Extending a Platform for IT-Security Exercises

Master's Thesis
in Information Coding

by

Johan Björn

LiTH-ISY-EX--11/4443--SE

Linköping, 2011
Extending a Platform for IT-Security Exercises

Master's Thesis

in Information Coding

by

Johan Björn

LiTH-ISY-EX--11/4443--SE

Supervisors: Jonas Hallberg and Johan Bengtsson

Swedish Defence Research Agency

Examiner: Viiveke Fåk

Linköping Institute of Technology

Linköping, 2011-04-12
### Abstract

The Swedish Defence Research Agency, FOI, has developed a platform that is used to train and study IT-security. This platform was used during the cyber Baltic shield, an international cyber security exercise. During the exercise, a number of teams acting as system administrators, tried to secure and defend the system of a fictive power supply company. Another team acted as a terrorist organisation with the goal to compromise the systems of the power supply companies and shut down their power generators.

FOI has also developed a security assessment method, named XMASS, which is implemented in a software tool called SANTA. This can be used to model a networked IT-system and get a picture of its current state of security.

This thesis aims to integrate the tool, SANTA, with the platform for cyber security exercises to get the ability to visualise a system and analyse its security during an IT-security exercise.

The thesis also identifies some problems with XMASS regarding how traffic mediators, for example firewalls, are modelled. A literature review is performed to get a picture of the current state of research on security assessment methods and leads to a proposition of a new model for traffic mediators.

### Keywords

IT-Security, IT-Security Assessment, IT-Security Exercise, Traffic Filtering, Firewalls
Abstract

The Swedish Defence Research Agency, FOI, has developed a platform that is used to train and study IT-security. This platform was used during the cyber Baltic shield, an international cyber security exercise. During the exercise, a number of teams acting as system administrators, tried to secure and defend the system of a fictive power supply company. Another team acted as a terrorist organisation with the goal to compromise the systems of the power supply companies and shut down their power generators.

FOI has also developed a security assessment method, named XMASS, which is implemented in a software tool called SANTA. This can be used to model a networked IT-system and get a picture of its current state of security.

This thesis aims to integrate the tool, SANTA, with the platform for cyber security exercises to get the ability to visualise a system and analyse its security during an IT-security exercise.

The thesis also identifies some problems with XMASS regarding how traffic mediators, for example firewalls, are modelled. A literature review is performed to get a picture of the current state of research on security assessment methods and leads to a proposition of a new model for traffic mediators.
# Contents

1 **Introduction** ........................................................................................................... 1  
  1.1 Motivation .............................................................................................................. 1  
  1.2 Problem Formulation ........................................................................................... 2  
  1.3 Contributions ........................................................................................................ 2  
  1.4 Disposition ............................................................................................................. 3  
  1.5 Terminology .......................................................................................................... 3  

2 **Background** ............................................................................................................. 5  
  2.1 XMASS ................................................................................................................... 5  
  2.2 NTE and SANTA ................................................................................................... 8  
  2.3 Cyber Defence Exercise ....................................................................................... 10  
  2.4 Traffic Mediators and Firewalls ......................................................................... 12  

3 **Design of the Session System** ........................................................................... 15  
  3.1 Overall Structure Analysis ................................................................................. 15  
  3.2 Token Handling ................................................................................................... 16  
  3.3 Extending SANTA ............................................................................................... 16  
  3.4 Use Cases ............................................................................................................. 17  

4 **Implementation of the Session System** ......................................................... 21  
  4.1 Extending SANTA ............................................................................................... 21  
  4.2 Keeping Plugin Separation .................................................................................. 21  
  4.3 Storage ................................................................................................................ 22  
  4.4 Communication ................................................................................................... 23  
  4.5 Record and Replay of Sessions ......................................................................... 23  
  4.6 Team Visualization ............................................................................................. 24  

5 **Modelling Traffic Mediators** ........................................................................... 27  
  5.1 Literature Review ................................................................................................. 27  
  5.2 Analysis of the Current Model .......................................................................... 28  
  5.3 Requirements Analysis ....................................................................................... 29  
  5.4 Make Them Comparable ................................................................................... 29  
  5.5 Multiple Traffic Mediators ............................................................................... 30  
  5.6 Summary of the New Model .............................................................................. 32  

6 **Conclusions** .......................................................................................................... 33  
  6.1 Discussion ............................................................................................................. 33  
  6.2 Results ................................................................................................................ 33  
  6.3 Future Work ......................................................................................................... 34  

References ..................................................................................................................... 35
Figures

Figure 1: Overview of the profiles and their relations (J. Hallberg et al. 2006) .......... 7
Figure 3: NTE with the XMASS plugin ........................................................................ 10
Figure 4: Overview of the setup during the BCS exercise (Hammervik et al. 2010) .. 11
Figure 6: Use cases overview .................................................................................. 18
Figure 7: Extended NTEFile structure .................................................................. 23
Figure 8: The session controls view ........................................................................ 24
Figure 9: The session team view ............................................................................. 25
Figure 10: Traffic mediators connected in series .................................................... 31
Figure 11: Traffic mediators connected in parallel .................................................. 32
1 Introduction

No one probably knows if or when a “cyber war” could be reality or how much damage it could cause. What we do know is that the risks should certainly not be ignored.

