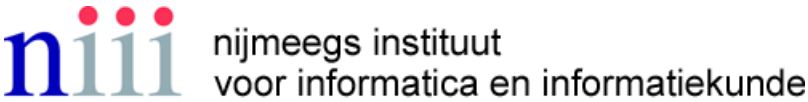
SURFnet Distributed IDS Proposal

Roberto Nebot Gozalbo

September 29, 2006

Supervisors: Engelbert Hubbers and Marco van Eekelen



Contents

1 Preface

The objective of this document is to describe a proposal for research. In the following sections a general overview of the system being investigated will be shown and information about the company which provides this service will be given.

The D-IDS proposal will provide a general guideline of the activities and the methods that need to be performed to evaluate the secureness of the system. Moreover, it will establish a basic timeline to perform the cited evaluations.

2 Introduction

2.1 D-IDS Overview

The SURFnet Distributed Intrusion Detection System is a type of IDS created by SURFnet as an open source project. It intends to get one step ahead to the other IDSs by solving many of the problems the other IDSs suffer from. Nowadays D-IDSs are based on the client/server approach where the client is the sensor and the server contains the IDS (mainly Snort[1]) and sometimes a honeypot. This kind of system presents some problems which can be summarized as four major disadvantages.

- The sensor needs to be upgradeable in order to add future honeypot and new signatures.
- The sensor may be vulnerable to the exploits used against honeypot and passive analysis software.
- The D-IDS will generate false positive alerts.
- Installing and running the sensor is not plug and play.

During the research project we intend to prove that the D-IDS approach of SURFnet can solve these problems. To realize this studies will be used the Distributed IDS design which SURFnet proposes and at the end it will be seen how secure the system is.

The figure 1 will provide a clear idea about what elements are involved in the SURFnet D-IDS.

In the figure the structure of the D-IDS developed by the SURFnet open source project is shown. The D-IDS is composed by the sensors which send the information collected on the network using vpn tunnels between server LAN and sensor LANs. The server LAN is usually composed by two computers, the main system that runs a Honeypot and the VPN services and the other computer that runs the web interface and the Postgres database. The vpn services are used to provide privacy between the sensor LANs and

[1]Snort is an open source network intrusion prevention and detection system utilizing a rule-driven language, which combines the benefits of signature, protocol and anomaly based inspection methods. With millions of downloads to date, Snort is the most widely deployed intrusion detection and prevention technology worldwide and has become the de facto standard for the industry. *Source:* `http://www.snort.org`

