Patch Management Security

Jan Vidar Simonsen

Master's Thesis
Master of Science in Information Security
30 ECTS
Department of Computer Science and Media Technology
Gjøvik University College, 2005

The MSc programme in Information Security
is run in cooperation with the Royal Institute
of Technology (KTH) in Stockholm.

Institutt for
informatikk og medieteknikk
Høgskolen i Gjøvik
Postboks 191
2802 Gjøvik

Department of Computer Science
and Media Technology
Gjøvik University College
Box 191
N-2802 Gjøvik
Norway

Abstract

This Master thesis focuses on the security aspect when deploying patches and we propose a methodology to ensure that the patching process is performed without any loss of the implemented security features.

We can observe nearly every week on Internet sites that one of the largest software vendors is taking a lot of critique for their amount of patch releases, and for the high frequency of discovered security vulnerabilities in their products. In general terms, it is clear that the increasing amount of vulnerabilities would cause a higher exposure of the systems, and higher exposure is causing a higher possibility for an attack. This is one of the many reasons to implement a Patch Management strategy that is adjusted to the organization.

But the procedures in several public standards and Patch Management strategies are not focusing on the overall security in an acceptable manner. If the patching is carried out without maintaining the security, it may lead to vulnerable configurations and weaken the security of the systems. The fact is that the number of patches is increasing every year, and if the deployment of these patches causes situations where the security is weakened, this can easily lead to another area of exposure to attackers and malicious users.

The original question was how we could adapt a method to focus on the security in the Patch Management procedures, and how this could ensure security when deciding to deploy the needed patches.

The decision was to add a step in the Patch Management procedures. This step should focus on mapping the security features based on data gathered from prior steps in this Patch Management process. The choice of method was to perform a connectivity study of all security related features for the systems, host based or network based, and gather this information to conduct a mapping of the security features formed as topology graphs. This step is supported by our metrics for determination of host-, and network-based security features. To gather the information for this connectivity study, the need for reliable data from prior steps is essential, and to support this aspect, metrics were developed to deal with the assets listing and vulnerability scanning procedures as well.

Finally, we propose an improved Patch Management strategy, which implements our connectivity methodology.

Keywords: patch management, network security, security metrics, graph theory.

Sammendrag

Denne Master rapporten fokuserer på sikkerhetsaspektet ved installering av patcher, og vi foreslår en metodikk for å kunne sikre at denne prosessen blir utført uten tap av noen av de implementerte sikkerhetsmekanismene.

Vi kan nesten hver dag observere på internet-siter at en av de største softwareleverandørene får mye kritikk for den store mengden av patcher de utgir, og for den høye frekvensen av oppdagelse av sikkerhetsrelaterte sårbarheter i produktene deres. Generelt sett er det klart at økning av sårbarheter kan kunne føre til høyere eksponering av systemene, og høyere eksponering fører til høyere sannsynlighet for at de blir angrepet. Dett er en av de mange grunnene til å implementere en Patch Mangagement strategi som er tilpasset organisasjonen.

Men prosedyrene i flere av de offentlige standardene omkring Patch Management strategi fokuserer ikke nok på sikkerhet generelt. Hvis denne jobben blir utført uten å opprettholde sikkerheten, kan det føre til sårbare konfigurasjoner og svekke sikkerheten til systemene. Faktum er at antallet patcher øker hvert år, og hvis installeringen av disse medfører situasjoner som svekker sikkerheten kan dette føre til enda et område for utnyttelse av angripere og ondsinnede brukere.

Spørsmålet var hvordan vi kunne tilpasse en metode som fokuserer på sikkerhet i Patch Management prosedyrene, og hvordan dette kunne ivareta sikkerheten under vurderingen om å installere patchene.

Avgjørelsen var å tilføre et steg i Patch Management prosedyren. Dette steget fokuserer på å kartlegge sikkerhetsmekanismene basert på data innhentet fra tidligere steg i Patch Management prosessen. Metoden som ble valgt var å utføre en undersøkelse for konnektivitet[1] for alle sikkerhetsrelaterte mekanismer for systemene, og samle denne informasjonen for å utføre en kartlegging av disse mekanismenes konnektivitet som en topologisk graf. Dette steget er støttet av våre metrikker for fastsetting av host-, og nettverksbaserte sikkerhetsmekanismer. For å samle informasjon omkring undersøkelsen, er behovet for korrekte og pålitelige data fra de tidligere stegene essensielt, og til denne delen har vi også utviklet metrikker som omhandler registrering av utstyr og utførelse av sårbarhets scanning.

Til slutt foreslår vi en forbedret Patch Management strategi som implementerer vår konneketivitets-metodikk.

Nøkkelord: Patch Management, nettverks-sikkerhet, sikkerhetsmetrikker, graf-teori.

[1] Oversettelse av ”Connectivity”

Preface

The work with this thesis has been performed during the last semester of the Master study of Gjøvik Univesity College. I feel that this work has suffered from my restricted possibility to perform tests in a network that implements a high number of computers and appliances, and the outcome of this thesis do not face my initial expectations.

I would like to thank everybody that has contributed to this thesis, especially my supervisor professor Slobodan Petrovic and my peer student and thesis-opponent Randi Gjerde.

I would also like to remind my girlfriend Ida and our son Even that I will probably return to the real world after this work is finished. Without your tremendous support and understanding this thesis would have been an even harder struggle, thank you both!

Gjøvik, June 23, 2005

Jan Vidar Simonsen

Table of contents

List of figures

List of tables

1 Introduction

1.1 Topics covered by this thesis

This thesis covers topics related to Patch Management and network security. This work also uses security metrics and graph theory to solve the defined problems.

1.2 Problem description

On September 14, 2004, the voice communication system in the Los Angeles air traffic control system was shut down [1]. This caused at least five cases of violation of the minimum distance between airplanes; fortunately the anti-collision system on board the plains managed to avoid a disaster. This shutdown happened because of a known, not patched, software flaw that was making it necessary to reboot the whole system every 30 days to avoid communication shutdown.

To patch the systems in time is one of the biggest challenges for the administrators in the battle against vulnerabilities. The trend shows that the number of reported security related incidents is decreasing [14], and that the time frames from vulnerability are detected until it is exploited is decreasing [15], and the number of vulnerabilities is increasing. To deal with this in an efficient way, the organizations are dependent on an effective patch management solution. If the administrators for the air traffic control system had a proper and effective patch management system, the incident could probably have been avoided. This shows the importance of being able to deploy patches to critical systems, and to be able to do this in a secure manner.

There are various problems related to patch management, in addition to the time factor. When patches are deployed to the vulnerable systems, they may also require reboot or even cause the system to unexpectedly shut down. If this happens to a computer that has any security related role, like an Intrusion Detection System (IDS) for example, this may be a threat to the overall security, and the patching process may cause an even more vulnerable system. This makes it necessary to identify the relations and connectivity for the security components in the network, and form a kind of map for nodes of importance. This knowledge can then be used to form a strategy for patch management or other kind of work that can cause problems for security or production.

1.3 Research questions

The biggest challenge to this work is to develop metrics to be used in conjunction with patch management. These metrics have to be as general as possible, but they also have to result in a specific guidance that can be useful when implementing or evaluating patch management. Because of this, it is necessary to do some exploration into the standards and recommendations for patch management, and identify basic requirements for the patch management process.

To be able to get the necessary background, the following questions are to be answered:

1. Which requirements should be stated for patch management?
2. Which strategies should be considered for patch management?
3. Is it possible to develop metrics to be used with patch management?
4. Will patch management cause any consequences to an organization?

1.4 Claimed contributions

We identify strategy models for patch management based on security, related to the most common requirements, and develop metrics to be used in a patch management process.

As for the metrics, we study graph connectivity as a security feature in a computer network, and apply this as metrics to assess security of patch management. These metrics can be used as a tool when planning for patch management, or evaluating effectiveness of an organization's patch management process.

1.5 Structure of this report

The rest of this report is structured as the following list:

Chapter	Description
Chapter 2 Method	Presents the chosen methods for this thesis
Chapter 3 Related work	Presents the literature and related work for patch management
Chapter 4 Patch Management and security features	Presents theory about patch management and graph theory
Chapter 5 Security Metrics for Patch Management and security connectivity	Presents the metrics developed in this thesis
Chapter 6 Experiments and results.	Presents the experiments and the experiment results
Chapter 7 The Metrics and Patch Management	Presents this thesis proposed patch management strategy
Chapter 8 Conclusion and further work	Presents the thesis conclusion and a further work proposal

Table 1: Report structure

2 Method

2.1 Introduction

The purpose of this thesis is to develop a framework that can be used in connection to Patch Management. The results focus on security and vulnerability and relate them to Patch Management, and they are presented to support the process of planning, implementing or evaluating Patch Management. To achieve a result that complies with working environments, some of the research questions are tested in a constructed network. Other research questions are answered based on findings from publicly accepted documents in the patch management field.

2.2 Methodology

A summary of the intended steps in this thesis workflow can be visualized in the form of the following list:

1. Gain a deeper understanding of the requirements to the Patch Management process and Patch Management tools
2. Study graph theory and relate the connectivity aspect to security and Patch Management
3. Develop suggestions for a Patch Management strategy and Patch Management procedures
4. Develop metrics for Patch Management and security connectivity
5. Test and evaluate the metrics.

For the steps 1 and 2 a literature study provides the needed theory basis and capability of a proper understanding of the most central areas within the studied discipline.

For the next step, step 3, involving strategy and procedures for the Patch Management area, a case study is required. A case study is defined in Creswell [24] as a study in which the researcher performs deep exploration of an event, operation, process or one or more people. This case study should support the author's need to gain a clearer overview of the earlier work in the field, and to gain a deeper understanding of the existing problems related to Patch Management and vulnerability migration.

As for the remaining steps, 4 and 5, a literature study about metrics is needed, and the test will be carried out as an experiment.

The combination of these methods as literature studies, experimental design and case studies form the methodology described in [24] as the "Mixed research approach".

3 Related work

The research questions in this thesis cover many different subjects, and as for the chapter of related work, there are many sources to refer to. We present selected sources here, not being able to cover everything. There may be other important references that will be referred to later in the thesis. The reason for this is to limit the size of this chapter.

3.1 Requirements for patch management

To get a base for the Patch Management, some general requirements for this process should be stated. These should be requirements regarding the management aspect, tools, how to deploy patches, security aspects and so on. These requirements cover many topics, and it may be difficult to produce a complete list. But some of the most important subjects are mentioned in this section.

3.1.1 General

Patch Management security.

Kevin Dunn [6] describes several vulnerabilities in connection with the use of automatic patch management tools. The author points out how easy a malicious user can launch an attack against patch management tools that do not use any authentication feature, and make the patching tool download a corrupted patch. He clearly claims that patch management solutions must implement security features as authentication, cryptography and integrity checks to provide the sufficient security. He also states that testing is essential and recommends that this job is best performed in a closed environment, offline.

The document from NIST [5] recommends that both parties involved in the downloading process of patches should be authenticated, and that cryptographic integrity verification of downloaded patches should be performed. [5] also states that the patches should be checked for viruses.

Most of the Patch Management tools support several of these subjects. But it is important for an administrator to be aware of these features, and to use them as well.

Coverage and efficiency

The Patch Management must cover all the organization systems and applications that need to be patched. It is not enough to just cover the operating system or a few applications. If there are any remaining vulnerabilities, it may be just what an attacker need to launch an attack. This is a major challenge for the Patch Management tools, and there are several applications that can be characterized as insufficient regarding this requirement. An example is the “Microsoft Software Update Services” [7] that do not cover third party software in any way. In general, lack of coverage is clearly a shortcoming for Patch Management tools, and it may result in the need for several patch management applications to perform patching of all operating system/applications in the network. That is not a desired situation.

The identified patches should be based on the products and drivers in use. Otherwise, patches may be installed that are not necessary for the specific usage of the application. This will involve detailed knowledge of the use of all applications, as stated in [5], and may be a complicated task.

Vulnerability scanning

In [32], Bishop states that vulnerabilities arise from computer system design, implementation, maintenance and operation. The author provides the following definition of vulnerability and related terms:

> "The specific failure of the controls is called a *vulnerability* or a security *flaw*; using that failure to violate the site security policy is called *exploiting the vulnerability*. One who attempts to exploit the vulnerability is called an *attacker*"

To be able to gain and sustain the control of the environment it is necessary to develop structured procedures of how to do this. Among many, vulnerability scanning is one of the recommendations in the document from NIST [5], where they also mention the shortcomings of such a strategy. In this document the authors state, that the results should be interpreted by an expert, and that the scanners are mostly detecting *surface vulnerabilities* that they define as:

> "A surface vulnerability is a weakness as it exists in isolation—that is, without any other vulnerability. The difficulty of identifying the risk level of vulnerabilities is that they rarely exist in isolation. For example, several "low-risk" vulnerabilities could exist on a particular network that, when combined, present a high risk. A vulnerability scanner would generally not recognize the danger of the combined vulnerabilities and thus would assign a low risk to each, leaving the network administrator with a false sense of confidence in his or her security measures. A more reliable way to identify the risk of vulnerabilities in aggregate is through penetration testing."

The scanners also represent the situation at the exact moment the scan is performed, and in a dynamic environment only one scan would not represent the actual reality. This task may also cause the need for additional software and scanning tools, but for most of the organizations depending on high security, such tools are most likely in use or planned to be used. These tools can also be used to verify that the patch removed the vulnerability.

Another document form NIST, describing network security testing [21], states that a combination of both networked and host based vulnerability scanners should be used.

Reports and documenting features

Documenting the patching and changes is necessary. The documented information must include relevant parts of the patch process in order to have informational value later on.

3.1.2 Standards and Recommendations

A standard is a collection of recommendations regarding a subject or a trade, and these recommendations have been generally recognized and accepted by the

professional community in the specific field. This can be both national and international research community. The standards are not by any means an expression to guarantee the user to achieve the purposes of the standard. Implementing the standard depends a great deal on the persons doing it, and without an explicit certification from an independent organization, the implementation may be of a great uncertainty. The standards may also contain shortcomings, or their recommendations do not fit everyone's needs. In spite of this, these standards and recommendations are very helpful as sources of information and guidance, and often serve as a reference. In this chapter we explore some standards and recommendations regarding information security and especially patch management.

The Standard of Good Practice for Information Security

The document [2] from ISF, deals with patching in section CI2.3.5 and CI2.3.6. These sections cover identification, evaluating, testing and deploying security patches. Awareness of vulnerabilities in firewalls is mentioned in section NW1.3.5, and some overall instructions are given to combat this. The total amount of recommendations for patching in this document is not enough to claim "best practice".

However, the document describes "Change management", and deploying patches is included in this concept. When performing Change Management, the standard divides the objects of change into three groups:

- Critical Business Applications, section CB2.3
- Networks, section NW3.2
- Computer Installations, section CI3.3

The overall purposes of these sections are almost equal for all the groups, and it is defined like this for the network section:

> "To ensure that changes are applied correctly and do not compromise the security of the network." [2]

The recommendations and procedures are divided in the same way for all the groups, and the text is almost identical. To do a short summary, the suggestions state that the job is carried out by authorized personnel, with the approval from the system's owner. The process and changes should be documented and logged, the changes should be tested, and there should be a possibility for rollback.

ISO/IEC 17799:2000

Another standard that treats patch management is the ISO/IEC 17799 [3]. In this document, the Section 10.5.1 "Change Control Procedures", deals with the procedures according to performing changes to IT systems. In general, this chapter contains all statements from the ISF-document, and has some additional statements. These statements are to identify all software, information, database identity and hardware that can require changes. This is probably a fundamental requirement in connection with patch management. The document also states that implementing changes should not disturb the business process.

In Section 10.5.2,"Technical review of operating system changes", the operating system is focused on. The recommendations here regard review of applications and integrity procedures to discover failure in connection with changes in the operating system, operating budget and to ensure that information about the changes is provided on time. Patching or changes to other applications are not mentioned.

Information Technology Security Evaluation Criteria

The criteria of evaluation in "Information Technology Security Evaluation Criteria" (ITSEC) [4], states that known vulnerabilities should be dealt with. This should be a part of the evaluating process of the Target of Evaluation (TOE) and meet the demands for content and presentation, as identification, consequences and methods for countermeasures. In addition, it is necessary to prove that the vulnerabilities are taken into account or completely removed. The document also describes procedures to check compliance of the requirements mentioned in this section. They state, in Section 3.37, that the vulnerability analysis should be independent, well informed and documented, and that a penetration test should be performed as a proof.

Even if this standard does not target the patch management topic directly, its methodology and baseline against vulnerabilities could be useful and important in the work related to this thesis.

[2] The Standard of Good Practice for Information Security [2]

Procedures for Handling Security Patches

The document "Procedures for Handling Security Patches" [5] from the National Institute of Standards and Technology (NIST), addresses the patch management directly. Different tasks are identified for the three phases before, during and after deploying patches to the system. In addition [5] proposes a way to organize and administer this work, but as it is claimed in the introduction in Chapter 3, these procedures are targeted at medium or large sized organizations. The size of a medium sized organization is not clear in this context, but it is probably an American measure (because of the document is produced by NIST), meaning that just few Norwegian organizations are of the referred size. Based on this assumption, the proposals for how to organize patch management may not fit to Norwegian business proportions. Despite this, the procedures stated in this document are very useful for the work related to this thesis, because the document deals with many topics that the thesis includes.

The work is divided into the following agenda:

1. Create and Maintain an Organizational Hardware and Software Inventory.
2. Identify Newly Discovered Vulnerabilities and Security Patches.
3. Prioritize Patch Application.
4. Create an Organization-Specific Patch Database.
5. Conduct Generic Testing of Patches.
6. Distribute Patch and Vulnerability Information to Local Administrators.
7. Verify Patch Installation Through Network and Host Vulnerability Scanning.
8. Train System Administrators in the Use of Vulnerability Databases.
9. Perform Automatic Deployment of Patches (When Applicable).
10. Configure Automatic Update of Applications (When Applicable).

Figure 1 shows the relationships between these proposed tasks. The PVG (Patch and Vulnerability Group) is a proposed group consisting of employees, which are especially involved in this kind of activities.

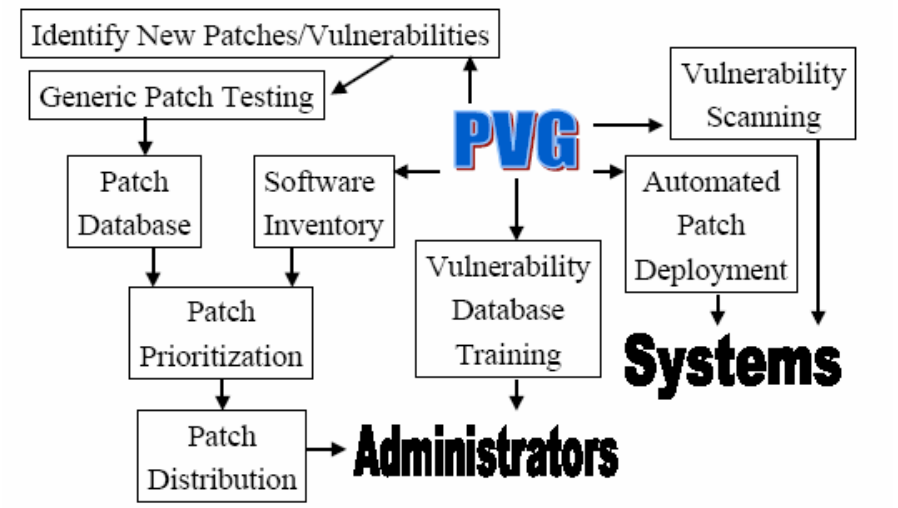


Figure 1: Relationships in Patch Management processes [5]

3.2 Strategies for patch management

[5] recommends an explicit policy for the patch management process. From this the procedures are formed, and these procedures will be the organization's action plan to defeat the vulnerability problem. This will probably involve a lot of administrative overhead in the start, and along the road these documents and procedures have to be updated for every little change in the IT systems. The document does not mention how to check compliance of the policy, other than documenting and logging.

