to find the communicating parts, encryption used, information exchanged etc.	[4] [28]

Table 2.2: Survey tools used by others

from. The tools can be very specific, as well as very general with a lot of

adjustment options. Table 2.3 shows some of the tools used by others that can be relevant in our case:

Tool name:	Description:	Reference:
iOpener	Identifies every unit in a SCADA system and is able to control it without authorization of the main control unit.	[29]
Metasploit	Metasploit framework is used for developing, testing and the use of exploit code on various systems. It is described in greater detail in Section 3.2	[6] [28]
Hydra	Is a fast network logon cracker. Hydra supports numerous protocols and it is known to be flexible, as well as fast.	[10] [33]
Fuzzers	Fuzzers are used for sending invalid input to an application to force abnormal behavior. The behavior can be studied to find vulnerabilities. Fuzzers can be configured with wide range of information sent.	[28]
Netwox	A suite of tools for various purposes such as e.g. spoofing and brute forcing. It is described in greater detail in Section 3.2	[26]
Wikto	Wikto is a tool used for server assessment. More information about it can be found in the description below.	[27]

Table 2.3: Attack tools used by others

The tool iOpener is used by Langer Communications with Ralph Langner as a representative. When he has problems with convincing asset owners about how important security is he runs iOpener on the asset owners systems. According to an article in the Digital Bond the program identifies every Programmable Logic Controller (PLC) in a SCADA system and is



able to control it by i.e. turning it on and off.[29] In practice this means that you can control various parts of the systems like i.e. engines, ventilation system, pressure units and so on remotely without authorization from the main controller unit. This is very alarming from a security perspective. This would be very interesting to try on our system, even though it is a PCS system, not a SCADA system. The similarities may be sufficient for it to work. The article also mentions that iOpener is made to not be vendor specific and works on equipment from several vendors.

Wikto is a tool used for server assessment. It is based on Nikto which is a server scanner but Wikto also has additional features such as brute force fuzzing, basic web server directory crawling and Google hacks to identify poor protection [27]. Even though there is no connection to a server in our system, the laboratory may have similar weaknesses as an web server and hence pose a threat if the vulnerabilities are exploited.

## Chapter 3

# Equipment

The equipment in the laboratory is delivered by Kongsberg Maritime. It has to be noted that this equipment is for laboratory purposes, so the system is a small part of the whole system. The laboratory system may also not correspond with the system used by Kongsberg Maritime in regards to the security mechanisms used. The live system used has higher requirements regarding security than the laboratory system has.

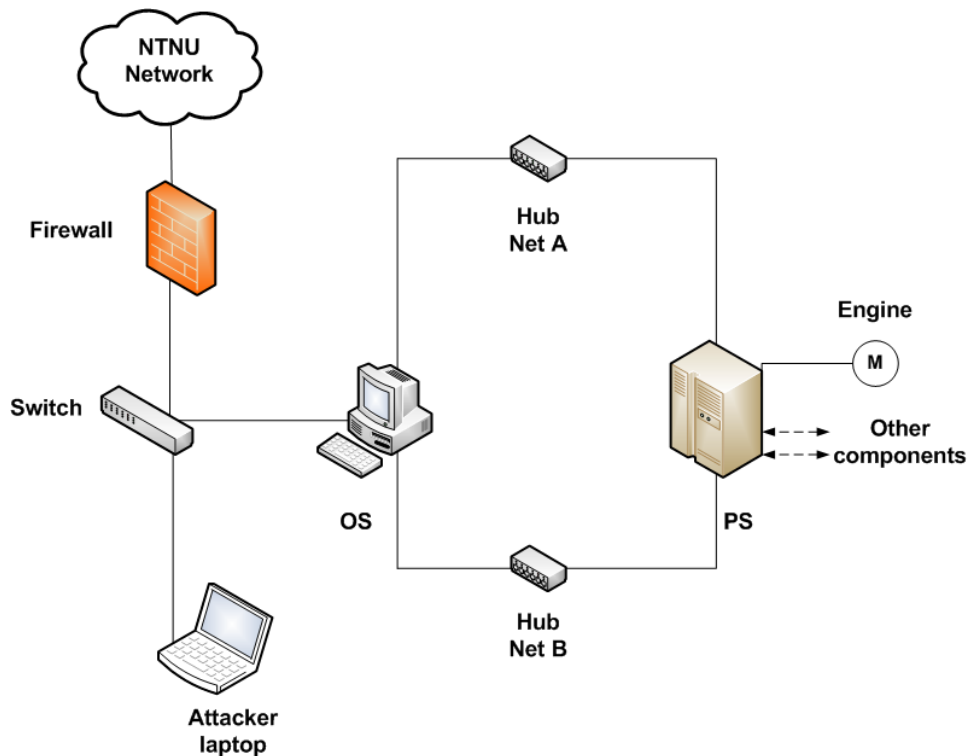


Figure 3.1: System overview

### 3.1 Laboratory equipment

The laboratory equipment consists of several units. The core unit is the Process Station (PS) AIM 1000 Albatross. It controls several components like an engine component, pressure valves and several other parts and sensors. The PS gathers information from the components and sensors continuously. In a full scale system there can be several PS stations scattered across the production plant to gather information on nearby machinery.

The Operator Station (OS) is using the operative system Windows Xp with Service Pack 2. OS exchanges data with the PS on various components and sensors. Through this communication the OS has continuous control over the units. The communication is exchanged through standard Ethernet with UDP being the main data carrier. Connections between all the components are wired with TP cables. The program used on the OS for control of the PS is AIM 2000. AIM 2000 displays an overview over various parts of the system along with the ability to simulate system behavior. We will limit ourselves to looking at the use of the analog and digital I/O of the PS controlled by the OS in this report. The control of the I/O is done through AIM 2000. The limitation is done because we are only working on an engine component. The choice of that component is based on the fact that it is easy to determine if it is turned on or not and in our case if the attack is successful or not. Other components connected to the PS could be used but could be harder to observe. The firewall used in the system is a hardware firewall. There is a software firewall in the OS as well but it was turned off for the purpose of this testing because the properties were configured in advance by the NTNU staff to not permit any communication with the outside world (which in this case is the rest of the NTNU network). The complete overview of the system is shown in Figure 3.1.

The OS has 3 network cards. One is used for network A (with the IP address 172.21.101.1). The second one is used for network B (with the IP address 172.22.101.1). The corresponding IP addresses for the PS are hence 172.21.100.1 on network A and 172.22.100.1 on network B. By listening (sniffing) on the information exchanged between OS and PS we determine that all the traffic is sent on a “virtual” network AB with the address 172.23.101.1 (and 172.23.100.1 for the PS interface). This is a reliability mechanism for the system. If one of the hubs (displayed in Figure 3.1) starts to malfunction or has to be shut down for some reason, the other one can continue the packet exchange and keep the system running. The last network card is used for network ADM. It has assigned a local IP address 10.122.10.5. The hardware firewall significantly limits the traffic through the ADM interface to the NTNU network and Internet. Only addresses with the NTNU address space of 129.241.—.— are allowed and all other addresses are unreachable. The hardware firewall also contains a Network Address Translator (NAT). NAT translates from local address space e.g. 10.122.10.5

to a 129.241.187.83 (NTNU address space). Due to the fact that the firewall was set up by NTNU and is not part of the laboratory system the security testing of the system will be done by an attacker on the inside of the firewall. Consequences of an attacker on the inside of the firewall will be covered in the discussion chapter. The IP address of the attacker is 10.122.10.3, which means that he is on the same subnet as the OS and can correspond with it easily. A detailed overview of the system is shown in Figure 3.2.

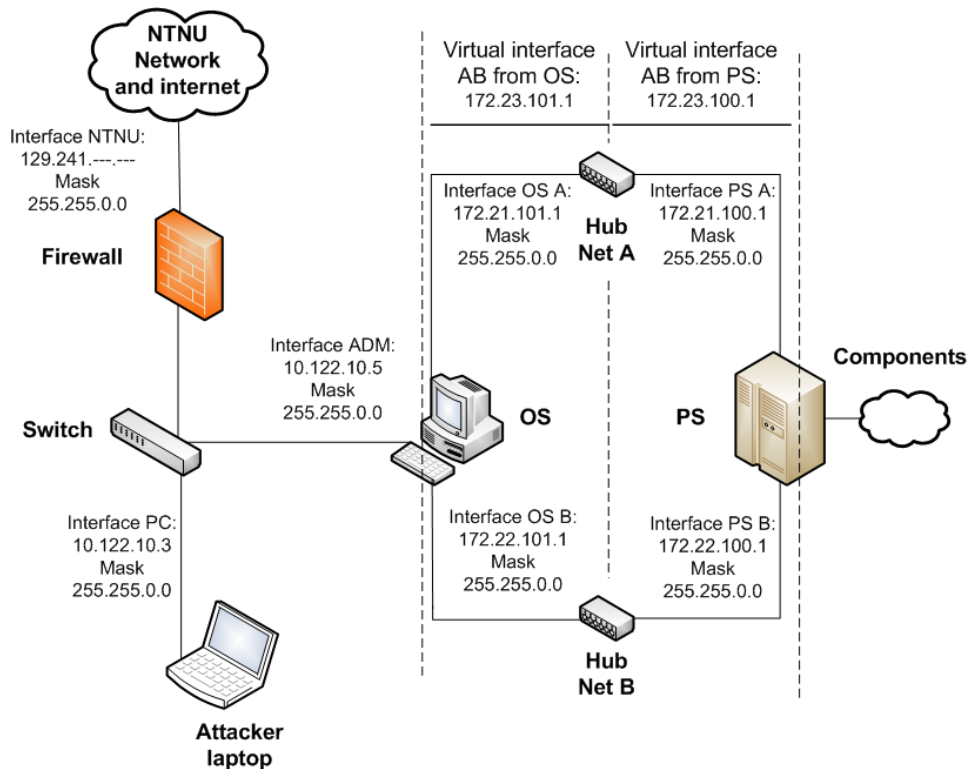


Figure 3.2: System overview with details

## 3.2 Security testing tools used

Below is a list of programs used. They are commonly available, multi platform and tailored for different purposes.

### **Cain & Abel**

Cain & Abel is mainly a password recovery tool. It can recover passwords by sniffing the network, cracking encrypted passwords using Dictionary, Brute-Force and Cryptanalysis attacks, recording VoIP conversations, decoding scrambled passwords, revealing password boxes, uncovering cached passwords and analyzing routing protocols [1].

### **Colasoft Packet Builder**

The Packet Builder is used for altering packets. It displays a clear interface of what the packet content is and enables a user to alter different parameters of the packet. When the modification of the parameters is made the program calculates and attaches a new checksum on the end [2].

### **Colasoft Packet Player**

This program is used for packet replay. It can open captured packets and trace files, then replay them back to the network. In our case we used Wireshark (described below) to capture the packets and Colasoft for the replay [3].

### **Metasploit**

Metasploit is a framework for developing, testing and using exploit code consisting of various tools. This allows various use for developers, testers and administrators to see if their systems are properly secured. Security researchers use it as well to find and expose flaws. The framework contain 320 exploit modules.

The use of Metasploit is usually done in two stages. First you select the exploit you want to use. The exploits are sorted by operating system, type of service or protocol and finally the version where the exploit can be used. An example is Windows/SMB/ms\_08\_067\_netapi which exploits a flaw in the windows operating system, using the Server Message Block (SMB) which is an application-level network protocol used for sharing access files, printers and serial ports. Finally the ms\_08\_067\_netapi exploits a flaw in the servers path stack corruption in version \_08\_067.

The second stage in using Metasploit is selection of the payload. This payload is basically the attack method you want to use. If you have an

exploit of a stack overflow e.g. then you can select to insert a command shell payload in the overflowed stack to try to control the targeted computer through this shell. Other settings has to be selected in the payload as well, such as the protocol this attack will run over (e.g. TCP) and the version of the underlying protocol (e.g. IPv6) [6].

## **Nessus**

Nessus is a vulnerability scanner. It used to be open source program until 2005 when they closed the source code and made a license. There is a free but limited “Home Feed” license available. Despite this it is still the leading vulnerability testing program with over 20,000 plugins, security checks and scripting language support for writing you own plugins according to sectools.org [7].

## **Newag and Netwox**

Netwox is a toolbox containing 222 tools. There are various tools regarding packet sniffing, scanning, brute forcing, tampering (spoofing) and many more. The tools can be run over various protocols like TCP, UDP, Address Resolution Protocol (ARP)<sup>1</sup> etc. The toolbox is oriented towards system administrators. Netwag is the graphical user interface for Netwox [8].

## **TightVNC viewer**

TightVNC is a client software used for testing of the VNC connection. TightVNC enables to setup of a connection to a remote VNC server. After the connection is set up a desktop view of the controlled machine is achieved, as well as control of its mouse and keyboard with the mouse and keyboard of the connecting client [11].

## **Wireshark**

Wireshark is a packet sniffing program that can be used for analysis and protocol development. This program was the most useful tool in our case. It captured network traffic and saved it for analysis. Filtering of the saved traffic was also done in this program to match our specifics and use. The analysis of the protocol was simplified by the use of Wireshark [12].

---

<sup>1</sup>ARP is used for finding the hardware address when only the IP address is known.

## Chapter 4

# System scan

### 4.1 Nessus scan

Nessus was used to map the vulnerabilities of the laboratory delivered by Kongsberg Maritime. The results and settings of the tests were as follows.

**Purpose:** The purpose of this scan was to find weaknesses in the system that can be exploited in other attacks. By finding a possible weakness or several weaknesses, an attacker can try to exploit them. The scan can also give ideas for similar attacks of a different kind. By executing a scan more information about the system can be obtained. A scan gives information about the system used, open ports, protocols used and possible weaknesses that can be found with regard to the information found about the system.

**System state prior to attack:** OS and PS are both turned on and functional but without any processes being run. The Engine is not running.

Nessus was configured to use 5 of the 6 possible port scanners like shown in Figure 4.1. The LaBrea tarpitted hosts check was not used due to the fact that it is designed to be a honey pot for attackers and malicious programs such as worms and viruses. A PCS system of this kind, such as the one tested, should not have anything that attracts attackers or malicious software and hence the test is pointless. A LaBrea honeypot is frequently used in Intrusion Detection Systems (IDS) to catch and hold on to attackers.

All the plugins available in the “home version” of Nessus were set to be tested. The reason for this was that all weak spots of the system should be discovered. A option for thorough tests was also set in the options to make them as complete as possible.

All the screenshots along with an explanation of the settings in Nessus are shown in Appendix B as they are a bit too detailed to be shown in this Section.

There were problems with finding the IP address of the OS from the

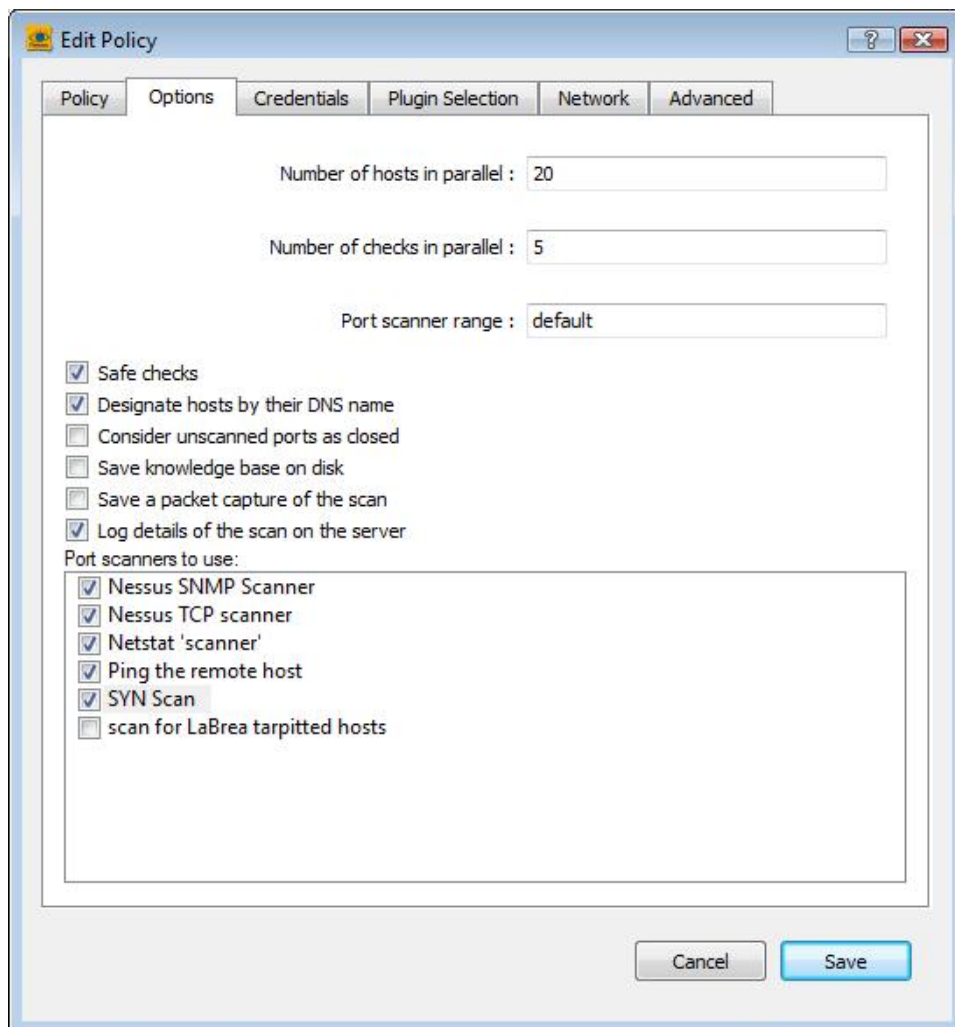


Figure 4.1: Options set in Nessus

outside of the hardware firewall. These problems are described in Appendix A

**Attack procedure:**

1. Connect to the same subnet as the OS (10.122.10.3).
2. Configure Nessus to look for known vulnerabilities.
3. Listen to the OS IP address (10.122.10.5)

*note:* In Nessus there are plugins that are simple programs which checks for a given flaw. There are currently 24525 different plugins in total that are



used by Nessus. The full version with extended plugins is called ProfessionalFeed and costs \$1200 per year, but because of limited resources only the HomeFeed could be used in our case. HomeFeed lacks many of the features like “Configuration Auditing”, “Sensitive Data Auditing” and “SCADA plugins” which could have been very useful in our case since PCS systems have similarities to SCADA systems. Still, despite not using the ProfessionalFeed with all the possibilities it gives, we got various results on the scans.