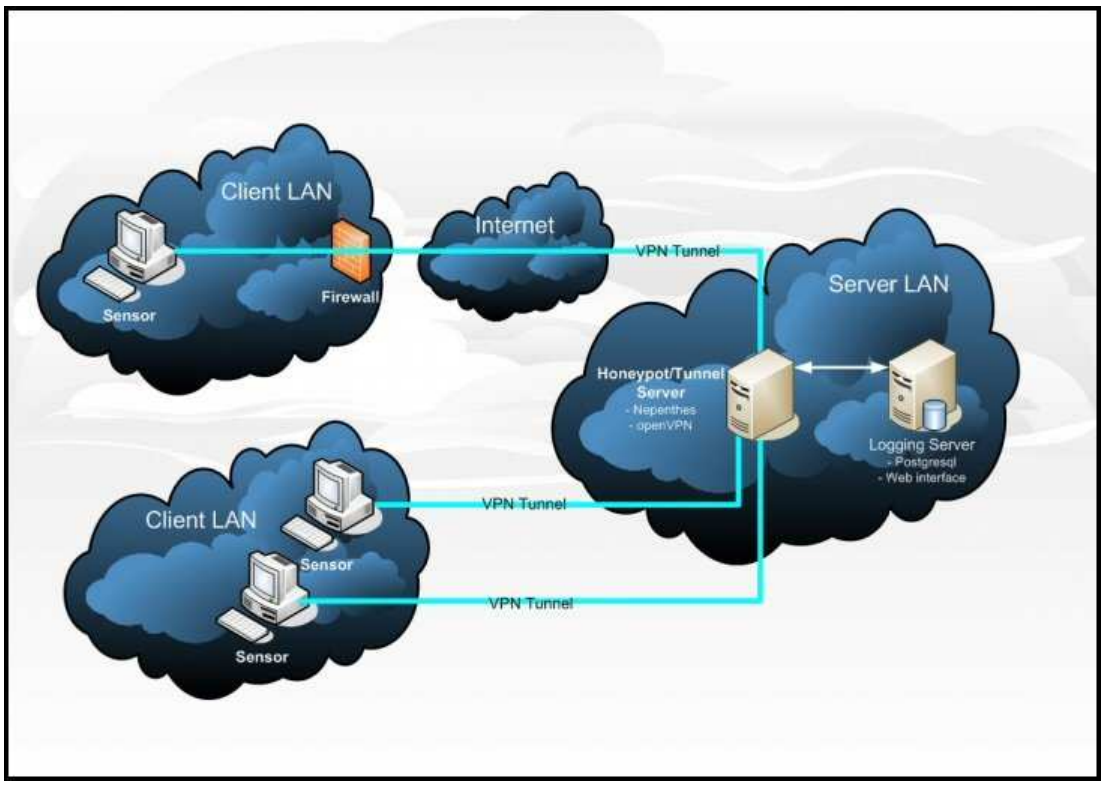


Figure 1: Distributed IDS obtained from *http://ids.surfnet.nl*

server LAN (honeypot). The web interface is used to provide an easy way to access the database information.

2.2 Project's founder background

The system that is being studied was released by a company called SURFnet as an Open Source project[2]. So, it is possible for everybody to download the system and to run a service of net analyzation using this IDS. On the webpage a complete guide to configure the entire system is available.

SURFnet is a type of ISP(Internet Service Provider). This company has deployed a high-grade computer network specially reserved for the education and research in the Netherlands. SURFnet offers to its connected institutes several services. See table 1 for an overview.

[2]You can download the sensor and the server software in `http://ids.surfnet.nl` or use the subversion repository in `http://sourceforge.net/projects/surfnetids/`

Offered Services	
Conferencing	Connection to SURFnet
Consultancy	Detective
Eduroam	Email
Experimental Services	FTP
Helpdesk	IP-Address and domain names
Discussion Lists	Multicast
News	Reports
Security	SURFnet Search Engine
SURFnet D-IDS	SURFnet Internet Access at home
SURFnet User Groups	Time Service
Video and Audio	

Table 1: Services offered by SURFnet. For more information see *http://www.surfnet.nl/en/index_en.html*

3 State of the art

3.1 IDS overview

In this section some security concepts will be provided and Internet historical notes will be given. Furthermore, IDSs development will be justified as a tool to solve some type of security problems. At the end of the section, IDSs will be classified according to a defined criteria, as well as, some issues discussed.

These days, since the use of Ethernet has become a standard, the use of networks in the world has increased every day. During the eighties each company had its own network but the creation of the first web browser[3] and the release of the web service by Tim Berners-Lee from CERN in 1989 changed something in the world and make possible the Vannevar's Bush[4] vision[5]. The following years during the nineties the use of Internet spread

[3]NCSA Mosaic was originally designed and programmed for Unix's X Window System by Marc Andressen and Eric Bina at NCSA. The system was released in 1993

[4]Vannevar Bush (March 11, 1890 - June 30, 1974) was an American engineer and science administrator, known for his political role in the development of the atomic bomb, and the idea of the memex–seen as a pioneering concept for the World Wide Web. Source: *Wikipedia.org*

[5]Vannevar Bush's essay *As We May Think*, first published in The Atlantic Monthly in July 1945, argued that as humans turned from war, scientific efforts should shift from increasing physical abilities to making all previous collected human knowledge more ac-

rapidly all over the world.

Nowadays, everyone wants to stay online, but this common use of Internet that is helping to evolve our society quickly has got a dark side. In the past, the security of systems was related only with people directly using the computers and with the devices that they plugged or introduced to it, but security also changed during the nineties with the spread of networks. All over the world the way attackers could compromise the security of systems and the velocity which they could spread their malware increased massively.

To protect ourself against these hazards, security experts developed new methods to make the networks safer. It's common in these days to deploy several security layers in networks. Modern secure networks contain at least a router with built-in firewall and another firewall to filter the input/output packets. Sometimes if the network provides services such as DNS, SMTP or WWW another element is been introduced, the DMZ (Demilitarized Zone) that is a subnetwork between the Internal network and Internet which can be accessed from outside and from inside but hosts in the DMZ may not connect with the internal network. See figure 2.