Instead we should be aware of the risks and continually look for ways to improve security of our IT-systems. It is also important to have the knowledge of how to defend a system against an on-going cyber attack.

1.1 Motivation

The Swedish Defence Research Agency, FOI, is one of the major participants in international cyber defence exercises (CDX). These exercises aim to simulate the attack and defence of computer networks. The last exercise, the Baltic Cyber Shield held in 2010 had the specific scenario to defend the computer networks of a power supply company from cyber attacks. The importance of such exercises increase by the rate which critical infrastructure get connected to the internet.

During exercises like these a lot of information has to be processed by the participants as well as the management teams. If some of this information could be visualised and presented in a structured way it could make it easier to comprehend.

One such kind of information is the security level of the networks of the defending teams. FOI has developed a security assessment method, called XMASS which can be used to provide and overview of the security state during exercises. This is however not yet a part of the system used in the exercises.

There are two scenarios where a better visualisation of the team networks and the assessed security values would be helpful. The first is during exercises to help the white team (see 2.3) to see the current state of security of the blue teams and to see trends in security value variations. If an event causes the security value to vary
significantly up or down, with a good visualisation it could be detected faster in order to find the cause of the event.

The other scenario is when the exercise is finished and the collected data is afterwards studied to analyse the performance of both the blue and the red team.

1.2 Problem Formulation

The first part of the problem formulation is centred on the security assessment tool; SANTA. A new architecture should be created where SANTA is an integrated part of the CDX platform and not just a stand-alone program. The tool should also be extended with the following functionality:

- Display the status for the blue teams during an exercise,
- update the status of the teams during exercises by handling events, so called “tokens” and
- record those events to be able to do a replay of the exercise

The second part of the work aims to improve the model for network traffic filtering in the security assessment method; XMASS. A literature study should be performed to:

- Give a picture of the current state of research considering modelling of traffic filters and
- lead to a proposition for a new model for traffic filtering.

1.3 Contributions

The results presented in this thesis contribute to the field of operational IT security with the following:

- An architecture integrating SANTA with the CDX platform.
- A tool supporting the participants (blue teams) and the management (white team) team with visual information on how the exercise is progressing.
- The possibility to record the events during an exercise to be able to replay it later in order to study the course of events and the performance of the blue teams.
- A new model for traffic filtering in the security assessment method XMASS.
1.4 Disposition
Chapter 2 contains background information relevant to this thesis. The security assessment method XMASS is described as well as the implementation of the method in the form of SANTA. The concept of cyber security exercises are also explained as well as techniques for modern firewalls and intrusion detection and prevention systems.

In chapter 3, the design of the extensions made to SANTA is described and motivations are given to the various decisions made during the design phase.

Chapter 4 describes the implementation of the session system which handles communication with the cyber security exercise systems. Problems during the implementation and their solutions are also described here.

Chapter 5 explains the problems with the current way traffic mediators are modelled in XMASS and continues with a proposal of a new model.

Chapter 6 concludes the report and suggests some future improvements.

1.5 Terminology

NTE
The New Tool Environment, a framework for development of security assessment method plugins.

XMASS
eXtended Method for Assessment of System Security, a security assessment method.

SANTA
NTE together with a plugin implementing XMASS.

CDX
Abbreviation for Cyber Defence Exercise.
2 Background

This chapter serves as a background to the work presented in this report. The security assessment method, XMASS, is described along with NTE, a framework for security assessment methods which implements XMASS as a plugin. This tool is to be used during the IT-security exercises, which are given brief insight into. Finally, various filtering and intrusion detection and preventions technologies are described as a background to the new model of traffic filtering in XMASS.

2.1 XMASS

The eXtended Method for Assessment of System Security or XMASS is, as the name suggests, a method for assessment of security in information-technology systems. XMASS was introduced by researchers at FOI and are described in (J. Hallberg et al. 2006). The method was later refined and the updated method is described in (J. Hallberg et al. 2007).

The method consists of five steps as stated by Hallberg et al (2006), these are:

- Systems modelling,
- Calculation of entity security profiles,
- Calculation of traffic mediator filter profiles,
- Modelling of inter-entity relations, and
- Security values computation.

System Modelling

In the modelling language of XMASS the basic building blocks are the entities. The entities are divided into two groups; traffic generators and traffic mediators. A traffic generator represents devices such as workstations, servers or networks, for example the internet. The traffic mediators on the other hand are not producing traffic
themselves but rather forwarding it between the traffic generators. They represent for example routers, switches and firewalls.

**Security Profiles**

XMASS builds on the assumption that the security of an entity can be described by a set of security features. The security features is composed by a number of security values. These values are referred to as the security profile of the entity.

One of the security features can for example correspond to how well the entity handles access control. In that case, the security values which the security feature is composed of would each represent one aspect of the quality of the entity’s access control. The security values are in the range \([0,1]\) where 0 means that the entity does not address the corresponding quality and 1 that it is perfectly handled.