In a document regarding patch management from SUN Microsystems [13], an overall strategy for this process is proposed. Figure 2 shows the tasks in a full patch management cycle. From the figure it is clear that there should be a central policy for all the tasks. The cycle starts with the "Discover" task.

This strategy is mostly in accordance with the standards and recommendations for patch management, and presents an easy to understand picture of the process. By this, the strategy proposal will be a good example of how to perform a patch management process. It should be mentioned that this is an approach designed for, and by, SUN Microsystems, and probably does not fit every organization. Also, the "Discover" task is not intended to be performed for every patch, and it may not be possible to perform rollback for all deployed patches in the "Verify & Back Out" task.

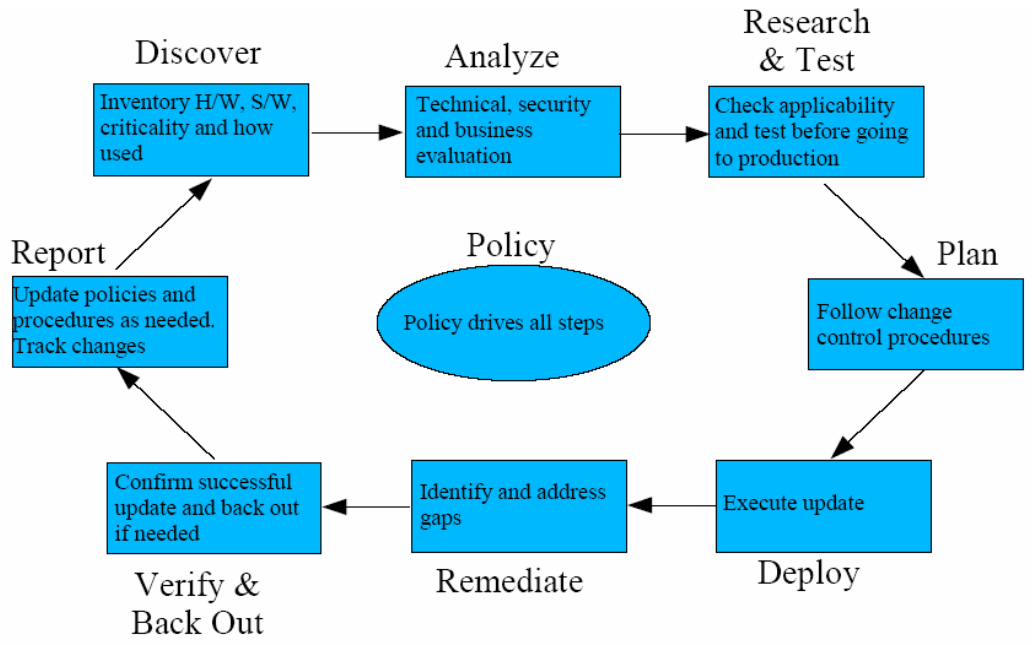


Figure 2: SUN Microsystems strategy proposal [13]

To prevent that the deployed patches disturb the production environment, or by other means take the patched system to a not preferred state, it is necessary to test the patches before deploying them. As stated in [5], there should be a test environment that mirrors the target system as much as possible. This includes, but is not limited to,

hardware, services, configuration, network and bandwidth. To meet all of these requirements is not realistic, as this would involve having an extra set of the whole system, or at least the business critical part of the system. The consequences of not having this test lab are that the patches are not tested properly, or they are not tested at all, and in turn the patch may cause problems or failure to the systems. How the testing should be performed is a strategic concern, and it should be stated in the policy.

Monitoring the systems is also an aspect that should be a part of the strategy. This is often performed as a security feature, but it may be taken into account for patch management as well. This may be performed as simple as reporting to the employee responsible for the patch management process as well as to the security employee. The persons involved in this have to have the proper understanding of threats and vulnerabilities and related issues to be able to understand the provided reports and documents.

Client patch deployment versus central patch deployment is another strategic concern. In [2], it is stated that there is a possibility to make the clients update the client computers by themselves. It is claimed that every user has the time to perform this task, and the administrators have not. The most common way is to perform this from a central point. In this manner, the administrators are able to gain a better control of the systems. It is not clear what the best strategy for this would be. It probably depends on the organization's needs, structure and priorities.

The strategies above are regarded as proactive responses and contribute to a more aggressive approach in the struggle against vulnerabilities. Another approach is to perform the patching after a failure appears or vulnerability is exposed. Even though the proactive approach intends to make the organization avoid situations of failure or exposure, it is necessary to plan for this reactive approach as well.

3.3 Security metrics for patch management

Payne proposes in [10] several steps to guide the development of security metrics. These steps are general proposals to developing and introducing a security metrics program to an organization. Our use of the metrics will not involve implementing them in an organization, but we have to consider few general aspects as well. Thus [10] may be a good support in the initial stage of developing the metrics.

The paper [11], describes some metrics for patch management. In [11], metrics targeted to express the percentage of installed patches are presented, as well as some metrics targeted to express the number of systems that encountered some difficulties after installation. But these metrics suffer from a brief and narrow presentation, and cover just a few (although important) aspects in the patch management area. In spite of this, this work will be useful as a basis for a part of the metrics developed in this thesis.

The document "Security Metrics Guide for Information Technology Systems" from NIST [12] is also a very descriptive guide for metrics, especially security metrics. The structure of the example metrics in this document served as a basis for the metrics defined in this thesis.

As we observed in the review of the standards in Section 3.1.2, some of the documents were stressing the importance of identifying the systems that have potential for patching. There may be several methodologies to gain control of this, and there are likely advantages in using several different methods to assure the correctness of the result. In this thesis, we propose a method to perform a mapping of the network topology, or a mapping of security features in the network, by using the connectivity aspect from graph theory.

To use graph theory on network connectivity, it is necessary to elaborate some of the theory in these areas. It is useful to have an understanding of the term connectivity. Harary [8] states that connectivity can be used as a specification on how many edges or vertices have to be removed to cause the graph to be disconnected. This applies to both vertices and edges. In wired computer networks, we consider a network of computers as connected, as long as they are able to communicate through the wires between them[3]. If a link between two nodes, i.e. two computers, physically exists, but the communication between the computers is not operative, they are not considered as connected. In the graph theory this is indicated by the removal of an edge between the two vertices. When a computer performs shutdown, reboots, or by other means fails to perform its tasks or communication, it will not be considered as a member of the network anymore. This is in correspondence to the removal of a vertex in the graph theory.

For the communicating computers, the connectivity aspect of network availability is directly related to how many links or computers have to be removed before the network is disconnected. Speaking in terms of usual topology of common organizational computer networks, this is a trivial matter. Often there are a few servers, firewalls, routers or links that the whole network depends on. These are to be considered as what F. Buckley and M. Lewinter name a Cut Vertex [9], and the "deletion" of such a vertex results in a disconnected network.

However, there may be several different aspects of connectivity in a computer network. Some of the participating computers often have different roles, performing different tasks. If a computers main functionality is to serve as a network IDS, the removal of this node may cause disconnection of the IDS traffic, but it probably does not make the network disconnected in terms of availability.

This can make it advantageous to prioritize the nodes in the network, and differ the prioritizing for different objectives. As explained, the shutdown of an IDS may not have any influence on network availability, and the production may continue. But for the task of performing network monitoring, the loss of an IDS could be crucial, and may result in a disconnected security graph within the network. Other prioritizing influences are e.g. the nodes processing capacity and bandwidth. This makes the prioritizing task more complex.

[3] Another kind of communication medium may also be used, but we focus on wired networks

3.4 Possible consequences of patch management for an organization

SUN Microsystems states in a patch management guide [13] that by following an effective patch management plan, an organization may gain these benefits:

- Increased availability
- Increased performance
- Better security
- Increased stability

These proclamations are probably colored by their business intentions and products, but it is not unlikely for this to be facts. On the other side, there may be some disadvantages related to implementing and performing patch management. Some of these may be administrative overhead if there are frequent changes to the systems, or if there are many guest computers not affected by the patch management procedures. Our approach is to examine this question regarding security impacts to the computer network based on the experiments of connectivity in networks.

In [5] the authors state some of the advantages from using automatic notification tools, like the reduced amount of vulnerability reports to handle and the effect of being notified directly. They also mention some disadvantages and point out that telling a third party of organizational details can be invasive.

4 Patch Management and security features

4.1 Strategy Models for Patch Management

4.1.1 Introduction

We had a look at SUN Microsystems patch management strategy model [13] in Chapter 3. This model is not related to security in particular, but is targeted at deployment of patches on the basis of cost effectiveness and availability.

In this thesis we target the patch management process in the aspect of system security. To be able to consider the different security needs for different organizations we have divided some overall patch management requirements into three categories. These categories are described in form of the requirements for different patch management tasks. A single organization may have different security needs for different tasks, and this can cause an organization to belong to different categories for the different tasks.

4.1.2 Aspects of Patch Management strategy

Different organizations are likely to have different requirements for many aspects within the patch management process. Some of the organizations may have rigid requirements in one area and not so rigid in another area. Because of this variety it is almost impossible to make a "best practice" list; it depends highly on the organization's different needs and requirements.

From the findings in Chapter 3, we can describe several of the most important aspects for patch management and try to map these in categories. First, we present and describe these aspects.

Testing environment

Testing the patch is a crucial part to ensure that the patch does not introduce failures or make the systems unstable. This is highly recommended for connectivity and availability reasons of important business systems. But it may be performed for other reasons as well, including confirmation of vulnerability removal, ensuring integrity and other security related reasons.

The test environment should be a replica of the actual environment, involving the replication of logical factors such as workload and physical factors such as temperature and so on. This is of course a very expensive and time consuming requirement. But these costs do not justify the absence of testing, and other solutions are likely to be satisfactory for test purposes. It may be possible to mirror some of the most important computers, and this might result in a more economic and efficient solution. The "most important computers" in this context may be computers involved in crucial business procedures, like an application server, or computers involved in network security tasks, like an IDS. Another solution that may be of interest would be the use of virtual environments. By this, an organization may replicate networks, but limited to the throughput of the virtual environment hosts. This is by far a more economic solution, but on the other side, one shortcoming is the missed possibility to test on the actual hardware. These solutions are considered as "offline", meaning that

the tests are not performed in the real production environment.

There is also a possibility to perform tests in the real environment where the patches are to be installed. The computers in this "online" testing may include some of the computers that do not have an impact in the organization's crucial environment, and make up a replica in terms of hardware and OS. There are shortcomings of this approach too, and it introduces a higher risk because the testing is performed in the production environment.

Test procedures

To ensure patch compatibility in the system, the patch has to be tested after installation. This testing should be performed in a test environment as stated above. The objective of the test procedures is to create an understanding of how the patch has affected the target system, including the impact on other applications. This makes it necessary to replicate normal use of the system, including applications as well.

After the test is done, it is important to continue to monitor the systems in search for anomalies caused by the installed patch.

Security

The patch management tools usually have some features for ensuring the integrity of the patch, to authenticate the provider and support setup for antivirus scanning. These are just the basic requirements. The tools must support encrypted connections and downloads, in order to prevent a malicious user to derive information from this traffic. If the patches are to be stored, e.g. in a patch database, before they are deployed or taken into test procedures, they must be stored in a secure manner. Some of the storage related protection procedures are access control, cryptology, and integrity check before deploying them. The tools are also subject to DNS cache poisoning when checking for new patches, and this must be taken care of.

Patch rollback

If the patch causes system failure, or for other reasons we need to remove it from the system, the patch has to support rollback. Primarily, rollback consists of replacing the files the patch has changed with a copy of the original files. But there may be more to this task. The patches may be complex and perform many changes and configurations. Because of this it is not possible to perform rollback for all patches.

Assets list

In order to know which patches to deploy, a thorough list of the inventory regarding all computers and network components in the organization must exist. In addition to this list, there must be an accurate description of each computer or component and its use. The strictest organizations may also include the users groups into this, stating which user group is to do what and how.

This list should contain at least the following:

- Computers and network appliances:
- Short description
- Operating system; version number, patch level and configuration
- Hardware and firmware; product vendor, serial number, drivers and description
- Software; version number; patch level; configuration;
- Each computer or components location, both logical and physical, IP-address/IP-range
- Running services
- Thorough description; system name, network alias, role in the network/organization, patch history, allowed/forbidden connections, main users, main vendors patch release site/connections.

Vulnerability scanning

Vulnerability scanning can detect if a patch actually removes the vulnerability or not, and it is recommended to perform scanning after installation of patches. It is also recommended to perform this kind of check in a periodical manner, to test for common vulnerabilities. However, the results from these tools are highly dependent on their use and the qualifications of the persons performing these checks.

These tools are also used to make a list of all the systems connected to the organization's network, as the systems that are about to be scanned make a response to the scanning. Again, the results from the scanning tools do not necessarily make up the whole truth about the situation, and the tools may not be able to detect all the connected systems in the network.

Prioritizing

In order to install several patches at the right moment, it may be advantageous to have a prioritized list of systems or patches. This will make the administrator's job easier, and the patches will be deployed in the right system at the right time.

There are several criteria to base the patching priorities on, of which some are listed below.

- Type/Class of patch[4]
  - Critical Update
  - Security Update
  - Service Pack
  - Hotfix
  - Software Update
- Patch characteristics
  - Rollback support
  - Reboot required

[4] This is a part of the Microsoft patch definitions [17]

- Computers and network appliances
  - Servers
    - Internal
    - "Internet facing", typical in a DMZ [5]
  - Clients
    - Internal
    - Laptops
    - Remote
    - Guests
- Network appliances or other
  - Firewall
  - Router/switches
  - IDS related
- Software
  - Operating system
  - Applications
- Criticality level, based on third party reports or newsletters from sites like:
  - CERT: http://www.cert.org/nav/index_red.html
  - CVE: http://cve.mitre.org/
  - Microsoft: http://www.microsoft.com/technet/Security/default.mspx
  - Anti virus vendors or other security related partners

4.2 Different patch management categories

From the criteria in the previous section we have categorized some requirements into three levels that will represent three patch management categories. The procedures are recommendations for the category, and are not meant as an exhaustive list of "proofs" or rigid requirements. Therefore there are procedures that are likely to be added or removed from the categories, to fit the organization's need, but each category's general principle will hold.

4.2.1 Highest

This category is meant to fit the organizations that have extremely high requirements for availability of their services. This strategy does not allow any patches that can compromise the availability or security in any aspect. An organization like this often runs critical systems, may have sensitive financial information, or is government related. Examples of organizations like this are flight control systems, banks or health services.

A summary of the characteristics is given below:

General:

- A Vulnerability Group manages the Patch Management procedures
- No unnecessary patches are to be deployed
- No new features are allowed through patches
- All software is strongly configured , and the use is restricted[6]

[5] Demilitarized Zone [16]

- No guest computers or remote connections are allowed
- A review of the assets list is performed regularly

Testing:

- All patches are tested in a replicated environment
- The test environment can be a replica of critical computers or subnets, and the workload and other variables are simulated as much as possible.
- Only offline testing

Security:

- Great care is taken in order to ensure the integrity of the patches
- Several different security features are implemented for network and host security
- The test environment is highly secured and monitored

Prioritizing:

- All deployed patches must support rollback
- Only patches that match the categories Security Update and Critical Update are allowed to be installed
- Patches requiring reboot are normally not allowed
- Computers and other components are prioritized (for patching) according to availability and security

Policy and documents:

- Strong policy driven procedures
- Strong document requirements
- Documents and results from procedures are reviewed and analyzed for evidence of patterns or systematic anomalies.

4.2.2 Moderate

This category implements most of the recommendations from the literature for installing patches in a networked environment. The category relates to those organizations that have an interest in secure and reliable information/services, but also have to take economic considerations when implementing patch management. The organizations that are most likely to implement this strategy are the large/mid sized businesses that do not consider availability as business critical, and those that have implemented strong security features for their critical information assets.

General:

- One or two employees handle the Patch Management procedures
- No unnecessary patches are allowed
- All changes are documented

Testing:

- Separate test environment not required, but recommended
- Virtual test environment is one solution
- Test environment can consist of some selected online computers

[6] This may reduce the real need for patches as the restricted use and configuration limits the software

- Critical patches are tested
- All critical patches must support rollback
- Online testing allowed, off-hours
- Prioritized computers and network components

Security:

- Automatic or tool based integrity check of all patches
- Only software with security risks is configured and restricted
- The computers in the test environment are highly secured and monitored

Prioritizing:

- Patches that match the categories Security Update and Critical Update are allowed to be installed
- Patches that match the categories Service Pack and Hotfix are allowed, but require special consideration.
- Patches requiring reboot are normally not allowed to be installed on "Cut node"-computers or appliances[7]
- Computers and other components are prioritized (for patching) to availability and security

4.2.3 Lowest

This category has not the same focus on security and the formal procedures as the other two above. As we can imagine, not all organizations are willing to spend money on matters that they do not find any obvious reason for, and there is a possibility that this also holds for patching and security. These procedures represent the minimum requirements for performing a patch management plan.

General:

- Automatic deployment of non critical patches
- All relevant patches are deployed
- Minimal documenting

Testing:

- No test of the patches required
- Untested patches have to be able to perform rollback.
- One member of the staff is available in the hours after a patch install.

Security:

- No configuration of software, no direct restrictions of software use

Prioritizing:

- Patches that match the categories Security Update and Critical Update are automatically installed
- Patches that match the categories Service Pack and Hotfix are allowed, but require rollback features.
- The remaining patches, from those above, are installed if they are considered necessary

[7] From the connectivity aspect in graph theory

- Patches requiring reboot are normally allowed at designated hours
- Computers and other components are prioritized to availability

4.3 From procedures to strategy

As we could see from the previous chapter, it is possible to implement Patch management at several different levels of influence on the organization. How to perform the Patch Management process is not an easy task to determine, but these procedures should be driven by a policy. The Patch Management Policy can be developed by identifying the organizational needs for patch management, and stating these requirements in a document. This policy is a part of the organization's strategic documents, and should be approved by the organization's strategic management, and equalized with other strategic IT-documents like the Security Policy.

From the Patch Management policy the organization can develop the procedures to ensure that the Patch Management is performed in a manner that support the statements in the policy. Such procedures may look like the categories we presented in the previous chapter, and it is obvious that these three categories must have three different policies. The difference in motive and implementation of the Patch Management makes it difficult to present a generic policy in this document. We shall therefore try to identify the most common steps for a Patch Management Strategy to be used when a policy is to be established or evaluated.