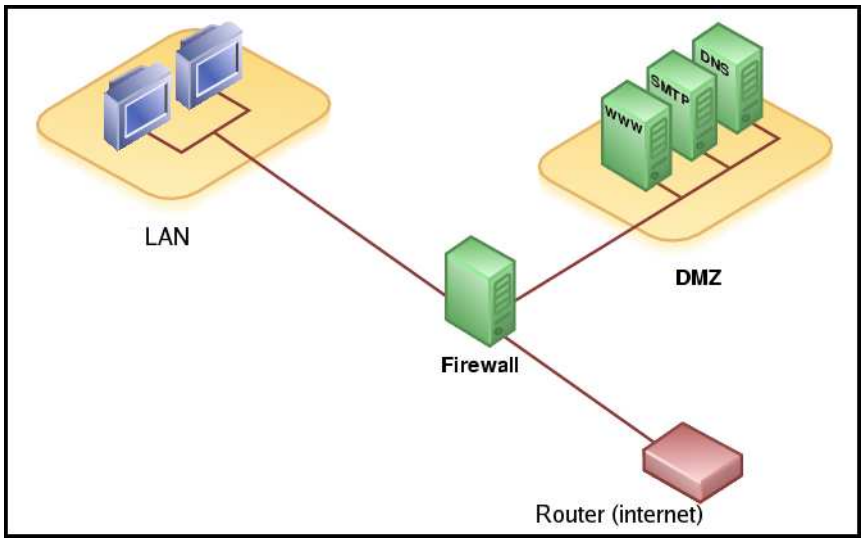


Figure 2: Structure of a Demilitarized Zone. Source: *http://en.wikipedia.org/wiki/Image:Demilitarized_Zone_Diagram.png*

cessible. You can find the essay at `http://www.ps.uni-sb.de/~duchier/pub/vbush/vbush-all.shtml`. Source: `http://en.wikipedia.org/wiki/As_We_May_Think`

It is possible to improve this organization but it is not the object of this study. There are several security layers in network but studies have revealed that most computer security incidents are caused by people who have privileges inside the network. (Sometimes they are just honest people that make mistakes.) On the other hand, there are malicious hackers and "script kiddies" that are able to pass through authentication controls and firewalls. An IDS offers us the opportunity to detect the attack at the beginning, when it finishes or while it is in progress. An IDS works like a smoke detector. An IDS is a device which collects data with its sensors to monitor activity in a network and detects malicious or suspicious events. If a defined rule of the IDS matches with an activity, then the IDS raises an alarm that it detected something suspicious.

IDSs can be classified according to the place that these systems have in the network.

- Network based: It is a stand-alone device attached to the network which monitors the traffic throughout the network. An example of this type of system is Snort.
- Host based: It is a system that runs in a host to protect that host. An example of this type of system is Tripwire.

Based upon the way the suspicious activities are found we can distinguish two types:

- Signature based
- Heuristic based

A signature based IDS will monitor the packets on the network and compare them against a database that contains signatures or attributes from known malicious threats. This is the way that many antivirus programs detect the malware. The problem occurs when a new type of attack appears. In this case the IDS is not able to detect the new threat. For example, Tripwire belongs to this type of IDS.

The heuristic based IDSs are also known as anomaly based. These IDS monitor network traffic and compare it against an established baseline. The objective of the baseline is to define what is "normal" on the net, what protocols are used, what amount of bandwidth is normally used and what

ports are used. When an abnormal use of a parameter is detected the system alerts the administrator. This kind of IDSs are able to detect malware hid to signature based approach. For example NIDES[6] is a heuristic based IDS.

The actual IDSs projects blend the two approaches.

After this short introduction of IDSs, their types and their functions we how focus on hiding its presence phenomenon of while. This is called stealth mode. At this moment the reader may think, why we need a stealth mode? The response is easy, during a while, imagine that an attacker break into a network, if this attacker knows the existence of an IDS perhaps he intend to compromise the availability of that system and in that case the IDS is not useful. For this reason, an IDS normally has two interfaces one monitors the network and the others sends the alerts.

But the current IDSs suffer some limitations. The most common limitation is the false positives. Sometimes the device raises an alarm and there is no problem in the network. This is what we call false positive. Other problem that an IDS experiments is the opposite case. Sometimes the system doesn't show that something goes wrong. Another limitation of the device is that the IDS doesn't run itself, the system raises alarms and mostly someone has to react against them. So to manage sensitivity of the IDS is critical. You should have to find the proper configuration, otherwise the system generates lots of false positives and you can not trust the system which implies that is not useful.

3.2 Other systems

There are some systems related with the IDS. Some security engineers would classify them as IDS as well, although they do not match the general definition.