The security profile of an entity describes only how the security of that entity is in isolation, i.e. when no relations or other devices are taken into consideration. This profile is then used together with the profiles of the connected entities and traffic mediators in order to calculate the system dependant security profile as seen in Figure 1.
Relations
An important part of the XMASS model is the modelling of relations between entities. The flow of information and network topology affects the security in a networked information technology system. There are also other aspects such as traffic filtering, intrusion-detection and prevention.

The effects of connections between entities are modelled in XMASS in a series of steps. There are two types of relations in XMASS; physical relations and logical relations. Physical relations correspond to wired or wireless communication. This relation is bidirectional and symmetric. Logical relations on the other hand are unidirectional and
asymmetric. They are used to model communication and dependencies between the realizations of the security features of entities.

The first step to determine what effect neighbouring entities have on an entity is to calculate the neighbour dependant security profile (NSP). This profile represents the effect a physical relation would have between two entities.

**Traffic Filtering**

In larger networks, entities are rarely directly connected to each other but rather through some router and/or firewall. The effect that such devices have on the network traffic is modelled in XMASS with filter profiles (FP).

Each traffic mediator has a filter profile which describes the filtering capabilities of the device. However, the effective filtering abilities are also dependent on the security profile of the traffic mediator since it could be attacked directly and have its filtering capabilities turned off. Thus, both the filtering profile and the security profile are used to calculate the effective filter profile (EFP).

**Multiple Filtering Devices**

The resulting profile for a traffic generator connected through a traffic mediator is calculated using the effective filtering profile of the traffic mediator and the security profile of the traffic mediator. This profile is called the equivalent security profile (ESP). The process can be repeated for any number of traffic mediators connected in series.

**Multiple Paths**

Sometimes there is more than one way of communication between two entities. One way to model such scenarios was proposed by Jonas Hallberg et al. and is described in (J. Hallberg et al. 2006). This method was later refined and the new method can be found in (J. Hallberg et al. 2007). With the refined method, in a case of several possible paths of communication the resulting security profile (RSP) is calculated as the element-wise minimum of the equivalent security profiles of the paths.

**System Dependent Security Profile**

The final step of profile calculations is the system dependent security profiles (SSP). These represent the security values of the entities with influence from their context. To calculate the system dependent security profiles the security profile of the entity is used along with all neighbour dependent security profiles and logical security profiles. The result is calculated element wise using an arbitrary function, for example Min, Max or Average.

2.2 NTE and SANTA

The New Tool Environment, NTE, was developed by Johan Bengtsson and Peter Brink using C# and .NET. Their work is described in (Bengtsson & Brinck 2008). NTE was created to help developers to implement different security assessment methods and provide them with enough functionality to let them focus on their specific assessment method instead of time consuming tasks as storage and version management.
Overall Structure
NTE is designed to be flexible enough to give the developer freedom to implement assessment methods they way he or she likes. As seen in Figure 2 NTE is a framework with plugins that it interacts with through interfaces. There are two kinds of plugins in NTE; they are tool plugins and database plugins. A tool plugin is an implementation of a security assessment method and a database plugin is an implementation, or rather encapsulation of a database.

Interfaces
To be able to interact with plugins in an independent way, NTE defines a set of interfaces which plugins must implement. For tool plugins the NTEToolPlugin interface is defined, it contains methods for project management such as create, open and export.

The database plugin has two interfaces, one that is used by the tool plugin to give it access to store and fetch functions. The other interface is used by NTE and is a little closer to the database to enable NTE to store and fetch systems to and from the database.

Projects and Systems
When working with NTE the user first has to create a project. A project consists of a tool plugin, a database plugin and a requirement collection. All of these components have to be present in order to create a project. When a project is created the user also has to create a system. It is up to the tool plugin to decide what a system is and what it contains. In the XMASS tool plugin, a system consists of the modelled network with entities and their relations and profiles. An XMASS system also holds the security profile templates, filtering profile templates and the physical relation profile.
XMASS Implementation

Along with NTE, an implementation of XMASS was also created as a plugin in NTE. NTE along with the XMASS plugin is referenced to as SANTA and this is the plugin used throughout this thesis.

A database plugin was also created along with NTE and is used as the default database plugin for XMASS projects. It encapsulates a SQLite database which is transparent to NTE and the tool plugin. An example of what SANTA looks like can be seen in Figure 3.

The NTE File

NTE defines a file structure to encapsulate the project database file and reference each project with its tool plugin and database plugin. Since NTE has no knowledge about the structure of the database it cannot access any of the information stored in it. This is why the plugin versions have to be stored in a separate file while all the data belonging to the system is stored in the database file.

2.3 Cyber Defence Exercise

The Baltic Cyber Shield (BCS) described in (Hammervik et al. 2010) and (Geers 2010) was an international cyber defence exercise (CDX) aiming to investigate how IT attacks and defence of critical infrastructure can be studied. The exercise was a
collaboration between the Cooperative Cyber Defence Center of Excellence (CCDCoE) and the Swedish Defence Research Agency (FOI).

Six blue teams participated in the exercise which had the role as system administrators on a fictive power generator company. Their task was to secure their network and actively fight intrusion attempts. A red team acted as a cyber terrorist organisation trying to compromise the computers of the blue teams and shut down the power generators. The exercise also included a white and a green team. The green team was responsible for the design and hosting of the infrastructure of the exercise as well as data collection, while the white team was managing the exercise, handling rules, scenario and scoring system. The setup of the exercise can be seen in Figure 4.