*second note:* The reason for connecting to the same subnet as the intended target scanned is that in this case the firewall hardware will cause limitations on the vulnerabilities scanned from the other side of it. This will lead to a incomplete view of the scan in regards to the visible vulnerabilities but may be more realistic in regards to attacking the system from the outside.

**Result:** The general results are shown in Figure 4.2 showing the scan time, the IP address of the scanned OS, number of open ports and the number of vulnerabilities connected to the open ports a long with a level of seriousness for the vulnerabilities. All the details of the scan (and configuration of the scan) were considered too large and are placed in Appendix B. The essence of the information gathered by Nessus for the OS are shown below:

#### Vulnerabilities<sup>1</sup>:

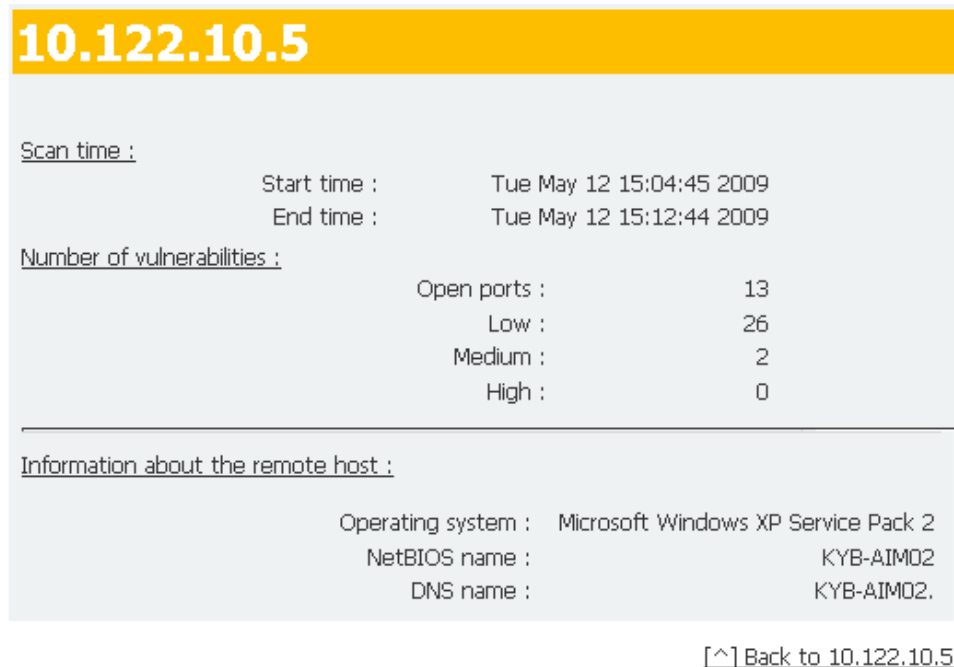


Figure 4.2: Nessus scan overview

1. It is possible to bypass the software firewall rules.
2. The OS can be controlled remotely.
3. Various information can be obtained using a NetBIOS request.
4. A user can log on through Windows Terminal Services.
5. The OS implements TCP timestamps.
6. Identification of the operating system used could be made.
7. Access to files, printers, etc between nodes on a network is possible.
8. It is possible to log on using a guest account or null session.
9. A file transfer protocol is active, which can pose a threat.
10. Two protocols that are in use enable exact time determination.
11. HyperText Transfer Protocol (HTTP) information was collected.
12. Information on the HTTP web server in use gathered.

**Vulnerability description:**

1. Since the TCP synchronizing (SYN) packets with the FIN flag are able to pass the firewall an intruder can bypass the software firewall by setting the FIN and SYN flags on a packet and establish a connection on the other side of the firewall. The firewall passes the packets since the FIN flag is set and the connection is made on the other side since the receiver acknowledges that this is a SYN packet and a connection has to be made.
2. The OS is running a Virtual Network Computing (VNC) server permitting a console to be displayed remotely. The VNC enables possibilities of remote control of the OS but at the same time creates a security vulnerability. A weak username and password can be used as a basis for an attack on the VNC server by brute force guessing or dictionary attacks.
3. Information about the OS was collected using a NetBIOS nbtscan requests. The information collected is shown in Figure 4.3 and contains information such as computer name, domain name and even the Media Access Control (MAC) address of the network card. The MAC address is the physical address of the hardware and can uniquely identify each node on the network.

---

<sup>1</sup>Sorted by seriousness. Some of the vulnerabilities found by Nessus were left out because they were insignificant

4. Terminal Services allows a Windows user to remotely obtain a graphical login (and therefore act as a local user on the remote host). An attacker may use this service to mount a dictionary attack against the OS to try to log in remotely. This service is vulnerable to Man-in-the-middle attacks and an attacker can steal the credentials of legitimate users by impersonating the Windows server.
5. The TCP timestamps implemented have a side effect which is that the uptime of the remote host can sometimes be computed.
6. Identification of the operating system used was possible. This was possible with a confidence level of 99%. The operating system used on the OS was Microsoft Windows XP Service Pack 2.
7. CIFS (Common Internet File System) is used to provide shared access to files, printers, etc between nodes on a network. There are known viruses and worms that exploit the protocol used by CIFS for a malicious purpose.
8. Since the OS is running a Windows based operation system it is possible to log on using the guest account, a NULL session (also know as anonymous users) or valid account information. The information about the account log in possibilities is shown in Figure 4.4.
9. TFTP (Trivial File Transfer Protocol) is in use. TFTP is often used by routers and diskless hosts to retrieve their configuration. Unfortunately it is also used by worms to propagate.
10. The OS responds to an Internet Control Message Protocol (ICMP) time stamp. This makes it possible for an attacker to determine the exact time and date of the OS. This may help him to defeat all your time based authentication protocols.
11. Network Time Protocol (NTP) is in use. It provides information about the current date and time of the OS and may provide system information.
12. Some information on the HTTP configuration could be extracted. The collected information is shown in Figure 4.5.

In addition information on the HTTP web server type and version. The results were that the OS is using a HTTP web server of the type: SentinelProtectionServer/7.1

**Pending issues:** Can we exploit this? If we can, then how?

**Plugin output :**

The following 4 NetBIOS names have been gathered :

KYB-AIM02 = Computer name  
KYBERNETIKK = Workgroup / Domain name  
KYB-AIM02 = File Server Service  
KYBERNETIKK = Browser Service Elections

The remote host has the following MAC address on its adapter :  
00:04:23:d0:6a:53  
CVE : CVE-1999-0621  
Other references : OSVDB:13577

Nessus ID : [10150](#)

Figure 4.3: Information collected through a NetBIOS request

**Description :**

The remote host is running one of the Microsoft Windows operating systems. It was possible to log into it using one of the following account :

- NULL session
- Guest account
- Given Credentials

**See also :**

<http://support.microsoft.com/support/kb/articles/Q143/4/74.ASP>  
<http://support.microsoft.com/support/kb/articles/Q246/2/61.ASP>

**Risk factor :**

none

**Plugin output :**

- NULL sessions are enabled on the remote host

Figure 4.4: Account log in possibilities

Port Number /and Protocol:	Description of port service:
69/UDP	The OS is running a TFTP which is often used by routers and diskless hosts to retrieve their configuration.
123/UDP	An NTP server is listening on this port, which enables exact determination of the date and time configured on the OS.
135/TCP	The OS is running a Windows RPC service which replies to Bind Requests.
137/UDP	The OS listens on port 137 and replies to NetBIOS nbtscan requests.
139/TCP	An SMB server is listening on this port.
445/TCP	This port is used for file and resource sharing on Windows 2000, Xp and 2003.
3389/TCP	Terminal Services on the OS are turned on, which allows a Windows user to remotely obtain a graphical login (and therefore act as a local user on the remote host).
5800/TCP	A VNC server is listening on this port. VNC allows users to control the host remotely.
5900/TCP	A VNC server is listening on this port as well. The difference is that port 5800 is listening for a web server enabling the use of the HTTP protocol.
6002/TCP	A web server is running on this port.
7000/TCP	Afs fileserver is in use on this port.
7001/TCP	Callback to the cache managers is in use on this port.
7777/TCP	Cbt services are in use on this port.

Table 4.1: Open ports on OS

```
Protocol version : HTTP/1.0
SSL : no
Pipelining : yes
Keep-Alive : yes
Headers :

Date: Tue, 12 May 2009 11:57:51 GMT
Server: SentinelProtectionServer/7.1
MIME-Version: 1.1
Content-Type: text/html
Keep-Alive:1
Content-Length: 2456
```

Figure 4.5: HTTP information

# Chapter 5

## Attacks

This Chapter contains the attacks and tests carried out on the system. Each attack is structured by a description of the purpose, preliminary state of the system, the method used, the results and pending issues that present themselves after an attack. Most of the attacks tried are based on the system scan in the previous chapter and the possibilities it gives. The following Sections describe both the successful and unsuccessful attacks carried out from a Internet perspective called “Outsider” perspective and an perspective of an insider of the system called “Insider”. The finishing Section sums up the attacks.

### 5.1 Attacks from the outside

The remainder of this chapter will be structured like shown in Figure 5.1. The attacks carried out are divided into attacks that can be done by a person that has access to the inside of the system e.g. an employee and attacks that can be carried out by a person that only has external access to the system e.g. terrorists, criminal groups etc. This division is done because an insider has different resources available than an external attacker, as well as the fact that inside attacks are the most common threat source to PCS, as described in the background Section 2.2.

Attacks from the outside of the system are very common in COTS like Microsoft Windows as well as other operating systems. Since COTS and PCS are merging the possible attacks are transferred to PCS as described in the background Section 2.1. Hence the attacks possible on COTS are also possible on PCS. These attacks will be described in greater detail in the following Subsections. When looking at this PCS system from the outside perspective the only visible component that is suitable to carry out attack on is the OS. The reason for this is that it has three network cards and information cannot be passed through to the two network cards that are connected to the PS without tampering with the OS. The attacks from the

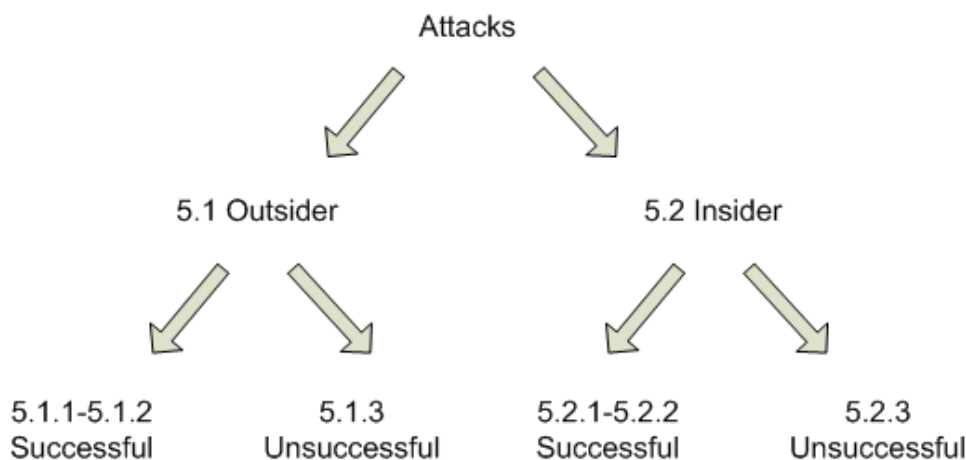


Figure 5.1: Remaining chapter division

outside perspective assume that no tampering with the OS has been done in advance. That is why the main part of the outsider attacks are focused on the OS. Although if an attacker gets control over the OS, tampering with the remainder of this PCS system can be done easily since the OS controls the other units.

### 5.1.1 Successful outsider attack: Gain control over the OS

**Purpose:** The main purpose of this attack is to gain control over the OS. If the control is gained we will explore the possibilities available for an attacker. To carry out this attack we will use some of the information previously collected by the Nessus scan.

**System state prior to attack:** OS running in idle mode.

#### Attack procedure:

- Run Metasploit Framework
- Search for Microsoft Windows Xp, Service Pack 2 vulnerabilities.
- Compare the available vulnerabilities found to the used services, open ports and vulnerabilities found in the Nessus scan.
- A match was found on SMB services and file and folder sharing.
- Configure Metasploit as follows<sup>1</sup>:
  - use windows/smb/ms08\_067\_netapi exploit.

<sup>1</sup>background on the Metasploit configuration is described in more detail in Section 3.2.



- set RHOST 10.122.10.5 (Remote host ip which is the OS ip in this case).
  - set RPORT 445 (port number on the host to connect to).
  - set PAYLOAD windows/meterpreter/bind\_tcp (Listens for a connection, injects the meterpreter server DLL for remote control of the OS).
  - set TARGET 0 (to make Metasploit detect the OS, or 3 to set it to Xp Sp2 english).
  - exploit (to start the execution of the program).
- pwd command was used to show working directory, which is C:\WINDOWS\system32 by default.
  - directory was changed to D:\ as displayed in Figure 5.2.
  - sysinfo command was used to get information about the system and the results are displayed in Figure 5.3.
  - download command was used to download a text file from the OS.
  - cat command could be used to display the text file in the command window as well.
  - terminated session.

**Result:** This attack gives the attacker the freedom to do whatever he or she wants. By configuring the Metasploit framework in a correct manner an attacker can virtually do anything he or she pleases through this attack. The exploited vulnerability enables both unlimited access to information about the system such as network adapters, operative system info, processes running, file system and registry amongst others, as well as execution and modification of these things. An attacker can e.g. remotely shut down the OS, execute a command on it, terminate processes, upload and download files etc.

Figure 5.2 displays the command window and the file system on disk D: along with information about file names, sizes and access rights. Figure 5.3 displays the information shown upon executing the sysinfo command. All the network cards on the OS are displayed with name, MAC address, IP address and netmask. Even the virtual network adapter that uses the network adapters with IP 172.21.101.1 and 172.22.101.1 is displayed here. The details of this virtual adapter are described by Szostak [34].

**Pending issues:** How do we protect the system against these types of attacks?

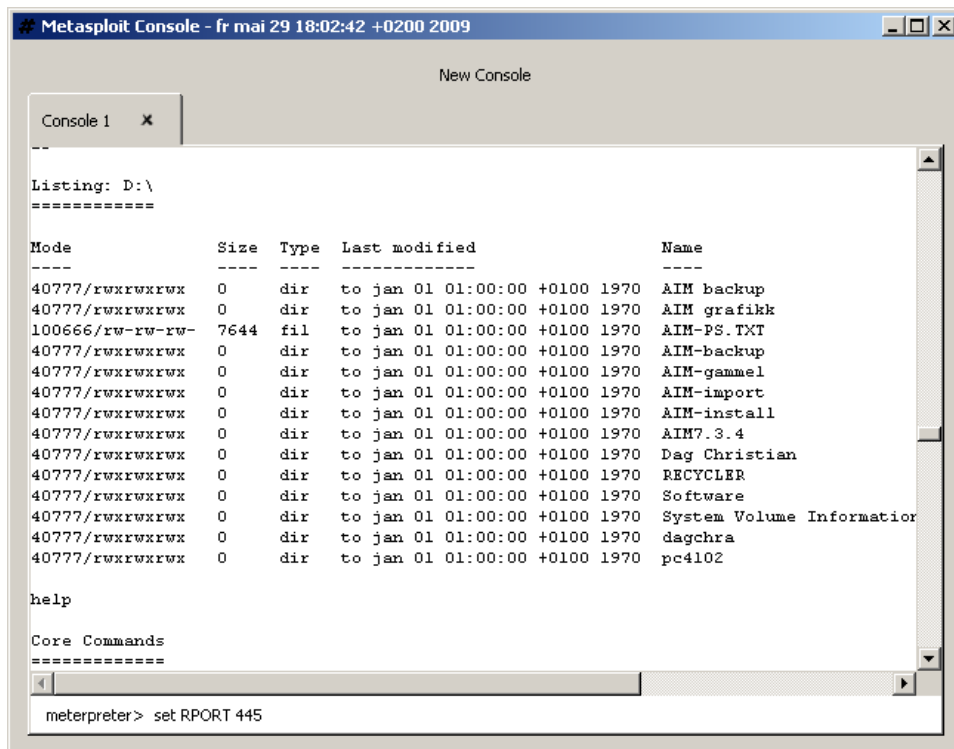


Figure 5.2: Directory displayed through Metasploit attack

### 5.1.2 Successful outsider attack: Denial of Service (DoS) on OS

**Purpose:** The purpose of this attack is to make the OS unavailable. This is done by sending more communications requests to the OS than it is able to handle. By doing this an attacker prevents legitimate users and units from using the system or makes the system so overloaded that it is practically impossible to communicate with it.