For example, the programs called system vulnerability scanners, the most well known is Nessus. These perform a port scan and after that try to exploit some vulnerabilities in the open ports reporting the vulnerabilities found. If we think carefully, these programs work in the same way that the IDSs collect information from a sensor in a network: if a rule matches with the stored patterns an alarm or notification is raised. However, these types of programs

[6]NIDES(Next-generation Intrusion Detection Expert System) was the result of a research started in the Computer Science Laboratory at SRI international in the early 1980s.

are not interesting to us. System vulnerabilities programs are not passive elements of the network like IDSs. Vulnerability scanners are used to find vulnerabilities and we are focus on programs that wait for attacks.

But there is another kind of systems that we are interested in. It is quite similar and called honeypot. A honeypot would be considered an IDS, in the sense that a honeypot(like bees to honey), attract attackers and then records all the activities that the attacker does against the trap system.

The idea of what a honeypot is, appeared in a Cliff Stoll book called the Cuckoo 's Egg in 1990, during the nineties the idea was developing and at the end of nineties appeared the first applications. Lance Spitzner defines a honeypot as *"an information system resource whose values lies in unauthorized or illicit use of that resource"*. In fact, a widely accepted definition for this research, could be defining a honeypot as a fictious vulnerable IT System used for the purpose of being attacked, probed, exploited and compromised.

It is possible to classify honeypots by the level of interaction. According to this criteria we can find:

- Low Interaction honeypot: The fictious system only emulates a part of (vulnerable) applications or operating system. The real interaction is not possible.
- Medium Interaction honeypot: The fictious system only provide a custombuild environment with limited system access
- High Interaction honeypot: This kind of systems provide a working operative system enabling the attacker to interact with the system at the highest way.

The honeypot offers us many advantages and disadvantages, but these are not relevant in this proposal. They will be discussed during the research. There is another aspect that should be mentioned, many legal issues are related with honeypots .

- liability
- privacy
- entrapment

Related with the liability, the organization or person which deploys the honeypot will be held liable if the honeypot is used to compromise other systems. The attacker actions using the honeypot could have legal repercussions to the organization. It is true that according to the definition of honeypot and it is shown on the figure 3 more than 90% of the Operating Systems are high interaction potential honeypots. As the reader can see, there are a lot of background in the topic to discuss.

The privacy in the honeypot is other complicated theme. Using honeypots, users' privacy it is invaded. In fact, the attackers privacy is also compromised. So, it is important to define which information would be captured.

In the case of entrapment, it is only applied to police and government agencies. Individuals and Organizations use the honeypot to learn about security and to make safer their systems.

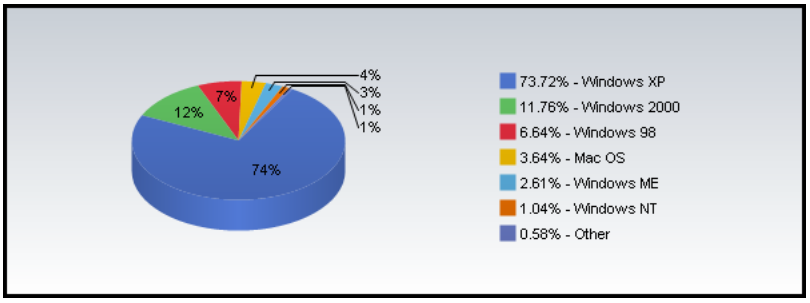


Figure 3: OS Market Share during 2005, According to Hitslink. Source: *http://marketshare.hitslink.com/*

3.3 Project Objectives

When a new system is designed and released, the programmers/designers and the users want that the resources from an organization will be used by the information system in the way that the designer specified in the program. This is in short terms what security means in the computer science field. If security is analyzed to a deeper level major features of the security bases can be seen as:

- Confidentiality

- Integrity
- Availability
- Authentication

If we could guarantee these core security basics we could state that a system is secure. So, the task in this project is ***to know how secure the SURFnet Distributed IDS system is***. According to the information shown in the SURFnet D-IDS webpage, we should prove that the main ideas of the new design proposed by SURFnet are more or less effective than in the typical Distributed IDS design.

The basic rules which is SURFnet IDS based on are:

- The sensor should run out-of-the-box.
- The sensor should be completely passive and therefor maintenance free.
- The D-IDS should not generate any false positive alerts.
- The sensor should be able to run in a standard LAN.
- Comparison of statistics generated by sensors and groups of sensors should be possible.

4 Justification

4.1 Social Relevance

There is a maxim in Security that says that to hide the information doesn't provide secureness. This shows the importance of projects such as the distributed IDS founded by SURFnet. However, to build a release a well-done open source project requires a lot of work, and often voluntary work so it is very important for these type of projects that people join and support them.