The uncommon step of stating the procedures first is not the way this should be managed in a real life environment, but we do it like this in this report in order to avoid that the resulting procedures are a product of the strategy, and not the requirements identified from the reviewed literature. On the other hand, one may state that this method makes the strategy proposal a product of the categories of procedures. With this in mind, we concentrate all our effort in making this section an independent part.

1 State how the Patch Management process is to be performed, the overall role of the person(s) responsible for this process' continuity.
2 Make a statement of how the organization's assets are to be identified, what should be included in this list and how to maintain the list. If such a list already exists, state the additional entries to be added to this list.
3 State the prioritizing of the installations that are subject to patching. State which situations may change this prioritizing, and suggest an overall guide of how to perform the changes.
4 Define a list of consistent criteria for the patches that are allowed to be installed
5 Define a list of the patches that are not allowed to be installed
6 Define requirements of testing and test environment
7 Define requirements for deployment of patches
8 Define the requirements for the tools in use in the Patch Management process
9 State overall requirements for documenting and reports from each step in the Patch Management process
10 Make some overall statements of how to perform Patch Management continuity and efficiency.

These steps are to be considered as a suggestion of strategy statements, and they are defined to be as generic as possible. By this generic nature, it may be possible for different organizations, with different patching requirements, to use this strategy proposal. The implementation of this strategy may be performed with big difference, but if this strategy is followed, the strategy's overall goals hold.

4.4 Graphs and connectivity

4.4.1 Basic graph theory

The literature on networks and graphs is vast, and we are going to limit this chapter to some of the subjects that relate directly to graph theory and networks. The theory in this chapter is mainly based on the book by Harary [8] with supplements from McHugh [19].

A graph $G=G(V,E)$ consists of a set of vertices $V=V(G)$ and a set of edges $E=E(G)$. The number of vertices, *V*, is $p \geq 1$, and the number of pairs of distinct vertices is $q \geq 0$. By this, the cardinality of V, denoted $|V|$, is called the "order of *G*", and the cardinality of *E*, denoted $|E|$, is called the "size of *G*". If a graph *G* contains *p* vertices and *q* edges, it is also referred to as a (*p*,*q*) graph. The vertices are usually shown as points and the edges are the connecting links between the vertices. Notation for an edge $e=\{u,w\}$ in *G*, that connects two vertices *u* and *w*, is *e* = *uw*.

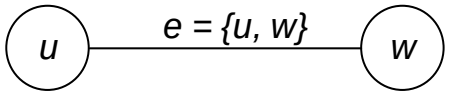


Figure 3: A simple graph

In the simple example shown in Figure 3, the two vertices $u \in V(G)$ and $w \in V(G)$, are said to be adjacent vertices, and the edge *uw* is incident to the two vertices.

The number of edges that are incident to a vertex $w \in V(G)$, ia called the vertex degree, denoted *deg (w)*, and the sum of the degrees is given by the following theorem:

Theorem 1 (See for example [8])

For a graph *G (V, E)* the sum of the degrees is:

$$\sum_{i=1}^{|V|} \deg(w_i) = 2 \mid E \mid \qquad (3.1)$$

□

As the vertices in the graph *G* can be arranged in an order like *v*1, *v*2, ..., *v*n, the corresponding vertex degree sequence is (*deg*(*v*1), *deg*(*v*2), ..., *deg*(*v*n))

The path from a vertex v_o in the graph *G(V, E)* to another vertex v_k in *G* is denoted as *P(V, E)*, where

$$V = \{v_0, v_1, \ldots, v_k\} \quad E = \{v_0v_1, v_1v_2, \ldots, v_{k-1}v_k\} \qquad (3.2)$$

This path can be denoted as an alternating sequence of vertices and edges, and is referred to as a path from v_1 to v_k:

$$v_1, e_1, v_2, e_2, \ldots, e_{k-1}, v_k \tag{3.3}$$

In the sequence above we see that all the vertices are distinct, and the vertices v_i and v_{i+1} are the endpoints of an intermediate edge e_i. We can also figure the length of the path by the number of edges of the path, denoted P_k, where k is the number of edges in the path.

A common way of performing a structured and organized representation of a graph is the *adjacency matrix*. This matrix shows the relation between the vertices as it indicates all incident edges. We define the adjacency matrix A of a graph $G\ (V, E)$ as the $|V| \times |V|$ matrix:

$$A(i, j) = \begin{cases} 1 & \text{if } (i, j) \in E(G) \\ 0 & \text{if } (i, j) \notin E(G) \end{cases} \tag{3.4}$$

4.4.2 Connectivity

We must apply a method of how to measure the "connectedness" of the security features for the hosts and network, and we use the classical connectivity theory.

The vertices $u \in V(G)$ and $w \in V(G)$, are connected in the graph G, if there exists a path from u to w in G. If every pair of vertices in G has this relationship, the graph G is a connected graph. If the removal of the vertex $u \in V(G)$ causes the graph G to be disconnected, meaning that the requirement is not true for all vertices in G, then the vertex u is a *cut-vertex*. This vertex-connectivity is denoted $\kappa(G)$, and represents the smallest number of vertices that can disconnect a graph. Similarly this is true for the edge that causes G to be disconnected, and this is called a cut-edge, where the edge-connectivity is denoted $\lambda(G)$.

In general, we say that a graph G is *k-connected* if it remains connected after removing $k-1$ vertices, where $k \geq 1$.

4.4.3 Graphs and computer networks

A computer network can easily be modeled as a graph, where the vertices in the graph represent the computers or network appliances, and the edges are the communication links between the communicating accessories. This can be an efficient way to represent a complex network, especially if the intention is to perform analysis such as a connectivity analysis.

To explain how the network-to-graph model is to be applied later in this thesis, we show in Figure 4 the theoretical topology model, the "hybrid mesh" topology [34]. This

model is then converted to a graph in Figure 5, and we also show the adjacency matrix for this graph.

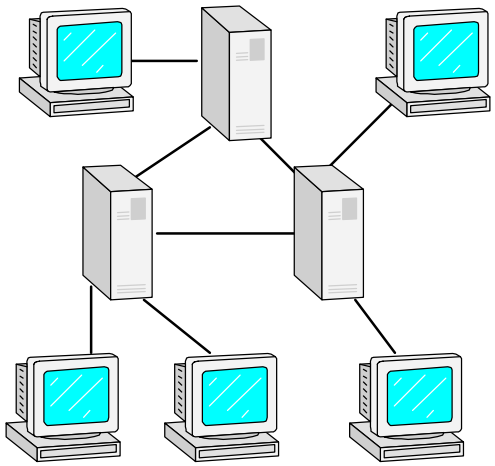


Figure 4: Example of the "hybrid mesh" topology

Figure 5 shows how the graph G (V, E) corresponds to the example of the hybrid mesh in Figure 4.

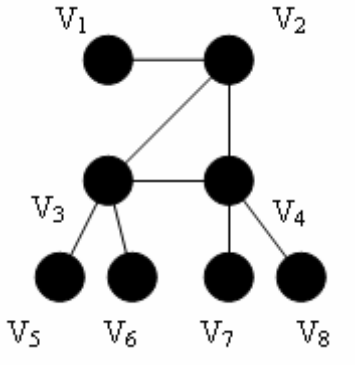


Figure 5: Hybrid mesh graph

V (G) = {V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8}
E (G) = {(V_1, V_2), (V_2, V_3), (V_2, V_4), (V_3, V_4), (V_3, V_5), (V_3, V_6),
(V_4, V_7), (V_4, V_8)}
Order ($|V(G)|$) = 8
Size ($|E(G)|$) = 8
$\kappa(G) = 1$ (V_2)
$\lambda(G) = 1$ (V_1V_2)

Adjacency matrix:

$$A = \begin{Bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{Bmatrix}$$

4.5 Connectivity as a security feature

The connectivity as a security feature is meant to assess the host- or network based security features and identify if they are distributed in any aspect. If the security features are distributed, it also involves how the related communications are performing at the loss of nodes in the connected network. A key element that influences connectivity as a security feature may for example be an IDS sensor, host or network based, that is logging to a database, or it may be a centralized logging system in general such as the syslog service [30]. This involves communication between the networked computers and nodes causing mutual dependability for the security feature. This led us to the terms "host based security connectivity" and "network based security connectivity".

These terms may be a little confusing, as the host based security features also make use of networked systems for communication. But the term reflects the security feature's main target, for example as the syslog service's main target is to distribute a host's log file entries to a centralized server, this will be "Host based security connectivity", and as a NIDS[8] target is to monitor network activity we talk about "Network based security connectivity". This is only for the purpose of separating the different security features, and by this separation we can identify the different dependencies of these security features.

4.5.1 Host based security connectivity

The need for software based security features for the hosts is already a fact. As we see, many of the virus and worm attacks are targeted at workstations and other client computers via e-mail. If the antivirus system on the mail server does not detect a message containing a worm, there is already a breach of security, and the infected mail may continue to its target. But the worm can still be deleted; the security now depends on the targeted hosts' security performance. If the antivirus program on the host is able to detect and stop the worm, the threat is eliminated. If it is not able to detect the worm, the worm may continue to do whatever it is programmed to do. But the worm may still be stopped. If the host has a software based firewall or host based IDS, the worm can be detected when it is trying to access files or establish a network connection. This is a common scenario, and if the host based security software is configured properly, there is a high probability that this worm will be stopped. The only problem is that without any notification to the network or system administrators

[8] Network-based Intrusion Detection System [16]

it will probably take a while for them to get information about the security incident. To deal with this important problem, Grance, Kent and Kim [18] propose to implement centralized logging, making the host's operating system and security functions log to a central log-database on the internal network. Some of the advantages from this would be more efficient monitoring and troubleshooting.

This example describes how hosts can participate in special communication links with other computers/appliances, and by this we can form a graph, based on services/security features between the hosts/appliances. Other services that will use the same principle of connectivity between hosts are e.g. remote management, such as patch management agents.

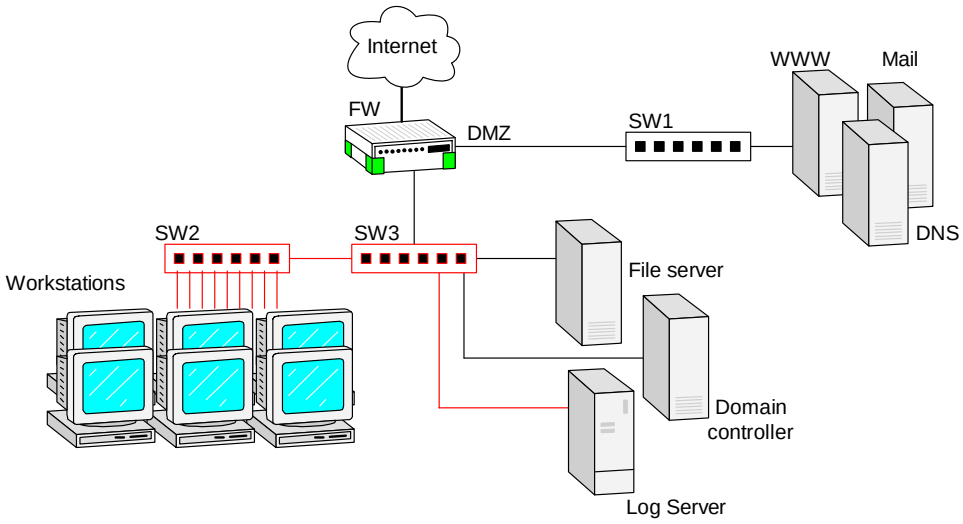


Figure 6: Small enterprise network

Figure 6 demonstrates the path for the syslog[9] [30] service for the workstations, as it is configured to communicate with a log server on the internal network. As this is just an example for explanation, we have chosen a relatively simple network topology. The graph for the syslog service can be shown in Figure 7.

[9] The original syslog service is not highly reliable and secure remote log-service, and it should not be used when security is essential. The Syslog-ng is a possible replacement [27], using TCP and SSH.

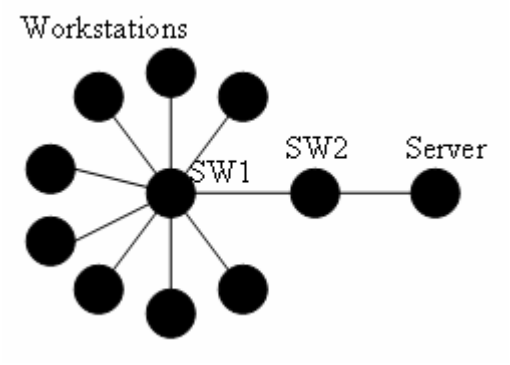


Figure 7: Syslog graph

The syslog feature depends on the communication to the log-server, and this makes it dependent on the network and the involved computers/appliances. We easily see which vertices may cause this communication to fail. By this registration, we can avoid to make other security features, like host based IDS, dependent on the same appliances. This can contribute to increasing the overall robustness of the system security features.

4.5.2 Network based security connectivity

Network based security connectivity follows the same principle as Host based security feature, but the security features are network based this time. This can for example be one or several NIDS-sensors logging to a central database. Again, it is the security feature's main target that is the subject for the term. As the NIDS are considered network based, they are sorted under the term "Network Based Security Connectivity".

5 Security Metrics for Patch Management and security connectivity

5.1 Introduction

The structure of our metrics is based on the structure presented in [12], but we have performed a few changes to adapt the form to our usage. This structure is presented in Table 2.

Metric ID	A unique identification for the metric
Metric subject	A short summary of the metric subject
Performance Goal	The desired goal of the metric. This can be a simple question.
Performance Objective	One or two questions that will help the user to accomplish the performance goal.
Metric	A description of the quantitative measure of the metric.
Purpose	A description of the overall purpose of the metric.
Implementation Evidence	A list of questions that functions as indicators for the metrics goal. The questions are either a Yes/No answer, or related to a number of devices that implement the questioned subject. For the YES/NO answer, YES = 1 point and NO = 0 points. For the numbered answer, the score is the number of devices that implement the questioned subject divided by the total number of devices in the network. This will form a result between 0 and 1. All questions are put in such a way that the highest score gives the best result.
Frequency	A frequency of metric evaluation. The performance goal will also indicate how often a business should execute this metric.
Formula	A formula to guide the user to get the metrics value, from the set of the data from the questions in the Implementation Evidence.
Data Source	A list of sources to collect data from, in order to answer the Implementation Evidence.
Indicators	A short description of the target and the goal of the metric.

Table 2: Description of the metric form

The metrics were developed to support some of the steps in the Patch Management process, and to make the organizations focus on security when performing the deployment of the patches. These metrics support the procedure of identifying the implemented security features, and give the possibility to organize this information into a structured map. By this, organizations will be able to identify if the patches that are about to be deployed affect the security features of the system.

The metrics are as follows:

1. Assets List
   The metric aims to support the identification of all equipment in use; and in addition there is a suggestion of how to register the collected information.
2. Vulnerability scanning
   Main goal of this metric is to support the use of vulnerability scanning of the systems. This is an essential task in the search for vulnerabilities and should be carried out correctly.
3. Host based security connectivity
   This metric is targeted at the host based security features. The connectivity aspect in search here is if some of the identified features are somehow remotely controlled or performs central logging.
4. Network based security connectivity
   This metric is targeted at the network based security features and their relation to other appliances.

5.2 Metric 1: Assets List

5.2.1 General

This is the metric that is to be evaluated first. The target is to identify all computers and appliances, and to register as much vital information as possible about them. It is vital to have a correct and updated list in order to be able to perform connectivity analysis later on, but, as we learned from Chapter 3, it is also a recommended part of the Patch Management process.

The implementation evidence is formed as questions, and we shall now account for every question to enlighten their reasons and motive.

The collected information from this metric should be organized in a record and indexed for searching purposes.

Any of the answers in this metric is rewarded with 1 for YES, and 0 for NO. The total score could range is from 0 to 8, in integers only.

5.2.2 Implementation evidence

1. Are all computers, appliances and networked devices described in the list?

All computers, appliances, devices that are or are not connected to the network should be registered in the list. This involves identifying all firewalls, routers, switches, portable handheld devices, wireless access points/receivers, printers etc.

2. For every computer or appliance, does the list describe the following?

We present a suggestion of the information that the list should contain for every entry:

- Unique identification
- Short overall description
- Operating system
  - Version number
  - Patch level
  - Configuration
- Software
  - Version number
  - Patch level
  - Special configuration
- Hardware and firmware
  - Product vendor
  - Serial number
  - Drivers
  - Description
- Services and port numbers
- Each computer or component's location, both logical and physical, IP-address/IP-range
- Security dependencies and communication to other computers or appliances.
- A thorough description;
  - System name
  - Network alias
  - Role in the network/organization
  - Patch history
    - Allowed/forbidden connections
    - Main users
    - Main software vendor's patch release sites.
  - Other comments

3. Is the list correlated with network scanning procedures or other techniques for revealing networked nodes?

The produced list should be used when performing some kind of scanning of the network to verify the list and to check for other networked nodes. Variance in result should cause examinations of reasons for this.

5.2.3 Information form

The data gathered from the metric should be organized in a database. These data should be presented in an orderly manner, and we have a proposal for the presentation form, shown in Table 3. Changes to this proposal should be made after the organization's need or requirement. The document by Herzog [25] proposes an information form for network and computers, but we found these forms too narrow for our needs. There may also be several predefined forms available from the Internet.

Identification information	
Short overall description	

Operating system

OS	
Version number:	
Patch level:	
Configuration:	

Software

Product	version number	patch level	configuration

Hardware and firmware

description	Product/vendor	drivers	serial number	Other information

Services

Service	port	protocol	Other information

Location

Physical location	Logical location	DHCP	IP/IP-range

Security dependencies and relationship to other computers or appliances.	

A thorough description

System name	
Network alias	
Role	
Main users	
Connections	
Patch history	
Patch release sites	

Table 3: Asset information form

5.2.4 Metric form

The metric Assets list is shown in Table 4.

<table>
<tr><td>Metric ID</td><td>SM-IL</td></tr>
<tr><td>Metric subject</td><td>Assets List</td></tr>
<tr><td>Performance Goal</td><td>Are all computers and appliances described in a list?</td></tr>
<tr><td>Performance Objective</td><td>Is the assets list complete, containing correct information, and is this list regularly updated?</td></tr>
<tr><td>Metric</td><td>Higher total score value gives a higher availability of the total security function.</td></tr>
<tr><td>Purpose</td><td>To support the creation of an Assets list</td></tr>
<tr><td>Implementation Evidence</td><td>1. Are all computers, appliances and networked devices described in the list?
YES:__ NO:__
2. For every computer or appliance, does the list describe the following:
a. Unique identification information
b. Short overall description
c. Operating system; version number, patch level and configuration
d. Software; version number, patch level, special configuration
e. Hardware and firmware; product vendor, serial number, drivers and description
f. services and port numbers
g. Each computer or component’s location, both logical and physical, IP-address/IP-range
h. Security dependencies and relationship to other computers or appliances. (E.g. IDS sensor/database)
i. Thorough description; system name, network alias, role in the network/organization, patch history, allowed/forbidden connections, main users, main vendors patch release site.
j. Other comments
#YES:__ #NO:__
3. Is the list correlated with network scanning procedures or other techniques for revealing networked nodes?
YES:__ NO:__</td></tr>
<tr><td>Frequency</td><td>For all the changes performed on the network topology, or periodically in accordance to the security policy/patch management policy</td></tr>
<tr><td>Formula</td><td><table><tr><td>Score</td><td>YES = 1 NO = 0</td></tr><tr><td>Formula</td><td>1 + # YES in 2 + 3</td></tr><tr><td>MIN</td><td>0</td></tr><tr><td>MAX</td><td>12</td></tr></table></td></tr>
</table>

Data Source	• Network topology • Network scanning or vulnerability tools • System administrators • System owners
Indicators	In order to know which patches to deploy, there must exist a thorough list of the inventory regarding all computers and network components in the organization. In addition to this list, there must be an accurate description of each computer or component and its use. The strictest organizations may also include the users groups into this, stating which user group is to do what and how. This list will also contribute to identify security relationship between computers, and by this contribute information to establish a security graph of the network.