**System state prior to attack:** OS running in idle mode.

**Attack procedure:** To carry out this attack the laboratory system was changed a bit. The change was done to be able to listen on the communication link between the attacking pc and the OS. Instead of a switch (as in the original set up of the system) a hub was used. A hub broadcasts every packet to all the ports on the hub, while a switch only sends packets on the port intended for the receiver. That is why a hub is much easier to eavesdrop on than a switch. The changed setup is displayed in Figure 5.4

Once the laboratory set up was changed a bit the attack procedure was carried out as follows:

```
Console 1 x
User has been idle for: 2 secs
sysinfo
Computer: KYB-AIM02
OS      : Windows XP (Build 2600, Service Pack 2).
ipconfig

Intel(R) PRO/1000 MT Dual Port Server Adapter #2 - Deterministic Network Enhancer Miniport?
Hardware MAC: 00:04:23:d0:6a:53
IP Address  : 10.122.10.5
Netmask     : 255.255.255.128

Broadcom NetXtreme Gigabit Ethernet for hp - Deterministic Network Enhancer Miniport?
Hardware MAC: 00:11:0a:a3:da:38
IP Address  : 172.21.101.1
Netmask     : 255.255.0.0

Intel(R) PRO/1000 MT Dual Port Server Adapter - Deterministic Network Enhancer Miniport?
Hardware MAC: 00:04:23:d0:6a:52
IP Address  : 172.22.101.1
Netmask     : 255.255.0.0

Kongsberg Maritime virtual Network Adapter - Deterministic Network Enhancer Miniport?
Hardware MAC: 44:4e:45:00:00:02
IP Address  : 172.23.101.1
Netmask     : 255.255.0.0
```

Figure 5.3: System info displayed through Metasploit

- Wireshark was started up on an eavesdropper pc to listen to the traffic between the OS and the attacker.
- Netwag toolbox<sup>2</sup> was set up on the attackers pc with the following configuration:
  - Synflood tool was used.
  - Destination IP was set to 10.122.10.5.
  - Destination port was set to 139 since it uses the TCP protocol.
  - The attack was initialized.

*Background for the attack:* A normal client-server running over the TCP protocol usually exchanges a series of messages. The client sends a synchronize message to the server. The server replies by an acknowledge message. Then the client sends an acknowledgement message as well. The exchange is called a three-way handshake and is commonly used as a basis in the TCP protocol. This DoS attack exploits this. The first two packets are sent, but the acknowledge message from the client is not sent. The lack of acknowledgement message from the client creates a half open connection on the server side which consumes resources on the server. If the client continues to send synchronize messages over time all of the servers resources are consumed [18]. The half open connection is illustrated in Figure 5.5.

**Result:** In this attack a flooding tool was used that exploits the half open

---

<sup>2</sup>Netwag is described in more detail in the equipment chapter

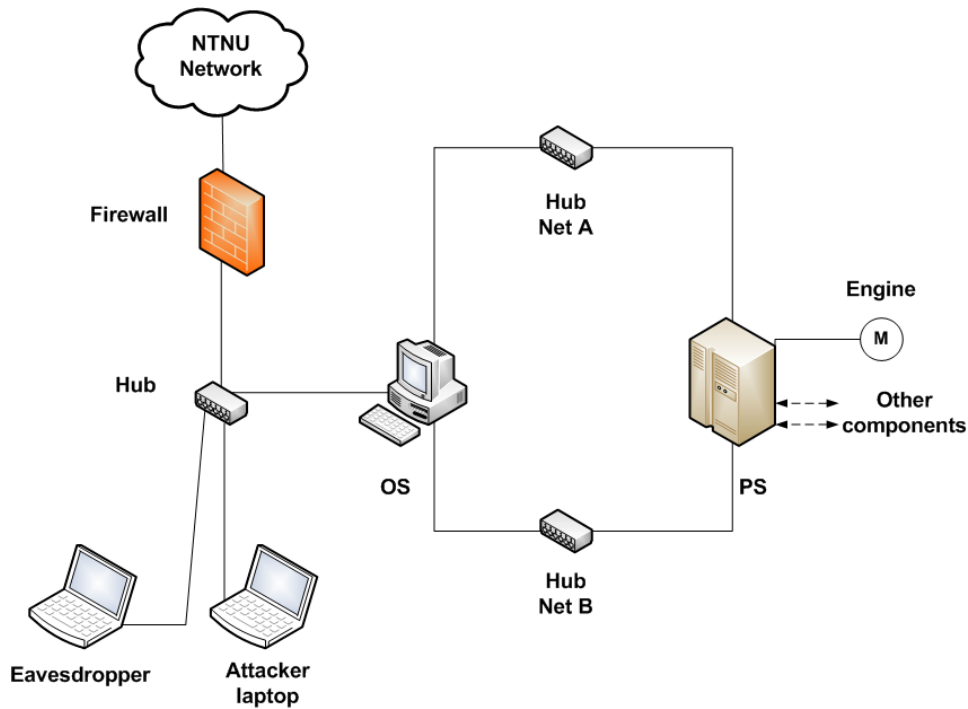


Figure 5.4: DoS attack setup

connections in the TCP protocol. The tool sets the SYN (synchronize) flag on each TCP request but in addition to not answering the SYN-ACK packets the Synflood tool makes fake IP source addresses (spoofs the addresses) for the sent packets so that the OS (which in this case is the server) sends the SYN-ACK messages to a fake address. The attacker sends packets in a very fast sequence and within 3-4 seconds the CPU usage on the OS is 100%. This CPU utilization leads to very slow responses on the OS and after some time it hangs for several seconds. In this test we were not able to force the OS to a complete halt but if this attack was carried out by several attackers at the same time (which is called a Distributed Denial of Service [DDoS] attack) it may have been possible. An overview over the traffic on the hub is shown in Figure 5.6. One thing that is worth noticing is the time and amount of packets sent. During 42.5 seconds over 100000 packets were sent to the OS. Another thing that is worth noticing is that the Synflood tool generates fake random source addresses.

Like mentioned earlier the CPU on the OS overloaded and working on 100% capacity after only 3-4 seconds due to the enormous amounts of half open connections created. The main process responsible for the overload was a System process running on the OS. The start of the attack along with the CPU usage history is shown in Figure 5.7.

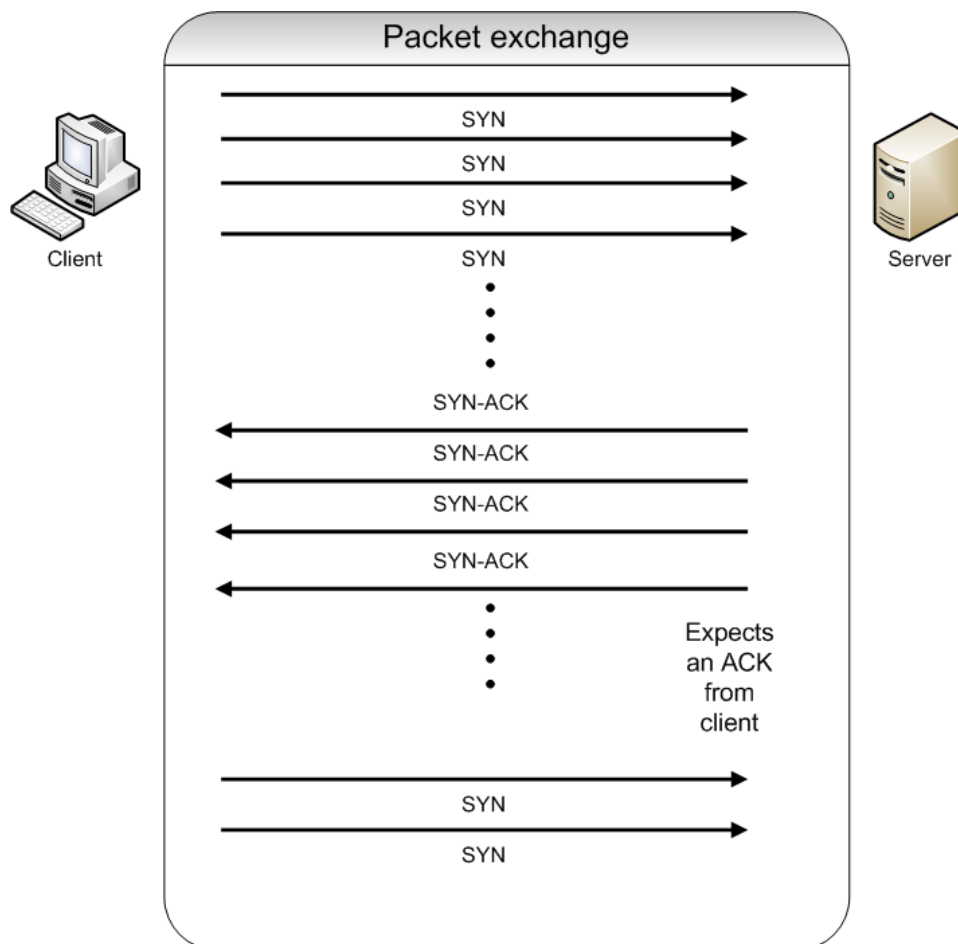


Figure 5.5: DoS attack exchange

**Pending issues:** How can we protect the system against this type of attack?

### 5.1.3 Unsuccessful outsider attack: Unsuccessful Metasploit attempts

**Purpose:** Exploit vulnerabilities using Metasploit framework. Although there was a successful attack using Metasploit from an outsiders perspective on the system, several of the other exploits that were tried did not give any results.

**System state prior to attack:** OS running in idle mode.

**Attack procedure:** Like in the successful Metasploit attack the procedure for this attack was to match possible Nessus vulnerabilities and open

No. -	Time	Source	Destination	Protocol	Info
99980	42.529442	221.129.127.198	10.122.10.5	TCP	58579 > netbios-ssn [SYN] Seq=0
99981	42.530439	135.19.47.94	10.122.10.5	TCP	39551 > netbios-ssn [SYN] Seq=0
99982	42.530440	68.57.48.113	10.122.10.5	TCP	32083 > netbios-ssn [SYN] Seq=0
99983	42.531439	155.136.159.51	10.122.10.5	TCP	42557 > netbios-ssn [SYN] Seq=0
99984	42.531440	33.78.209.161	10.122.10.5	TCP	42622 > netbios-ssn [SYN] Seq=0
99985	42.531440	218.12.219.133	10.122.10.5	TCP	42796 > netbios-ssn [SYN] Seq=0
99986	42.532440	243.211.106.198	10.122.10.5	TCP	13120 > netbios-ssn [SYN] Seq=0
99987	42.532441	8.45.19.175	10.122.10.5	TCP	14289 > netbios-ssn [SYN] Seq=0
99988	42.533440	222.124.188.42	10.122.10.5	TCP	22900 > netbios-ssn [SYN] Seq=0
99989	42.533441	185.16.187.213	10.122.10.5	TCP	12236 > netbios-ssn [SYN] Seq=0
99990	42.534453	10.159.102.238	10.122.10.5	TCP	47174 > netbios-ssn [SYN] Seq=0
99991	42.534454	115.2.7.128	10.122.10.5	TCP	56660 > netbios-ssn [SYN] Seq=0
99992	42.535440	213.184.132.19	10.122.10.5	TCP	29391 > netbios-ssn [SYN] Seq=0
99993	42.535441	107.13.105.211	10.122.10.5	TCP	6576 > netbios-ssn [SYN] Seq=0
99994	42.535442	158.70.90.245	10.122.10.5	TCP	38720 > netbios-ssn [SYN] Seq=0
99995	42.536442	206.158.165.32	10.122.10.5	TCP	39280 > netbios-ssn [SYN] Seq=0
99996	42.536443	223.128.112.54	10.122.10.5	TCP	1tctcp > netbios-ssn [SYN] Seq=0
99997	42.537445	69.217.136.75	10.122.10.5	TCP	20410 > netbios-ssn [SYN] Seq=0
99998	42.537446	233.182.134.254	10.122.10.5	TCP	18383 > netbios-ssn [SYN] Seq=0
99999	42.538439	138.104.101.42	10.122.10.5	TCP	16089 > netbios-ssn [SYN] Seq=0
10000	42.538440	172.16.102.36	10.122.10.5	TCP	32000 > netbios-ssn [SYN] Seq=0

```

...0 ... = Acknowledgment: Not set
... 0... = Push: Not set
... .0.. = Reset: Not set
... ..1. = Syn: Set
... ...0 = Fin: Not set
Window size: 1500
Checksum: 0x25c7 [correct]

```

Figure 5.6: DoS attack traffic

ports to the Metasploit exploits. The following exploits were tried:

- Since the version of the TFTP running on the OS could not be determined the three exploits regarding TFTP had to be tried, which were the following:
  - AT-TFTP v1.9 exploit.
  - TFTP32 version 2.21 exploit.
  - TFTPWIN threaded TFTP Server exploit.
- Buffer overflow in NTP.
- SMB authentication requests.
- VNC remote desktop server.
- Windows Xp with Service Pack 2.

**Result:** Regarding the TFTP running on the OS the reason for the attacks being unsuccessful may be that the version running on the OS is different than the version where the exploitation was possible.

SMB authentication requires that the user of the OS authenticates by clicking on the path (which is \\Server\Share). This can be embedded in a link on a web page or an email. This resembles a fishing attack where an attacker is forging a legitimate site to force the target to enter sensitive information that is sent to the attacker or click on a page that has malicious motives. This attack on SMB was unsuccessful because the connection through SMB could not be established.

VNC remote desktop attack was tried by trying to exploit the weakness in the RPC protocol that the blaster exploited in 2003 (as described in the background chapter 2.2)but the system is patched up enough to prevent this type of attacks.

Various attacks regarding exploitation of vulnerabilities for Windows XP with Service Pack 2 were also tried through Metasploit but there were limited results. The only vulnerability that could be successfully exploited was the one described in Subsection 5.1.1.

**Pending issues:** Are there other Metasploit attacks that could work?

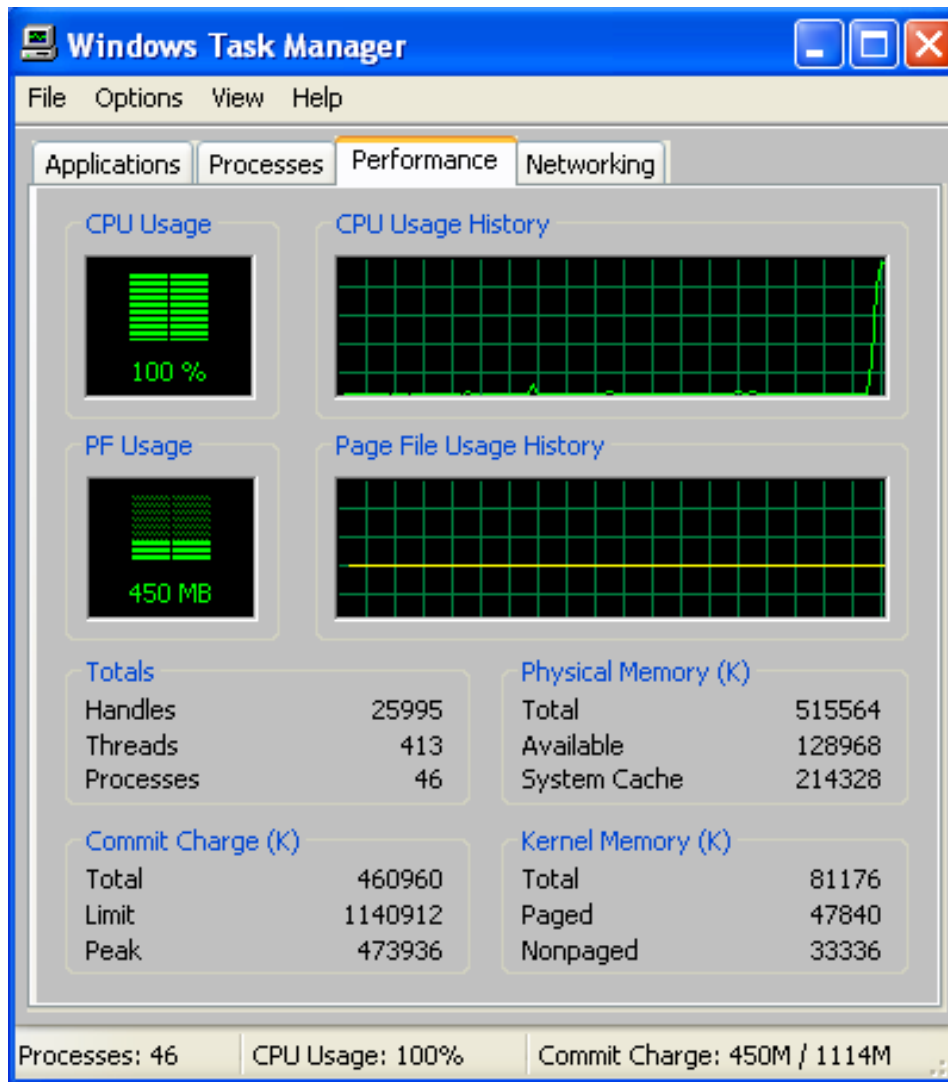


Figure 5.7: CPU usage on the OS



## 5.2 Attacks from the inside

On this PCS there are numerous vulnerabilities that an inside attacker can exploit. Some of the vulnerabilities are presented in the previous Section 4.1. An insider has different premises than an attacker from the outside. In this system an inside attacker bypasses the hardware firewall and has much easier access to system resources. The successful and unsuccessful attacks carried out from the insider perspective will be described in details in the two next following Subsections.

### 5.2.1 Successful insider attack: VNC server access for the OS

**Purpose:** The VNC enables remote control of the OS like mentioned in Section 4.1. With insider access to the OS the password enabling the access to the VNC can be broken. If the password is compromised anyone can get remote control over the OS and then also the whole PCS system.

**System state prior to attack:** OS running in normal operational mode.

**Attack procedure:** Based on the Nessus scan we know that a VNC server port is listening on port 5900 over the TCP protocol.

- An insider uses gets access to the registry of the OS.
- The insider fetches the hashed password for VNC which is stored in the registry in: HKEY\_CURRENT\_USER\Software\ORL\WinVNC3\Password (the hashed password is stored in hexadecimals with the total of 8 bytes, but due to privacy reasons could not be given in this thesis).
- Cain & Abel program is used, password cracking is enabled and the hashed password found in the registry is used as input in the VNC-3DES password cracker, which creates an output password: \*\*\*\*\*.
- The output passwords authenticity has to be verified. TightVNC viewer client is used to connect to the OS remotely with the host IP address: 10.122.10.5, port number: 5900 and password: \*\*\*\*\*.