If we focus on how open source works, we can cite Linus's Law according Eric S.Raymond that states that *"given enough eyeballs, all bugs are shallow* [7] *"*. There are some criticism that could discuss this affirmation, but 21 years of work in the same way back the open source method of work. You only need a solid software engineering methodology to decide which contributions can take part in the project.

[7] This cite is from his essay *"The Cathedral and the Bazaar"*

As was said in section 3.3, testing the security of this project will provide more reliability. Whatever on the result of this thesis research a trustworthy system will be provided in the end. However, in the case of catastrophic results for the project we will demonstrate that the product approach is not relevant. This is not totally negative because in that case, we can concentrate the community efforts in other systems with different designs or rebuilding the design of that system to improve it.

But the main social aspect that concerns our project is the security aspect. According to the FBI Computer Crime Survey[8] the 87% of computers in the US territory have experimented some kind of security incident during last year. It can be seen in Figure 4 that 83.7% of the organizations questioned experienced virus problems in 2005. It is true that some of these problems could be solved using more secure operating systems. Anyway, this type of malware costs the companies 12 million dollars during 2005 as it is seen in figure 5. This is a study from United States. So, security is not a myth but a big problem. For this reason, anything that reduces the number of incidents and the total amount of money wasted is a worthy goal. The D-IDS is not a definitive weapon against the security problems, but if their projected objective becomes a reality, it could help to reduce part of these statistics.

4.2 Scientific Relevance

In real world software is being developed with some time and money restrictions. The immediate consequence of that model is that it is a software with less quality. On the other hand the software "normally" contains bugs and so. If we mix both affirmations, we obtain poorly developed software contains more bugs. Normally not all the bugs have security implications but there are some which could compromise a system. With that reflection we should take care of the bugs if we want to obtain secure software. See *Fatal Defect: Chasing Killer Computer Bugs*[9] for examples of this kind.

[8]The 2005 FBI Computer Crime Survey addresses one of the highest priorities in the Federal Bureau of Investigation. These survey results are based on the responses of 2066 organizations. The purpose of this survey is to gain an accurate understanding of what computer security incidents are being experienced by the full spectrum of sizes and types of organizations within the United States.

[9]The book comments some tragedies caused by bugs. The bugs of the book are not caused by security exploited bugs but could be a good illustrative example about bug consequences.

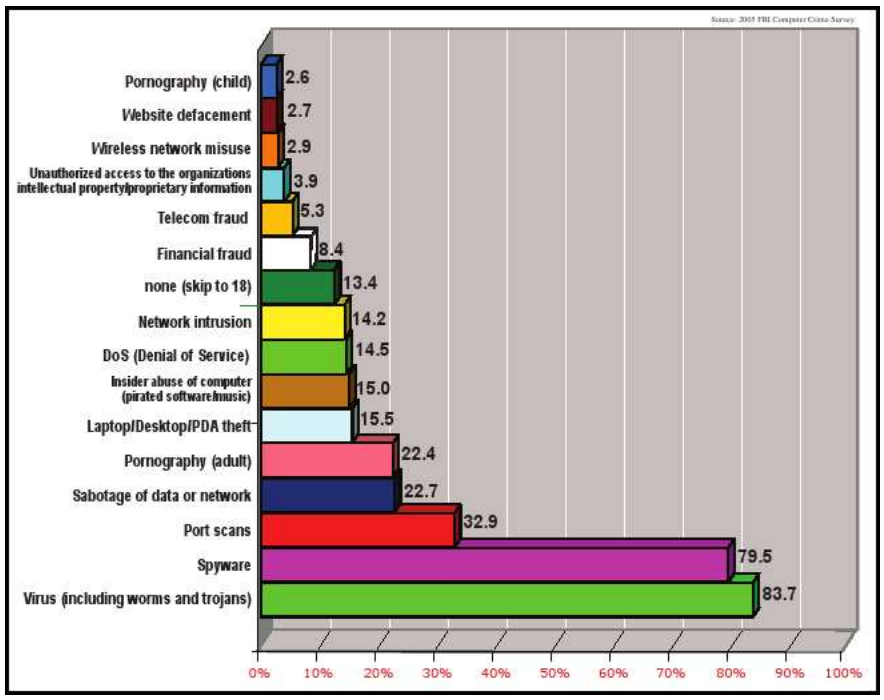


Figure 4: Security Incidents during 2005 in United States. Source: *FBI*

In most applications exist some types of bugs that have security implications. Using the exploitable bugs an attacker should be able to compromise the availability, authentication, integrity or confidentiality of the system. Most security engineers considers only three. These engineers consider that authentication is part of confidentiality. These four characteristics represent the secureness of the system. The suppression of one of these security principles will compromise the system.