Table 4: Metric 1: Assets List

5.3 Metric 2: Vulnerability scanning

5.3.1 General

As we saw in Section 3.1, the purpose of performing vulnerability scanning is to reveal both existent and potential vulnerabilities and report this in an orderly fashion. Using automated tools increases the possibility of controlling and identifying a larger amount of systems, services and potential vulnerabilities. Some drawbacks are related to the use of these tools, but this is a matter that has to be dealt with prior to the measuring of this metric.

The results from this scanning should be correlated with the metric "Assets List", and if there are any differences this should be looked into. If vulnerabilities or not approved/known services are discovered by the vulnerability scanning, these should be reported.

Any of the answers in this metric is rewarded with 1 for YES, and 0 for NO. The total score could range is from 0 to 9, in integers only.

5.3.2 Implementation evidence

1. Is vulnerability check of all the systems in the network performed?

The vulnerability scans should be performed on every computer that is connected to the network. If the target is of high importance for security or availability, or for other reasons should not be scanned by a common vulnerability scanner, the use of alternative vulnerability checks should be considered. This can be e.g. host-based scanners or a manual check of the target.

2. Are these scans performed at every host/computer periodically?

There should be a scheme that regulates when every computer is to be scanned. In such a scheme it will be possible to separate targets based on an analysis of potential impact if compromised, vulnerability history, importance of availability etc.

3. Are network based vulnerability scanners used?

These tools can be difficult to configure, and they may produce false alarms, both false negative and false positive. In spite of these difficulties they are efficient tools, and are not to be excluded in the security work.

To use a network based scanner is also important for the purpose of detecting connected systems. This approach depends on several factors, such as if the scanning tool uses ICMP or TCP ping, [22], and several other factors.

4. Are host based vulnerability scanners used?

Network based scanners are common and easy to use, but they may have negative influence on the target. Situations like bandwidth consumption, target availability problem and false positives are some of the common problems caused by these kinds of scanners. Because of this, it may be an advantage to use a host based scanner for some targets or situations. These can often be managed remotely.

5. Are vulnerability scanners from at least two different vendors used?

It will always be the safest solution to use security products from several vendors. It is a recommended strategy for antivirus, firewalls etc [20]. This is also true for vulnerability scanners. The motive for this is that what one product is not capable of detecting, the other product may be.

6. Are the rules/signatures/plug-ins for the vulnerability scanner updated before use?

Like other rule- or signature based security features, the vulnerability scanners have to be updated before use to ensure completeness. For most of the tools this is a trivial matter, and may even be automated.

7. Are the vulnerabilities discovered in the scan added to the list of vulnerabilities to be patched?

This is an important question, but as usual it can be a difficult process to divide false positives from the real vulnerabilities. The goal is that the vulnerabilities are the target of investigation, and if they are real they should be reported and dealt with.

8. Are these scans targeted at hosts after the installation of patches to confirm vulnerability removal?

The intention of many patches is to remove vulnerabilities or to perform some kind of other change to the applications or operating system. But as we want to be sure that the vulnerability is removed and that the changes did not cause or produce other vulnerabilities, we have to perform a vulnerability scan after the systems are patched.

9. Are manual actions and reports for detected vulnerabilities performed if no patch is available?

This question targets if there is some kind of alternative response to vulnerabilities that do not have a released patch. The criticality of the vulnerable system will decide how important it is to perform an alternative plan.

5.3.3 Metric form

The metric is shown in Table 5.

Metric ID	SM-VS
Metric subject	Vulnerability scanning
Critcal Element	Is a vulnerability scan of the systems performed frequently?
Subordinate question	Is the vulnerability scan targeted at every computer in the system? Are the results from the vulnerability scan taken into account for Patch Management?
Metric	Provides the results from vulnerability scans to perform patching of the vulnerable computers. Performing vulnerability scan in a correct and effective manner
Purpose	Measures the completeness and effectiveness of the vulnerability scan
Implementation Evidence	1. Is vulnerability check of all the systems in the network performed? **YES:__ NO:__** 2. Are these scans performed at every host/computer periodically? **YES:__ NO:__** 3. Are network based vulnerability scanners used? **YES:__ NO:__** 4. Are host based vulnerability scanners used? **YES:__ NO:__** 5. Are vulnerability scanners from at least two different vendors used? **YES:__ NO:__** 6. Are the rules/signatures/plug-ins for the vulnerability scanner updated before use? **YES:__ NO:__** 7. Are the vulnerabilities discovered in the scan added to the list of vulnerabilities to be patched? **YES:__ NO:__** 8. Are these scans targeted at hosts after the installation of patches to confirm vulnerability removal? **YES:__ NO:__** 9. Are manual actions and reports for detected vulnerabilities performed if no patch is available?

<table>
<tr><td></td><td>YES:__ NO:__</td></tr>
<tr><td>Frequency</td><td>All systems should be scanned periodically

All patched hosts should be scanned immediately after the patches are installed</td></tr>
<tr><td>Formula</td><td><table><tr><td>Score</td><td>YES = 1 NO = 0</td></tr><tr><td>Formula</td><td>1+2+3+4+5+6+7+8</td></tr><tr><td>MIN</td><td>0</td></tr><tr><td>MAX</td><td>8</td></tr></table></td></tr>
<tr><td>Data Source</td><td>• Network topology
• Network scanning or vulnerability tools
• System administrators
• System configurations
• Security Policy
• Patch Management Policy</td></tr>
<tr><td>Indicators</td><td>This metric measures if and how the vulnerability scanning is performed and relates this process to the Patch Management. The target for this metric is to identify if the vulnerability scan is a part of the Patch Management in any way. High score can be reached without correspondence with the metric’s target, but the highest score cannot be reached this way.</td></tr>
</table>

Table 5: Metric 2: Vulnerability Scanning

5.4 Metric 3: Host based security connectivity

5.4.1 General

As we have explained the term "security connectivity" in Section 4.5, we now develop a metric for targeting the "host based security connectivity". This metric's main target is to help the user to identify the security features on all different hosts, and to examine if and how these features are communicating with other computers/appliances in the network. The data from this metric can be arranged to identify how different security features make use of other appliances and computers. This can in turn be used in security and availability strategies.

Some questions require YES/NO answers, and are also here rewarded with 1 for YES, and 0 for NO. The total score is decimal numeral and the range is from 0.0 to 8.0.

5.4.2 Implementation evidence

1. How many hosts/appliances are connected to the network?

To be able to compute the various percentages later, we should have a total number of computers/appliances that are taken into account for this. These appliances can be everything from firewalls, servers, workstations and other devices that contribute to the communication in the network, or any device that can be configured in any aspect. The typical non-managed layer 2 switches will not be counted here, as they are not considered a part of any security feature, nor do they support any security features.

2. How many networked hosts have enabled any kind of host security feature (e.g. software firewall, IDS and so on)?

This question targets every host, server, firewall or other device that is connected to the network and that also has a configured security feature, such as a host based IDS, or performs logging of any kind. This number ought to be nearly as high as in the first question for this metric, and we can compute the percentage based on the number from the previous question. This will be the base of how many computers/appliances can potentially contribute to the host based connectivity feature.

The user should also be sure of why those hosts not counted in this question not are secured in any way.

3. How many networked hosts have enabled any kind of logging feature (e.g. system logging, logging from software firewall, IDS and so on)?

If there exists some security related software it should be performing logging, and the logs should be secured from tampering. Even the logs from the system itself should be secured, as they may contain vital information e.g. after an attack or for troubleshooting purposes.

The purpose of this question is to make the user aware of several logging features in the security software that may be in use, and configure them properly. This applies also to the system log features. After this, the accumulated number may be used to compute the percentage based on the number from this metric's first question.

4. How many of the hosts security features perform communication to other appliances (e.g. logging or centralized management/update)?

There may be communication between hosts and computers/servers/appliances that is related to some of the security features on the hosts. This could be a centralized antivirus/IDS-signature server, central management consoles, log features and so on. If this kind of communication exists, this information should be registered for use in the final question in this section.

The number here will be an estimate of how many of the computers/appliances that host security features probably depend on other equipment to work.

5. How many of the systems log features log to other appliances (e.g. log servers or database)?

We have separated the log function as this may not be considered as a security feature. We do it like this to avoid any misunderstanding or confusion. The target of this question is the same as the previous question; to identify communication for security/management purposes for use in the connectivity analysis.

6. Does an alternative route for the communication exist?

This is an important question, because here the user is meant to identify any redundant communication paths for the hosts. This means that where there are several paths for the communication to be routed, this must be registered. Any security feature that has an alternative path for communication to its target is subject to a higher connectivity and may be more robust against removal of a device in the network.

7. Is it possible for the hosts to communicate to an alternative secure source?

To ensure the best availability and performance of the security features, there should be alternatives for them. In this question we try to identify if an alternative source for the hosts security feature to communicate with is implemented, such as a redundant log server. The question is not concerned with the communication path as it was in the previous question. Redundant solutions increase the connectivity for the security feature.

8. How many hosts have an alternative configuration to ensure security?

Here we probe for how many systems have an alternative configuration in place, to be used in abnormal situations. This may be in the case of an attack, when performing patching of important security related computers/appliances, etc. The number should be nearly as high as the total number of hosts.

9. Are the hosts and appliances that are part of a distributed security or management system identified and form a model of reference to be used to analyze the host's security connectivity?

The intention is that the user should model a graph or a topology map for the previously identified communication, and this will be the base to perform connectivity analysis. This map can be used when performing expansion of the network, to identify

how the security features depend on other devices, which security features are available, if any devices are not working and so on.

5.4.3 Metric form

The metric is shown in Table 6.

Metric ID	SM-HSC
Metric subject	Host security connectivity
Performance Goal	Is the connectivity identified for the host's security features?
Performance Objective	Are the host security features distributed in any aspect? Are there alternative solutions to sustain security if the security connectivity is broken?
Metric	Higher total score value gives a higher availability of the total security function.
Purpose	To identify and elaborate the connectivity of the host security features.
Implementation Evidence	1. How many hosts/appliances are connected to the network? #____ 2. How many networked hosts have enabled any kind of host security feature (e.g. software firewall, IDS and so on)? #____ 3. How many networked hosts have enabled any kind of logging feature (e.g. system logging, logging from software firewall, IDS and so on)? #____ 4. How many of the host's security features log to other appliances other than to themselves (e.g. log servers or database)? #____ 5. How many of the systems log features log to other appliances other than themselves (e.g. log servers or database)? #____ 6. Are the log servers/DB available from several segments of the network? **YES:__ NO:__** 7. Is it possible for the hosts to log to an alternative secure source? **YES:__ NO:__** 8. How many hosts have an alternative configuration to ensure security? #____ 9. Are the hosts that are part of a distributed security or management system identified to form a model of reference

<table>
<tr><td></td><td>to be used by the system/network administrators?
YES:__ NO:__</td></tr>
<tr><td>Frequency</td><td>For every changes performed on the network topology, when causing disconnected security graphs (restarting/patching systems), performing patch management, or periodically in accordance to the security policy/patch management policy</td></tr>
<tr><td>Formula</td><td><table>
<tr><td>Score</td><td>Yes = 1, No = 0</td></tr>
<tr><td>Formula</td><td>(2/1)+ (3/1)+ (4/1)+ (5/1)+6+7+(8/1)+9</td></tr>
<tr><td>MIN</td><td>0</td></tr>
<tr><td>MAX</td><td>8</td></tr>
</table></td></tr>
<tr><td>Data Source</td><td>• Network topology
• Network scanning or vulnerability tools
• System administrators
• System configurations
• Security Policy
• Patch Management Policy</td></tr>
<tr><td>Indicators</td><td>This metric measures the connectivity level of the host system’s security features and alternative configurations to ensure security. The target of this metric are the security features for all networked hosts and systems, including special network or security appliances like firewalls and so on. Only by evaluation of the systems as a whole, the measurement can reach the highest (acceptable) score.</td></tr>
</table>

Table 6: Metric 3: Host based security connectivity

5.5 Metric 4: Network security connectivity

5.5.1 General

This metric follows the theory of the connectivity as explained in Section 4.5, and the logic of this metric is the same as for the "host based connectivity" metric. The main target of this metric is to help the user identify the network security features, and the appliances that these features make use of for function purposes. The data from this metric can be arranged to identify how different security features make use of other appliances and computers. This can in turn be used in security and availability strategies.

Any of the answers in this metric is rewarded with 1 for YES, and 0 for NO. The total score could range is from 0 to 8, in integers only.

5.5.2 Implementation evidence

1. Does the network include network monitoring features or other network related security appliances (e.g. NIDS, firewall etc)?

This question is targeting the overall network security appliances, making the user identify these appliances.

2. Are the network security appliances covering all segments of the network?

The network security must be covering most of the segments of the network; this is to ensure that all possible paths are monitored or filtered. This is often a question about money, but if e.g. redundant paths are making it possible to circumvent/bypass security features, the security can be compromised.

3. Do any of the network security appliances overlap each other?

This question tries to identify if there are any security features that function in a way that creates an overlapping related to other security features/appliances.

4. Do the network security appliances perform centralized logging (e.g. log servers or log database)?

There could be several benefits from making e.g. the firewalls or different IDS sensors log to a centralized log database. If centralized logging is performed, the user should also identify all appliances that are involved in the communication paths.

5. Are the log servers/databases available from several segments of the network?

If there are redundant paths between segments and/or the centralized log server/databases, it can affect the connectivity for the security features.

6. Are the computers and servers related to security features separated from the production network?

The main purpose of this question is to identify if there is an additional management network for the security related communication. This could affect the connectivity for the security features, e.g. if two overlapping or similar security features use two

separate paths for logging/management.

7. Are the "cut nodes" identified for the network security features?

The "Cut nodes" are those vertices that cause the graph to be disconnected if they are removed. Similar to this, we are using this term for the nodes that can cause the security related communication to fail if they shut down. These nodes are important to identify to avoid several security features depending on the same node.

8. Is special attention given to the network activity when performing shutdown or separation of security related "cut node" computers or appliances?

This question is meant to identify if there are alternative procedures when devices/computers that are related to security are down or are not performing as they should. This could be if e.g. the NIDS is out of order, and there exists no other monitoring of the network. In such a situation, the alternative plan could be to increase the logging in the firewalls, maybe turn on logging for every incoming connection etc. Such a configuration would produce a great amount of log data, and the impact of monitoring may decrease if the logging continues for more than a short period, because of large records and a large amount of data. But the point is that this records the network traffic in periods of vulnerability for just this subject.

5.5.3 Metric form

The metric is shown in Table 7.

Metric ID	SM-NSC
Metric subject	Network security connectivity
Performance Goal	Is a connectivity graph for network security defined?
Performance Objective	Are the network security functions of the systems distributed in any aspect? Are various security functionalities designed for availability?
Metric	Higher total score value gives a higher availability of the total security function.
Purpose	To identify and elaborate the connectivity of the network security features.
Implementation Evidence	1. Does the network include network monitoring features or other network related security appliances (e.g. NIDS, firewall etc)? **YES:__ NO:__** 2. Are the network security appliances covering all segments of the network? **YES:__ NO:__** 3. Do any of the network security appliances overlap each other? **YES:__ NO:__** 4. Do the network security appliances perform centralized logging (e.g. log servers or log database)? **YES:__ NO:__** 5. Are the log servers/databases available from several segments of the network? **YES:__ NO:__** 6. Are the computers and servers related to security features separated from the production network? **YES:__ NO:__** 7. Are the "cut nodes" identified for the network security features? **YES:__ NO:__** 8. Is special attention given to the network activity when performing shutdown or separation of security related "cut node" computers or appliances? **YES:__ NO:__**

<table>
<tr><td>Frequency</td><td>For all the changes performed on the network topology, or periodically in accordance to the security policy/patch management policy</td></tr>
<tr><td>Formula</td><td><table>
<tr><td>Score</td><td>YES = 1 NO = 0</td></tr>
<tr><td>Formula</td><td>1+2+3+4+5+6+7+8</td></tr>
<tr><td>MIN</td><td>0</td></tr>
<tr><td>MAX</td><td>8</td></tr>
</table></td></tr>
<tr><td>Data Source</td><td>• Network topology
• Network scanning or vulnerability tools
• System administrators
• System configurations
• Security Policy
• Patch Management Policy</td></tr>
<tr><td>Indicators</td><td>This metric measures the connectivity level of the system's security features. The main purpose of this metric is the security features for all networked hosts and systems, including special network or security appliances. Only by evaluation of the systems as a whole, the measurement can reach the highest (acceptable) score.

Question 8 is meant to identify any special security related procedures performed (only) when some of the public/private servers/database/sensors are out of function and causes any communication link to be disconnected.</td></tr>
</table>

Table 7: Metric 4: Network Security Connectivity

5.6 Discussion

One of the reasons to develop security metrics is to support the performance of measurement to a security related area of evaluation [12]. The results from the metrics are the target of an interpretation that will create information about how e.g. the newly implemented security feature has improved the security of the organization. To ensure the highest validity and reliability of the results, the metrics should be adjusted to fit the organization they are evaluated in. These adjustments constitute the base of the metric difficulties, and metrics' that are not optimized for their use may result in incorrect results and misleading decisions.

Optimization of metrics is an organizational related process, making the development of general metrics very difficult. If general metrics contain specific areas that are not adjusted to the implementing organization, these metrics may mislead the organization to misinterpret the goal of measure for the metric, resulting in incorrect measures for the organization. Organizations could develop their own specific metrics, but for the metrics in this thesis, they should be as general as possible to be useful for as many organizations as possible.

To develop general metrics has been a central goal for this thesis, and they had to be rewritten several times before we had removed any specific parts related to of Patch Management and security. As an example we can make the metrics for vulnerability scanning targeted at finding vulnerabilities just for Patch Management, or we can develop the metrics to support general aspects of vulnerability scanning. Our decision was the general metric solution, because even if the vulnerability scanning task is performed as a part of the Patch Management process, it does not change the vulnerability scanning procedures or criteria. The metrics' relation to Patch Management is to perform the process as a whole, and not as single steps within the process. This is also in compliance with the thesis' main goal, to focus on the security aspect of Patch Management, and we do this by using metrics that support our agenda of this thesis' subject.

The results were metrics that also are useful to for other security related tasks, not only the steps in the Patch Management process as they were originally designed for.