**Result:** The attack is successful and a new window comes up enabling remote control over the OS with a real-time view and control over the mouse pointer, keyboard and a view of what is seen on the screen. This enables control of the OS and hence the whole PCS system in this case. Turning the engine unit on remotely using this type of attack was not a problem.

The main vulnerability that is exploited in this attack is that the stored VNC password is encrypted using triple Data Encryption Standard (3DES)

and stored in the registry. DES and 3DES has known weaknesses that the program Cain & Abel utilizes to decrypt the password. One of the main purposes of a hash is that it has to be irreversible and hence a password should not be found if only the hash value of the password is available. Due to the weaknesses of 3DES the plaintext password corresponding to the hash value can be found using Cain & Abel. More details about the use of Cain & Abel is found in Appendix B.2

In practice remote control can also be easily discovered because control of the mouse and keyboard is lost but if the attack is done when the OS operator is away the attack may not be so easily discovered. Access to the registry of the OS is also usually not something an employee has, but if the administrator makes a simple mistake by not turning the OS of or restricting the access to it in a physical way an employee might just get hold of the hashed VNC password. Then by using the method described above the employee might get total control over the system. Figure 5.8 shows remote control over the OS using VNC and if the I/O value of the motor is changed it turns the engine on and off remotely.

**Pending issues:** How do we protect the OS against these types of attacks?

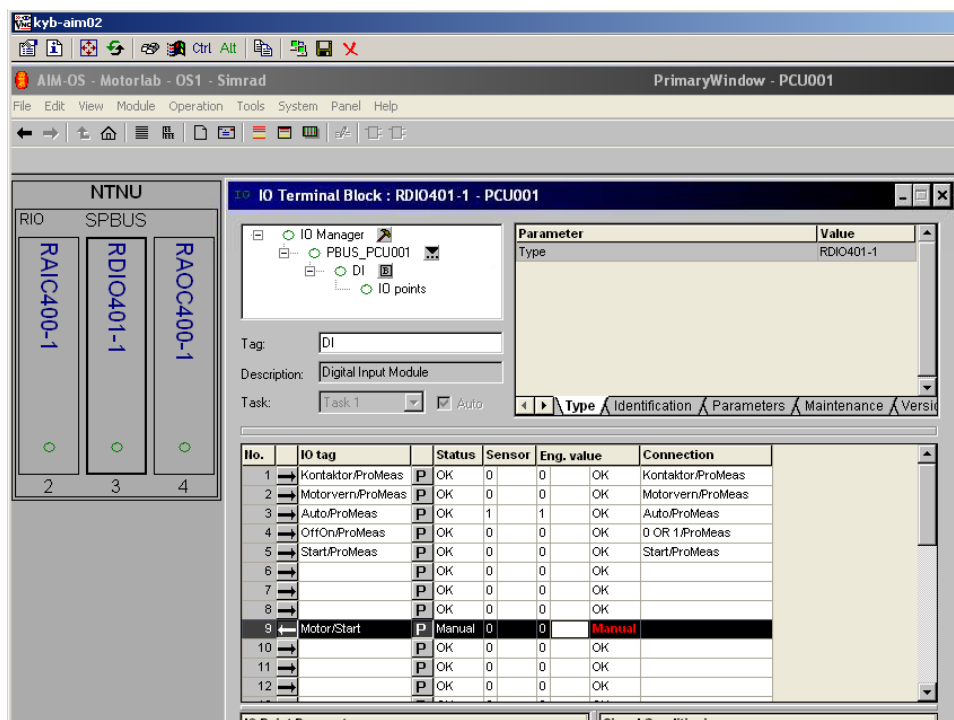


Figure 5.8: Remote control of OS using VNC

## 5.2.2 Successful insider attack: Packet replay through a switch

**Purpose:** This attack is based on packet replay. Packets sent between the OS and PS are captured by an attacker. Then the attacker sends the same packets to see if it is possible to recreate the same behavior of the system. For this purpose the engine unit is used and the packets sent from the OS to PS are tried captured, then replayed. The engine is used due to the fact that it is easy to see if the replay attack is successful or not because it is turned on or off.

**Basis:** The basis for this attack is that it was tried by Szostak [34] on this system in the previous semester. One of the topics that are relevant is if this type of replay attack will work on a switch instead of hubs used in this setup. A switch would be more realistic in a PCS used in the industry, hence this type of attack would be more probable in a real life setup. The setup of the system in this type of attack is shown in Figure 5.9.

**System state prior to attack:** The OS and PS are both turned on

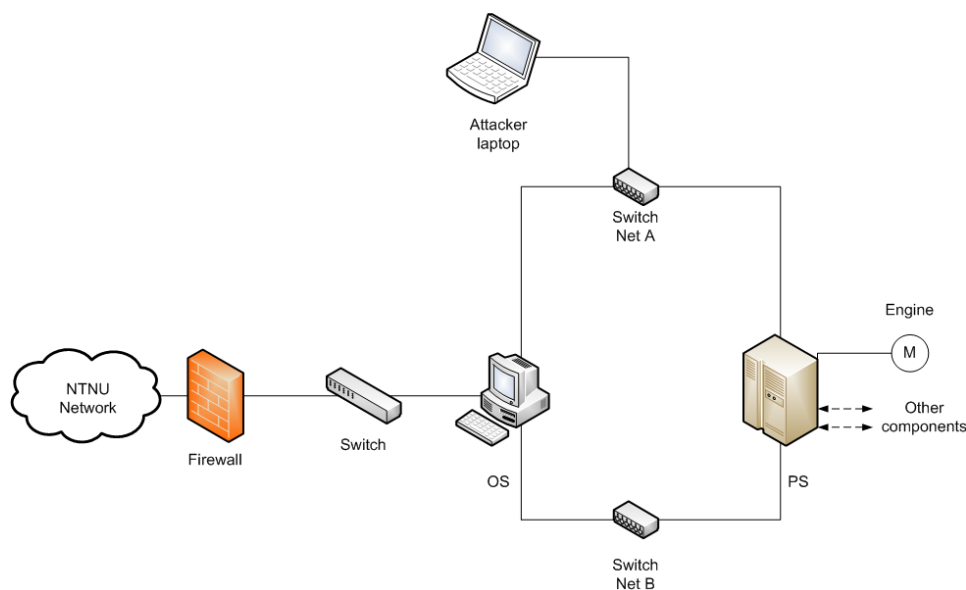


Figure 5.9: Insider attack

and functional but without any processes being run. The Engine is not running either.

### Attack procedure:

1. Substitute the switch on net A with a hub and listen (sniff) to the traffic with Wireshark.

2. Turn the engine on and off using the OS.
3. Change the substituted hub back to a switch.
4. Filter the packets and sort out the packets turning the engine on and off and save them.
5. Disconnect the OS from the internal network
6. Use Colasoft packet player to replay the captured packets over the switch on net A.

**Result:** The attack is successful. An attacker is able to turn the engine on and off but there are a few assumptions that has to be made here. As described by Szostak [34] there are security mechanisms in the OS preventing replay attacks to some extent. The mechanisms does not always seem to work as intended and if many attacks are carried out in series during a short period of time the OS goes into a error state, which it recovers from after 10-15 seconds. In that case the attacks are successful, but if the OS does not go into a error state then the replay attacks are prevented by the OS. In that case the only solution is to disconnect the OS or make it unable to respond.

The other assumption that has to be made in order for this attack to be carried out is that the packets between the OS and PS has to be sniffed in some way. The first point in the attack procedure is to substitute the switch with a hub. Since a hub forwards each packet received on all the ports it makes it a lot easier to listen on than a switch, which only forwards packets to the intended recipient port. If an attacker has access to the inside of the system an insertion of a hub between the switch and the OS can be made, so that the packets can be sniffed. An alternative is to replace the switch with the hub but that may be more difficult to do in practice.

The success of this attack, although with some assumptions, proves that these types of systems are vulnerable to replay attacks even if hardware that can make it hard to accomplish this types of attacks is used.

**Pending issues:** How can we protect the system against this type of attacks?

### 5.2.3 Unsuccessful insider attacks: Force OS forwarding

**Purpose:** In an attempt to be able to communicate with the inside network from the outside (which is the Internet perspective in our case) there were some modifications tried on the OS in order to be able to forward packets from the outside interface to the inside. Since Windows has the packet forwarding turned on by default some modifications had to be done.

**System state prior to attack:** System running in normal operational

mode.

**Attack procedure:**

- The attacker connects to the outside interface.
- OS modifications by an insider:
  - Registry value found in: HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters
  - The IPEnableRouter value has to be changed from the default 0 to 1.
  - Windows Xp has to be rebooted in order for this change to take effect.
- Some modifications of the captured packet turning the engine on were required in order to send the out to the OS with the PS as destination.
- Various modification attempts were made in the Packet Player Builder tool. The tampered parameters where the following:
  - Source Ethernet MAC address of the packets
  - Destination Ethernet MAC address of the packets
  - Time To Live (TTL).
  - Source IP address of the packet
  - Destination IP address of the packet
  - Checksum was recalculated automatically by the program
  - Saved the packet in a file.
- The saved file was used in Colasoft Packet Player in an replay attack attempt.

*Background:* This method of enabling IP forwarding is based on K. Salah and M. Hamawi article [32]. The IP forwarding is done by enabling it in the registry by changing the IPEnableRouter value to 1. Then packets can be sent through the computer and it works as a ordinary router. That is exactly what we need in order to send packets from the outside network to the inside network.

The TTL field determines how many “hops” are allowed. For every host the package passes through the TTL field has to be decreased by 1. This field is made to prevent packets from circulating in the system with an end-less loop, which could flood the system. The TTL field was originally set to 1 in the packet because the packet were only intended to be sent between the OS and PS.

No.	Delta Time	Source	Destination	Protocol	Size
1	0.000000	10.122.10.3:10000	172.23.100.1:11000	UDP	163

Ethernet II Header		[0/14]
Destination Address:	02:41:4C:42:11:01	[0/6]
Source Address:	00:1D:72:C6:C5:FC	[6/6]
Protocol:	0x0800	[12/2]
IP - Internet Protocol		[14/20]
Version:	4	[14/1] 0xF0
Header Length:	5	(20 Bytes) [14/1] 0x0F
Time To Live:	4	[22/1]
Protocol:	17	(UDP) [23/1]
Checksum:	0x3AEC	(Correct) [24/2]
Source IP:	10.122.10.3	[26/4]
Destination IP:	172.23.100.1	[30/4]

Figure 5.10: Replay packet fields edited

**Result:** Despite various modifications tried such as changing the source IP and MAC address of the packet to the IP and MAC address of the attackers computer, changing the TTL from 1 to 4 and recalculating the checksum this attack failed. This modification of the packet is shown in Figure 5.10. The packet was sent out from the attackers computer, but the OS did not forward it, despite the enabling of packet forwarding through the registry change.

The reason for this may be the same seen in attempts to try to replay the packet from the internal network, like shown in Subsection 5.2.2. In some cases the OS prevents replay of packet by a security mechanism. Since no documentation on the system could be received from Kongsberg Maritime it is very hard to clearly state the security mechanism in use. In this case there may be a lower layer protocol (under UDP/IP) that prevents packets from the external network to go into the internal network. Like seen in Subsection 5.2.2 when the OS was not operational the replay attacks were successful, but in this case the option of disconnecting the OS is not possible because the OS is used as a packet forwarding unit.

**Pending issues:** Could this attack have been done in another way to make it successful and bypass the security mechanism in the OS?

### 5.3 Summary

To sum up this Chapter an overview of the attacks is presented here. The attacks are divided into two tables, containing outsider and insider attacks. Outsider attacks are shown in Table 5.1, while insider are shown in Table 5.2. A name of each attack, the outcome, and a short description are shown.

Attack name:	Outcome:	Description:
Gain control over the OS	Success	A weakness in the SMB protocol was exploited using Metasploit. A shell was planted in the OS so that an attacker could control the OS through that shell and enabled endless possibilities for remote abuse.
Denial of Service	Success	Half open connections in a three-way handshake in TCP consume resources. This is exploited by flooding the OS with half open connections so that resources are consumed and the OS becomes unable to respond to legitimate requests.
Other Metasploit attempts	Unsuccessful	Gain control over the OS was a successful attack using the Metasploit framework. Since the framework contains 320 modules the open ports and vulnerabilities found in the Nessus scan were compared and tried out.

Table 5.1: A summary of outsider attacks

Attack name:	Outcome:	Description:
VNC server access for the OS	Success	The password for the VNC server on the OS is stored in the registry, in a hash encrypted 3DES. This is a weak encryption enabling the password to be found and an attacker can take remote control over the OS along with mouse, keyboard and real time screen view.
Packet replay through a switch	Success	This attack assumes previously captured packets between the OS and PS. A replay of those packets can result in an insider forcing system behavior that is not initialized by the OS.
Force OS forwarding	Unsuccessful	Since the OS has 3 network cards communication between the outside and the inside network is hard. A registry value on the OS was attempted changed to be able to communicate between the networks.

Table 5.2: A summary of insider attacks



# Chapter 6

## Discussion

### 6.1 Answers to research questions

In this section we will try to answer the research questions presented in Section 1.1. Although the answers to these questions are spread throughout the thesis, we will discuss them and try to provide appropriate answers to them jointly in the following Subsections.

#### 6.1.1 Is it possible to get useful information about the system that can be used in attacks later on?

In order to be able to attack any ICS or ICT system, information about the system must be collected in advance. If the answer to this question would have been no, then attacks on the system would be very difficult to carry out, since an attack in such a case would have to have been based on general knowledge about typical systems and system implementations only, rather than specific system knowledge and details about the specific set up and parameters used in the case we are investigating. However, in our case the answer to this question is yes and the thorough answer is mainly found in Chapter 4. Here is a summary of some of the main points from that chapter

For the purpose of finding information about the system a scanning tool was used. The tool used was Nessus and it was configured to look for known vulnerabilities such as open ports, services running, protocols available etc. Known vulnerabilities in Windows along with all of the components that belongs to it were also scanned for. All the available plugins in the HomeFeed version of Nessus were tried, which gave the results shown in Chapter 4.

This thesis focuses on attacks from the Internet that can be made on a PCS. For this system, the main component that can be attacked is the OS. The reason for this is that it is strategically placed as the first and only component seen from an Internet perspective. Therefore the OS was the main and only target of the Nessus scan. A direct connection to the other components of the system from an Internet perspective could not be made

for reasons we will describe in further detail in the following Subsection 6.1.3.

The results of the Nessus scan showed 13 open ports, 2 medium level vulnerabilities and 26 low level vulnerabilities. In addition the scan collected information about the OS, such as the operating system, computer name, workgroup and even the physical MAC address. All this information was used in Chapter 5 for the attacks.

As often is the case, the technical system information we obtained from the relevant industry regarding the laboratory system were sparse. Hence we had to settle for the system information obtained by Nessus. This was very unfortunate because if documentation on the components could be obtained, it could have given ideas on ways to attack the system. Especially information about the interactions of the system and protocol information.

Despite the fact that documentation could not be obtained, the information collected in the Nessus scan was enough to carry out several successful attacks on the PCS.

### **6.1.2 Can the system be penetrated from the Internet and in that case what damage can an attacker do?**

This question was the main motivation for writing this thesis. To be able to answer it, the scan of the system had to be used as a background for possible vulnerabilities in the system to exploit and it gave ideas about how to penetrate the system. System information from the scans such as open ports, known vulnerabilities and protocols used were matched up against tools like Metasploit framework, DoS attacks, replay attacks, vulnerability databases and other sources of information. After countless attempts some of the attacks were successful. The attacks along with the configuration of various attack tools used, procedures and results are described throughout Chapter 5. The main points along with a discussion are presented here.

As mentioned in the previous subsection, the main point of the attacks was the OS. The OS could in fact be penetrated from the Internet. For this the Metasploit framework was used and a vulnerability in the SMB protocol was exploited to plant a shell that was controlled by the attacker. This was a very serious security breach and enabled the attacker access to information about the system, processes running on it, registry and much more. The attacker could even execute commands on the OS, terminate processes, shut down the OS, restart the OS and even upload and download files. This gave many possibilities for exploitation.

The successful Metasploit attack shows how important patching of the operating system is and the possibilities a successful attack of this kind opens up, such as remote control of the attacked OS through a shell. Another attack that cannot be fixed by a simple patching of the system and is much more difficult to protect against is the DoS attack. The Netwag toolbox

faked the source IP address and made it random for each packet, in addition to the fact that it made a half open connection by sending a SYN and not replying to it by an ACK, which consumed resources on the OS. DoS attacks are in general very hard to protect against. One way is to filter and shut out the flooding source addresses, but in this case this is not possible. A DDoS attack which is done by several attackers at the same time is even more dangerous and can make the target totally flooded so it cannot respond to legitimate requests.

In addition to looking at penetration from the Internet perspective some attacks were described that require an inside attacker as well. We have learned from our background studies presented in Section 2.2 that insiders are accountable for over 50% of the security breaches in an ICS. Therefore the knowledge of an insider (like an employee) can be priceless in attacking our PCS.

Since VNC enables remote control of the OS any user with the password of the VNC server on the OS can control it and all of the units controlled by the OS. In our case this means total control of the laboratory system. By exploiting that the password is stored as a hash in the registry and that the hash is compressed by a 3DES encryption an insider can decrypt it. Cain & Abel program enables reversing of a hash encrypted with 3DES. In our case this was successful and we could get remote access and control over the OS. If an insider could obtain this password he or she could use it for private purposes or even sell it. The value of such information could be very high, depending on the system that was remotely controlled by the VNC. Terrorist groups, foreign spies or attackers with malicious intents would probably be very interested in gaining control over an OS in e.g. the power grid. An example of this type of industrial espionage was described in Section 2.2.