To exploit these four security basics can be used different types of techniques such as *Buffer Overflow*, *SQL Injection*, and so on. This is what attackers do to compromise the secureness of a system. Other ways of attack are scripts created by an attacker with the concrete exploit knowledge, however these types of attacks are not relevant because one of the aims of the master thesis is learn how to perform this kind of attacks.

This explained technique could be used for attack, but you also can use them to defend yourself against the attackers. If you know how your attackers compromise your system you should find solutions to fix this bugs and avoid

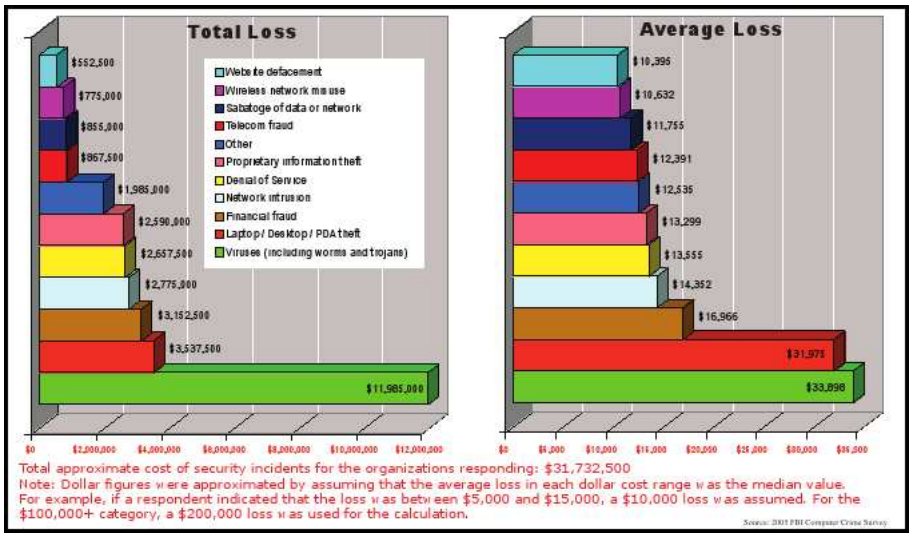


Figure 5: Estimated loses due to security incidents during 2005 in United States. Source: *FBI*

this risk.

On the other hand, a honeypot is part of IDS and the honeypot is a new approach to tracking hackers developed in the last 10 years. There are a lot of issues related with honeypots. We intend to answer some of them, but mainly implications about the SURFnet IDS's honeypot approach.

With our thesis we should find the answer to these questions:

- *How does the SURFnet D-IDS work?*
- *What kind of security aspects are relevant for this system?*
- *Which security aspects are concerned with each part of the system?*
- *Which risks can we identify?*
- *Are the identified risks handled properly?*
- *What about honeypots?*
- *Which implications have the inclusion of a honeypot in the system?*
- *What hacking techniques exists?*

- *Which techniques are suitable for our system?*
- *How can I compromise the security of the system?*

5 Thesis Development

5.1 Process

In section 3.3 we have shown the main aim of the project. However, it is not easy to reach this. To develop a serious security evaluation of a program, you must follow a very strict guideline to obtain useful and practical results.

Unlike in other areas of Computer Science where we have well known methodology, in this security area we don't have a methodology yet. For this reason we will need to come up with one.

In section 3.3 also are provided the four basic aspects that define security. It is important that the D-IDS respects this basic principles. Otherwise, the evaluation will conclude that the IDS is not secure.

The primary task of this project consist of understanding how the distributed IDS works. When this will be understood, the following task to be performed will be to find which parts of the system are related to the four security principles. After this it will be clear how the system works and where the possible security risks are. It is needed for this project that a research of general basic hacking techniques will be performed. Once this is done the testing of the system will be started. According to the results which will be obtained in this test we evaluate the system with respect to a list of criteria we will define.

5.2 Project Phases

The project has been broken down into seven subsequent phases. In the following section each phase will be explained in more detail.

The table 2 resumes the project phases and the main tasks involved in each project phase. The table 3 provide each phase deliverables.

The current project will be performed during eighteen weeks. The following section explains each phase in a more deeper approach.

Project Phase	Tasks
1. Orientation	Master Thesis Proposal.
2. Preliminary Research	Research about the system and security in general.
3. Security Risk List	Identify which security systems are crucial to this IDS
4. Research Vulnerabilities	Research about the general hacking techniques.
5. System Testing	Intend to compromise the security of the system.
6. Evaluation	Evaluation of the system based on a definite criteria.
7. Project Finalization	Finalization of the thesis and the presentations.