We have also tried to make the metrics both quantitative and objective in accordance to the statements in [31]. Our metrics are using quantitative measurements to produce a single value that states the level of success for the particular metrics, and an objective measure to produce the results in the sub-questions within every metric. This makes it possible to avoid qualitative and subjective interpretations, and makes the metrics more robust for generic use in various environments and situations.

5.7 Summary

We suggest some metrics to be used in the Patch Management process. These metrics are not intended to cover all aspects in this process, but the use of these metrics provides information about the security features and how they relate to the surrounding environment. This information supports the mapping of the security

features and their dependencies on other equipment for functioning properly. This mapping can be used in the Patch Management process to identify how the deployment affects these security features, and if changes must be performed to ensure that the environment is as protected as intended.

6 Experiments and results

6.1 Introduction

This chapter describes how the experiments were performed, and presents the results from these experiments. The main goal of the experiments is to test the metrics on different network topologies that implement different security solutions. To test the security, we exposed the network to an attack. The attack was simulated with a vulnerability scanner, and the objective of this approach was to register if any of the scanning went undetected. When this happened, we could conclude that the security was not optimal. Registrations were correlated with the outcome from the metrics, and served as an indicator of how the metrics performed in different security configurations.

The results from the three configurations are presented in Section 6.3 and summarized in Section 6.4. All details for the experiment can be found in Appendix A.

6.2 Experimental design

We performed several experiments with changes to the network topology and security configuration. First we started with the typical "Small Business" network topology, meaning a single firewall with servers in a DMZ and an internal network. The security features, beside the firewall, were a NIDS sensor logging to a MySQL database. From this design we added security features such as more NIDS sensors, implementation of centralized logging, configuring HIDS[10] on some of the hosts etc. We also performed changes to the network topology, and these modifications were influenced by several topologies found in [20], where the designs were carried out with security in mind.

For every configuration, we performed a vulnerability scan with the Nessus [23] vulnerability scanner. This scanner had identical configuration for every scan. The scanning to simulated an attack, and as the scanning triggered some of the implemented security features, we treated those registration entries as attack evidence. We did not include the details from this data, as we were mostly interested in whether the security features were detecting the scan or not. From the registrations we could conclude if the security features were monitoring and working properly.

Also, for every change in the design or security configuration, we applied our metrics. The data from these metrics were used as a basis to produce the security graphs and the relevant graph data.

6.3 Results

6.3.1 Configuration 1

The topology, shown in Figure 8, is meant to be a model of a larger corporate network. There are several components missing, such as e-mail and other usual services, but the main goal of this model is to implement some security features for testing the metrics.

The IDS computer has 2 sensors that cover both channels from the firewall. By this

[10] Host-based Intrusion Detection System [16]

configuration, in theory, all communication that is going in and out from the firewall is monitored. These sensors log to a MySQL database, containing the analyzing tool ACID [28], which ships with Snort IDS [35].

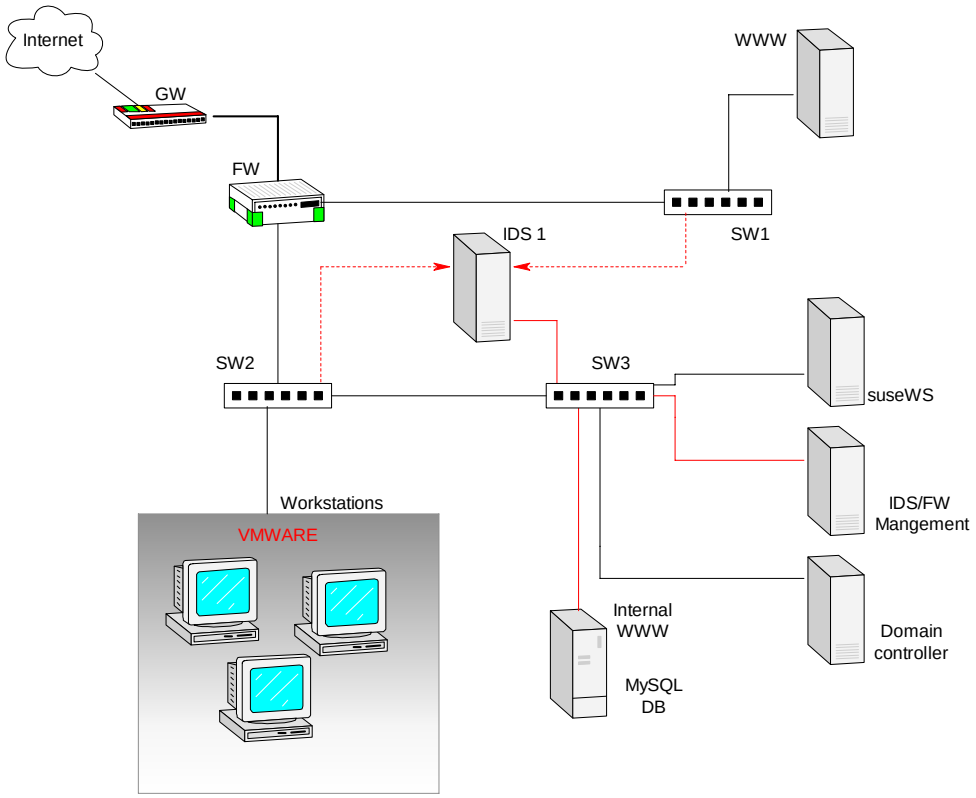


Figure 8: Network topology, configuration 1

The metric results summarized

As we see in Table 8, the metrics' evaluated for configuration 1 gained a score of 19.1.

Metric	Score	Max score	Difference
Assets List	9	11	2
Vulnerability scanning	3	5	2
Host security connectivity	2.1	8	5.9
Network security connectivity	5	7	2
TOTAL:	19.1	31	11.9

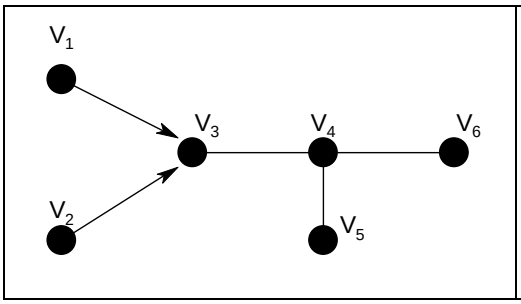
Table 8: Metrics result, configuration 1

Security graphs.

From the data gathered in the process with the metrics, we can now model a graph for the security features of the network, shown in Table 9.

The only security feature that was performing any centralized logging was the NIDS monitoring feature. As we see from the graph below, it is not a very robust graph in

terms of availability. If any of the vertices IDS, SW3 or MySQL database fails, no security monitoring is performed in the network. The directed edges v_1v_3 and v_2v_3 indicate that the communication link only works one way. For this example it is the transmit feature that is blocked for the NIDS sensors, making them "receive-only".



V_1	SW1
V_2	SW2
V_3	IDS
V_4	SW3
V_5	IDS/FW management
V_6	MySQL database

Table 9: Security graph, configuration 1

From this, we can compute the connectivity and other graph data.

$V(G) = \{V_1, V_2, V_3, V_4, V_5, V_6\}$
$E(G) = \{(V_1, V_3), (V_2, V_3), (V_3, V_4), (V_4, V_5), (V_4, V_6)\}$
Order ($|V(G)|$) = 6
Size ($|E(G)|$) = 5
$\kappa(G) = 1$ (V_3, V_4)
$\lambda(G) = 1$ (any edge)

6.3.2 Configuration 2

The topology in this experiment, shown in Figure 9, is based on the topology in experiment 1; but this time the workstations have the Prelude Hybrid IDS [26] installed. This host based IDS perform logging to the Prelude Manager [29], which in turn logs to a database (at the same host). This computer is placed at the same location as the other management computers.

The IDS has the same configuration as before, and it still performs logging to a MySQL database.

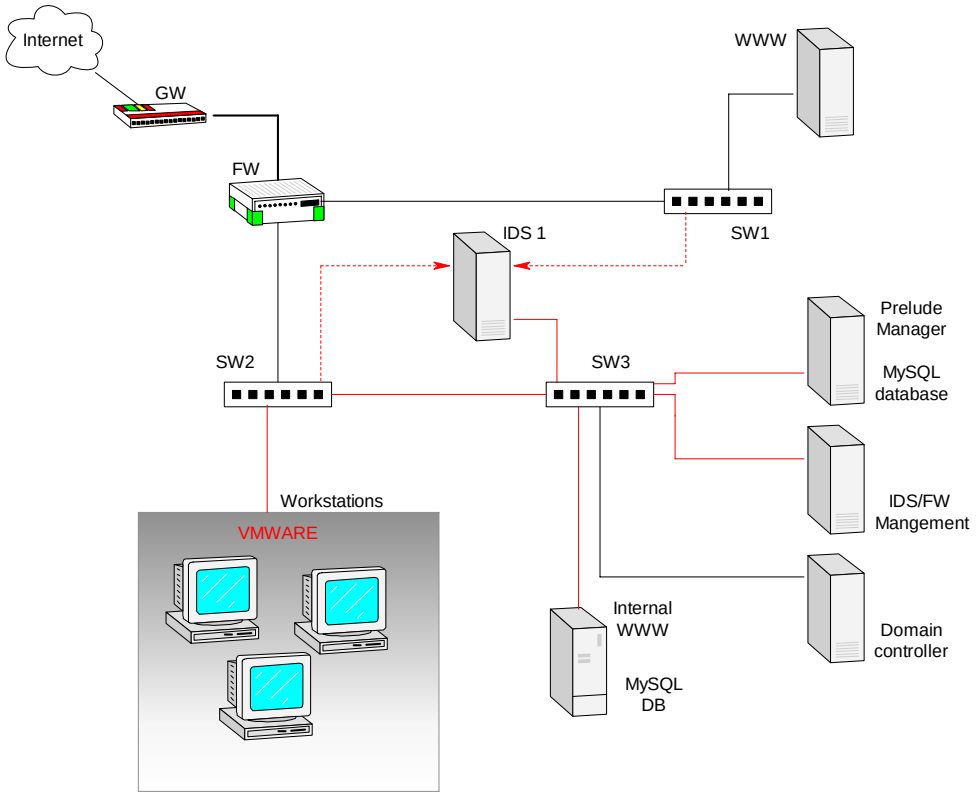


Figure 9: Network topology configuration 2

The metric results summarized

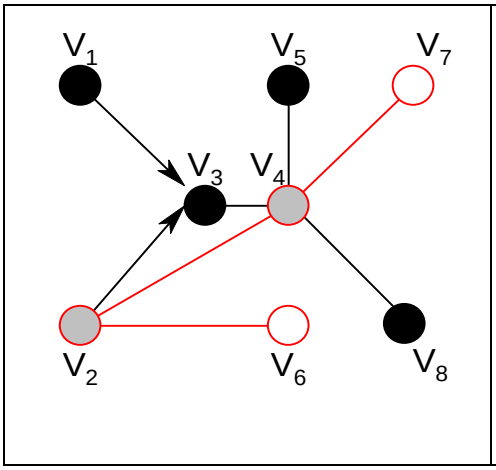
As we see in Table 10, the metrics' evaluated for configuration 2 gained a score of 19.5. This is a slightly higher score compared to configuration 1, and is a step towards higher security.

Metric	Score	Max score	Difference
Assets List	9	11	2
Vulnerability scanning	3	5	2
Host security connectivity	2.5	8	5.5
Network security connectivity	5	7	2
TOTAL:	19.5	31	11.5

Table 10: Metric results, configuration 2

Security graphs

The changed configuration caused another security graph to appear in the network, in addition to the other NIDS-graph from the first configuration. Because both of these security features are implemented in the same network, using the same appliances, the graphs are overlapping each other. We have colored the NIDS-graph black, the HIDS-graph red with white nodes, the nodes used by both features are filled grey, and displayed them joined. As we see from the graphs in Table 11, both of the security features are dependent on one switch, V4.



V_1	SW1
V_2	SW2
V_3	IDS
V_4	SW3
V_5	IDS/FW management
V_6	Prelude IDS sensors (workstations)
V_7	Prelude Manager
V_8	NIDS MySQL database

Table 11: Security graph configuration 2

In Table 12 we show the vertex connectivity and other graph data for this configuration.

The data from the NIDS graph ***G***	The new HIDS graph ***H***
$V(G) = \{V_1, V_2, V_3, V_4, V_5, V_6\}$ $E(G) = \{(V_1, V_3), (V_2, V_3),$ $(V_3, V_4), (V_4, V_5), (V_4, V_6)\}$ Order ($\lvert V(G) \rvert$) = 6 Size ($\lvert E(G) \rvert$) = 5 $\kappa(G) = 1$ (V_3, V_4) $\lambda(G) = 1$ (any edge)	$V(H) = \{V_4, V_6, V_7\}$ $E(H) = \{(V_4, V_6), (V_4, V_7)\}$ Order ($\lvert V(H) \rvert$) = 3 Size ($\lvert E(H) \rvert$) = 2 $\kappa(G) = 1$ (V_4) $\lambda(G) = 1$ (any edge)

Table 12: Graph data, configuration 2

From this data we see that the vertex V4 is the “cut node”, and that this is one out of two cut-nodes for the NIDS graph, and a “shutdown” of this node would stop both of the implemented security features.

6.3.3 Configuration 3

The topology in this experiment, shown in Figure 10, is a theoretical design of a business that monitors the network with several IDS configurations, both NIDS and HIDS. They also make use of remote system logging for many different servers and proxies. The design is based on the awarded[11] example in [20].

The IDS monitoring systems use a separate network to perform remote logging to the database. The other computers’ remote log systems use the regular production network.

The workstations are monitored by the Prelude Hybrid IDS. This involves remote logging to the manager that transfers the processed events to a database. This manager is also responsible for communicating with the manager placed inside the DMZ. This

[11] This design was awarded honors by the GCFW Advisory Board [33][33]

solution makes remote logging from a DMZ safer to use, as it is the manager inside the secured network that initializes the communication. The DMZ manager is not allowed to initialize any communication. The workstations are also preconfigured to use the syslog-ng [27] if required. We assume that there are 25 workstations connected to the network on SW6.

The proxy servers and the servers on SW5, as well as other servers, are configured to use the syslog-ng, which is logging to the log server on SW4.

All servers in DMZ have Prelude Hybrid IDS installed. This configuration logs to the Prelude manager in the DMZ, which in turn logs to the internal Prelude manager as mentioned before.

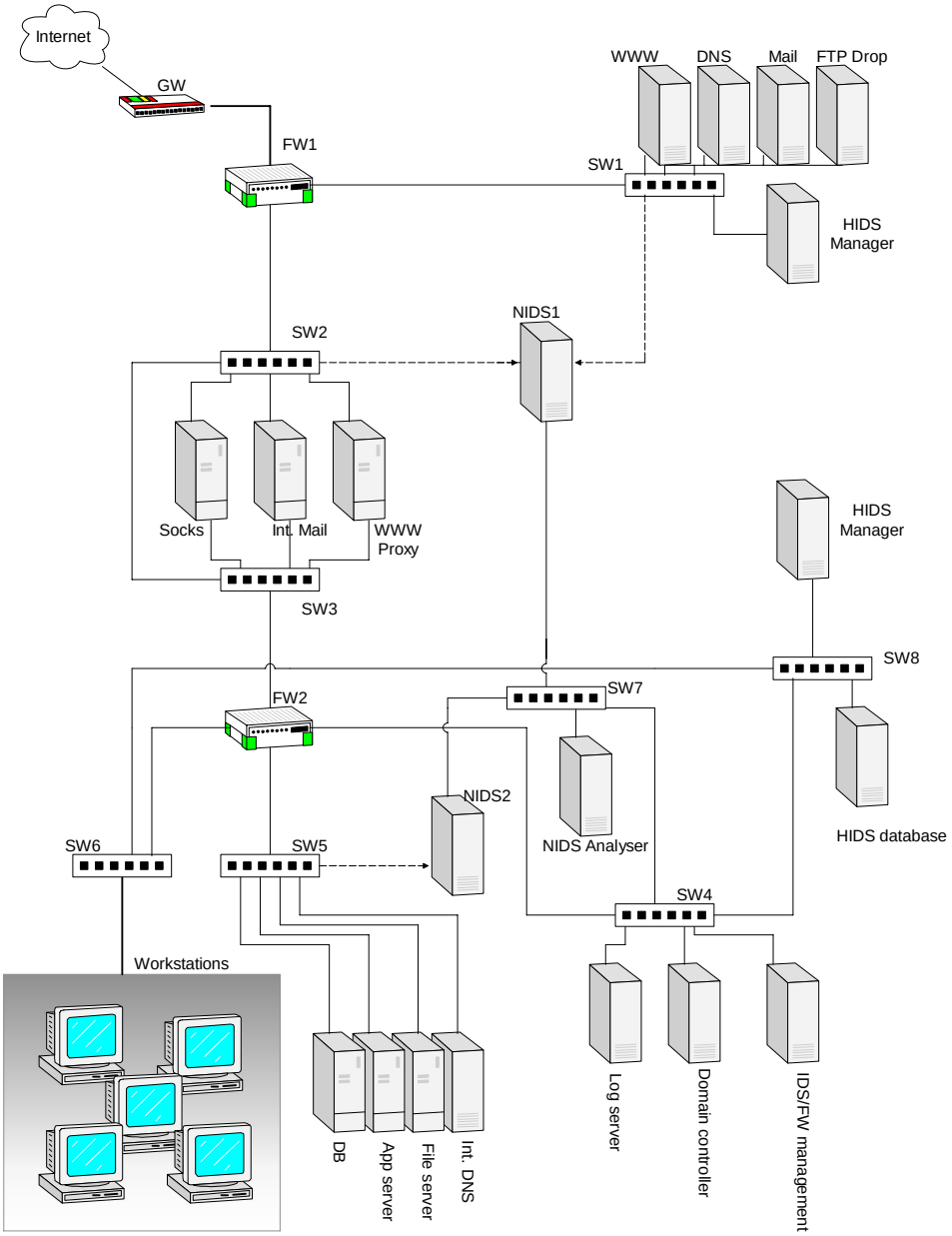


Figure 10: Network topology, configuration 3

The metrics results

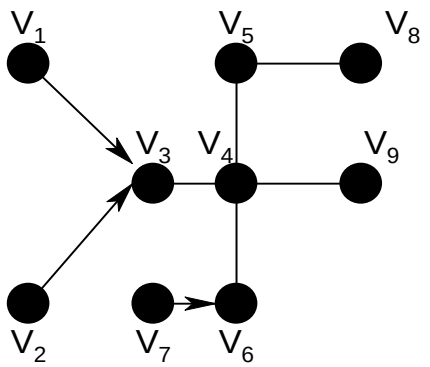
As we see in Table 13, the metrics' evaluated for configuration 3 gained a score of 30.2. This is the highest score for all the experiments.

Metric	Score	Max score	Difference
Assets List	11	11	0
Vulnerability scanning	5	5	0
Host security connectivity	7.2	8	0.8
Network security connectivity	7	7	0
TOTAL:	30.2	31	0.8

Table 13: Metric results, configuration 3

Security graphs

Because of the many security features in this configuration, we also identified just that many security graphs. We therefore decided not to join all graphs into one, but to present the graphs separately. Table 14 shows the graph for the NIDS feature.