Another successful insider attack was done by demonstrating that the replay attack is also possible through a switch, not only a hub. There are two assumptions in this case, which are that the packet intended for replay has to be captured in advance and the second one is that the OS has to be non-responding. The second assumption can be achieved by continuously repeating the replay attack (although there is a certain point of uncertainty in doing that as described by Szostak [34]). The first assumption can be achieved by adding a hub e.g. between the OS and the other components so that the traffic can be listened to and captured to be replayed on a later point in time. An insider has access to the system and can fulfill these assumptions. If this replay attack is successful an insider can control vital parts of a PCS system like an engine unit in our case or in an industrial case a valve to a nuclear power plant for instance. Unsupervised control of an insider in this case can have disastrous consequences.

Now we will try to sum up the various types of attacks from both an Internet perspective and an insider perspective. There are many different possibilities of cyber attacks. By merging COTS with ICS there has been

opened up opportunities for exchanging of information between ICS that are separated by huge distances and remote control of industrial plants that was never possible before.

The merging of COTS and ICS also results in introduction of several of the weaknesses of COTS into previously isolated ICS. As shown in this thesis, a PCS which is a part of an ICS can pose a threat to the security. Although the security mechanisms in this laboratory PCS may not resemble the ones that are used in the industry today, the ICS control parts of the critical infrastructure, and the security requirements for such systems should be a priority.

Not only the use of COTS is responsible for the change of the threat scenario and the consequential to the change of the security in an ICS. Also the introduction of new information transfer technologies such as wireless devices, usb pins and other units can produce backdoors into systems that are normally considered trustworthy. Therefore companies have to have clear guidelines and rules on how to use and secure the new technologies.

### **6.1.3 What kind of security mechanisms can be found looking at the system from an outside attacker using the Internet?**

To answer this question we have to look at the whole system. The PCS described here has many components that prevent intrusion. The main component that provides security is the hardware firewall. It is configured by NTNU, so it is not part of the system provided by Kongsberg Maritime but still it protects the system against intrusions from the outside network. The firewall is configured so that it limits the communication between the OS and the Internet to addresses that are within the NTNU domain (129.241.—.—). In that way students using the laboratory have enough information to reach internal sites and e-mail, but cannot enter potentially harmful sites. On the other hand communication from the Internet (even with a NTNU domain address) to the system is not possible. The hardware firewall prevents all communication from the outside to the PCS. Even simple ping requests are not possible from the outside as described in Appendix A. This is done to separate the system from the outside world. The hardware firewall can be considered a security mechanism although it did not come with the system originally, but similar systems in the industry are usually operating behind firewalls.

Another security mechanism in the system are the three different network cards on the OS. There is no way to communicate with the PS from the Internet without reconfiguring some of the settings on the OS. The reason for this is that Windows does not allow communication transfer between network cards on a machine by default. It is possible to turn it on but as described in Subsection 5.2.3 this forwarding of information between network

cards did not lead to a successful attack. So even if an attacker could somehow bypass the hardware firewall it would be very difficult to enter or even communicate with the inside network (between the OS and PS).

There are also security mechanisms on the inside network that prevent attacks. As described in Subsection 5.2.2 there is a mechanism in the OS that prevents replay attacks. Due to the fact that sometimes the replay attack works and sometimes it does not, it is very hard to determine what security mechanisms are embedded in the OS without having any documentation available about it. The security mechanisms may be on a lower layer than UDP/IP because the forwarding of replay packages from the outside interface to the inside was unsuccessful.

To summarize, the security mechanisms in this system are present but there are ways to bypass them, for example by exploiting a weak password hash in the VNC.

## **6.2 Reflections after the attacks**

This section contains some reflections after the attacks. The unanswered questions throughout the thesis are discussed here, things that could have been done to find out more and things that could have been done different by the industry. A suggestion on further work wraps up this Section.

### **6.2.1 Unanswered questions**

Through the thesis there were some unanswered questions. From Section 4.1 the questions were “Can we exploit this? If we can, then how?” regarding the system scans of the OS made through Nessus. In Chapter 5 some of the vulnerabilities found in the scan were successfully exploited but still the majority of the vulnerabilities were not exploited. One of the reasons may be that most of the vulnerabilities found were low level vulnerabilities, which means that they are not considered as critical for the security of the system. They may pose a threat to the security of the system but it is considered very low.

In Chapter 5 there were several unanswered questions. In Subsection 5.1.1 remote control of the OS is gained by exploiting a vulnerability in the SMB protocol and the question asked is “How do we protect the system against these types of attacks?”. In this case the answer is simple. A patch of the operating system has to be applied in order to prevent that type of attacks. Operative system vulnerabilities constantly are found and spoken of in various forums on the Internet. Software tools are made to exploit those vulnerabilities. Therefore the only way to protect against this type of exploitation is to keep track of the new patches available and update the system on a regular basis.

In Subsection 5.1.2 a DoS attack was described. Afterwards the same question of “How can we protect the system against this type of attack?” was raised. This time the answer to the question is not that simple. Filtering and exclusion of IP addresses that are flooding the network is one way the server side can protect itself against it, but there are two problems with that type of protection. The network is flooded with packets even though the server does not reply to the packets. The second problem is that an attacker can forge the source address and hence the blocking becomes less effective. According to RFC 2267 [20] one way to protect against these types of attacks is to make the Internet Service Provider (ISP) restrict the source addresses of the packets sent by the user to only the address space allowed by the user. This would effectively eliminate DoS attacks but requires filtering by the ISPs. An active detection mechanism implemented in the network by the ISPs would also make it easy to localize the attackers that flooded the network or attempted to flood it with DoS attacks.

Subsection 5.1.3 raised the question “Are there other Metasploit attacks that could work?”. Since there are 320 modules available in Metasploit that can be used for exploitation, the answer to the question can be said to be yes, with high probability. Due to the limited time available, only some of the modules of Metasploit were tested. The ones selected were based on the Nessus scan results.

Now the focus is changed to attacks from the inside. In Subsection 5.2.1 the question “How do we protect the OS against these types of attacks?” was raised again. The answer to it would be to use a different type of VNC server or just disable the VNC server. Since the VNC server used, stores the password in the registry and uses a weak encryption, it poses a serious threat to the security of the OS and also the whole system in this case since the system is indeed totally controlled by the OS. If a VNC server is required, then another type should be used that preferably does not use 3DES encryption but more secure encryption standards like Advanced Encryption Standard (AES) or others. Preferably, the VNC server should be disabled because it enables remote control over a vital element in a PCS and even the best security cannot prevent human errors which can have very serious consequences. One obvious hazard is that the password may get into the hands of an attacker with a malicious intent.

In the last successful attack in Subsection 5.2.2 the question “How can we protect the system against this type of attacks?” was asked for the last time. One of the possibilities is to eliminate the necessary preconditions that were assumed present for the attack. Physical protection of the connection cables between the OS and PS for the purpose of preventing sniffing on them is one possibility. Another is to improve the replay protection in the OS so that it is impossible for an attacker to carry out this attack. The replay protection in the OS worked sometimes, but other times the OS went to a state where it recovered from an error. Therefore the results of these attacks

could not be determined with absolute certainty. If the OS prevented the replay attacks every time it would prevent this types of attacks.

The question asked in Subsection 5.2.3 was “Could this attack have been done in another way to make it successful and bypass the security mechanism in the OS?”. Due to limited time there was not much room to experiment any further with this type of forwarding attack interconnecting the outside and inside networks of the PCS. An alternative to making this attack successful and bypassing the security mechanism in the OS would be to physically connect an additional network node element (like a hub or a switch) instead of the OS. This is shown in Figure 6.1. It would require physical access to the system and would eliminate the OS as the only component that is connected to the outside network. In practice this remake of the network setup would should be discovered pretty quickly, at least in an organization with a functioning quality control management.

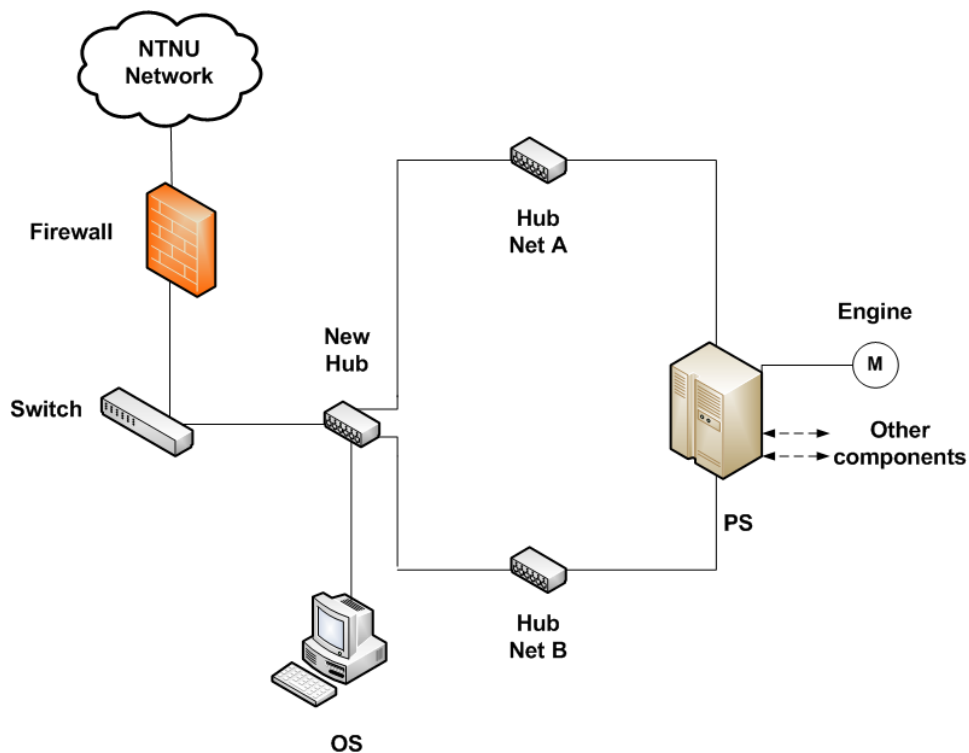


Figure 6.1: System remake to make forwarding between networks possible

### 6.2.2 What could have been done to find out more?

Since this was a laboratory setting, the system had very strict firewall settings as described in the former Subsection 6.1.3. The network administrators along with the staff responsible for the laboratory has set very strict

settings on the hardware firewall to protect it from attacks from the Internet. This prevented testing from the outside of the hardware firewall, but in a real life setting in the industry the hardware firewall could be less strict as in this laboratory setting. Communication with the outside world would be possible to a greater extent. Then it would have been very interesting to do penetration testing on the firewall to see what kind of restrictions are used in the configuration and this information could be used as a basis for further vulnerability testing.

In this thesis the Nessus scan provided more than enough information for testing of various vulnerabilities. For more extended testing of the system several additional tools could have been used. The tools from the background Section 2.3 could be used for more thorough scanning and system survey, such as Nmap, IP stack integrity checker and Ethereal.

The attack possibilities on ICS and especially COTS products such as Microsoft Windows are known to the public and exposed by the media on a large scale. Albeit the attack possibilities are numerous, for this thesis they were restricted to the possibilities shown in the nessus scan. There are certainly possibilities for other attacks. Some of the other attack tools that could be used are the once described in the background Section 2.3. Tools such as Hydra, Fuzzers and Wikto would be very interesting to test in greater depth on this laboratory. Some of the tools already used such as Metasploit and Netwox would also be interesting to do further testing with, but due to time constrains only some of the possibilities the tools allow were tried. The iOpener tool would be very interesting to test as well, but unfortunately we were unable to obtain it.

### **6.2.3 What could have been done different by the industry?**

In this subsection we present some improvements the industry can do based on the results from this thesis. Some of the improvements are so general that they can be applied to most ICT systems, while some are more specific for ICS and PCS.

Unpatched systems can be abused by attackers. This is a real threat to many ICT systems today because there is usually a time difference between a vulnerability is found in a system, e.g. an operative system, and the time until a patch for it is developed and installed by the system administrator. In some cases the administrator is not even aware that there is a new patch to apply, like in the example from background Section 2.2 with the nuclear plant attacked by a worm, due to a unpatched system. Hence administrators have to constantly keep track of the latest patches and be aware of the vulnerabilities their systems may have.

Firewall filtering rules should be set. This should be done to minimize the possible ways to exploit the system. The open ports of the firewall should be monitored and given special attention. In our laboratory system



the VNC server posed a vulnerability threat. Close monitoring of the VNC, or totally disabling it, would be advisable. Since the system described in the thesis is only a laboratory system used mainly by students, the consequences of a security breach in the system may not be so great. On the other hand, if a system used in the industry was running a similar VNC server and was victim of an attack, it could have other results. An ICS may have control over parts of the critical infrastructure and if used for malicious reasons, there can be very serious consequences.

DoS attacks are a threat for many ICS. A cooperation with the ISP could be made to monitor the nodes on the network and if possible shut out or prevent an attackers trying to carry out DoS attacks. This can be done by the ISP by reject packets with a source address other than the client's. If large amounts of traffic is generated, monitoring systems can be used to shut out the possible attacker.

In our system a replay protection mechanism was implemented and used in the OS. The results of the attack in Subsection 5.2.2 showed that the security mechanism in the OS did not work every time. This may be an issue because it allows an attacker to replay packets and potentially control units without the authorization of the OS. However the laboratory system does not fully resemble the system used by Kongsberg Maritime, and may be lacking some of the security mechanisms implemented in the live systems.

IDS systems can be used by the ICS to monitor the networks and discover possible attacks. This requires additional equipment and training of personnel, but the benefits of having an IDS can be very great. In addition log records should be made continuously to ensure that if intrusions are detected, they are also documented. The log records could be used later on to reset the system to a previous state if changes are made. Log records themselves are obvious targets of attacks as well, so they too should be protected by the IDS systems.

Tight security on the computer systems is not enough to prevent unwanted incidents from happening. Guidelines have to be defined to tell the users of the system how to act and prevent their behavior from compromising the security of the whole system. An open door to the server room or a weak password are examples of things that can be used by an intruder as a basis for an attack.

#### **6.2.4 Further work**

This topic has partly been covered in the previous Subsections but can be summarized here. The following topics could have been gone thoroughly into:

- System scans (Nmap, IP stack integrity checker, Ethereal).
- New attack tools tried (Hydra, Fuzzers, Wikto and iOpener).
- Used toolboxes (such as Metasploit and Netwox) tried in other type of attacks.
- Other attacks tried such as DDos attack.
- Fuzzing framework used to force abnormal behavior from the system.
- IDS set up and tested.

## Chapter 7

# Conclusion

PCS are used to control parts of the critical infrastructure of society, such as electric utilities, petroleum, water, waste, chemicals and pharmaceuticals amongst others. If the PCS become victims of cyber attacks, this can have severe consequences. The consequences may involve health and safety of human lives as well as having a huge impact on national and global economy. Since the merging of COTS and PCS, the previously isolated PCS now face new types of threats due to well-known flaws in COTS, as well as being connected to the Internet. Therefore the focus on securing PCS and ICS in general should get increased attention.

In this thesis the laboratory system used was a scaled down PCS that could be tested on without any serious consequences. The laboratory system was delivered by Kongsberg Maritime. The OS is the first unit an attacker from the outside has contact with and it is used for controlling the other components of the system, therefore the OS is the main source of attention in this thesis. A scan was made on the OS to map the vulnerabilities of the OS. The scan was used as a basis for the attacks. Attacks were divided into attacks from the outside (Internet) and attacks from the inside.

Under the circumstances of the testing on the laboratory PCS, many of the attacks tried were successful. A shell was planted in the OS, so an attacker could control it remotely, DoS attack flooded the OS and forced it to halt for a few seconds, VNC password was found enabling remote view and control of the OS, replay of packets was successful on the inside of the system making a man in the middle scenario possible. Despite the fact that the laboratory system may not have all the security mechanisms implemented, as the PCS systems in the industry does, the fact that the attacks on the laboratory system are possible may seem a bit disturbing.

To prevent from the types of attacks described in this thesis steps has to be taken. Some of the prevention steps can be to regularly patch the system, use firewall filtering, monitor nodes in case of DoS, IDS monitoring and guidelines on system use.

# Appendix A

## Lab problems

### Lab work 14.3:

There were problems with the internet connection. When I tried using the OS for connecting to the internet it was not possible. To solve this problem, the internet cable from the OS had to use another socket than the one used originally. I discovered that certain sockets in the wall enabled internet connection while some were only meant for local use. The socket in the wall was changed from 754 to 753. Then connection to the web pages of NTNU was possible but other pages on the internet was still not possible. The reason for this may be restrictions of connections set by the administrators of the system.

These restrictions may be a problem in further work and has to be discussed with the administrators of the system to see if it may cause problems for the tests I am going to run on the system.

The IP on the outgoing connection in the network settings of the OS is set to 10.122.10.5 which is a local IP address. When I tried to ping it from my computer the ip was translated to 129.241.187.1 which seems more correct. This may be a problem as well because I am not sure how to determine if that is the correct IP address and even if it is there seems to be a firewall turned on that may prohibit some of my tests as well.

Another thing discovered was that if OS and my computer are connected to sockets next to each other the connection fails on both of them. To solve this OS is connected to socket 753 and my laptop is on 755.