Table 2: Project phases which is divided the project and the tasks and deliverables which will be performed at the end of each phase

Project Phase	Deliverable
Orientation	Proposal
Preliminary Research	Introduction and Background sections
Security Risk list	List of Security Risk crucial for this IDS
Research Vulnerabilities	General Hacking Techniques section
System Testing	Attack Results section
Evaluation	Criteria Establishment and Evaluation Section
Project Finalization	Master Thesis, Presentation

Table 3: Deliverables

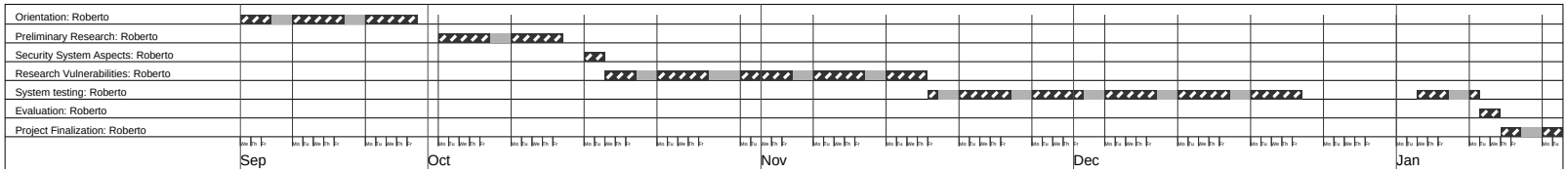


Figure 6: Master thesis Gantt diagram

5.3 Project Schedule

During this section the figure 6 will provide more information about the project planning.

5.3.1 Orientation

This is the initial phase. During this phase, the present document will be written. In the following phases, this document will be used as a general guideline.

5.3.2 Preliminary Research

During this phase, the literature about general security and system characteristics will be studied. The objective of this phase is to obtain the background necessary to understand how the system works and which security basics are involved with the system.

In this phase each element involved in the system will be studied separately at first. Once this is done, the integration of the elements will be studied.

This phase also includes the specific research about honeypot.

At the end of this phase, introduction and background sections of master thesis will be written.

5.3.3 Security Risk list

This is one of the most important phases in the project. The aim of this phase is to provide a concrete list of security risk crucial to this IDS. To perform this research the elements of the system will be separated. Then, the security basics shown in figure 3.3 will be examined for each part.

The information about which security risks are crucial for our project will be discussed in this phase. And the conclusion will be used to develop the subsequent research.

At the end of this phase, a concrete list of security risk to this IDS will be written.

5.3.4 Vulnerabilities Research

This phase will focus on research of the main hacking techniques useful to intend to compromise the security of the system according to the conclusions obtained in the last phase.

This research includes also a research about vulnerabilities present in the system and to look for scripts that will be used in the system testing.

At the end of this phase, a section about general hacking techniques will be written.

5.3.5 System Testing

The aim of this phase is to compromise the security basics shown in the security risk list using the background obtained in the previous research.

During this phase, the scripts found in the research will be used to compromise the security too. However, every script used will be explained in the master thesis.

At the end of this phase, the advantages claimed in the Distributed IDS will be tested and the attack results section will be written.

5.3.6 Evaluation

In this phase, first a list of criteria to perform the evaluation will be established. And according to the background that we obtained during the realization the project a final conclusion about the project will be exposed.

5.3.7 Project Finalization

During this phase, the Master Thesis Document will be finished an then a presentation will be written to explain the results of the research. All the results of the thesis will be reported to developers to improve the D-IDS.

5.4 Hardware

To reproduce the SURFnet D-IDS two computers will be needed, one of these will act as a sensor(this computer would be able to boot from a usb stick) and the other one will act as a server. In the SURFnet diagrams the

server system is composed of two computers, one of them runs a honeypot and the other a database. However, it is possible to configure the database and the honeypot in only one computer.

For the sensor will be needed also 1 GB usb stick to install the Knoppix remastered distribution provided by the SURFnet project.