Each of the blue teams where provided a pre-built computer network consisting of 28 virtual machines running on 20 physical PC servers.

At the end of the exercise one of the blue teams was declared the winner by the white team. The team which had been the most successful on protecting their system was chosen as the winner.

![Figure 4: Overview of the setup during the BCS exercise (Hammervik et al. 2010)](image)

**F-REX**

FOI which were responsible for the green team used a tool called F-REX to reconstruct the data collected during the exercise. The purpose is to later be able to analyze the exercise closely.
2.4 Traffic Mediators and Firewalls

Firewall technology is maturing quickly since its first appearance around 1985 (Evolution of the Firewall Industry). Today there are several “generations” of firewall architectures. Traffic mediating devices have gone from simple routers to advanced traffic filters and intrusion detection and prevention systems.

2.4.1 Filtering Techniques

In (Scarfone & Hoffman 2009) are some of the commonly used filtering techniques described. They are:

- Packet filtering,
- Stateful inspection and
- Application Firewalls

Packet Filtering

Packet filtering is the most basic form of network traffic filtering. It consists of a set of rules which allows or denies packages based on information contained in the package headers. This information includes:

- The packet’s source IP address
- The packet’s destination IP address
- The protocol being used, for example TCP, UDP or ICMP
- Possibly source and destination ports for certain protocols

A device which only implements packet filtering does not have any knowledge about the state of sessions or connections. It can only decide on each package in isolation.

Stateful Inspection

Stateful inspection is similar to packet filtering but includes tracking of the state of connections. A table is used to keep track of the state of each connection. When a package is received the firewall first checks against the state table to determine if the packet’s state matches its expected state. Protocol specific checks can also be made, for example could a firewall with Stateful inspection keep track of TCP package numbers and drop packages that are out of sequence.

Application Firewalls

An application firewall has the ability to inspect the content of packages in addition to the packages headers. This is sometimes referred to as deep packet inspection. The benefit with this technique is that the data flow can be analyzed at the application level which allows for blockage or restriction of certain applications. For example, could files attached to emails be analyzed and blocked if certain criteria are met.
Another useful feature is input validation, such as maximum length of arguments. For example a username that is 1000 characters long is suspicious in most systems.

2.4.2 Intrusion Detection and Prevention

In addition to the earlier described filtering techniques, there is also intrusion detection and prevention (IDPS) technologies that do not filter traffic directly but rather analyze packages and traffic flow patterns to detect incidents. There are several detection methodologies and they are usually combined for a broader and more accurate detection. The primary methodologies described in (Scarfone & Mell 2007) are signature-based detection, anomaly-based detection and stateful protocol analysis. The Stateful protocol analysis as described in (Scarfone & Mell 2007) is basically the same as “Application Firewalls” described in 2.4.1.

Signature-Based Detection

Signature-based detection uses patterns or signatures to identify known threats. One example of a signature is an email with a specific title with an attached file where the name ends with “.exe”. A signature-based detection system could easily block all such mails. One drawback with this technique is that the signatures must be updated frequently in order to detect new threats.

Anomaly-Based Detection

Anomaly-based detection is based on profiles that represent the normal behavior of the system. The profiles can include the behavior of hosts, network connections and applications. The profiles are created by monitoring the system and its characteristics in normal use, sometimes referred to as a training period. Once the profiles are in place, the IDPS use it to find anomalies in the various parts of the system. If, for example, Web activity consumes an average of 13% of the total network capacity during work hours and suddenly that traffic increase significantly, an alarm could be sent to the administrator.
3 Design of the Session System

This chapter describes the design of the session management system which SANTA is to be extended with. This extended functionality is referred to as the session system. The overall structure is first described including communication and token handling. Lower level code structure is then analysed with respect to the requirements.

3.1 Overall Structure Analysis

During a live CDX there is lots of information to handle for the all the teams and it can be hard to get an overview of the current state of the game. This might, for the same reason, be a problem for the white team when they are to set the scores for the blue teams.

The already present functionality in SANTA would be useful to visualize the state of the blue teams during an exercise. The information produced by SANTA could be used to see trends in security variations and to compare the teams to each other with respect to their current state of security.

Today there are no connection between SANTA and the system used for recording and scoring during a CDX. To be able to use SANTA in the exercises, events and information must be transmitted to, and received by, SANTA. Since F-REX is already collecting all of the important information during an exercise, to be able to reconstruct it later, the easiest way to obtain this information would be if SANTA could communicate with F-REX.

Integration With F-REX

One idea that came up during the design phase of the session system was to integrate SANTA in F-REX as a plugin. This would be natural since F-REX is already the central part of recording and replaying sessions. A problem, however, is that F-REX uses its own way to store data that is not compatible with the way NTE is storing...
data. This makes the integration hard since it would require a major rewrite of the storage handling in SANTA.