### Labwork 1.4:

Breakthrough on the possible IP address. Using the nslookup command

in the command prompt window on the OS I got the following information:

Default Server: kyb-ad01.ad.itk.ntnu.no  
Address: 129.241.10.17

After further testing it turns out that this is the IP address of the dns server, not the OS. This test gave birth to a new idea on how to determine the external IP address. A ping was tried on the OS name kyb-aim01 with the address: kyb-aim01.ad.itk.ntnu.no. The request timed out (hence no reply from the OS) but I did not get any error messages about the address, which was shown as

129.241.187.83

### **Labwork 12.5:**

This IP address also turned out to be the wrong one since the OS name is kyb-aim02 not kyb-aim01 according to the Nessus scan. When I tried to ping kyb-aim02.ad.itk.ntnu.no only the local IP address of the OS (10.122.10.5) came up. Therefore the global IP address could not be determined due to strict security configurations in the hardware firewall. When a ping request is attempted on the OS from the outside of the firewall it is unreachable, but if the ping is sent from the same subnetwork (hence behind the firewall) the ping returns the local IP address.

## Appendix B

# Details about tools used

### B.1 Nessus scan details

Here are the details of the configuration settings applied to Nessus, a long with the full results and comments:

The options for the scan were set to make a safe scan of the system, without the risk of shutting it down or creating any sort of complications for it. The LaBrea scan was not used due to the fact that a PCS system should not have honey pots that attracts attacks on it. The options set for the scan are shown in Figure B.1.

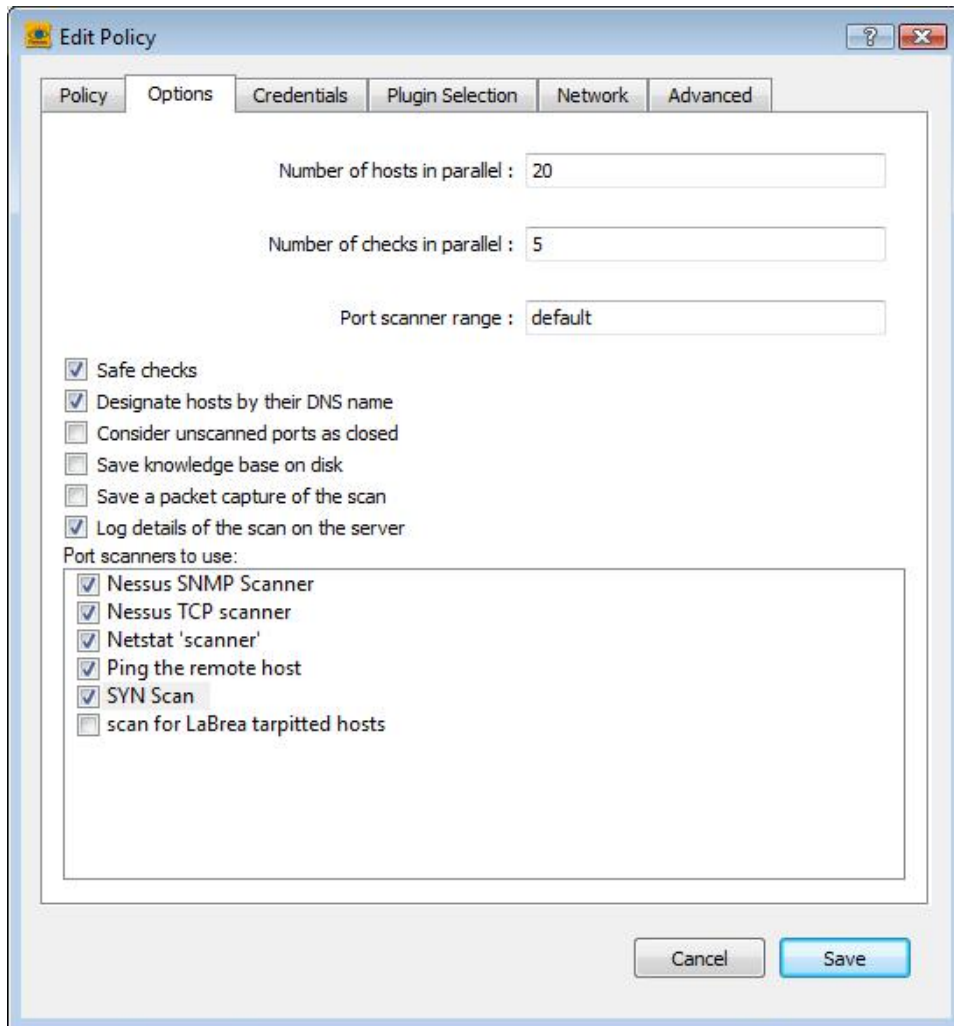


Figure B.1: Nessus scan options

Figure B.2 shows that all the plugins available in the “HomeFeed” version of Nessus are used. Therefore it makes the testing a bit slow but since the test it is done on only one host it is manageable.

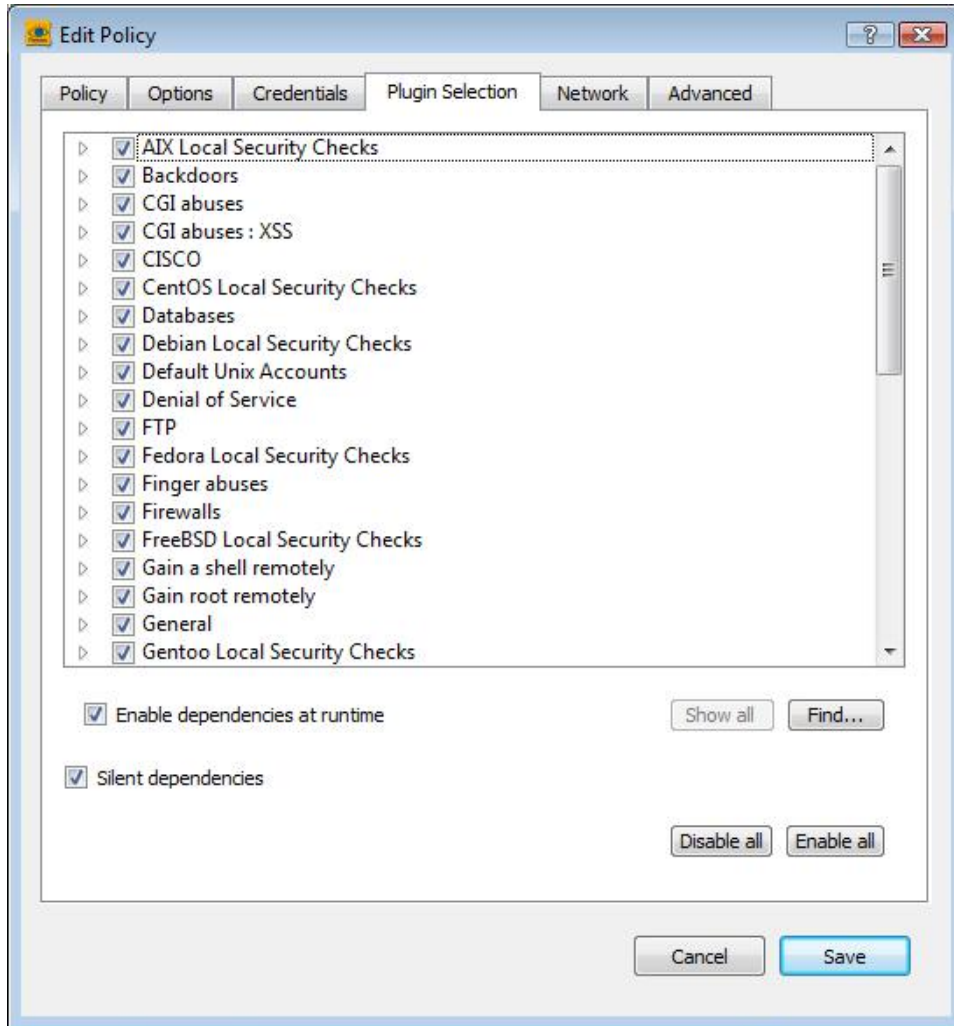


Figure B.2: Nessus scan plugins used



In case a congestion in the scan the total number of scans ran in parallel is reduced. This option was turned on during the scan like shown in Figure B.3.

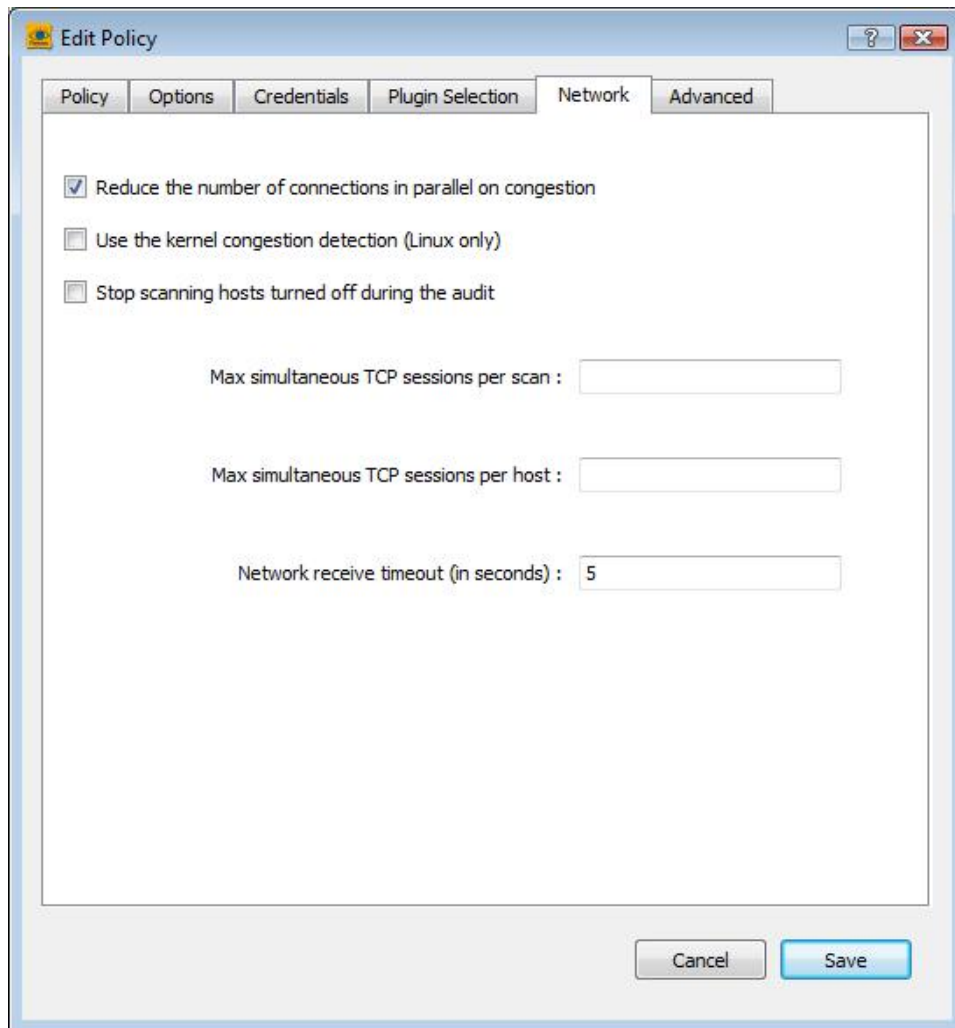


Figure B.3: Nessus network options

CGI scanning is enabled. It searches for well-known vulnerabilities in web servers and off-the-shelf web application software. Thorough tests is also turned on to make the testing more detailed. This option usually slows the system down a bit but since there is only one host scanned in our case it does not have a big impact on the scan time. This is shown in Figure B.4.

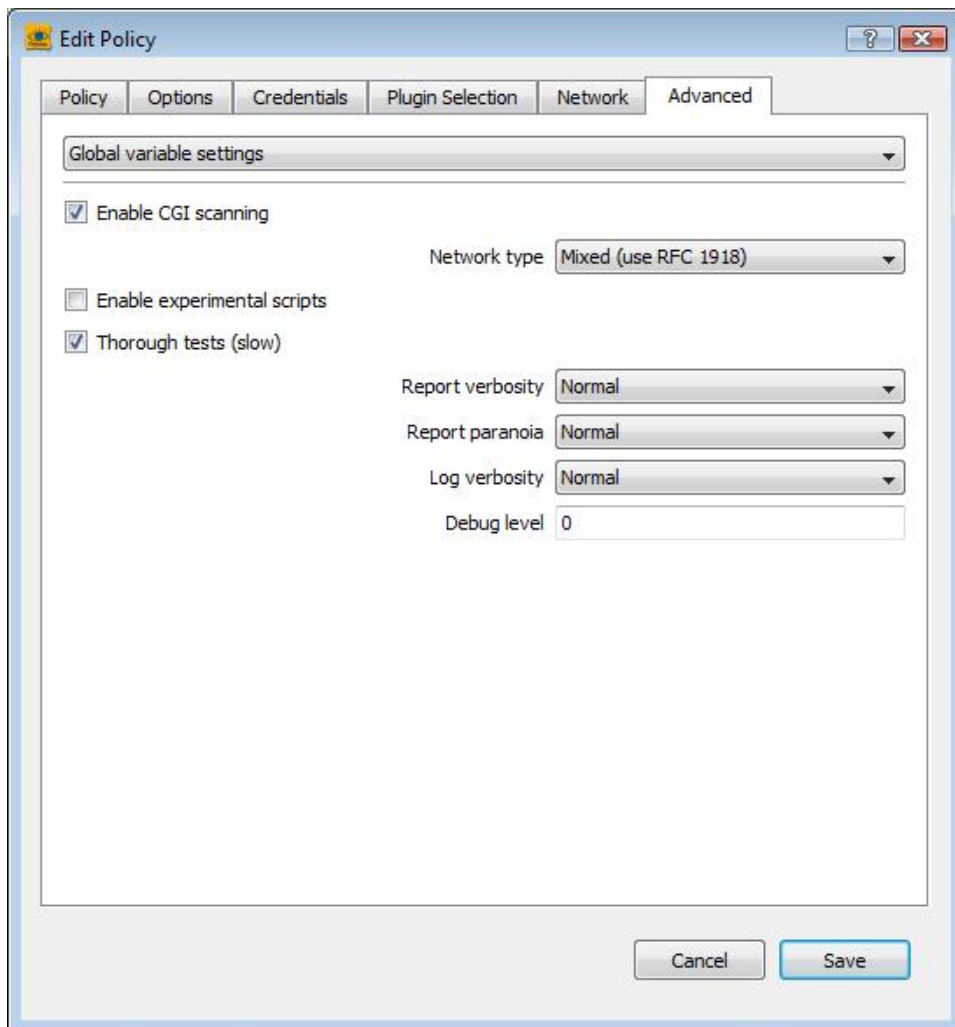


Figure B.4: Advanced Nessus options

The overview result of the scan is shown in Figure B.5. It shows 26 low level vulnerabilities, 2 medium level vulnerabilities and 13 open ports found on the OS. A more detailed description of each one will follow.

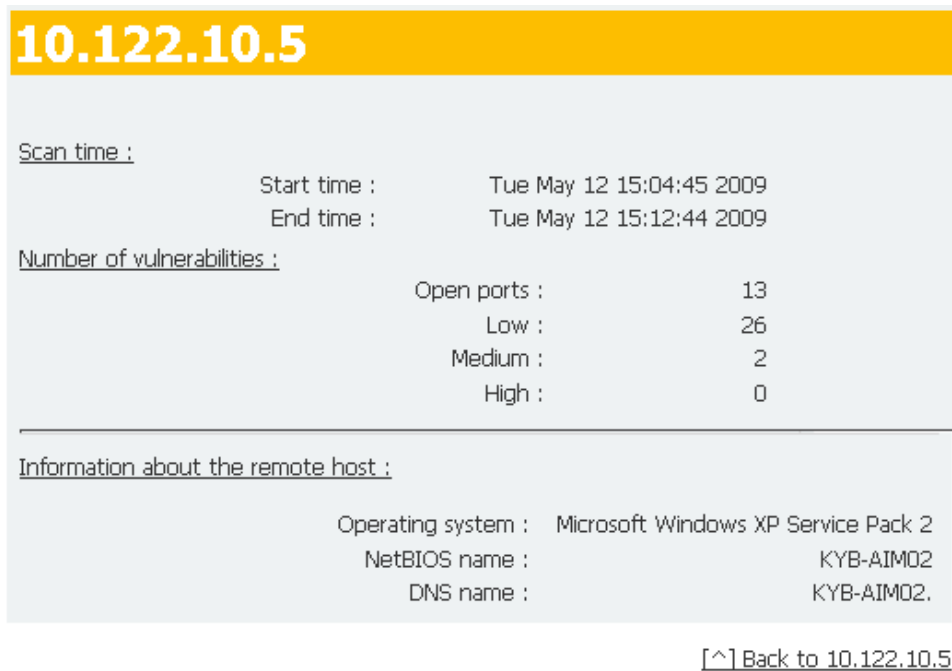


Figure B.5: Nessus scan overview

First of the two medium level vulnerabilities is shown in Figure B.6. It states that it may be possible to bypass firewall rules since the remote host does not discard TCP SYN packets which have the FIN flag set.

**Remote host replies to SYN+FIN**

**Synopsis :**

It may be possible to bypass firewall rules.

**Description :**

The remote host does not discard TCP SYN packets which have the FIN flag set.

Depending on the kind of firewall you are using, an attacker may use this flaw to bypass its rules.

**See also :**

<http://archives.neohapsis.com/archives/bugtraq/2002-10/0266.html>  
<http://www.kb.cert.org/vuls/id/464113>

**Solution :**

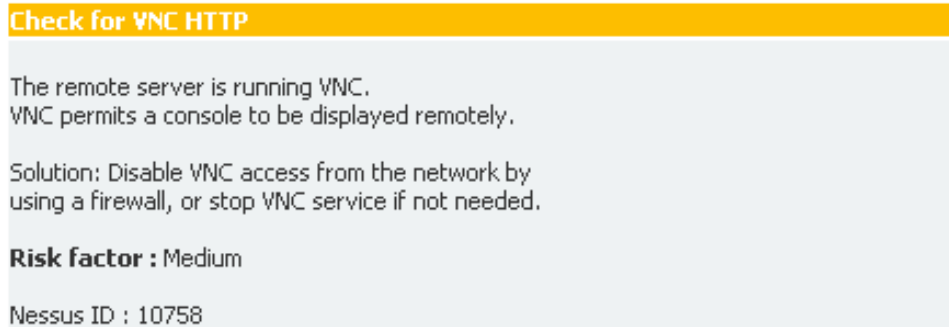
Contact your vendor for a patch.