V_1	SW1
V_2	SW2
V_3	NIDS1
V_4	SW7
V_5	SW4
V_6	NIDS2
V_7	SW5
V_8	IDS/FW management
V_9	NIDS Analyzer

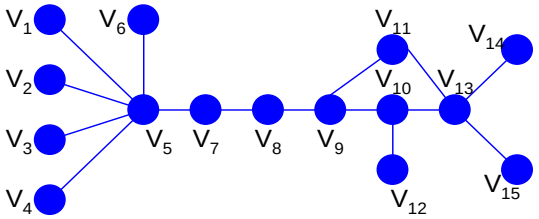
Table 14: NIDS graph, configuration 3

In Table 15 we show the vertex connectivity and other graph data for the NIDS graph.

The data from the NIDS graph ***G***
$V(G) = \{V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9\}$
$E(G) = \{(V_1, V_3), (V_2, V_3), (V_3, V_4), (V_4, V_5), (V_4, V_6), (V_4, V_9), (V_6, V_7), (V_5, V_8)\}$
Order ($\lvert V(G) \rvert$) = 9
Size ($\lvert E(G) \rvert$) = 8
$\kappa(G) = 1$ (V_3, V_4, V_6, V_5)
$\lambda(G) = 1$

Table 15: NIDS graph data, configuration 3

The next graph, showed in Table 16, represent the Prelude HIDS that is installed on several hosts.



V_1	WWW
V_2	DNS
V_3	Mail
V_4	FTP
V_5	FW1
V_6	HIDS Manager
V_7	SW2
V_8	SW3
V_9	FW2
V_{10}	SW6
V_{11}	SW4
V_{12}	Workstations
V_{13}	SW8
V_{14}	HIDS Manager
V_{15}	HIDS database

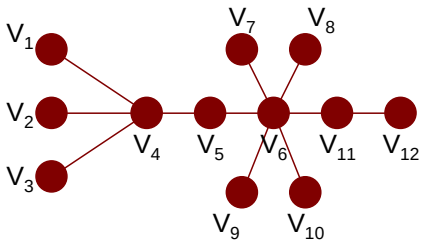
Table 16: HIDS graph, configuration 3

In Table 17 we show the vertex connectivity and other graph data for the HIDS graph.

The data from the Prelude HIDS graph ***H***
$V(H) = \{V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9, V_{10}, V_{11}, V_{12}, V_{13}, V_{14}, V_{15}\}$ $E(H) = \{(V_1, V_5), (V_2, V_5), (V_3, V_5), (V_4, V_5), (V_5, V_6),$ $(V_5, V_7), (V_7, V_8), (V_8, V_9), (V_8, V_9), (V_9, V_{10}), (V_9, V_{11}), (V_{10}, V_{12})$, $(V_{10}, V_{13}), (V_{11}, V_{13}), (V_{13}, V_{14}), (V_{13}, V_{15})\}$ Order ($\lvert V(H) \rvert$) = 15 Size ($\lvert E(H) \rvert$) = 13 $\kappa(H) = 1$ ($V_5, V_7, V_8, V_9, V_{10}, V_{13}$) $\lambda(H) = 1$

Table 17: HIDS graph data, configuration 3

As system logging is an important monitoring feature, we consider this as well. Table 18 shows the graph of the syslog-ng service.



V_1	SOCKS
V_2	Int.Mail
V_3	WWW Proxy
V_4	SW3
V_5	FW2
V_6	SW5
V_7	DB
V_8	App Server
V_9	File Server
V_{10}	Int. DNS
V_{11}	SW4
V_{12}	Log Server

Table 18: Syslog-ng graph, configuration 3

In Table 19 we show the vertex connectivity and other graph data for the syslog-ng graph.

The data from the Prelude HIDS graph ***I***
$V(I) = \{V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9, V_{10}, V_{11}, V_{12}\}$
$E(I) = \{(V_1, V_4), (V_2, V_4), (V_3, V_4), (V_4, V_5), (V_5, V_6),$ $(V_6, V_7), (V_6, V_8), (V_6, V_9), (V_6, V_{10}), (V_6, V_{11}), (V_{11}, V_{12}) \}$
$\text{Order} (\lvert V(I) \rvert) = 12$
$\text{Size} (\lvert E(I) \rvert) = 11$
$\kappa(I) = 1 \ (V_4, V_5, V_6, V_{11})$
$\lambda(I) = 1$

Table 19: Syslog-ng graph data, configuration 3

6.4 Experiment conclusions

From the experimental results we can see that the metrics perform in accordance with the increased number of host and network security features. This implies that we can use the metrics score to identify the security level of the implemented security features in the network.

We also see that for small networks the effect of gathering data to form a graph of the security features is somehow a trivial matter. By this we mean that the methods pay better off for more complex networks, as the security features are easily controlled in small networks. This was shown with the last configuration, where several security features were involved. Even if this network is not to be considered as a complex network, we can clearly see the benefits of the method for presenting the security features as graphs. This gives us an overview of every security feature in the network, and gives the user an opportunity to find mutual dependencies for different security features.

The key for developing correct security graphs is to perform well on the first two

metrics. These metrics make the baseline of the data the user should organize with the two last metrics. The metrics' scores for the three configurations are the following:

M1 = (9, 3, 2.1, 5)

M2 = (9, 3, 2.5, 5)

M3 = (11, 5, 7.2, 7)

The maximum score for the optimal configuration is M = (11, 5, 8.0, 7)

Visualized in Figure 11, we can see the difference between the first two configurations and the last one. Configuration number 4 is the highest possible score.

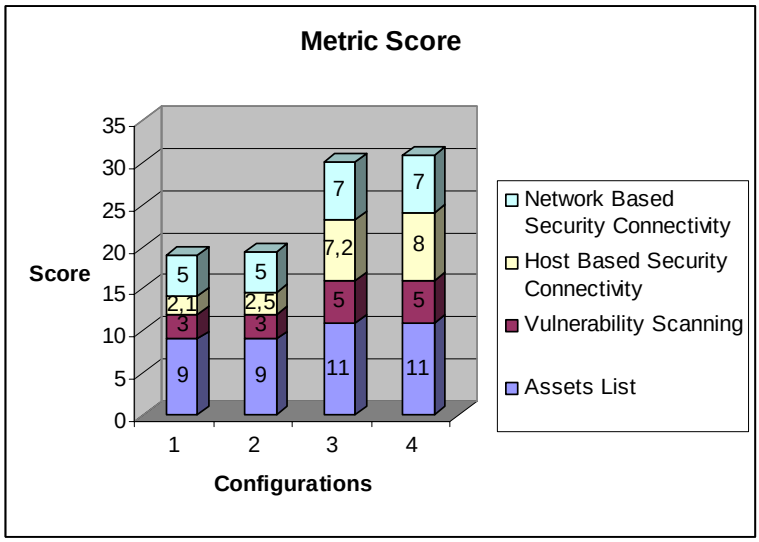


Figure 11: Metric score

This led us to the conclusion that the metrics are useful to make a plot of the security features, and the measured results can be expressed in four scales in the form M_{total} = (M1, M2, M3, M4), where the range for every metric is:

- M1 = [0, 11], integer only
- M2 = [0, 5], integer only
- M2 = [0, 8.0]
- M2 = [0, 7], integer only

The total score, M_{total}, from these metrics is an expression of the security level for the hosts and network. This is also an indicator of the system security level if one or several security features are not performing as they should.

6.5 Summary

In this chapter we have shown how we designed the experiments and how the metrics performed for different security solutions. The metrics seemed to perform as expected for the chosen solutions, where increased security features resulted in a higher total score.

7 The metrics and Patch Management

7.1 Introduction

This chapter describes the process of using the metrics, and suggests how this process can be integrated into a Patch Management strategy. We have divided the four metrics into two sections, where the first section involves the use of the two metrics' "Assets List" and "Vulnerability Scan". The results from this section are essential for the next section where the two remaining metrics should be used, and wrong or missing information from the first section may propagate through the next section and cause the outcome from the metrics to be misleading.

7.2 Processing the metrics

If the data from the "Assets List" and "Vulnerability Scan" do not correlate, this may be if e.g. there are registrations from the Vulnerability Scan that are not registered in the assets list, or vice versa, this must be dealt with. The control might involve updating the assets lists, or redoing the whole process of these two first metrics. Regardless of how the user correlates and corrects the data, the main goal is that the output data from the two metrics are correct, complete and consistent. The process is shown in Figure 12.

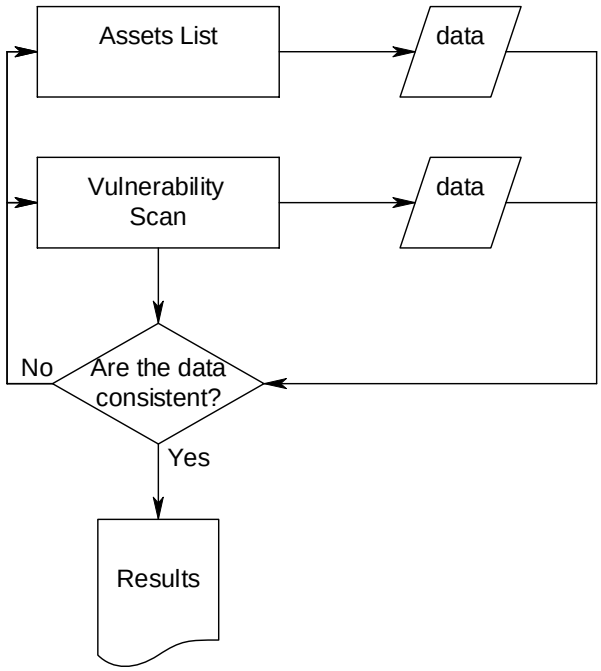


Figure 12: Part one of the metric process

The data output from the two last metrics are also subject to a correlation process, and if there are results that do not add up, this has to be investigated further. In the worst scenario, the user can decide to evaluate the metrics again. Figure 13 shows the

evaluation process for these metrics.

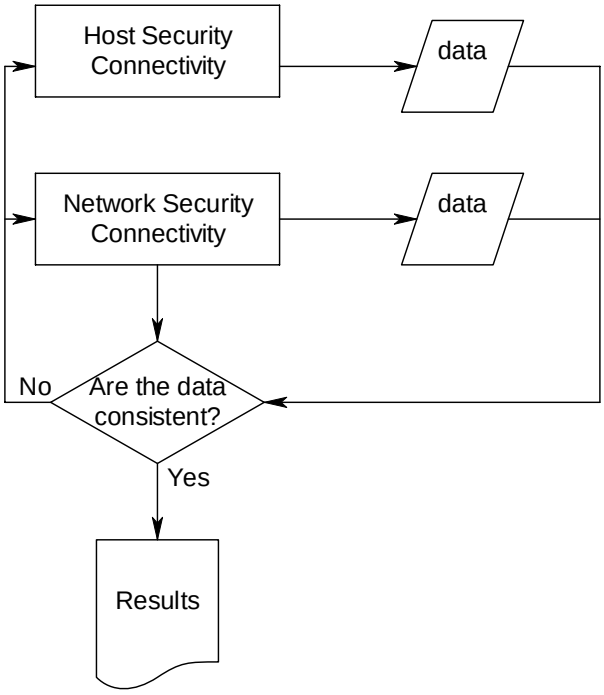


Figure 13: Part two of the metrics process

The results from this part are subject to a registration process, where the goal is to make a map of how the security related communication flows in the network. In turn, this map can help make up the decisions about several points in the Patch Management strategy. For example, a relevant decision is which computers to patch at which time, identifying which devices are safe to patch without loosing any of the security related services.

7.3 The Patch Management Process

There may be several ways of performing the Patch Management process, including the decision on which sub-processes are to be included in this process. These decisions are related to every organization's evaluation of their different needs and strategy, so it will not be possible for us to state a fixed Patch Management process that fits into every organization. Therefore, we present a Patch Management process that is based on the findings of best practice in Chapter 3, and the strategy proposed in Section 4.3. We shall then adapt our method for connectivity analysis to the proposed process.

We think that the model proposed by SUN Microsystems [13] makes the best basis for our understanding of which tasks will be a part of the Patch Management process. We therefore modify this model to fit our findings, and demonstrate how our metrics can be used in this process.

The process is shown in Figure 14, and the steps are organized in a logical sequence. As we see, the upper three steps are aligned on one axis. This is to point out that these

steps make up the basis for the action plan.

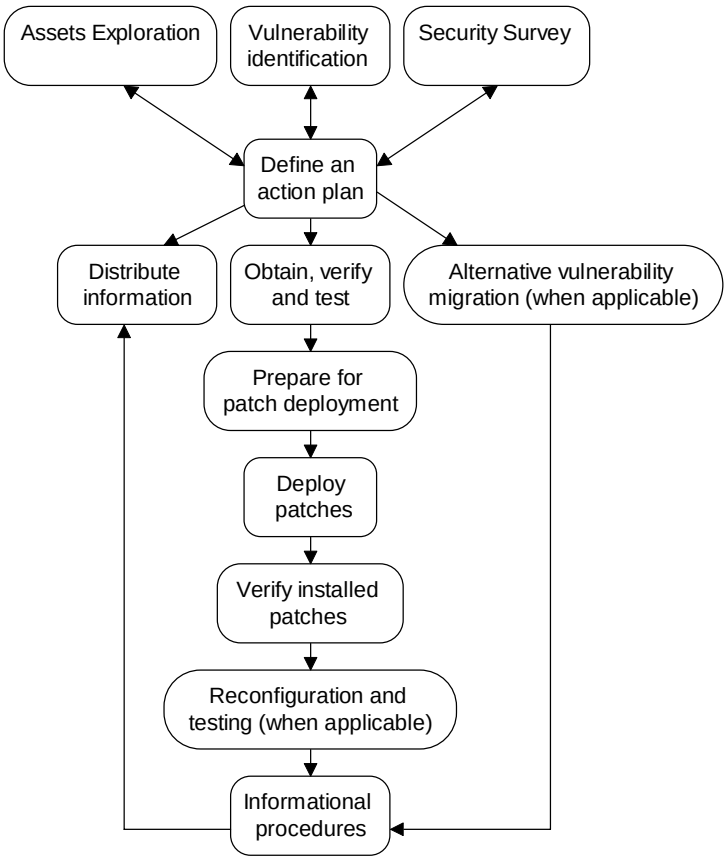


Figure 14: Patch process overview

The sequence follows a natural progress in the deployment process, and points out what step follows. But one aspect is missing in this diagram, and that is the Patch Management policy compliance check. The reason for leaving it out of the diagram is to make the diagram as simple as possible. That said, we stress the importance of checking that the actions performed in every step are in compliance with the policy. If the actions are in violence of some aspect in the policy, the consequences could be disastrous in terms of system security and stability.

7.4 The Patch Management process explained

7.4.1 Assets Exploration

As we have seen, some activities that scrutinize all of the organization's equipment related to the computer environment must be performed. It is also important to cover the software aspect of the environment, like operating systems and common applications; these are most often the target of the Patch Management. The information derived from this process is important to organize, to support quick availability and readability. It is also very important that this task is performed in a way that ensures the quality of the derived information. Wrong or missing information

from this task may result in wrong decisions/actions later in the process.

Our metric “Asset List” fits perfectly into this task, giving an indication to what to look for and how to think about this task. This report also suggests a register form, in which the derived information can be registered. But the most effective method to ensure quick and accessible information is to develop and use a database. The suggested register form, presented in Section 5.2.3, may be used as a template for this database.

7.4.2 Vulnerability identification

This would be the most important task of the Patch Management process. If the actual vulnerabilities are not discovered, the whole process is useless. This should be performed repeatedly, focusing on the discovery of new vulnerabilities. This involves more than making a sweep of the known systems with a vulnerability scanner, or managing the known patches that are released from the software vendors. It is also necessary to investigate if there are vulnerabilities that have been discovered but not released patches for. By doing this, the systems can be configured to isolate the vulnerability until the patches are made public. The discovery of new vulnerabilities produces vital information for the system administrators, so they know the indicators of an attack that exploits this vulnerability.

This information is usually found on the software vendors’ internet site, but it is also important to turn to additional sources such as mailing lists, third party security/vulnerability related internet sites/newsletters/mailing lists etc as mentioned in Section 4.1.2, under “Prioritizing”.

As for the discovery of vulnerabilities for the hosts in the network, it is necessary to use some kind of automatic tool. The most common method is to use one of the many security or vulnerability scanners available on the market. The results from these scanners are highly dependent on how they are used, and the improper use may cause wrong information. Our metric “Vulnerability Scanning” may support this part of the task.

7.4.3 Define an action plan

When the vulnerabilities are identified, it is necessary to analyze the influence of exposure. The primary outcome from this analysis is to decide if the vulnerability ought to be dealt with or not. A common example of this is if the system is highly dependent on availability and the vulnerability identified is not security/critical rated. The outcome may be not to jeopardize availability, and to leave the system un-patched to a further notice.

If the decision is to remove the vulnerability, the necessary patches should be identified and registered before they are applied. We propose a summary of some simple steps as a guide to this process.

- Patch identification

  This usually involves getting information from trusted software vendors or third party Internet sites. Information from untrusted Internet sites or mail should not be trusted; it has to be verified from trusted sources before it can be taken into account.

- Patch examination

When the necessary patches are identified, they need to be examined before they are obtained. The first step is to check if the patches are removing all identified vulnerabilities or not. Failing to remove all vulnerabilities will cause further exposure of the systems, and the whole process must be performed again. This will be time and money consuming, and may cause even more instability to the systems.

Next, it is important to check if the identified patches are in accordance with the requirements that are stated for the patches. This is usually stated in the Patch Management policy or in other strategic documents. If a patch does not meet the stated requirements, it must be taken into account, resulting in a decision on removing the vulnerability or not. How to execute such cases is situation dependent, but it must involve the approval from the superior technical management.

The next step is to make an examination of the consequences of deploying the patches. This involves a brief study of which systems and applications may be affected, the rank of importance of these systems, and how a possible failure may compromise the systems. By doing this, one can reveal if the deployment affects other services or makes business critical systems unstable.

- Prioritizing

After the consequences in the previous step are stated, and all patches are approved, a prioritizing of the patches can be performed. The prioritizing in this task is not to be mistaken with the prioritizing stated in the policy document. The patches are now prioritized according to the overall prioritizing in the policy, and the result should be stated as a list, and function as a guide for the order in which the patches are to be deployed.

- Obtaining the patches

These regulations are to ensure that the patches are obtained from trusted vendors. Other statements are only necessary if there are several sites or sources to obtain the needed patches from. Any configurations of the patching tools that are necessary to perform must be mentioned. The policy contains statements for which authentication requirements must be used when obtaining patches. A method for performing a control of this aspect, is e.g. to check the digital certificate and verify the PGP signatures.

- Verifying the patches

To verify that the downloaded files are not corrupted by any means, an integrity check of the downloaded patches must be performed. Many of the automated tools perform this check, but there should be alternative methods of doing this check. Manual verification of the cryptographic checksum, e.g. a Message Digest 5 checksum (MD5) [36], is one possibility, but this also involves a trusted site to gain the originals from. The patches should also be scanned by antivirus tools, and they must be stored at a secure location.