Instead, it was decided that SANTA should remain a separate program and communicate with F-REX using a web service plugin in F-REX. The advantage with a web service is that it has a standardised protocol for communication. This allows for development of a test version of the web service outside of F-REX that can later on be integrated with F-REX as a plugin. A lot of the code can also be automatically generated by Visual Studio when a web service is created in .NET. This helps to speed up the development of the session system. Figure 5 shows an overview of the communication.

3.2 Token Handling

The events that will be sent to SANTA will be in the form of tokens. These tokens represent actions made by either the blue team (a security enhancement) or by the red team (a security breach). For example, if one of the blue teams installs antivirus software on a computer, this will be reported and sent as a token to SANTA. The token contains a set of values corresponding to the changes that the token will have on the security values of the entity it is applied to. These values are then used to update the model of the network to get a new overall security value reflecting the new assessed state of the system.

Some support is already implemented in SANTA to a level where tokens can be loaded from a file and manually attached to an entity belonging to a network setup. The extended version of SANTA should be able to automatically handle those tokens. This requires that the tokens are extended with information such as what team it should target as well as the affected entity.

**Token Communication**

A file format for the tokens should be decided on, it should be easy to handle and also easy to implement in the current version of SANTA. Since some methods for handling tokens in XML format were already implemented, it was decided that XML will be used for the communication in the session system as well. XML is also a good choice because of its simple structure and extendibility.

3.3 Extending SANTA

Since SANTA is already an existing piece of software, care must be taken to not break the current structure. The development of the extension will probably also go smoother if the established practices are followed.
Separation of Plugin and NTE Framework

One of the requirements when extending SANTA with a session management system was to keep the separation of the plugin structure and the NTE framework as much as possible. It should, on the other hand, be possible for new plugins to also implement the session system if that is desired. In summary, to use or not use the session system should be something that is optional and not forced by the code structure.

Storage

NTE provides an interface for storage in an encapsulated SQLite database. This database should, if possible, be used to store relevant session data. To do so will probably save time and effort in contrast to developing a new storage solution for the sessions.

Session Visualisation

The interface should be extended to display several things. First there should be a visualisation of each of the teams that are active in the session. This should contain a view of the network associated with each team, similar to how the network is currently visualised in XMASS (see Figure 3). The interface should also display the security value of the system for each of the teams. This display should be updated in real time as the session runs. During the recording of a session the incoming tokens should be listed in a log to get a clear picture of what is happening.

Session Management

An interface to manage the recording and replay of the session should also be present. This should include the controls necessary to start a new session or continue an interrupted session. There should also be controls for replay a saved session with options for how fast the replay should run and ability to jump to a specific point in time. The requirements on the interface are further captured in the use cases in section 3.4.

The session system should have functionality for creating new sessions, save sessions and delete earlier sessions. The use cases in section 3.4 describe the requirements in this field in further detail.

3.4 Use Cases

To better capture the requirements of the interface a number of use cases were created during the design phase. Figure 6 show an overview of the included uses cases.
3.4.1 Create New Session

**Precondition**

NTE is started and a system is loaded using the XMASS plugin.

**Basic Flow**

1. The user chooses “New” from the session menu. The program asks if the user wants to save the current system to make sure that no changes will be lost.

2. A dialog box is opened where the user chooses a name for the session and selects the number of teams that will participate.

3. A new dialog opens and for each team the user specifies a name and selects a network configuration from a list of available configurations.

4. The session is created and the interface changes, a new window is opened for each team displaying the network for that team and the overall security value of the system, similar to the current display in XMASS. A control panel for the session is added to the main window containing a log displaying event during the session. The control panel also contains controls to start and replay the session.
Alternative Flow
In step 4, instead of displaying each team in a new window the teams are displayed in tabs in the main window.

3.4.2 Save Session

Precondition
A session is created or loaded.

Basic Flow
1. The user chooses “Save” from the session menu.
2. The session is now saved

Alternative Flow
Instead of step 1, any time when a session is about to be closed, the system asks if the user first wants to save the current session.

3.4.3 Load Session

Precondition
NTE is started and a system loaded using the XMASS plugin.

Basic Flow
1. The User selects “Load Session” from the menu.
2. A dialog box is opened displaying available sessions for this system. The name and number of teams is displayed for each session and also the time when the session was created and when it was last recorded. The user selects one of the sessions from the list and clicks on “Load”.
3. The session is loaded and the interface changes, a new window is opened for each team displaying the network for that team and the overall security value of the system, similar to the current display in XMASS. A control panel for the session is added to the main window containing a log displaying event during the session. The control panel also contains controls to start and replay the session.

3.4.4 Record Session

Precondition
A session is created or loaded.
**Basic Flow**

1. The session control panel is displayed where the user enters the connection settings for the web service which provides input during the session. The user then clicks on “Connect”.

2. If the server is available a connection is now established, this is indicated by a status label.

3. The user clicks on “Record” and the program is now listening for incoming events (tokens) from F-REX.

4. Whenever a token is received from F-REX, the token will be added to the token list in the main window. The status of the team affected by the token is also updated to reflect the new state.

**Alternative Flow**
In step 2, if the server is unavailable for some reason, an error will be displayed instead.

---

3.4.5 Replay Session

**Precondition**
A session is loaded, there are some events recorded on the session.

**Basic Flow**

1. The session control panel is displayed where the user selects if the session should be automatically controlled by an F-REX or run manually.