**Risk factor :**

Medium / CVSS Base Score : 5.0  
(CVSS2#AV:N/AC:L/Au:N/C:N/I:P/A:N)  
BID : 7487  
Other references : OSVDB:2118  
Nessus ID : 11618

Figure B.6: Syn and Fin can be set to bypass firewall rules

The second medium level vulnerability is displayed in Figure B.7 and indicates that the VNC server running on the Os may be security risk.



**Check for VNC HTTP**

The remote server is running VNC.  
VNC permits a console to be displayed remotely.

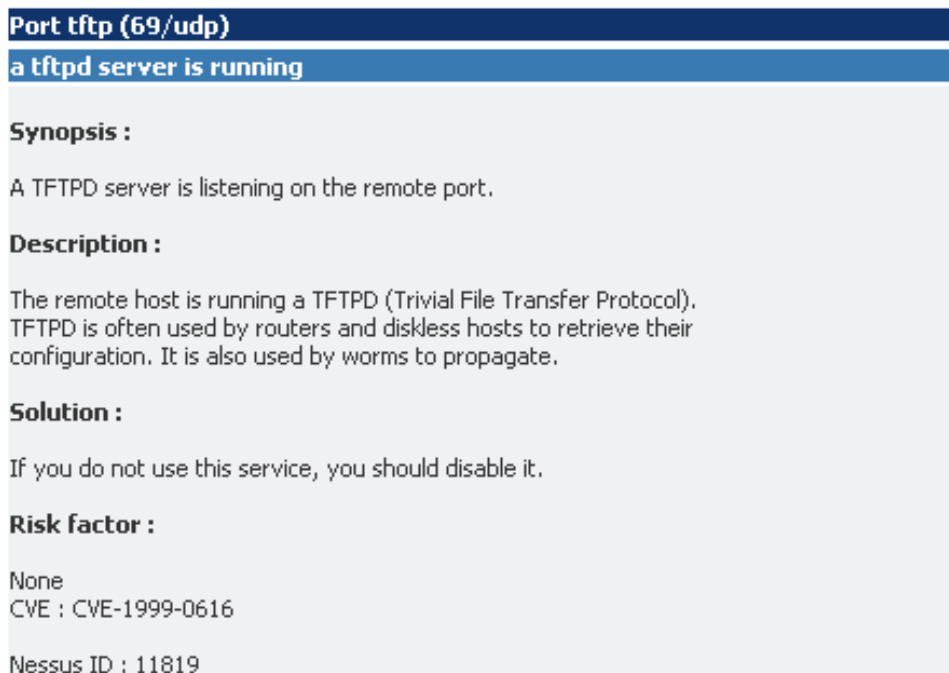
Solution: Disable VNC access from the network by using a firewall, or stop VNC service if not needed.

**Risk factor :** Medium

Nessus ID : 10758

Figure B.7: VNC server

Remainder of the low level vulnerabilities are sorted by the port number used. A TFTPd is listening on port 69 like shown in Figure B.8.



**Port tftp (69/udp)**

**a tftpd server is running**

**Synopsis :**

A TFTPd server is listening on the remote port.

**Description :**

The remote host is running a TFTPd (Trivial File Transfer Protocol). TFTPd is often used by routers and diskless hosts to retrieve their configuration. It is also used by worms to propagate.

**Solution :**

If you do not use this service, you should disable it.

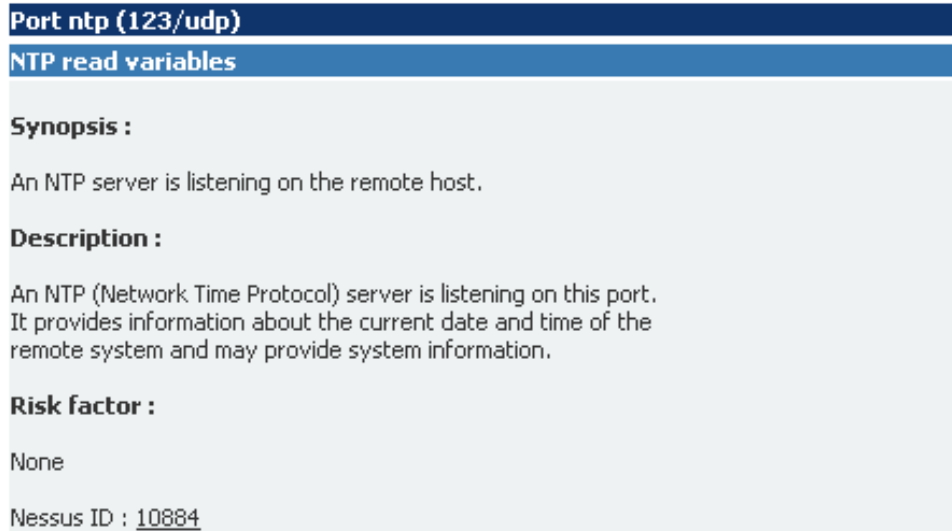
**Risk factor :**

None  
CVE : CVE-1999-0616

Nessus ID : 11819

Figure B.8: TFTPd is running on Port 69

Figure B.9 shows that a NTP server is listening on port 123.



**Port ntp (123/udp)**

**NTP read variables**

**Synopsis :**

An NTP server is listening on the remote host.

**Description :**

An NTP (Network Time Protocol) server is listening on this port. It provides information about the current date and time of the remote system and may provide system information.

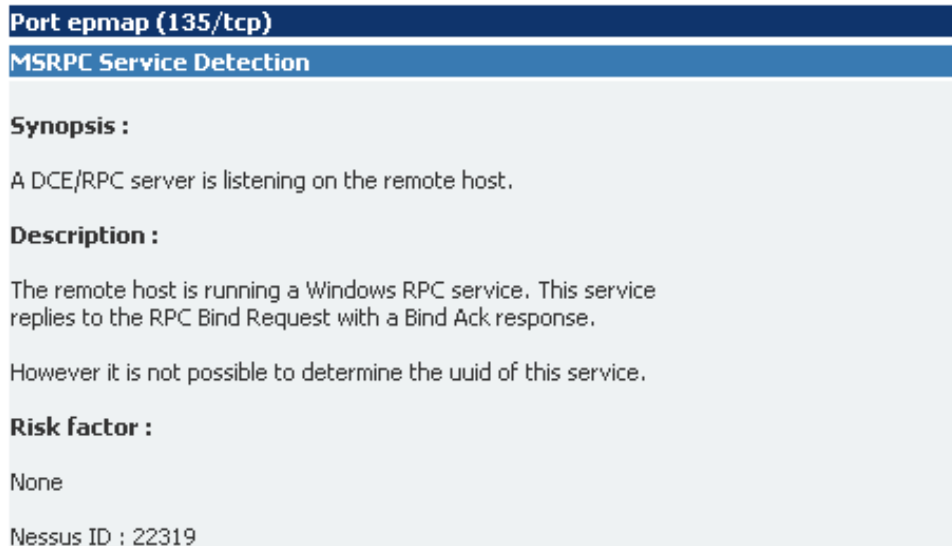
**Risk factor :**

None

Nessus ID : [10884](#)

Figure B.9: NTP server is listening on port 123

Port 135 is used by a Windows RPC service like shown in Figure B.10.



**Port epmap (135/tcp)**

**MSRPC Service Detection**

**Synopsis :**

A DCE/RPC server is listening on the remote host.

**Description :**

The remote host is running a Windows RPC service. This service replies to the RPC Bind Request with a Bind Ack response.

However it is not possible to determine the uuid of this service.

**Risk factor :**

None

Nessus ID : 22319

Figure B.10: Windows RPC service is in use

The OS uses port 137 for NetBIOS nbtscan requests like shown in Figure B.11.

**Port netbios-ns (137/udp)**

**Using NetBIOS to retrieve information from a Windows host**

**Synopsis :**

It is possible to obtain the network name of the remote host.

**Description :**

The remote host listens on udp port 137 and replies to NetBIOS nbtscan requests. By sending a wildcard request it is possible to obtain the name of the remote system and the name of its domain.

**Risk factor :**

None

**Plugin output :**

The following 4 NetBIOS names have been gathered :

KYB-AIM02 = Computer name  
KYBERNETIKK = Workgroup / Domain name  
KYB-AIM02 = File Server Service  
KYBERNETIKK = Browser Service Elections

The remote host has the following MAC address on its adapter :  
00:04:23:d0:6a:53  
CVE : CVE-1999-0621  
Other references : OSVDB:13577

Nessus ID : [10150](#)

Figure B.11: NetBIOS in use



Figure B.12 shows that a SMB server is running on port 139.

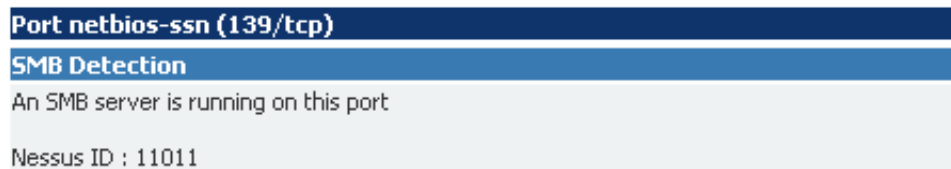


Figure B.12: SMB in use

Port 445 is used for file and resource sharing and allows information about the OS. This is shown in Figure B.13

**Port microsoft-ds (445/tcp)**

**SMB Detection**  
A CIFS server is running on this port  
Nessus ID : 11011

**SMB NativeLanMan**

**Synopsis :**  
It is possible to obtain information about the remote operating system.

**Description :**  
It is possible to get the remote operating system name and version (Windows and/or Samba) by sending an authentication request to port 139 or 445.

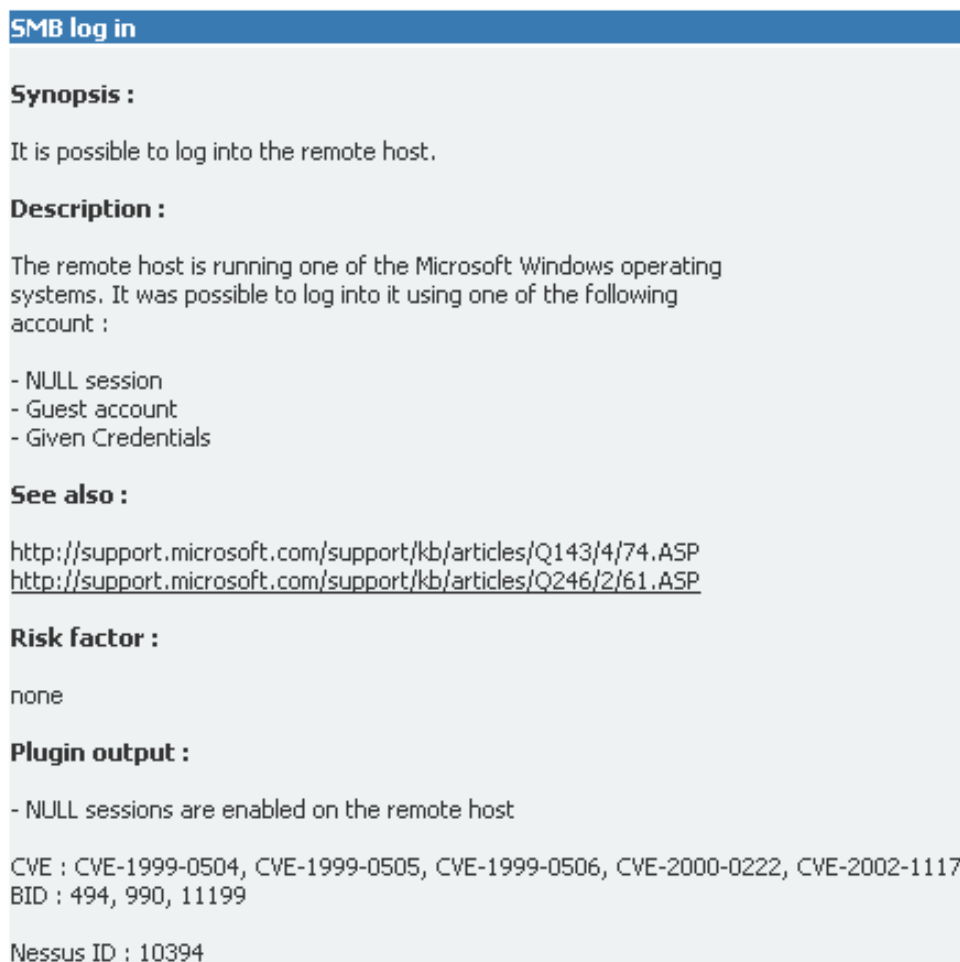
**Risk factor :**  
None

**Plugin output :**  
The remote Operating System is : Windows 5.1  
The remote native lan manager is : Windows 2000 LAN Manager  
The remote SMB Domain Name is : KYBERNETIKK

Nessus ID : [10785](#)

Figure B.13: Port 445 overview

Other services that port 445 is used for are shown in Figure B.14 and Figure B.15



**SMB log in**

**Synopsis :**

It is possible to log into the remote host.

**Description :**

The remote host is running one of the Microsoft Windows operating systems. It was possible to log into it using one of the following account :

- NULL session
- Guest account
- Given Credentials

**See also :**

<http://support.microsoft.com/support/kb/articles/Q143/4/74.ASP>  
<http://support.microsoft.com/support/kb/articles/Q246/2/61.ASP>

**Risk factor :**

none

**Plugin output :**

- NULL sessions are enabled on the remote host

CVE : CVE-1999-0504, CVE-1999-0505, CVE-1999-0506, CVE-2000-0222, CVE-2002-1117  
BID : 494, 990, 11199

Nessus ID : [10394](#)

Figure B.14: SMB log in

## SMB NULL session

### Synopsis :

It is possible to log into the remote host.

### Description :

The remote host is running one of the Microsoft Windows operating systems. It was possible to log into it using a NULL session.

A NULL session (no login/password) allows to get information about the remote host.

### See also :

<http://support.microsoft.com/support/kb/articles/Q143/4/74.ASP>  
<http://support.microsoft.com/support/kb/articles/Q246/2/61.ASP>

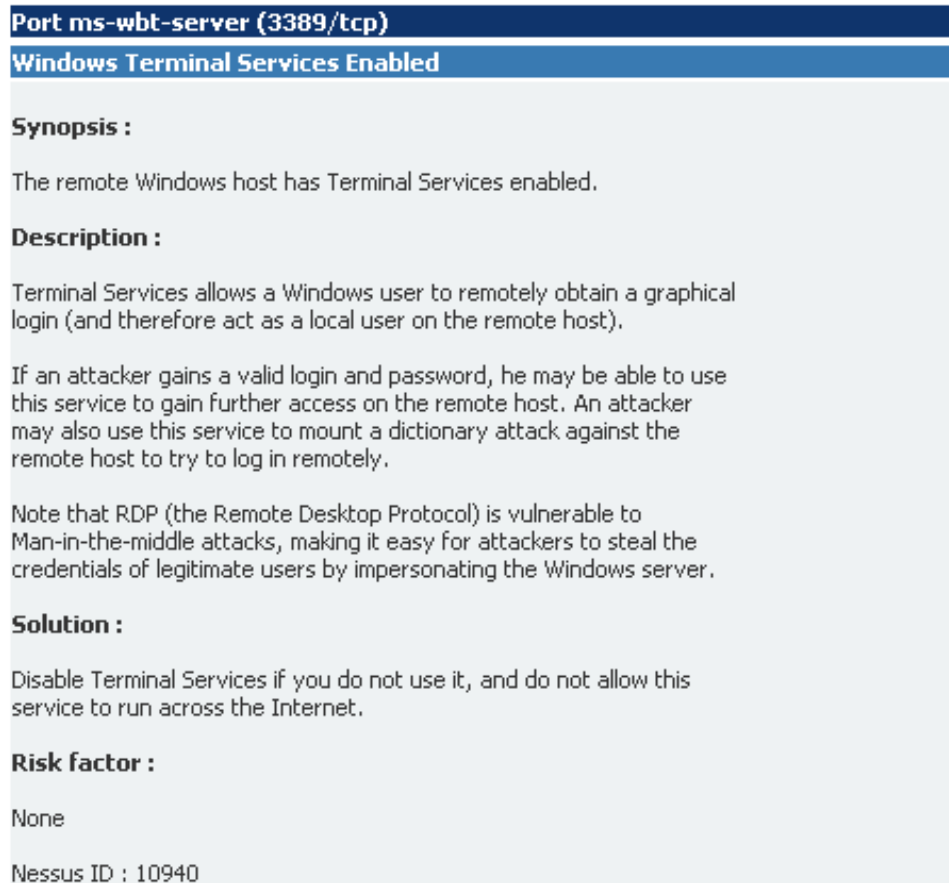
### Risk factor :

None  
CVE : CVE-2002-1117  
BID : 494

Nessus ID : [26920](#)

Figure B.15: SMB null session log in

For port 3389 Windows Terminal Services are enabled, like displayed in Figure B.16



**Port ms-wbt-server (3389/tcp)**

**Windows Terminal Services Enabled**

**Synopsis :**

The remote Windows host has Terminal Services enabled.

**Description :**

Terminal Services allows a Windows user to remotely obtain a graphical login (and therefore act as a local user on the remote host).

If an attacker gains a valid login and password, he may be able to use this service to gain further access on the remote host. An attacker may also use this service to mount a dictionary attack against the remote host to try to log in remotely.

Note that RDP (the Remote Desktop Protocol) is vulnerable to Man-in-the-middle attacks, making it easy for attackers to steal the credentials of legitimate users by impersonating the Windows server.

**Solution :**

Disable Terminal Services if you do not use it, and do not allow this service to run across the Internet.

**Risk factor :**

None

Nessus ID : [10940](#)

Figure B.16: Windows Terminal Services running

Port 5800 is listening for a VNC web server. HTTP settings are enabled, as shown in Figure B.17.

**Port vnc-http (5800/tcp)**

**Service detection**  
A web server is running on this port.  
Nessus ID : [22964](#)

**HyperText Transfer Protocol Information**

**Synopsis :**  
Some information about the remote HTTP configuration can be extracted.

**Description :**  
This test gives some information about the remote HTTP protocol - the version used, whether HTTP Keep-Alive and HTTP pipelining are enabled, etc...  
  
This test is informational only and does not denote any security problem

**Solution :**  
None.

**Risk factor :**  
None

**Plugin output :**  
Protocol version : HTTP/1.0  
SSL : no  
Pipelining : no  
Keep-Alive : no  
Headers :  
  
Nessus ID : [24260](#)

Figure B.17: VNC server running

A VNC server is listening on port 5900 but with slightly possibilities, as shown in Figure B.18.

**Port vnc (5900/tcp)**

**Service detection**

A vnc server is running on this port.

Nessus ID : [22964](#)

**Check for VNC**

**Synopsis :**

The remote host is running a remote display software (VNC).

**Description :**

The remote server is running VNC, a software which permits a console to be displayed remotely. This allows users to control the host remotely.

**Solution :**

Make sure the use of this software is done in accordance with your corporate security policy and filter incoming traffic to this port.

**Risk factor :**

None

**Plugin output :**

The version of the VNC protocol is : RFB 003.008

Nessus ID : [10342](#)

Figure B.18: VNC on port 5900

A Sentinel Protection server is running on port 6002. This is shown in Figure B.19

**Port X11:2 (6002/tcp)**

**Service detection**  
A web server is running on this port.  
Nessus ID : [22964](#)

**HTTP Server type and version**

**Synopsis :**  
A web server is running on the remote host.

**Description :**  
This plugin attempts to determine the type and the version of the remote web server.

**Risk factor :**  
None

**Plugin output :**  
The remote web server type is :  
SentinelProtectionServer/7.1

Nessus ID : [10107](#)

Figure B.19: Sentinel web server running



Additional information about the HTTP protocol in use on port 6002 is shown in Figure B.20



**HyperText Transfer Protocol Information**

**Synopsis :**  
Some information about the remote HTTP configuration can be extracted.

**Description :**  
This test gives some information about the remote HTTP protocol - the version used, whether HTTP Keep-Alive and HTTP pipelining are enabled, etc...  
  
This test is informational only and does not denote any security problem

**Solution :**  
None.

**Risk factor :**  
None

**Plugin output :**  
Protocol version : HTTP/1.0  
SSL : no  
Pipelining : yes  
Keep-Alive : yes  
Headers :  
  
Date: Tue, 12 May 2009 11:57:51 GMT  
Server: SentinelProtectionServer/7.1  
MIME-Version: 1.1  
Content-Type: text/html  
Keep-Alive: 1  
Content-Length: 2456

Nessus ID : [24260](#)

Figure B.20: Protocol in use on the Sentinel server

The last three ports (7000, 7001 and 7777) are used for a fileserver, a callback to the cach manager and Cbt services. Very little information could be recieved about them from the Nessus scan, like shown in Figure B.21

**Port afs3-fileserver (7000/tcp)**

**Port afs3-callback (7001/tcp)**

**Port cbt (7777/tcp)**

Figure B.21: Information on the last three open ports

Additional information could also be received not regarding the use of ports, such as the fact that ICMP protocol was in use. This is shown in Figure B.22

**Port general/icmp**  
**icmp timestamp request**

**Synopsis :**  
It is possible to determine the exact time set on the remote host.

**Description :**  
The remote host answers to an ICMP timestamp request. This allows an attacker to know the date which is set on your machine.  
  
This may help him to defeat all your time based authentication protocols.

**Solution :**  
Filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14).

**Risk factor :**  
None

**Plugin output :**  
This host returns non-standard timestamps (high bit is set)  
The ICMP timestamps might be in little endian format (not in network format)  
The difference between the local and remote clocks is 4390 seconds

CVE : CVE-1999-0524  
Nessus ID : [10114](#)

Figure B.22: ICMP protocol

Some information was fetched by Nessus over the TCP protocol. The information is displayed in Figure B.23 and B.24. The information fetched tells:

- the OS can be pinged.
- timestamps are in use.
- the host name is displayed.
- the operative system type a long with a certainty level.
- information about the scan.

The screenshot displays a series of scan results in a structured format. It begins with a header 'Port general/tcp' in a dark blue bar. Below this is a section titled 'Ping the remote host' in a lighter blue bar, containing the text 'The remote host is up' and 'Nessus ID : 10180'. The next section is 'TCP timestamps' in a dark blue bar. This section includes a 'Synopsis :' section with the text 'The remote service implements TCP timestamps.', a 'Description :' section with the text 'The remote host implements TCP timestamps, as defined by RFC1323. A side effect of this feature is that the uptime of the remote host can sometimes be computed.', a 'See also :' section with the URL 'http://www.ietf.org/rfc/rfc1323.txt', and a 'Risk factor :' section with the text 'None'. This section also includes 'Nessus ID : 25220'. The final section is 'Host FQDN' in a dark blue bar, containing the text '10.122.10.5 resolves as KYB-AIM02.' and 'Nessus ID : 12053'.

**Port general/tcp**

**Ping the remote host**

The remote host is up

Nessus ID : 10180

**TCP timestamps**

**Synopsis :**

The remote service implements TCP timestamps.

**Description :**

The remote host implements TCP timestamps, as defined by RFC1323. A side effect of this feature is that the uptime of the remote host can sometimes be computed.

**See also :**

<http://www.ietf.org/rfc/rfc1323.txt>

**Risk factor :**

None

Nessus ID : 25220

**Host FQDN**

10.122.10.5 resolves as KYB-AIM02.

Nessus ID : 12053

Figure B.23: TCP information

**OS Identification**

Remote operating system : Microsoft Windows XP Service Pack 2  
Confidence Level : 99  
Method : MSRPC

The remote host is running Microsoft Windows XP Service Pack 2

Nessus ID : [11936](#)

**Information about the scan**

Information about this scan :

Nessus version : 3.2.1.1  
Plugin feed version : \$Date: 2005/11/08 13:18:41 \$  
Type of plugin feed : CV5  
Scanner IP : 10.122.10.3  
Port scanner(s) : synscan  
Port range : default  
Thorough tests : yes  
Experimental tests : no  
Paranoia level : 1  
Report Verbosity : 2  
Safe checks : yes  
Optimize the test : yes  
Max hosts : 20  
Max checks : 5  
Recv timeout : 5  
Scan Start Date : 2009/5/12 15:05  
Scan duration : 458 sec

Nessus ID : [19506](#)

Figure B.24: Additional TCP info

A UDP traceroute is shown in Figure B.25. Since the attacking pc is connected through a switch the trace route only shows the two addresses (of the attacker and the OS), without any additional hops required.

```
Port general/udp
Traceroute
For your information, here is the traceroute from 10.122.10.3 to 10.122.10.5 :
10.122.10.3
10.122.10.5

Nessus ID : 10287
```

Figure B.25: UDP trace route

## B.2 Cain & Abel details

Cain & Abel was in our case used of decryption of the VNC password. As described in Subsection 5.2.1 the password was stored in the registry of the OS. The password is encrypted using triple DES algorithm, hashed and stored in Hexadecimals. Cain & Abel uses rainbow tables to nearly instantly find the password. Rainbow tables are tables of already cracked hashes [15]. This makes it very efficient to find the cleartext passwords from hashed passwords.

Figure B.26 shows the overview of the Cain & Abel password finding. The password hash along with the cleartext of the password was censored due to privacy reasons of the laboratory.

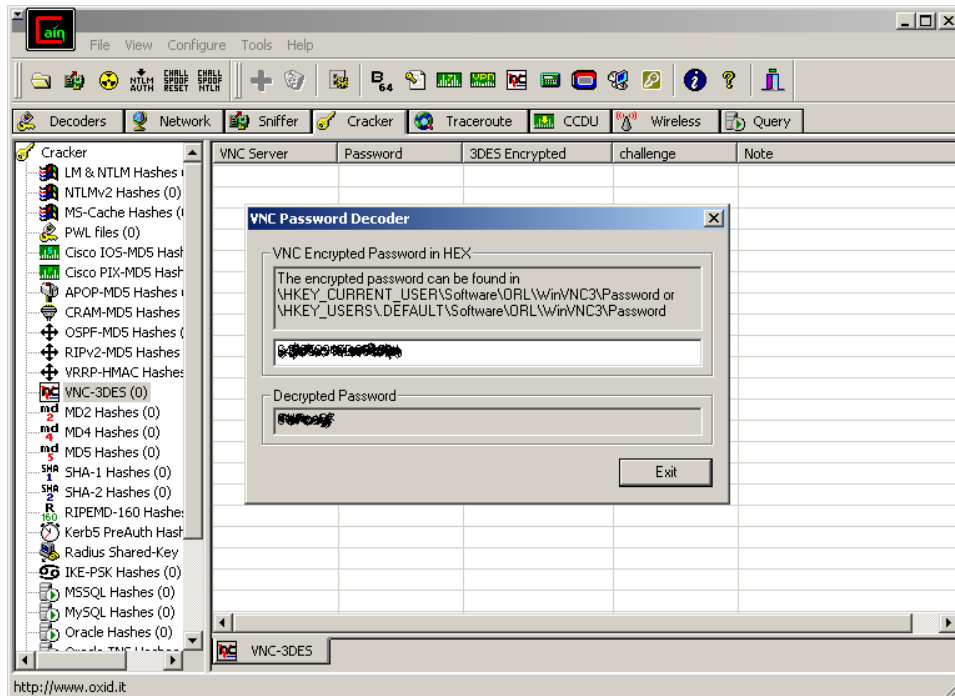


Figure B.26: 3DES reverse decryption

# Bibliography

- [1] *Cain and Abel user manual*, [http://www.oxid.it/ca\\_um/](http://www.oxid.it/ca_um/) Last used: 12.6.09.
- [2] *Colasoft packet builder*, [http://www.colasoft.com/packet\\_builder/](http://www.colasoft.com/packet_builder/) Last used: 12.6.09.
- [3] *Colasoft packet player*, [http://www.colasoft.com/packet\\_player/](http://www.colasoft.com/packet_player/) Last used: 12.6.09.
- [4] *Ethereal: A network protocol analyzer*, <http://www.ethereal.com/> Last used: 12.6.09.
- [5] *IP Stack Integrity Checker*, <http://isic.sourceforge.net/> Last used: 12.6.09.
- [6] *Metasploit homepage*, <http://www.metasploit.com/> Last used: 12.6.09.
- [7] *Nessus scanner*, <http://www.nessus.org/nessus/> Last used: 12.6.09.
- [8] *Netwox network toolbox*, <http://www.laurentconstantin.com/en/netw/netwox/> Last used: 12.6.09.
- [9] *Nmap homepage*, <http://nmap.org/> Last used: 12.6.09.
- [10] *THC Hydra*, <http://freeworld.thc.org/thc-hydra/> Last used: 12.6.09.
- [11] *TightVNC homepage*, <http://www.tightvnc.com/> Last used: 12.6.09.
- [12] *Wireshark homepage*, <http://www.wireshark.org/> Last used: 12.6.09.
- [13] E. Byres and D. Hoffman, *The myths and facts behind cyber security risks for industrial control systems*, Tech. report, International Society of Automation, 2002, [http://www.isa.org/CustomSource/ISA/Div\\_PDFs/PDF\\_News/Glss\\_2.pdf](http://www.isa.org/CustomSource/ISA/Div_PDFs/PDF_News/Glss_2.pdf) Last used: 12.6.09.



- [14] CERT/CC, *CERT advisory CA-2003-20 W32/Blaster worm*, Tech. report, Carnegie Mellon University, 2003, <http://www.cert.org/advisories/CA-2003-20.html> Last used: 12.6.09.
- [15] C. Clark, L. Chaffin, A. Chuvakin, S. Paladino, R. Ford, D. Dunkel, S. Fogie, M. Gregg, C. DeRodeff, and C. Schiller, *Infosecurity 2008 threat analysis*, Syngress, 2007, ISBN 1597492248, 9781597492249.
- [16] A. Clem, S. Galwankar, and G. Buck, *Health implications of cyber-terrorism*, Tech. report, Prehospital and Disaster Medicine, 2003, <http://pdm.medicine.wisc.edu/18-3pdfs/272C1em.pdf> Last used: 12.6.09.
- [17] D.P. Duggan, *Penetration testing of industrial control systems*, Tech. report, Sandia Report SAND2005-2846P, 2005.
- [18] W. Eddy, *TCP SYN flooding attacks and common mitigations*, 2007, <http://tools.ietf.org/html/rfc4987> Last used: 12.6.09.
- [19] J. Falco, K. Stouffer, A. Wavering, and F. Proctor, *IT security for industrial control systems*, Tech. report, National Institute of Standards and Technology (NIST), 2002, <http://www.isd.mel.nist.gov/documents/falco/ITSecurityProcess.pdf> Last used: 12.6.09.
- [20] P. Ferguson and D. Senie, *Network ingress filtering: Defeating denial of service*, 1998, <http://www.faqs.org/rfcs/rfc2267.html> Last used: 12.6.09.
- [21] J. Gardner and D. Weber, *Converting serial networks to ethernet communications*, International Society of Automation (2007), [http://www.isa.org/InTechTemplate.cfm?Section=Article\\_Index1&template=/ContentManagement/ContentDisplay.cfm&ContentID=65606](http://www.isa.org/InTechTemplate.cfm?Section=Article_Index1&template=/ContentManagement/ContentDisplay.cfm&ContentID=65606) Last used: 12.6.09.
- [22] R.L. Glass, *The software-research crisis*, IEEE Software, Volume: 11, Issue: 6, 1994, [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=329400](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=329400) Last used: 12.6.09.
- [23] S. Gorman, *Electricity grid in U.S. penetrated by spies*, The Wall Street Journal (2009), <http://online.wsj.com/article/SB123914805204099085.html> Last used: 12.6.09.
- [24] M. Hentea, *Improving security for SCADA control systems*, Interdisciplinary Journal of Information, Knowledge, and Management, Volume 3, 2008, <http://ijikm.org/Volume3/IJIKMv3p073-086Hentea361.pdf> Last used: 12.6.09.

- [25] E. Levy and I. Arce, *Crossover: Online pests plaguing the offline world*, IEEE Security and Privacy, vol. 1, no. 6, 2003, <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01253573> Last used: 12.6.09.
- [26] S. Luders, *TOCSSiC a teststand on control system security at CERN*, CERN-TOCSSIC, no. 2005-001, Revision 2.0, 2005.
- [27] C. McNab, *Network security assessment*, O'Reilly, 2007.
- [28] M.R. Permann and K. Rohde, *Cyber assessment methods for SCADA security*, The 15th Annual joint ISA POWID/EPRI Controls and Instrumentation Conference, 2005.
- [29] D. Peterson, *Langner awareness demonstration tool*, Digital Bond (2009), <http://www.digitalbond.com/index.php/2009/02/10/langner-awareness-demonstration-tool/> Last used: 12.6.09.
- [30] K. Poulsen, *Slammer worm crashed ohio nuke plant network*, Security Focus (2003), <http://www.securityfocus.com/news/6767> Last used: 12.6.09.
- [31] P.A.S. Ralston, J.H. Graham, and J.L. Hieb, *Cyber security risk assessment for SCADA and DCS networks*, Tech. report, ScienceDirect from International Society of Automation transaction, 2007, [http://www.sciencedirect.com/science?\\_ob=MIimg&\\_imagekey=B6V3P-4P59S0G-1-1&\\_cdi=5736&\\_user=586462&\\_orig=search&\\_coverDate=10%2F31%2F2007&\\_sk=999539995&view=c&wchp=dGLbVzW-zSkWA&md5=1dbf7eb715c793e25aa4ad6a5008eb3d&ie=/sdarticle.pdf](http://www.sciencedirect.com/science?_ob=MIimg&_imagekey=B6V3P-4P59S0G-1-1&_cdi=5736&_user=586462&_orig=search&_coverDate=10%2F31%2F2007&_sk=999539995&view=c&wchp=dGLbVzW-zSkWA&md5=1dbf7eb715c793e25aa4ad6a5008eb3d&ie=/sdarticle.pdf) Last used: 12.6.09.
- [32] K. Salah and M. Hamawi, *Comparative packet-forwarding measurement of three popular operating systems*, Journal of Network and Computer Applications, 2009, [http://www.sciencedirect.com/science?\\_ob=ArticleURL&\\_udi=B6WKB-4W15KXD-1&\\_user=586462&\\_rdoc=1&\\_fmt=&\\_orig=search&\\_sort=d&view=c&\\_acct=C000030078&\\_version=1&\\_urlVersion=0&\\_userid=586462&md5=6cc830b48085953fb604525920e4de90#implicit0](http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6WKB-4W15KXD-1&_user=586462&_rdoc=1&_fmt=&_orig=search&_sort=d&view=c&_acct=C000030078&_version=1&_urlVersion=0&_userid=586462&md5=6cc830b48085953fb604525920e4de90#implicit0) Last used: 12.6.09.
- [33] J.T. Sørensen, *Security in industrial networks*, Master's thesis, NTNU, 2007.
- [34] R. Szostak, *Project assignment security in process control systems*, Tech. report, Department of Telematics Faculty of Information Technology, Mathematics and Electrical, 2008.