- Test plan

The patches should be tested in a similar environment before they are deployed in the real production environment. This is to reveal if there are any negative consequences in relation to the deployment or use of the patches or patched systems. The test procedures should be stated in the Patch Management policy, and the required test environment for carrying out the tests should already exist. The policy should also state how to deal with any abnormal results from the tests.

- Deployment plan

Finally the patches are examined, obtained and verified. The only thing left is to decide how to deploy them. This decision is up to the requirements stated in the prioritizing plan, but now we make statements of how to deploy (automated, manual, tools etc.) and when (time of day). In any case, there must be created a deployment plan that identifies the systems, gives a statement of which patches are to be deployed on that particular system, when to deploy them and what tools to use/how to perform the deployment. If there are some other requirements to the deployment, e.g. reboot or manual registry changes, it must be specified here as well.

If no patches are released, and the vulnerability must be dealt with, special actions are necessary; this is mentioned in Section 7.4.6.

7.4.4 Security Survey

Based on the information from the previous tasks, we can now examine if the vulnerability or deployment plan affects the implemented security features, and strengthen or weakens the security of the system. To be able to do this, we need a connectivity map for every security feature and their use of equipment, as explained in Section 4.5. This map will reveal the coverage and dependencies of the security features. It is not necessary to perform this task every time vulnerabilities are discovered, only the first time, and if there are changes in the network or to the configurations of the security features.

If the survey identifies any changes to be made, this must be stated in a report. This report must contain information that makes it possible to perform the necessary changes prior to the deployment of the patches.

This task can be supported by evaluation of the metrics described in Sections 5.4 and 5.5.

7.4.5 Obtain, verify and test

In the action plan, we have stated what to do in relation to obtaining and verifying the patches. Almost all Patch Management tools support this, and may contribute to an automated processing of these requirements. When the patches are ready to be deployed, they should be tested as stated in the test plan.

7.4.6 Alternative vulnerability migration (when applicable)

If there are no patches for the identified vulnerabilities, and the risks are considered as potentially high, some actions must be carried out to protect the systems. This will normally involve the process of removing the vulnerable service/system from the possible exposure from an attacker. But in most cases the service/system exists for a purpose, and that is the reason for not removing them. In these cases, where the decision is to expose the systems even if the possibilities to be attacked are high, it is crucial to implement a tight monitoring of the vulnerable systems.

This task must not be mistaken to be as "easy" as the patching deployment process, and there is also a much higher risk of failing to protect the systems by choosing an alternative procedure like this.

To implement monitoring of a system with a known vulnerability requires information about the vulnerability, the vulnerable system and the signatures of a possible attack. Information about the vulnerability is gained through the task "Vulnerability identification". If the information from this task is not satisfactory, the information must be gathered from other reliable sources. In many cases, the information will contribute to protecting the vulnerable systems, and how to configure the monitoring. Important aspects are at which ports the vulnerability is located, in which process or application the vulnerable service is, and if there are already identified automated tools or attacks that the vulnerability exposes to. This information is in many cases sufficient to get an idea of what kind of network traffic a possible attack might generate.

From the gathered information it is possible to produce a profile of a possible attack, and by this profile one can configure the security features to filter these anomalies. In addition, it has to be stated what to do if the security features are triggered. The simplest way of preventing the attack is to remove the attacked service from the network. But other possibilities are to be considered too, e.g. deceiving the attacker to leave traces of evidence for legal matters.

It is likely that the attacker will try to cover the attack, e.g. hiding the attack traffic in legal traffic. In such cases the security features may not detect the attack efficiently, and the attacker may succeed in compromising the system.

7.4.7 Distribute information

Before any patches are deployed, the information about the deployment should reach the users and owners of the affected systems. This information should contain detailed information about what vulnerabilities are to be patched, the consequences of exposure of these, the necessary steps to perform the deployment, when the deployment is scheduled, and what to do if there are any problems after the deployment. In addition to the staff that is affected by the deployment, this information should be distributed to the management and the proper system-, network-, and security administrators. Because this information contains data about vulnerabilities related to the systems that could be misused, the information should be marked as confidential and access should be restricted.

Several advantages can be achieved from distributing the information to all involved

parties. One is that the information about disruptions in the production environment is important, to avoid loss of data and availability. Also, if the users of the systems are aware of this they will understand what is going on. The next aspect is that if the users learn about the threats and exposure of the systems, the possibility for them to be more security aware is increasing. In turn, this might contribute to a more secure environment. A positive side effect is that this information is the verification that the security work is carried out, and that this work is made visible for the management.

7.4.8 Prepare for Patch Deployment

Before the deployment is performed, it might be necessary to perform some changes to ensure stability and security. If some alternative configuration to the security features in the step “Security survey” were identified, these may be executed now. There is also the possibility that the Patch Management tools require some additional configuration. This may be the case if it is necessary to switch between automatic/manual deployments of the patches. But before any of these changes are carried out, the most important thing is to perform a backup of the affected systems. This involves verification that there exists a reliable and correct backup of the systems that are about to be patched or changed. This step is important because this creates the last possibility to return the affected systems into the prior stable state.

7.4.9 Deploy Patches

Finally the deployments are to be performed. This procedure would involve the necessary steps needed to deploy the patches to the systems correctly and in the correct order. It is important to follow the directions in the deployment plan, and not to take any shortcuts. A deviation from this plan may cause cancellation of the functionality of the patches and instability of the systems.

7.4.10 Verify Installed Patches

When the patches are installed and all the systems seem to be stable and in working order, it is necessary to verify that the patches were installed and that the identified vulnerabilities were removed. The easiest way to control this is to perform a vulnerability scan like the one in the step “Vulnerability identification”, by using a vulnerability scanner with the formerly identified vulnerabilities in mind. But these verifying procedures should support the possibility to identify which patches are installed on the system, because even if the tool does not report that a vulnerability is present, it is not a proof that the patch was installed properly and that the vulnerability is removed. Many of the network based vulnerability scanners do comply with this requirement, as they are just capable of checking the application version and patch level. But the same may hold for the agent based security scanners, so the user must be aware of the pitfalls in this procedure.

Regardless of which technology is used, the important thing is to verify that the patch is properly installed, the vulnerability is removed and that no new vulnerabilities are introduced.

7.4.11 Reconfiguration and testing (when applicable)

If changes have been carried out to some parts of the systems prior to the deployment, these changes should be reset into the original configuration.

The last step in this procedure is to verify that the systems operate as intended. This usually involves verifying that the most used applications work properly, and that no anomalies are detected. This can be a time consuming task, especially if many systems were patched. A simple solution is to make the users of these systems report any detected misbehavior, but this has its downsides in that the users are not trained in what to look for and that this may cause failures when detecting the anomalies.

7.4.12 Informational procedures

The last part in all the steps in this patch deployment procedure is to ensure that the information from the previous steps is gathered, processed, reported and stored. The gathered information should be verified to ensure that all data are correct. This is important for later use if a similar situation arises. Furthermore, all data should be stored in a database, and reports should be generated as stated in the Patch Management policy. The last step to carry out for the deployment process is to make sure that final information about the deployment is given to the proper management and staff.

7.5 Discussion

The Patch Management process is highly dependent on a patch management policy that states all the right decisions for the environment and how to patch the systems. This may be very hard to comply with, as the nature of the dynamic environment makes the policy an ever changing dynamic document. The consequences of an incorrect policy may lead the vulnerability related work to fail, and the system may be exposed without anybody's knowledge. It may also cause the deployment of patches to cause instability in the systems, and decrease the security features.

The steps suggested in this chapter are based on the findings in the literature study. It is assumed that the policy supports all the right decisions that should be supported, and we make no suggestions on how the policy related statements should be. This is a very time and resource saving move, and it makes the process easier to develop. One of the consequences of this is that the reliability of the process may suffer from not being based on a environment policy, and some of the statements within the steps may, because of this, not be realistic. But we have based our work on the statements in several publicly accepted and reviewed documents, mentioned in Chapter 3, and by doing this, we assume that our statements might be used in a live environment.

Another aspect we do not describe here is how to join our procedures with an existing procedure. Our suggestion assumes that no established Patch Management process exists before implementing our process or that the original process is turned down. But the steps in the process are described in as general manner as possible to support the possibility of joining an existing process into the process described here. By doing this, the suggested descriptions will also apply as a guide to evaluating an existing process, and support the joining of another patch management strategy steps into achieving a better performance for deploying patches.

7.6 Summary

In this chapter we have described how the metrics are related to each other, and we

also describe a suggestion of how to apply them. The explained sequence is important to follow, as the results from the steps may depend on the previous step.

In the Sections 7.3 and 7.4, we describe a Patch Management process, and relate our metrics to this process. The steps in this process are also focusing on the vulnerabilities and information about the vulnerabilities, in addition to acting as a guide to the deployment of patches.

The discussion part is targeted at the importance of a valid policy to act as a general guidance, and how our Patch Management proposal can be joined to an existing Patch Management process.

8 Conclusion and further work

8.1 Conclusion

The study of the standards and public documents showed that Patch Management procedures do not automatically comply with the ideal thought of ensuring security no matter what activity is performed in the network. There may be several reasons for this, such as if this is subject for another management department, e.g. the security strategy documents cover this topic or whatever requirements that are related to security or special configurations. The absence of security instructions in the Patch Management procedures may cause actions that decrease the security when performing deployment of patches. This gives the need for security related instructions and guidance to be stated in the Patch Management strategy/procedures, and they must be stated as clearly as possible to ensure efficiency and compliance with the instruction's goals. A situation like this is likely to affect the security in a positive way.

This thesis shows how to improve the security related aspects of Patch Management by adding an additional security related step into the Patch Management procedure, and focusing on vulnerabilities as well as the patches and deployment.

The metrics we developed were designed to support stages in the Patch Management process, but they are not directly related to the deployment of the patches. The metrics are an administrative support for the preparing of the patch deployment. This involves gathering information about all assets in the computer environment, and systematizing this data.

The experiment showed that the metrics measure how the systems implement different security features, and respond with higher score when the systems have a better basis for security. The procedure for developing the security connectivity mapping stands as the most original suggestion in this work. The experiments showed that there are several positive consequences of the method, as we might easily map the different security features and see their relationship to each other, and which appliances may result in a disconnection with most impact on security. But the experiments also showed that the method might have best effect on large networks, as the implemented security features tend to be more complex. Simpler networks with a smaller number of security features are easier to control, and there is a possibility that the proposed Patch Management procedures are just overhead actions. But these networks will probably need a Patch Management strategy anyway, although they may need a more compressed strategy, so the proposed strategy might stand as a good starting point for these purposes.

8.2 Further work

We only tested the metrics in a controlled environment, and the results may have been under the influence of ourselves. The results showed that the increase of security leads to a higher score, but the metrics should have been tested in an organization where we had no influence on the results. This would remove the uncertainty of how the proposed Patch Management procedure affects the security of the systems of patch deployment, and we could have had a stronger conclusion about how the performance

of the procedures could act as a guide for implementing a security focused Patch Management strategy.

Further, the whole Patch Management strategy should be tested in a live environment to check for compliance with the theoretically claimed improvements. Some of the difficulties with performing tests like these are that the results from the new strategy are somehow not comparable with other strategies unless they are exposed to the same vulnerabilities and the same attacks. We can even state that human differences (system operators or administrators) may compromise the reliability of the comparison. This makes the test procedures more difficult to implement, and many tests probably must be performed over time, to be able to generalize the results.

The procedures should be supported by a tool that guides the operator through every single step, and that collects the correct data into a database for Patch Management and vulnerability information improvement. The tool/tools should be possible to configure as the Patch Management policy statements differ for various organizations.

References

[1]. L.Geppert, Lost radio contact leaves pilots on their own, IEEE Spectrum November 2004, URL: http://21405.gel.ulaval.ca/references/spectrumNov2004-RadioContact.pdf, visited 25.1.2005.

[2]. The Information Security Forum (ISF), The Standard of Good Practice for Information Security, ISF, version 4, 2003, side 135. URL: http://www.isfsecuritystandard.com/index_ie.htm, visited 17.1.2005.

[3]. Code of Practice for Information Security Managment, ISO/IEC 17799:2000, Int'l Organisation for Standardization, Geneva; Dec. 2000.

[4]. Information Technology Security Evaluation Criteria, June 1991, F/GB/D/ NL, URL: http://www.ssi.gouv.fr/site_documents/ITSEC/ITSEC-uk.pdf, visited 19.1.2005.

[5]. P.Mell, M.C.Tracy, Procedures for Handling Security Patches - NIST Special Publication 800-40, Draft 3.0 August 2002, URL: http://csrc.nist.gov/publications/nistpubs/800-40/sp800-40.pdf, visited 23.1.2005.

[6]. Kevin Dunn, Automatic update risks: can patching let a hacker in?, Network Security, Volume 2004, Issue 7, (July 2004) page 5-8, URL: http://ics.stic.gov.tw/Redirect/?f=110&i=8196, visited 1.2.2005

[7]. Patch Management Using Microsoft Software Update, URL:http://www.microsoft.com/technet/itsolutions/techguide/msm/swdist/pmsus/pmsus251.mspx , visited 1.2.2005

[8]. F. Harary, Graph Theory, Addison-Wesley, Reading, Massachusetts, 1969.

[9]. F. Buckley, M. Lewinter, A Friendly Introduction to Graph Theory, Prentice Hall, 2003.

[10]. S.C. Payne, A Guide to Security Metrics, 11. Juli 2001, URL:http://www.sans.org/rr/whitepapers/auditing/55.php, visited 12.1.2005

[11]. Kenneth J. MacLeod, Patch Management and the Need for Metrics, URL:http://www.sans.org/rr/whitepapers/bestprac/1461.php, visited 12.1.2005

[12]. M.Swanson, N. Bartol, J. Sabato, J. Hash, L. Graffo, Security Metrics Guide for Information Technology Systems - NIST Special Publication 800-55. URL: http://csrc.nist.gov/publications/nistpubs/800-55/sp800-55.pdf, visited 1.2.2005

[13]. SUN microsystems, Solaris Patch Management: Recommended Strategy, URL: http://docs-pdf.sun.com/817-0574-12/817-0574-12.pdf, visited 2.2.2005

[14]. CERT/CC Statistics 1988-2004, URL: http://www.cert.org/stats/cert_stats.html, visited 2.2.2005

[15]. Bruce Schneier, Closing the window of exposure: Reflections on the Future of Security, 2000, URL: http://online.securityfocus.com/guest/3384, visited 2.2.2005

[16]. Dr. E. Cole, Dr. R. Krutz and J. W. Conley, Network Security Bible, Wiley Publishing, Inc, 2004.

[17]. Microsoft, Description of the standard terminology that is used to describe Microsoft software updates, URL: http://support.microsoft.com/default.aspx?kbid=824684, visited 17.02.2005

[18]. T. Grance, K. Kent, B. Kim, Computer Security Incident Handling Guide - NIST Special Publication 800-61, January 2004, URL: http://csrc.nist.gov/publications/nistpubs/800-61/sp800-61.pdf, visited 23.1.2005.

[19]. James A. Mc.Hugh, Algorithmic Graph Theory, Prentice Hall, 1990.

[20]. S. Northcutt et.al., Inside Network Perimeter Security, New Riders, 2003

[21]. J. Wack, M. Tracy, M. Souppaya, Guideline on Network Security Testing - NIST Special Publication 800-42, October 2003, URL: http://csrc.nist.gov/publications/nistpubs/800-42/NIST-SP800-42.pdf, visited visited 23.2.2005.

[22]. C. McNab, Network Security Assessment, O'Reilly, March 2004

[23]. Nessus Open Source Vulnerability Scanner Project, URL: http://www.nessus.org/, visited 23.02.05

[24]. J.C. Creswell, Research Design, Qualitative, Quantitative, and Mixed Methods Approaches, Second Edition, SAGE Publications, July 2002

[25]. P. Herzog, OSSTM 2.1 Open Source Security Testing Methodology manual, The institute for Security and Open Methodology, ISECOM, 2003

[26]. The Hybrid IDS Framework, URL: http://www.prelude-ids.org/, visited 23.2.05

[27]. syslog-ng© system logger, URL: http://www.balabit.com/products/syslog_ng/, visited 23.02.05

[28]. Analysis Console for Intrusion Databases (ACID) Website, URL: http://www.andrew.cmu.edu/user/rdanyliw/snort/snortacid.html, visited 23.02.05

[29]. Prelude Manager – Prelude Hybrid IDS, URL: https://trac.prelude-ids.org/wiki/PreludeManager, visited 23.02.05

[30]. Request for Comments #3164 - The BSD Syslog Protocol, URL: http://www.faqs.org/rfcs/rfc3164.html, visited 23.02.05

[31]. R. B. Vaughn, R. Henning, A. Siraj, Information Assurance Measures and Metrics - State of Practice and Proposed Taxonomy, Proceedings of the 36th Hawaii International Conference on System Sciences, IEEE, URL: http://csdl.computer.org/comp/proceedings/hicss/2003/1874/09/187490331c.pdf, visited 23.020.5

[32]. M.Bishop, Computer Security: Art and Science, Addison Wesley Professional, 2003.

[33]. GCFW Advisory Board, URL: http://www.giac.org/certified/boards/gcfw.php, visited 23.02.05

[34]. B. Hill, Cisco: The Complete Reference, McGraw-Hill/Osborne, 2002

[35]. Snort Intrusion Detection System, URL: http://www.snort.org/, visited 2.2.2005

[36]. RSA Laboratories, What are MD2, MD4, and MD5?, URL: http://www.rsasecurity.com/rsalabs/node.asp?id=2253, visited 2.2.2005

Appendix A: Experiment results

1.1. Configuration 1

This configuration is described in Section 6.3.1.

1.1.1. Metric 1: Assets list

The data gathered when using the metric form are presented in this section. We use the form from Section 5.2.3 for the presentation. Some of the fields in the form are not accounted for, this is mainly because this task was very time consuming, and gathering all the details was not regarded as vital for the experiments.