**Flow Automatic Replay**

1. The user enters the connection settings for the web service which controls the replay of the session, the user then clicks “Automatic Replay”.

2. The replay is now controlled by F-REX and the state of the session will be updated when new timestamps are received.

**Flow Manual Replay**

1. The user chooses replay speed and clicks on “Manual Replay”

2. The session will now be replayed at the selected speed. The user can also choose to step to the previous or next event. A timeline with a slide bar is used to select a specific point in time.
4 Implementation of the Session System

This chapter describes the implementation of the session system. Problems that came up during the work are also explained along with solutions to those problems.

4.1 Extending SANTA

The design of NTE works well when new plugins are developed. However, when extending the functionality outside the scope of the plugin structure, it gets a bit tougher. A few problems of this nature arose during the implementation of the session system.

One example of such a problem is that the part which calculates the security profiles and values in the XMASS model assumes that there is always only one system active at a time. This became a problem when the multiple system views for the blue teams should be implemented, since there now had to be several systems active in parallel. To handle this problem, a refactoring of the original code calculating the security values had to be done.

4.2 Keeping Plugin Separation

In SANTA, the plugins are loaded through an interface called NTEToolPlugin, which contains functions for initializing and controlling the plugin. This interface could have been modified to also initialize and control the session system. However, that would force all plugins to implement methods for the session system leading to incompatibility with already existing plugins.

Instead a new interface called SessionInterface was created. This interface should be implemented along with the NTEToolPlugin for all plugins which use the session system. Every time a plugin is loaded NTE will now check if the SessionInterface is implemented by the plugin and, if so, initialize the session part also.
For all plugins not implementing the `SessionInterface`, as the case is with all older plugins, the session part will just be skipped.

### 4.3 Storage

A session is in many ways a more complex and larger data structure than a system in SANTA. A session contains several teams where each team has a system attached to it. This system needs to be copied and stored along with the session since the system could otherwise be modified and the replay of the session would no longer be correct.

A system is built up of many parts and relations between them which must be copied along with the session. Unfortunately this revealed a performance problem in the database implementation in NTE. The issue was already mentioned in (Bengtsson & Brinck 2008) as a problem with the original implementation. At that time the amount of data handled by the database was small and the issue was not seen as a serious problem. However, the session systems handles several systems at the same time and it turned out to cause a major slow down on the program. To store three or four systems could take as much as 30-60 seconds on a normal computer. Waiting times of this length could easily be a source of annoyance for the user of the program.

The problem was identified to be caused by the way that the database handles relations. When the data is saved, all affected relations are first removed and then the new relations are inserted again. This behaviour together with the large amount of data leads to slow save operations.

Because of the nature of the problem, a major refactoring of the database implementation would probably be required to solve the issue. This was however not possible due to the limited time available. Instead an alternative way to store the session data was chosen.

NTE uses a file structure called the `NTEFile` where the database is stored in a serialized form together with some other properties. This file is then serialized using the built in serialization method in C# and written to the disk. This file was extended with a field to also include the sessions. For this to work, all structures in the session had to be serializable by implementing the `Serializable` interface. The extended structure of the `NTEFile` is shown in Figure 7. Only the extended part is shown here, the original version can be seen in (Bengtsson & Brinck 2008).
4.4 Communication

To be able to test the functionality of the session system regarding recording and remote replay of sessions a simple web service was created. This web service aims to reflect the server side of the communication as if it was implemented in F-REX. The implemented functionality of the web service is rather simple. It can send events in the form of tokens to simulate input from a real exercise. It can also send timestamps during replays to simulate automatic replay controlled by F-REX.

Since the web service by its nature cannot push data to the client, the client has to continuously ”ask” the web service for new data. If any new data is found, it will be returned to the client in the response message from the web service. This is implemented with a timer that asks the web service for new data with a certain frequency, for example once every second.

4.5 Record and Replay of Sessions

When a token is received on the client, it will first be converted from XML form to an instance of GameToken. The token will then be matched to the team and entity that it targets. When the entity is found the token is added to that entity and the corresponding system will have its security values updated with the changes from the token applied. To be able to track all tokens in a session the token is stored in the active session as well as in the targeted entity. Finally the log containing all received tokens is updated with the new token.
Figure 8 show the control panel with the log displaying the received tokens at the top. The bottom left area contains connection settings for the web service and the bottom left area holds manual replay controls.

![Figure 8: The session controls view](image)

### 4.6 Team Visualization

The view for each team was created using a modified version of a GraphPanel which is a component in NTE. Since the system is not supposed to be editable during sessions the GraphPanel had to be modified to disable that functionality. Figure 9 show the view of one of the teams in a session. The other teams and the control panel can be accessed from the tabs at the top of the window. The same layout which is used in XMASS is also used for the team views to not confuse the user.
Figure 9: The session team view
5 Modelling Traffic Mediators

This chapter first describes the literature review preceding the formulation of the new model for traffic mediators. It continues to describe the problems that exist with the way that traffic mediators are currently modelled in XMASS and proposes a new model that solves some of the problems.