6 Contact Information

Name: Roberto Nebot Gozalbo
Company: Radboud University Nijmegen
Function: Student
E-mail: melkor.kp@gmail.com
Telefon: +31 - 0643654920
Relation: Project Owner

Name: Marco Van Eekelen
Company: Radboud University Nijmegen
Function: University Lecturer, Computer science faculty
E-mail: M.vanEekelen@cs.ru.nl
Telefon: +31 - 24 3653410
Relation: Supervisor, Security of Systems

Name: Engelbert Hubbers
Company: Radboud University Nijmegen
Function: University Lecturer, Computer science faculty
E-mail: E.Hubbers@cs.ru.nl
Telefon: +31 - 24 3652713
Relation: Supervisor, Security of Systems

Name: Kees Trippelvitz
Company: SURFnet
Function: Developer
E-mail: Kees.Trippelvitz@surfnet.nl
Relation: SURFnet IDS developer

Name: Jan van Lith
Company: SURFnet
Function: Developer
E-mail: Jan.vanLith@surfnet.nl

Relation: SURFnet IDS developer

Name: Rogier Spoor
Company: SURFnet
Function: Developer
E-mail: Rogier.Spoor@surfnet.nl
Relation: SURFnet IDS project leader

References

[1] Surfnet website (English version). "http://www.surfnet.nl/en".

[2] Surfnet D-IDS website "http://ids.surfnet.nl"

[3] Dieter Gollman *Computer Security* John Wiley and Sons, 1999.

[4] Charles P. Pfleeger, Shari Lawrence Pfleeger *Security in Computing: Third Edition* Prentice Hall PTR ISBN: 0-13-035548-8

[5] Charlie Kauffman, Radia Perlman and Mike Speciner. *Network Security: private comunications in a public world*. Prentice Hall ,2002

[6] Jack Koziol et al. *The Shellcoder's handbook: Discovering and Exploiting security holes* John Willey & Sons ISBN: 0764544683

[7] Lance Spitzner *Honeypots: Tracking Hackers* Addison Wesley ISBN: 0-321-10895-7 Know your enemy learning about Security Threats: 2n edition

[8] SURFnet Public IDS service http://publicids.surfnet.nl:8080

[9] *Hakin9: Hardcore IT Security Magazine* Software Wydawnictwo Sp. z o.o. num 17 (2006) ISSN: 1731-2930

[10] SecurityFocus. http://SecurityFocus.com/infocus/1532

[11] Gulcas. http://gulcas.org/?q=node/145

[12] Hervé Debar, Marc Dacier and Andreas Wespi. *Research Report, a revised taxonomy for Intrusion Detection Systems*. IBM. www.cc.gatech.edu/~wenke/ids-readings/IDS_taxonomy.ps

[13] Thorsten Holz, *Honeypots know your enemy*. Universität Mannheim. 2006. honeyblog.org/junkyard/2006_honeynet_cebit.pdf

[14] Lawrence R. Halme and R. Kenneth Bauer. *Intrusion Detection FAQ. AINT Misbehaving: A Taxonomy of Anti-Intrusion Techniques*. 1995. www.cs.swarthmore.edu/~kuperman/cs97/papers/halme1995AINT.pdf

[15] SURFnet publications. 2005. www.surfnet.nl/publicaties/surfworks2005/indi-2005-009-15.pdf

[16] Georg Wicherski. *Medium Interaction Honeypots*. 2006. www.pixel-house.net/midinthp.pdf

[17] Stuart Staniford-Chen, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, D. Zerkle. *GrIDS: A Graph-Based Intrusion Detection System for Large Networks*. 1996. http://citeseer.ist.psu.edu/rd/0%2C58218%2C1%2C0.25%2CDownload/http://citeseer.ist.psu.edu/cache/papers/cs/5717/http:zSzzSzic.arc.nasa.govzSzpeoplezSzfrankzSznisc.96.pdf/staniford-chen96grids.pdf

[18] Rogier Spoor, Jan van Lith and Kees Tripplevitz. *SURFnet IDS a Distributed Intrusion Detection System*. www.terena.nl/activities/tf-csirt/meeting17/surfnet-ids-spoor.pdf

[19] Krisztian Piller and Sebastian Wolfgarten. *Honeypots forensics, No stone untumed or logs, What logs?*. 21C3, December 2004, www.ccc.de/congress/2004/fahrplan/files/135-honeypot-forensics-slides.ppt

[20] Lance Spitzner. *Honeypots*. www.cansecwest.com/core02/honeypots-0.2.ppt

[21] Fabien Pouget. Institut Eurécom. 2006. www.terena.nl/activities/tf-csirt/meeting17/wombat-pouget.pdf

[22] *2005 FBI Computer Crime Survey*. www.digitalriver.com/v2.0-img/operations/naievigi/site/media/pdf/FBIccs2005.pdf

[23] Cliff Stoll. *The Cucko's Egg: Tracking a Spy Through the Maze of Computer Espionage*.Pocket Books, 2005.