Identification information	Web server 1
Short overall description	Linux web server

Operating system

OS	Fedora Core 3
Version number:	kernel 2.6.9-1.667
Patch level:	
Configuration:	

Software

Product	**version number**	**patch level**	**configuration**
Apache web server	2.0.52 (fedora)		default
OpenSSH	3.9p1		default
Iptables	1.2.11		Rule set added for ssh, http and https

Hardware and firmware

description	**Product/vendor**	**drivers**	**serial number**	**Other information**
OEM computer	Packard Bell PB Club300xe			
CPU	Cyrix 300 MHz			
RAM	196 MB SDRAM			
Hard disk	IBM 12 GB			
Network adapter	3COM	(fedora)	MAC: 00-10-4B-62-13-BE	

Services

Service	port	protocol	Other information
SSH	22	SSH1/SSH2	
Web server	80	http	
Web server	443	https	SSLv2

Location

Physical location	Logical location	DHCP	IP/IP-range
	DMZ on Dlink Router	no	192.168.0.108

Security dependencies and relationship to other computers or appliances.	n/a

A thorough description

System name	www-FC3
Network alias	
Role	web server
Main users	Administrators
Connections	Http/https from Internet allowed. SSH from internal network allowed.
Patch history	
Patchrelease sites	http://download.fedora.redhat.com/pub/fedora/linux/core/updates/3/i386/

Identification information	Network IDS computer 1
Short overall description	Snort sensor on Linux

Operating system

OS	Fedora Core 3
Version number:	kernel 2.6.9-1.667
Patch level:	
Configuration:	

Software

Product	**version number**	**patch level**	**configuration**
OpenSSH	3.9p1		default
iptables	1.2.11		Rule set added for ssh
Snort IDS	2.3.2		Logging to MySQL database

Hardware and firmware

description	**Product/vendor**	**drivers**	**serial number**	**Other information**
CPU	1200 MHz			
RAM	196 MB			
Hard disk	Maxtor 8 GB			
Network adapters	1. Unex 2. Unex 3. Micro-star		MAC: **1:** 00-10-A7-1E-33-B5 **2:** 00-11-2D-13-93-CB **3:** 00-10-DC-99-7B-FF	 Promiscuous mode Promiscuous mode Management

Services

Service	**port**	**protocol**	**Other information**

SSH	22	SSH1/SSH2	

Location

Physical location	Logical location	DHCP	IP/IP-range
	Connected to SW3	no	192.168.0.122

Security dependencies and relationship to other computers or appliances.	The snort sensors are logging to MySQL database on 192.168.0.110

A thorough description

System name	
Network alias	
Role	Network IDS, 2 sensors
Main users	Administrators
Connections	SSH from management segment allowed
Patch history	
Patch release sites	http://download.fedora.redhat.com/pub/fedora/linux/core/updates/3/i386/

Identification information	DC1
Short overall description	Domain Controller

Operating system

OS	Suse Linux Enterprise Server 9
Version number:	2.6.5-7.97-default
Patch level:	
Configuration:	Domain Controller

Software

Product	version number	patch level	configuration
OpenSSH	3.8p1		default

Hardware and firmware

description	Product/vendor	drivers	serial number	Other information
CPU	1300 MHz			
RAM	768 MB			
Hard disk	IBM 25 GB			
Network adapter	Sunrich		MAC: 00-0A-CD-06-85-81	

Services

Service	port	protocol	Other information
SSH	22	SSH1/SSH2	
Portmap	111	RPC	
LDAP server	389	LDAP	
VNC server	5900		

Location

Physical location	Logical location	DHCP	IP/IP-range
	Connected to SW3	no	192.168.0.126

Security dependencies and relationship to other computers or appliances.	

A thorough description

System name	SLES
Network alias	
Role	Domain Controller
Main users	Administrators
Connections	
Patch history	
Patch release sites	http://www.novell.com/de-de/linux/suse/

Identification information	DB1
Short overall description	Internal MySQL database

Operating system

OS	Fedora Core 3
Version number:	2.6.10-1.770_FC3
Patch level:	
Configuration:	

Software

Product	version number	patch level	configuration
Apache web server	2.0.52 (fedora)		default
MySQL	3.23.58		Database for several snort sensors
OpenSSH	3.9p1		default
iptables	1.2.11		Rule set added for ssh, http, https and MySQL

Hardware and firmware

description	Product/vendor	drivers	serial number	Other information
CPU	Intel Celeron 200 MHz			
RAM	128 MB			
HD	IBM 12 GB			
Network adapter	Intel		MAC: 00-D0-B7-5A-95-2D	

Services

Service	port	protocol	Other information
SSH	22	SSH1/SSH2	
Web server	80	http	
Web server	443	https	SSLv2
MySQL	3306	mysql	

Location

Physical location	Logical location	DHCP	IP/IP-range
	SW2	no	192.168.0.110

Security dependencies and relationship to other computers or appliances.	IDS sensor 1, 2 and 3 log to this database

A thorough description

System name	DB
Network alias	
Role	Internal database for several IDS
Main users	Administrators
Connections	SSH from internal addresses

	HTTP from internal addresses SQL queries from internal addresses
Patch history	
Patch release sites	

Identification information	SuseWS
Short overall description	Suse Linux 9.2

Operating system

OS	Suse Linux 9.2
Version number:	2.6.8-24.11-default
Patch level:	
Configuration:	

Software

Product	version number	patch level	configuration
OpenSSH	3.8p1		Default
VNC	3.3		
Nessus	2.2.2		Default

Hardware and firmware

description	Product/vendor	drivers	serial number	Other information
CPU	1300 MHz			
RAM	512 MB			
Hard disk	IBM 40 GB			
Network adapter	D-Link		MAC: 00-05-5D-E6-18-01	

Services

Service	port	protocol	Other information
SSH server	22	SSH1/SSH2	
Portmap	111	RPC	
Nessus daemon	1241	TLSv1	Password authentication
VNC server	5900		

Location

Physical location	Logical location	DHCP	IP/IP-range
	Connected to SW3	yes	192.168.0.0/24

Security dependencies and relationship to other computers or appliances.	Computers on the internal network may use the nessus-server for scanning purposes.

A thorough description

System name	suseWS
Network alias	
Role	Management computer
Main users	Administrators
Connections	SSH from internal network allowed VNC from internal network allowed
Patch history	
Patch release sites	http://www.novell.com/de-de/linux/download/updates/

Identification information	WinWS
Short overall description	Windows XP workstation

Operating system

OS	Microsoft Windows
Version number:	XP SP2
Patch level:	SP2
Configuration:	

Software

Product	**version number**	**patch level**	**configuration**
Zonealarm	5.0		

Hardware and firmware

description	**Product/vendor**	**drivers**	**serial number**	**Other information**
CPU	AMD XP 2400+			
RAM	1 GB			
Hard disk	Maxtor 150 GB			
Network adapter	Nvidia Marvell		MAC: 00-11-2F-25-D8-4F 00-11-2F-25-CB-E2	

Services

Service	**port**	**protocol**	**Other information**

Location

Physical location	**Logical location**	**DHCP**	**IP/IP-range**
	Connected to SW2	yes	192.168.0.0/24

Security dependencies and relationship to other computers or appliances.	

A thorough description

System name	
Network alias	
Role	Workstation
Main users	
Connections	
Patch history	
Patch release sites	

The score for this metric is shown in the following table.

Assets List	**Yes**	**No**
1. Are all computers, appliances and networked devices described in the list?		**X**
2. For every computer or appliance, do the list describe the following: 2.1. Unique identification information	**X**	
2.2. Short overall description	**X**	
2.3. Operating system; version number, patch level and configuration	**X**	
2.4. Software; version number, patch level, special configuration	**X**	
2.5. Hardware and firmware; product vendor, serial number, drivers and description	**X**	
2.6. services and port numbers	**X**	
2.7. Each computer or components location, both logically and physically, IP-address/IP-range	**X**	
2.8. Security dependencies and relationship to other computers or appliances. (E.g. IDS sensor/database)	**X**	
2.9. Thoroughgoing description; system name, network alias, role in the network/organization, patch history, allowed/forbidden connections, main users, main vendors patch release site/connections.		**X**
3. Are the list correlated with network scanning procedures or other techniques for revealing networked nodes?	**X**	

As we see there are many details to gather for every computer and appliances. The score from the metric form managed to score the total of 9 out of 11.

1.1.2. Metric 2: Vulnerability scanning

For this metric, we are not able to answer all the questions, as we are not an organization with predefined strategy and procedures. For those questions we left blank we will not take this in account when computing the score. This means that this metric has a total score of 5, instead of the original 9. We got a total of 3 out of 5 points, obviously not an optimal score.

Vulnerability scanning	Yes	No
1. Is vulnerability check of all the systems in the network performed?	**X**	
2. Are these scans performed at every host/computer periodically?		
3. Are network based vulnerability scanners used?	**X**	
4. Are host based vulnerability scanners used?		**X**
5. Are vulnerability scanners from at least two different vendors used?		**X**
6. Are the rules/signatures/plug-ins for the vulnerability scanner updated before use?	**X**	
7. Are the vulnerabilities discovered in the scan added to the list of vulnerabilities to be patched?		
8. Are these scans targeted at hosts after the installation of patches to confirm vulnerability removal?		
9. Are manual actions and reports for detected vulnerabilities performed if no patch is available?		

1.1.3. Metric 3: Host Based Security Connectivity

This metric has a range from 0 to 8 points, but full score is not possible for us to achieve with the equipment we use. One reason for this is that the switches are not managed, and it is not possible for them to perform any logging or alternative configuration. We do not change the range of this metric in spite of the limitations of the equipment.

This formula is used for computing the score is the following:

S = (2/1) + (3/1) + (4/1) + (5/1) + 6 + 7 + (8/1) + 9, where the 6, 7 and 9 are either 0 or 1 point.

For this configuration we scored: (8/11) + (2/11) + (1/11) + (0/11) + 0 + 0 + 0 + 1 = 2.09 ≈ 2.1

Host based security connectivity	#	Yes	No
1. How many hosts/appliances are connected to the network?	**11**		
2. How many networked hosts have enabled any kind of host security features? (e.g. software firewall, IDS and so on)	**8**		
3. How many networked hosts have enabled any kind of logging features? (e.g. system logging, logging from software firewall, IDS and so on)	**3**		
4. How many of the hosts security features log to other appliances other than to themselves? (e.g. log servers or a database)	**1**		
5. How many of the systems log features log to other appliances other than themselves? (e.g. log servers or a database)	**0**		
6. Are the log servers/DB available from several segments of the network?			**X**
7. Is it possible for the hosts to log to an alternative secure source?			**X**
8. How many hosts have an alternative configuration to ensure security for a short period in time?	**0**		
9. Are the hosts that are part of a distributed security or management system identified to form a model of reference to be used by the system/network administrators?		**X**	

1.1.4. Metric 4: Network Based Security Connectivity

This metric has originally a range from 0 to 8, in integers only. But for our use we can just score 7 points as one question is not possible for us to answer. We adjust the range for this metric.

The formula is 1 for YES, 0 for NO, leaving a score on 5 out of 7.

Network security connectivity	Yes	No
1. Does the network include network monitoring features or other network related security appliances? (E.g. NIDS, firewall etc.)	**X**	
2. Are the network security appliances covering all segments of the network?	**X**	
3. Do any of the network security appliances overlap each	**X**	

other?		
4. Do the network security appliances perform centralized logging? (E.g. log servers or log database)	**X**	
5. Are the log servers/DB available from several segments of the network?		**X**
6. Are the computers and servers related to security features separated from the production network?		**X**
7. Are the "cut nodes" identified for the network security features?	**X**	
8. Is special attention given to the network activity when performing shutdown or separation of security related "cut node" computers or appliances?		

1.1.5. The metrics results

The following table sums up the metrics score for configuration 1.

Metric	Score	Max score	Difference
Assets List	9	11	2
Vulnerability scanning	3	5	2
Host security connectivity	2.1	8	5.9
Network security connectivity	5	7	2
TOTAL:	19.1	31	11.9

1.2. Configuration 2

This configuration is described in Section 6.3.2.

1.2.1. Metric 1: Assets list

The forms are skipped for this part due to the small number of changes and the many pages these forms consumed. One of the two main changes is that the SUSE WS has implemented Prelude IDS, using the Prelude Manager to collect the sensor data. This computer also has MySQL database installed for the managing part. The other change is that the workstation has installed the Prelude IDS system, and functions as a sensor. Otherwise the configurations are the same.

The metric form will be the same as for the previous experiment, and we are leaving it out.

The score from the metric form managed to score the total of 9 out of 11.

1.2.2. Metric 2: Vulnerability scanning

This metric is also identical to experiment configuration 1, so there is no point of repeating it.

We got a total of 3 out of 5 points, obviously not an optimal score.

1.2.3. Metric 3: Host Based Security Connectivity

This metric has a range from 0 to 8 points, but full score is not possible for us to achieve with the equipment we use. One reason for this is that the switches are not managed, and it is not possible for them to perform any logging or alternative configuration. We do not change the range in spite of the limitations of the equipment.

This formula is used for computing the score:

S = (2/1) + (3/1) + (4/1) + (5/1) + 6 + 7 + (8/1) + 9, where the 6, 7 and 9 are either 0 or 1 point.

For this configuration we scored: (8/11) + (4/11) + (2/11) + (1/11) + 0 + 0 + 0 + 1 = 2.45 ≈ 2.5

The score is not that far from the previous score (2.1). This may look as if the effect of implementing host based security feature does not have much influence. But, this is a consequence of not having the logical number of workstations as an organization normally has. This causes the score to increase by only 1/11 ≈ 0.1 points. If for example the number of workstations were 80% of the total devices registered it would have increased the score by 0.8 points.

Host based security connectivity	#	Yes	No
1. How many hosts/appliances are connected to the network?	**11**		
2. How many networked hosts have enabled any kind of host security features? (e.g. software firewall, IDS and so on)	**8**		
3. How many networked hosts have enabled any kind of logging features? (e.g. system logging, logging from software firewall, IDS and so on)	**4**		
4. How many of the host's security features log to other appliances other than to themselves? (e.g. log servers or database)	**2**		
5. How many of the systems log features log to other appliances other than themselves? (e.g. log servers or database)	**2**		
6. Are the log servers/DB available from several segments of the network?			**X**
7. Is it possible for the hosts to log to an alternative secure source?			**X**
8. How many hosts have an alternative configuration to ensure security for a short period in time?	**0**		
9. Are the hosts that are part of a distributed security or			

management system identified to form a model of reference to be used by the system/network administrators?		**X**	

1.2.4. Metric 4: Network Based Security Connectivity

This metric has originally a range from 0 to 8, in integers only. But for our use we can just score 7 points as one question is not possible for us to answer. We adjust the range for this metric.

The formula is 1 for YES, 0 for NO, leaving a score on 5 out of 7.

Network security connectivity	**Yes**	**No**
1. Does the network include network monitoring features or other network related security appliances? (e.g. NIDS, firewall etc.)	**X**	
2. Are the network security appliances covering all segments of the network?	**X**	
3. Do any of the network security appliances overlap each other?	**X**	
4. Do the network security appliances perform centralized logging? (e.g. log servers or log database)	**X**	
5. Are the log servers/DB available from several segments of the network?		**X**
6. Are the computers and servers related to security features separated from the production network?		**X**
7. Are the "cut nodes" identified for the network security features?	**X**	
8. Is special attention given to the network activity when performing shutdown or separation of security related "cut node" computers or appliances?		

1.2.5. The metrics results

The following table sums up the metrics score for configuration 2.

Metric	**Score**	**Max score**	**Difference**
Assets List	9	11	2
Vulnerability scanning	3	5	2
Host security connectivity	2.5	8	5.5
Network security connectivity	5	7	2
TOTAL:	19.5	31	11.5

1.3. Configuration 3

This configuration is described in Section 6.3.3.

1.3.1. Metric 1: Assets list

This topology has many computers and appliances. It would be too space consuming to add all assets lists into this report. We assume the forms are in place for the rest of this process, and that they are correct.

1.3.2. The metric form

We assume that the metric form is checked as it is for the bets interest of the users. We also assume the maximum score, meaning 11 out of 11 possible points.

Assets List	Yes	No
1. Are all computers, appliances and networked devices described in the list?	X	
2. For every computer or appliance, does the list describe the following: 2.1. Unique identification information	X	
2.2. Short overall description	X	
2.3. Operating system; version number, patch level and configuration	X	
2.4. Software; version number, patch level, special configuration	X	
2.5. Hardware and firmware; product vendor, serial number, drivers and description	X	
2.6. services and port numbers	X	
2.7. Each computer or component's location, both logical and physical, IP-address/IP-range	X	
2.8. Security dependencies and relationship to other computers or appliances. (e.g. IDS sensor/database)	X	
2.9. A thorough description; system name, network alias, role in the network/organization, patch history, allowed/forbidden connections, main users, main vendors patch release site/connections.	X	
3. Are the list correlated with network scanning procedures or other techniques for revealing networked nodes?	X	

1.3.3. Metric 2: Vulnerability scanning

We assume the organization is aware of the benefits of combining the vulnerability scanning to the patch management process, and we let them score the highest score in the metric too. This means 10 out of 10 points.

Vulnerability scanning	Yes	No
1. Is vulnerability check of all the systems in the network performed?	**X**	
2. Are these scans performed at every host/computer periodically?	**X**	
3. Are network based vulnerability scanners used?	**X**	
4. Are host based vulnerability scanners used?	**X**	
5. Is host based vulnerability scanning on servers or on "cut edge" computers performed?	**X**	
6. Are vulnerability scanners from at least two different vendors used?	**X**	
7. Are the rules/signatures/plug-ins for the vulnerability scanner updated before use?	**X**	
8. Are the vulnerabilities discovered in the scan added in the list of vulnerabilities to be patched?	**X**	
9. Are these scans targeted at hosts after the installation of patches to confirm vulnerability removal?	**X**	
10. Are manual actions and reports for detected vulnerabilities performed if no patch is available?	**X**	

1.3.4. Metric 3: Host Based Security Connectivity

The hosts are monitored by a HIDS or the syslog-ng service. There are preconfigured hosts that can implement the second one if there are some changes in the environment.

This formula is used for computing the score:
S = (2/1) + (3/1) + (4/1) + (5/1) + 6 + 7 + (8/1) + 9, where the 6, 7 and 9 are either 0 or 1 point.

For this configuration we scored: (47/55) + (47/55) + (45/55) + (45/55) + 1 + 1 + (45/55) + 1 = 7.16 ≈ 7.2

Host based security connectivity	#	Yes	No
1. How many hosts/appliances are connected to the network?	**55**		

2. How many networked hosts have enabled any kind of host security features? (e.g. software firewall, IDS and so on)	**47**		
3. How many networked hosts have enabled any kind of logging features? (e.g. system logging, logging from software firewall, IDS and so on)	**47**		
4. How many of the hosts security features log to other appliances? (e.g. log servers or database)	**45**		
5. How many of the systems log features log to other appliances? (e.g. log servers or database)	**45**		
6. Are the log servers/DB available from several segments of the network?		**X**	
7. Is it possible for the hosts to log to an alternative secure source?		**X**	
8. How many hosts have an alternative configuration to ensure security?	**45**		
9. Are the hosts that are part of a distributed security or management system identified to form a model of reference to be used by the system/network administrators?		**X**	

1.3.5. Metric 4: Network Based Security Connectivity

This network has 2 NIDS computers, one with 2 sensors and the other with 1 sensor. They log to the same database for more efficient monitoring and statistics. The firewalls have the possibility to enable a more general logging if a situation requires it.

The formula is 1 for YES, 0 for NO, leaving a score on 7 out of 7.

Network security connectivity	**Yes**	**No**
1. Does the network include network monitoring features or other network related security appliances? (e.g. NIDS, firewall etc)	**X**	
2. Are the network security appliances covering all segments of the network?	**X**	
3. Do any of the network security appliances overlap each other?	**X**	
4. Do the network security appliances perform centralized logging? (e.g. log servers or log database)	**X**	
5. Are the log servers/DB available from several segments of		

the network?	**X**	
6. Are the computers and servers related to security features separated from the production network?	**X**	
7. Are the "cut nodes" identified for the network security features?	**X**	
8. Is special attention given to the network activity when performing shutdown or separation of security related "cut node" computers or appliances?	**X**	

1.3.6. The metrics results

The following table sums up the metrics score for configuration 3.

Metric	Score	Max score	Difference
Assets List	11	11	0
Vulnerability scanning	5	5	0
Host security connectivity	7.2	8	0.8
Network security connectivity	7	7	0
TOTAL:	30.2	31	0.8