5.1 Literature Review

Preceding the work with the new model for traffic mediators a literature review was performed. The purpose of the study was to get a picture of the research done on security assessment in general and modelling of traffic filtering devices in particular. It also aimed to give an understanding of how firewalls and intrusion detection and prevention systems work, and the technologies they use.

The primary sources used for this study was the IEEE Xplore Digital Library and the ACM Digital Library.

Apparently a lot of research has been done regarding security quantification. However, one interesting paper was found which studied the research done on quantified security (Verendel 2009). This paper examines 90 papers regarding quantified security between 1981 and 2008. It questions if the methods represents security in an operational setting as described in the following statement:

“…we find that a majority of the methods use assumptions that are neither conceptually obvious nor empirically well-tested in operational security. There is even counterevidence in some cases, so work using such assumptions without further validation may be based on faulty assumptions and using methods may give wrong results.” (Verendel 2009)

While this might be true also for XMASS which was one of the papers included in the said study the purpose of XMASS is not mainly to quantify operational security
but rather to be able to see security trends in a system and to compare different setups to each other.

Any research regarding modelling of what effect firewalls has on security in a system was not found, at least not in a form similar to XMASS and the model used there. The study did however give a better understanding of how firewalls and intrusion detection and prevention systems work. This understanding was necessary to be able to understand the problems with the current model of traffic filtering and to find a possible solution to those problems. Two reports describing this subject, (Scarfone & Hoffman 2009) and (Scarfone & Mell 2007) where used as the base for the proposed model.

5.2 Analysis of the Current Model

In a real world scenario if two (or more) firewalls are connected in series, the effect of the second firewall is dependant on the functionality and configuration of the two devices. The benefit of a second firewall, identical to the first, would be very low or non-existent since no traffic that is blocked in the first firewall will ever reach the second. In a scenario where the first firewall is compromised, a second firewall could add some security. But, most likely, the second firewall could easily be compromised as well since they are identical, and thus have the same security holes.

On the other hand, if the two firewalls are different, meaning they have different filtering capabilities and/or different configurations then the second firewall could increase the security of the system since the two firewalls would complement each other. The increase in security would be proportional to the “difference factor” of the two firewalls.

The current model does not consider how different mediators are, or if they are different at all. Any firewall connected after the first one will further increase the security. The effect decreases for each device due to how the calculations are made, but each device would still have some effect.

Another problem is the difficulty to compare the filtering capabilities of the traffic mediators. This is because of how the filtering functional requirements are currently modelled.

Each feature is given as a decimal in the range [0,1] representing how well that feature is fulfilled by the device. But it does not say anything about how this feature is implemented. Take for example two firewalls with packet filtering. One of them filters all incoming traffic, while the other filters all outgoing traffic. Both are given the fulfilment value 0.5. To compare the two values will not tell anything about how similar they are with respect to their filtering capability.
5.3 Requirements Analysis

To compare two filtering devices there must be a way to differentiate them from each other. The firewalls must have properties that can be compared to each other unambiguously.

In a perfect world, a model of a network would be a complete mirror image of the real network, showing all properties, flaws and strengths of the system. The model would, of course, also be possible to create without much effort. But on the other hand, in a perfect world, we would not have to model networks since no security would be needed, right?

Unfortunately, we don’t live in a perfect world so we need security, and we cannot model a network perfectly with no effort. The question is where the line should be drawn, when is the model too far from the reality, when is it too complicated to create or use?

To summarize; with the new model, it should be possible to:

- Compare traffic mediators to each other to be able to tell if and how they complement each other.
- Keep the rest of the model unchanged to not complicate the model too much and make it harder to use.

5.4 Make Them Comparable

To be able to compare traffic mediators to each other, with respect to their security enhancing features, a distinct set of properties must be defined for each device. The properties should be comparable to each other unambiguously. The properties should also correspond to the current features defined in XMASS to be able to use them without further modifications to the model.

Each filter functional requirement in the filter profile (FP) is divided into one or more blocks were each block represents a property within the requirement. This can be either a specific configuration or a level of fulfilment of a feature. Each requirement can be divided into as many blocks as desired, independent of the number of blocks used for the other requirements. The number of blocks used for each requirement and their interpretation should be specified when a new system model is created.

The contribution of each block to the total value of the requirement could be either weighted or unweighted. For example, if one block is considered to be more important than the others, it would be given a higher weight.

If no specific weights are used the weight for all blocks of a requirement are equal to one divided with the number of blocks for that requirement.

In XMASS the effectiveness of a traffic mediator is not only dependant on the filtering capabilities but also the security profile of the device. To not lose this
property, the \textit{security profile} (SP) should affect the \textit{effective filter profile} (EFP) in the new model as well. This is calculated as a weighted average of the security profile in the same way as the current model when calculating the EFP, that is:

\[ K = \sum_{i=1}^{N} w_i \cdot SP^e_i \]

Where \( SP^e \) is the security profile of traffic mediator \( e \), \( w \) is a vector of the same size as the profile, where

\[ \sum_{k=1}^{N} w_k = 1 \text{; and } N \text{ is the size of the vectors.} \]

The final value of a block depends on if the block is fulfilled or not. A block that is not fulfilled is given a value of zero. If the block is fulfilled, the value is calculated using the weight of the block and a weighted average of the security profile of the traffic mediator, that is:

\[ B_{ij} = W_{ij} \cdot K \]

Where \( B_{ij} \) is the block with index \( i \) from requirement \( j \) and \( W_{ij} \) is the weight of that block. \( K \) is the weighted average of the security profile.

\section*{5.5 Multiple Traffic Mediators}

There are two cases to consider when handling multiple devices. In the case of multiple devices connected in series the new model works quite different from the current one. The other case, where multiple paths are considered is not affected by the new model but a short description of how the new model is applied to such cases is given.

\textbf{Devices Connected in Series}

In the case where there is more than one traffic mediator with filtering capabilities connected in between two traffic generators their combined influence on the security has to be calculated. To come around the current problem where each new device increase the security regardless of the filtering capabilities and configuration of the other devices, a new way to calculate their combined effect is given. Figure 10 show one setup where multiple filtering devices are connected in series.
Instead of the earlier approach where the result was aggregated iteratively, the devices are now seen as one unit with the combined filtering capabilities of all involved devices. A common data structure in form of a matrix $X$ is used to build up the result. The matrix is of the form $M \times N$ where $M$ is the maximum number of blocks for any requirement and $N$ is the number of requirements. This matrix is first filled out by the block values from the first traffic mediator. It is then updated for each of the other considered traffic mediators by comparing each block in the matrix to the corresponding block of the traffic mediator and select the one with the higher value. Like so:

$$X_{ij} = \text{Max}(X_{ij}, B^*_{ij})$$

Where $X_{ij}$ is the value for the block at position $i,j$ in the matrix and $B^*$ is the block value at the same position for the considered traffic mediator.

The rows can then be summarized to form the EFP of the combined device which then is used in the subsequent steps of the calculations as described in XMASS.

**Devices Connected in Parallel**

This case is not changed from the current model. In the case where several parallel paths exist each of the paths are first calculated separately to get the EFP for each of the parallel paths. The results are then compared to each other just like before to get the resulting security profile (RSP).

Figure 11 show an example of two paths available between computer A and computer B. Both of the paths feature several filtering devices connected in series. In this case path A and path B would first be calculated separately and the results would be compared to select the least secure path.
5.6 Summary of the New Model

The following stages are performed to calculate the effect traffic filtering devices have on security using the new model:

- Decide the number of blocks for each *filter functional requirement* (FFR) and what they represent.
- Assign for each traffic mediator which blocks are fulfilled on that device.
- Calculate for each block its contribution value using the profile of the mediator and if used, the weight for the block. If a block is not fulfilled it is assigned the value of zero.
- With several devices in series: Obtain the maximum value for each block among all considered devices.
- Calculate the EFP values by summarizing the blocks corresponding to each requirement.
- Use the result in subsequent calculations in XMASS.
6 Conclusions

This chapter contains a discussion regarding the work presented in this thesis. The result is compared to the requirements set up in chapter one. Finally, some future improvements are suggested.

6.1 Discussion
During the work with the integration of SANTA into the CDX platform and the extensions made to SANTA a few problems with the design of the original program was found.

The first problem was the slow execution of save operations by the database. The underlying implementation was not suited for the larger amount of data being handled by the session system.

Another problem was the separation between the framework NTE and its plugins. Since the session system was not planned for when the program was first designed, care had to be taken to not break the structure. This also made the session system harder to implement.

All those problems were hard to foresee and was discovered during the implementation phase of the work. The result was that more time had to be used on this phase to find workarounds and solutions to the problems which reduced the time available to implement further improvements.

The conclusion to draw from this is that some research should probably be done on the current state of the program before any plans and development is made.

6.2 Results
The first chapter of this thesis describes the requirements on the implementation of the session system. The integration with the CDX platform is implemented in
SANTA and the only thing remaining is to integrate the web server with F-REX which must be done by someone with knowledge in F-REX development. The server side is however tested and a simulation of F-REX behaviour is confirmed to work.

Visualisation and handling of event during exercises are also successfully implemented in SANTA.

The new model for traffic filtering in XMASS allows for a more fine-grained modelling of traffic mediators and setups with several traffic filtering devices connected in series is now specifically handled. The rest of the model is unchanged and is only slightly harder to use. The extra work comes from the need to specify the blocks for the filtering requirements in the new model.

### 6.3 Future Work

One major improvement to SANTA would be to update the database functionality to better suit the new needs regarding storage. The current design has proven to be too slow to be used effectively. If this problem could be solved it would enable the session system to use the database instead of the custom storage solution which currently is implemented.

The current implementation of the session system has only one kind of view over the systems where all information is displayed. This view is suited for the white team who wants to have the full picture over all the teams. However, it could be useful to have a different kind of view intended for the blue teams where only their own network is display, possibly with some further restrictions on the information available. This view could be used by the blue teams to get feedback on their actions and see their progress in the exercise.

Regarding the new model, an evaluation of the new model could be performed too see how it behaves in comparison with the old model. Hopefully it would prove the new model to be more flexible and give the user more control over the behaviour of the model.
References

Evolution of the Firewall Industry. Available at:


