

Data Loss Prevention Systems and Their Weaknesses

Tore Torsteinbø

Supervisors

Michael Sonntag (JKU)

Vladimir A. Oleshchuk (UiA)

This Master's Thesis is carried out as a part of the education at the University of Agder and is therefore approved as a part of this education. However, this does not imply that the University answers for the methods that are used or the conclusions that are drawn.

University of Agder, 2012

Faculty of Engineering and Science

Department of Information Technology

WARNING!

This document contains sensitive information and is only for internal distribution to trusted parties.

Abstract (English)

Data loss prevention (DLP) has grown in popularity for the last decade and is now becoming a mature technology. With the growing amount of digitally stored assets, the need for enterprises to detect and prevent data loss is increasing. DLP software that analyses traffic, detects and blocks unauthorized use of confidential data is therefore a result of this growing need, but do these security products live up to their own claims?

This thesis will look at how effective DLP is at preventing different types of data loss depending on the various factors involved, such as nature of the attack and the technical knowledge of the attacker. Through examples from real DLP software we will outline the various components that make up a modern DLP solution and how they work together to protect the data of an organization.

We hypothesize that current DLP products are insecure and pose a security risk to the environment they are installed in. This is not the fault of DLP itself, but how the technology has been implemented by security vendors. If an attacker can exploit a weakness and compromise a DLP system, the system can be turned against itself and be used to accelerate data theft by providing information about the whereabouts of all sensitive data, if not the data itself. This automated way of stealing data is much faster, and in many cases more accurate than manually scavenging the network for files. As a result the time span to detect and stop an attack reduced and might leave the victim in a worse situation than if no DLP was present in the first place.

Abstract (German)

Data Loss Prevention (DLP) hat in den letzten 10 Jahren stark an Popularität gewonnen und ist jetzt zu einer ausgewachsenen Technologie geworden. Mit der wachsenden Anzahl an digital gespeicherten Datensätzen wächst auch der Bedarf an der Ermittlung und Vorbeugung des Verlustes dieser. DLP Software, welche Traffic analysiert und unauthorisierte Zugriffe auf vertrauliche Daten aufspürt und blockiert, ist somit ein Resultat der wachsenden Notwendigkeit. Aber werden diese Sicherheitsprodukte den Ansprüchen gerecht?

Diese Arbeit wird zeigen, wie effektiv DLP dem Verlust von Daten, basierend auf der Art und Weise des Angriffes und dem technischen Wissen des Angreifers, vorbeugen kann. Mit Hilfe von Beispielen von realer DLP Software werden wir gezielte Komponenten, welche eine moderne DLP Lösung ausmachen, aufgreifen und zeigen, wie diese zusammen arbeiten, um die Daten einer Organisation zu schützen.

Wir stellen die Hypothese auf, dass DLP Produkte unsicher sind und ein Sicherheitsrisiko für die Umgebung darstellen, in denen sie installiert sind. Dies ist nicht die Schuld des DLP selbst, sondern die der Sicherheitsanbieter, welche die Technologie umgesetzt haben. Wenn ein Angreifer eine Schwachstelle ausnutzt und somit das DLP System kompromittiert, kann das System gegen sich selbst gerichtet werden und beschleunigt somit den Datendiebstahl mit dem Zurverfügungstellen von Kenntnissen über den genauen Ort von empfindlichen Daten, wenn nicht sogar die Daten selbst. Dieser automatisierte Weg des Datendiebstahls ist wesentlich schneller und in vielen Fällen auch genauer als das manuelle Plündern von Daten im Netzwerk. Das Resultat ist ein kürzere Zeitspanne, in welcher ein Angriff ausgeführt und gestoppt werden könnte und somit könnte das Opfer in einer schlechteren Situation zurückgelassen werden, als wenn das DLP System niemals präsent gewesen wäre.

TABLE OF CONTENTS

List of Figures.....	vii
List of Tables	viii
Acknowledgement.....	ix
1 Introduction	1
1.1 Background	1
1.2 Definitions	1
1.3 Objectives.....	1
1.4 Methodology	2
1.5 Limitations	3
1.6 Structure.....	3
1.7 Previous Work	3
2 Data Loss Prevention Architecture	5
2.1 What Is Data Loss Prevention?	5
2.1.1 Definition	6
2.1.2 The Various Terms.....	6
2.1.3 A Brief History.....	6
2.2 Threat Elements.....	8
2.2.1 Accidental Data Loss.....	8
2.2.2 Insider Attacks	9
2.2.3 External Attacks	11
2.3 The DLP Technology	13
2.3.1 Policies.....	13
2.3.2 Core Technologies.....	14
2.3.3 Data Classification	18
2.3.4 Data Protection	23
2.3.5 Feature Examples from Bypassing a DLP.....	30
2.3.6 Technological Shortcomings	31
2.3.7 Alternative Uses	31
3 Vulnerabilities in DLP Systems.....	33
3.1 Threat Scenario	34
3.1.1 Background	34

Data Loss Prevention Systems and Their Weaknesses

3.1.2	The Threat Materialized	34
3.1.3	Summary.....	37
3.2	Finding the Flaws.....	38
3.2.1	Test Environment.....	38
3.3	Evaluating MyDLP.....	39
3.3.1	Penetrating the Application	41
3.4	Evaluating Trend Micro DLP.....	49
3.4.1	Penetrating the Application	51
3.5	Attacking the DLP	59
3.6	Results	64
4	Discussion.....	65
4.1	Attacking Update Mechanisms and Service Channels.....	65
4.2	Security of Other Products.....	65
4.3	Security Vendors as Front Figures for Computer Security	66
4.4	Integrating DLP into Business Workflow.....	67
4.5	DoS Attacks.....	68
5	Conclusion	69
5.1	Future Work.....	70
6	Bibliography.....	71
	Appendix I: Web Application Hacking Checklist.....	77

LIST OF FIGURES

Figure 1: The process of an attack [2].	2
Figure 2: Data states in a DLP with examples in parentheses [4].	5
Figure 3: Edited version of Gartner's 2008 Hype Cycle for Data and Application Security [16].	6
Figure 4: Motives behind external attacks [26].	11
Figure 5: A basic DLP implementation	13
Figure 6: Policy overview	14
Figure 7: Endpoint DLP - Enforcing policy [10].	15
Figure 8: Trend Micro DLP blocking copying of the sensitive word "masteroppgave".	24
Figure 9: Trend Micro USB device configuration.	25
Figure 10: The DLP blocks Skype from loading a sensitive file.	25
Figure 11: Analyzing e-mail communication.	26
Figure 12: If desired, sensitive e-mails can be allowed to specific domains.	27
Figure 13: Step 1 - Company workstation compromised.	35
Figure 14: Step 2 - DLP management server compromised.	36
Figure 15: Step 3 - DLP endpoint agents compromised.	36
Figure 16: Database user table	52
Figure 17: Trend Micro DLP - Role creation	57
Figure 18: server.ini file size value.	61
Figure 19: Starting a new scan with OpenDLP.	61
Figure 20: Deploying remote agents with OpenDLP	62
Figure 21: Overview of running agents.	62
Figure 22: Results from running agent.	63

LIST OF TABLES

Table 1: MyDLP - Running network applications and daemons.....	39
Table 2: MyDLP - Web technologies used.....	40
Table 3: Files with PHP \$_GET, \$_REQUEST or \$_POST parameters.	44
Table 4: Trend Micro DLP - Running network applications and daemons.....	49
Table 5: Trend Micro DLP - Web technologies used	50
Table 6: Trend Micro DLP - Endpoint update deployment.	60

ACKNOWLEDGEMENT

Thanks to Michael Sonntag and Vladimir Oleshchuk, my supervisors, for their support. Thank you to my mother, father, sister and brother for their care; as well as my little nephew that brightens up the day.

Thank you to members of the international security community, particularly Andrew Gavin, Vic Vandal and Rich Mogull, for their help and insight.

1 INTRODUCTION

1.1 BACKGROUND

As companies integrate more and more technology into their way of conducting business, the IT systems become more complex. For security vendors, this gives opportunities to launch products that protect against new threats. Still, in their race to be the first to release these products, quality is put on hold for higher profit margins. Even after the products are released, little effort is put into patching these vulnerabilities, except in the case where third party security researches threatens public disclosure.

Although this development is present in many types software, we believe that companies specializing in security products should lead by example by not releasing products that will introduce further vulnerabilities into an already complex IT environment. To back up our claims we will evaluate various Data Loss Prevention products and demonstrate how an attacker can use these products against themselves. Although our evaluation will not represent all Data Loss Prevention products, our results will be combined with previous conducted research that has already documented severe vulnerabilities in multiple security products.

1.2 DEFINITIONS

Data Loss Prevention (DLP): See chapter 2.1.1.

Web application: An application that is accessed by using a web browser to communicate with a web server.

Management console: A web application that allows administration of the DLP systems.

Targeted attack: An attack where an organization is intentionally targeted. Often more sophisticated than “targets of opportunity”, which usually rely on scripts scanning for easily exploitable vulnerabilities.

Sensitive data: Data, which according to company policy is considered sensitive and needs to be protected.

Data Exfiltration: The process of transferring data out from a network to a remote location; often associated with the term data theft.

1.3 OBJECTIVES

This thesis focus on two areas of data loss prevention; the system itself and exploitation of said system. For better structure, the objectives related to each area have been separated into the two paragraphs as seen below. This structure will be prevalent through the rest of the thesis.

1. Explain data loss prevention, its features and limitations. How can the various DLP elements, such as potential attackers, data types and policies be classified? How effective is DLP in preventing an attacker from stealing information?

2. Evaluate and discover weaknesses in current DLP software that can assist attackers in stealing data, and if possible create an exploit to demonstrate said weaknesses.

1.4 METHODOLOGY

This thesis has been conducted as an exploratory research combining both secondary and original research. Academic papers, internet sources, as well as communication with experts in the field were used to explain DLP and the underlying technology. Since the feature set across various DLP products on the market today vary greatly, we concentrated on the most common and fundamental aspects of the technology.

A large part of this project was spent on evaluating DLP products and the underlying security of these products. As such, it was important to establish guidelines that could be followed when conducting the research. The nature of vulnerability research is often interpreted as quite unorthodox because it in many cases requires the tester to take new approaches for an attack to be successful. This is still well within the lines of the traditional scientific method, as long as the examination and correlation of test results is done in a consistent and reliable manner.

The methodology given in the 2008 edition of the book *The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws* written by Dafydd Stuttard and Marcus Pinto was followed for our DLP product evaluation [1]. The methodology approaches web application hacking in a structured manner. It also does not limit testing to specific technologies, and can easily be used to evaluate everything from PHP to Flash based web applications. The reason for choosing this methodology is that a large part of the attack surface present in the evaluated DLP products is the web application. Additionally, the methodology is a good starting point for evaluating other remotely accessible services as they share the same security principles. For our vulnerability research we have adapted the evaluation list from the book. This version of the list can be found in Appendix I.

How easily vulnerabilities can be uncovered is based on the knowledge, experience and creativity of the tester, as well as available resources, such as time and tools. The process of attacking and exploiting can be illustrated in five simple steps as shown in Figure 1. This thesis will be structured accordingly when evaluating the DLP products.



Figure 1: The process of an attack [2].

In a real situation “intelligence gathering” implies researching the target that will be attacked. In our case this is DLP products and as such, sound knowledge of the technology and its inner workings must be obtained. We then construct a threat scenario to show the relevancy of conducting vulnerability research and how it maps to reality. Any discoveries from the research are then used to exploit the system, which is quickly succeeded by data exfiltration and cleanup (control and post exploitation in Figure 1).

1.5 LIMITATIONS

Many of the features of DLP overlap with the security features of firewalls, intrusion detection systems and certain endpoint software. This will be mentioned briefly in those cases where an overlap exists, but in general this thesis will focus only on the features and responsibilities unique to DLP.

License, price and functionality, will decide which DLP solutions will be evaluated. As can be seen in chapter 3, only DLP systems that offer a free downloadable trial version were tested.

It is important to stress that functionality varies across different DLP products. Multiple examples will be given in this thesis to illustrate functionality, but that does not necessarily mean this functionality is present in all DLP products. Limitations of the technology itself are further explained in chapter 2.3.6.

1.6 STRUCTURE

Two separate chapters have been written to focus on each of the two paragraphs presented in objectives (chapter 1.3).

Chapter 2 addresses the first paragraph and is structured as follows: Chapter 2.1 defines what a DLP is and gives a general introduction to the technology. Chapter 2.2 explains the various threats involved in data loss and investigates how effective a DLP is in addressing them. Chapter 2.3 is the biggest sub-chapter and goes into the architecture of a DLP system; explaining content discovery methods and how data is being protected, as well as limitations of the system and examples of how it can be bypassed.

Chapter 3 addresses the second paragraph and is structured as follows: Chapter 3.1 introduces the attack scenario and illustrates how a DLP system can be turned against itself. This attack is then demonstrated in chapter 3.5 following an in-depth evaluation of the two DLP products; MyDLP (chapter 3.3) and Trend Micro DLP (chapter 3.4). The results from our discoveries will be summarized in chapter 3.6.

In chapter 4, discoveries and dilemmas brought to light from the previous two chapters, will be discussed. This includes discussion of different types of attacks and weaknesses, as well as a look into why DLP and other security products fail at security.

The conclusion and recommendation for future work is found in chapter 5 and 6 respectively.

1.7 Previous Work

Although many studies have been done on DLP systems, they mostly relate to methods for classifying and detecting sensitive content (see [3] [4] [5]), or best practices and guides related to the implementation of DLP (see [6] [7]). These and many other sources were used to provide background information for how DLP works and how it is implemented.

The idea of turning a DLP system against itself is inspired from Andrew Gawin's open source tool OpenDLP and his presentation "Gone in 60 Minutes: Stealing Sensitive Data from

Data Loss Prevention Systems and Their Weaknesses

Thousands of Systems Simultaneously with OpenDLP” [8]. The tool was originally developed to help with data exfiltration during penetration tests and provides capabilities that are further explored in chapter 3.5.

2 DATA LOSS PREVENTION ARCHITECTURE

2.1 WHAT IS DATA LOSS PREVENTION?

As organizations progress into a more technological environment, the amount of digitally stored data increases dramatically. As a consequence, keeping track of where it is stored is no longer as easy as before. The modern workforce naturally creates and uses data sensitive to the organization to do their job. This data is then used across services such as e-mail, business applications and cloud-services, as well as being accessed from multiple devices, including laptops and mobile phones. In many cases it is even hard for the users to manage the amount of data they deal with themselves, and the (ir)responsibility doesn't end there. In addition, a user also needs to keep track of how sensitive data is and who should be allowed to access it.

DLP is a recent type of security technology that works toward securing sensitive data in an automated and non-intrusive fashion. Through policies a DLP system automatically makes sure no sensitive data is stored, sent or accessed where it shouldn't be, while still allowing users to use the tools and services they choose and need to fulfill their tasks. Unlike traditional white- and blacklisting, the DLP only blocks the actions where sensitive data is involved, e.g. sending e-mails is perfectly acceptable, but not if they contain sensitive data. DLP can also be set to handle different levels of sensitivity and document access control. To quote George Lawton: "DLP systems keep people from deliberately or inadvertently sending out sensitive material without authorization" [9].

In addition to protecting sensitive data, a modern DLP should be adaptive, mobile and as minimally intrusive as possible [7]. Adaptive means that it can work in different environments and be configured to meet the needs of a wide range of different businesses. Mobile means that it can still protect the data, even when the device is used outside the company network. The products today only fulfill this to a certain degree. DLP is still maturing, but unlike a few years ago, most vendors have standardized on the core functionality that defines a modern DLP solution.

In the following chapters the definition of DLP will be given and the underlying data protection technology will be explained. This includes protecting multiple states of data (see Figure 2), and supporting a wide range of devices and platforms; all of which makes DLP itself complex.

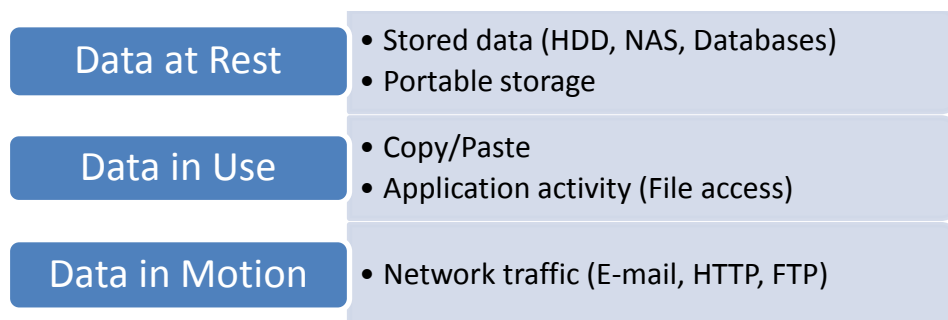


Figure 2: Data states in a DLP with examples in parentheses [4].

2.1.1 DEFINITION

For this paper we will use the DLP definition provided by Securosis – A leading research and advisory firm that has been following the development of DLP closely.

Definition:
 “Products that, based on central policies, identify, monitor, and protect data at rest, in motion, and in use through deep content analysis” [7].

While the term DLP is abused by many for describing encryption, device control, DRM, identity & access management, and other technologies that deal with data loss, these are not content-aware. This means they lack the ability to dynamically apply a policy based on content determined at the time of operation. To tell if something is a DLP one should focus on this content-aware and analytical aspect [10].

2.1.2 THE VARIOUS TERMS

As a recent newcomer in the security field, DLP goes under multiple names, including Extrusion Prevention [11], Data Leakage Prevention (DLP) [9], Information Leak Detection and Prevention (ILDP) [3], Content Monitoring and Filtering (CMF) [7] and lastly, Content Monitoring and Protection (CMP) [7].

Today, most vendors, including those that will be evaluated by us, have standardized on the term Data Loss Prevention [12] [13]. This is the term we will use for the rest of this thesis.

2.1.3 A BRIEF HISTORY

Although existing prior to 2006, DLP started blooming around the fall of this year. At this time larger vendors started to acquire smaller companies specializing in data security, a trend that continued far into 2007 [14] [15]. During 2008 and the beginning of 2009 DLP was on everyone’s lips, as illustrated by Gartner’s 2008 hype cycle seen in Figure 3. After this, the technology slowly faded into the background, as is common for many security trends.

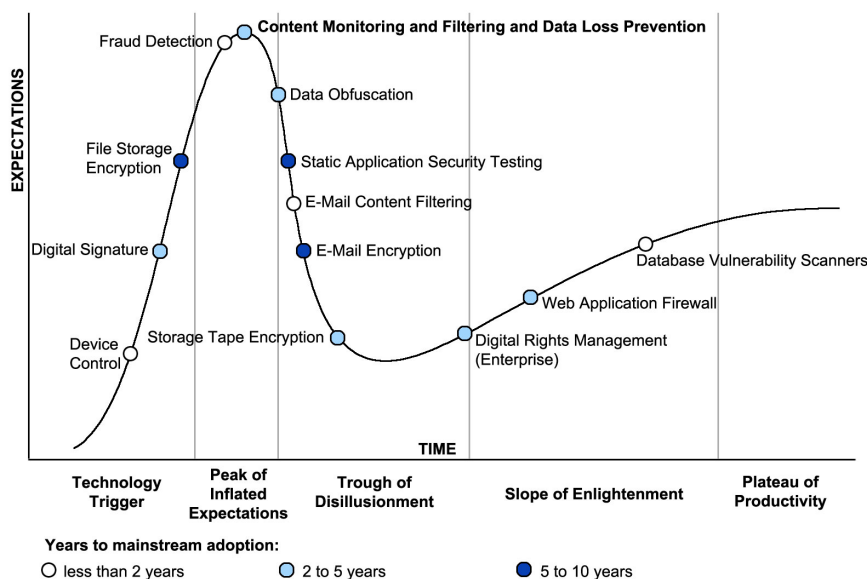


Figure 3: Edited version of Gartner's 2008 Hype Cycle for Data and Application Security [16].

By comparing DLP today with the late 2008 Hype Cycle, it is safe to say that DLP solutions are slowly crawling up the last part of the slope of enlightenment (Figure 3). At the same time, additions to, or new variations of DLP, such as cloud DLP and mobile DLP have also appeared. These variations follow their own path and will not be discussed further in this thesis.

The largest vendors today are Websense, Symantec, RSA, Palisade Systems, NextLabs, McAfee, Fidelis Security Systems, Code Green Networks and CA Technologies [13]. All these vendors offer products with more or less the core features associated with a DLP. This includes: network monitoring, e-mail monitoring, file system monitoring, and endpoint protection. To discover content the products use wordlists, regular expressions and partial file hashing. As for machine learning algorithms, only Symantec offers this feature in their DLP product [12]. These technologies are covered in depth in chapter 2.3.

2.2 THREAT ELEMENTS

Before presenting the inner workings of a standard DLP suite, we would like to summarize the threats DLP is responsible for preventing. Loss of data comes in many different forms and knowing how they happen moves us one step further in preventing them. To give a better overview we have categorized types of data loss as follows: Accidental data loss, insider attacks and external attacks. We also mention what threats exist outside the scope of DLP to illustrate some of the limitations of the technology.

2.2.1 ACCIDENTAL DATA LOSS

A typical cause for accidental data loss is employees unfamiliar with company policies. In other words, they do not recognize the sensitivity of the documents they are working with or they overestimate their own knowledge regarding computer security. Common examples include:

- Sending sensitive documents unencrypted via e-mail.
- Saving documents to a USB storage device for further transportation.
- Uploading files to online services.

Another cause that results in insecure handling of sensitive material is lack of proper training. If a user does not know how to mark a file as sensitive or encrypt it, how can it be protected?

As soon as data leaves company property, control over it is lost. One can only hope the e-mails were sent over a safe connection to a trusted third party, or that the documents were encrypted before being saved to a USB-drive that later went missing. DLP tries to remedy such situations by only allowing data to flow through authorized channels defined by company policies. It also automates tasks such as sensitive file classification and encryption, although this should not be seen as an excuse to omit proper employee training.

To illustrate our point and show how accidental data loss occurs, two examples have been picked.

Example 1: Virgin Media

The British telecommunication and broadcast company lost in summer 2008 financial details of 3000 people. The incident happened after an employee burned the sensitive data to a CD. The data itself was unencrypted and meant for transfer between Virgin offices. The disk has since been lost [17].

According to company policy all such data must be encrypted and transferred over FTP. Use of other media types is prohibited. Even though a policy was in place, people often act on their own without understanding the consequences of their actions [18].

The second example illustrates how accidental data loss occurs when the wrong e-mail attachments are sent.

Example 2: Recruitment Agencies

In Britain, a Manpower employee sent an e-mail to 60 of his co-workers. The unencrypted attachment contained personal information for 400 job seekers. Luckily for Manpower, they did not get fined for the incident [19].

Similarly, Manpower's British competitor Hays, had a similar incident in 2011. In this case an employee at Hays accidentally e-mailed the rates that RBS pays for 3000 contractors to 800 of the bank's employees. This of course led to some criticism from trade union Unite, as it was revealed some of the contract staff received £2,000 per day [20].

A survey of erroneous e-mail transmission conducted in 2008 reports that 66.2% of business users have sent e-mails erroneously. Further on, the Japan Network Security Association revealed in 2007 that data loss through e-mail, based on publically reported incidents in Japan, equaled 9.8% of the incidents [21].

One of DLPs greatest strengths is combating accidental data loss. When coupled with concise policies and proper integration, a DLP system can easily stop accidental leaks and at the same time help users treat sensitive data correctly. If we look at the first example, it is not clear if the user knew about the company policy regarding FTP transfers. Maybe other users were also burning CDs and the user acted like everyone else. If a DLP was in place, the system could stop the CD burning from starting and refer the user to the company policy describing how sensitive files should be treated. This same procedure can also be applied to the contents of e-mails. If a sensitive attachment is sent, the e-mail is blocked. If more flexibility is needed you could specify recipients that are allowed to receive certain types of sensitive files. This of course depends on the capabilities of the DLP product in place.

More information regarding detection and blocking of leaks will be presented in chapters 2.3.3 and 2.3.4 respectively.

2.2.2 INSIDER ATTACKS

A good definition of an insider attack is the one given by Webopedia.

Definition:

“Insider attack: Any malicious attack on a corporate system or network where the intruder is someone who has been entrusted with authorized access to the network, and also may have knowledge of the network architecture” [22].

The documents released on WikiLeaks.org are examples of insider attacks with the purpose of publically leaking documents. Of course, the motivation behind these types of attacks is not always related to political activism. Other reasons for conducting insider attacks are typically financial gain or holding a grudge against the employer [23].

The CyberSecurity Watch Survey is an annual report which provides statistics related to insider attacks. In the 2011 survey 43% of 607 respondents reported that an insider incident had occurred. Of these, 57% resulted in unintentional exposure of private and sensitive data. If we subtract the unintentional accidental data losses from the amount of insider incident, we end up with 18,5% malicious insider attacks, and of these 82% suffered theft of intellectual property. From an economic perspective the report tells us that costs suffered from insider attacks is comparable to those caused remotely, even though insider attacks only accounts for 27% of the total attacks [23].

For insider attacks the effectiveness of a DLP-system is difficult to measure. First one has to consider the technical knowledge of the attacker. Then one has to consider what access this individual has to various systems. If the attacker is someone from the IT department, the DLP will most likely be bypassed. Additionally such an attack will be more devastating as IT workers usually have, or know, how to get access to sensitive data.

A DLP system is not an access control system in the traditional sense, and should not be seen as a replacement to one. It is up to the companies themselves to make sure users only have access to what they need to do their job, and separate data according to level of sensitivity. The role of a DLP comes into play when a user is working with sensitive data. The DLP does not disallow the user to view the sensitive content (which is what an access control system would do), but instead makes sure the user does not treat the data in an irresponsible manner (e.g. sending it attached to e-mails or uploading it to public file-shares).

Depending on how the DLP is bypassed, the actions of the attacker do not necessarily go unnoticed. Before finding a proper way of bypassing the system the attacker might have triggered alarms while discovering a proper bypass. If IT security is warned of such malicious activity, actions can be taken before the stolen data is distributed further by the attacker. In cases where the breach is not stopped the DLP can still be invaluable when looking for forensic evidence. Most DLP products keep logs of who accessed what when, that can be used when pursuing further legal action. In both these cases the DLP does not prevent the data loss directly, but is still valuable in comparison to having no DLP at all.

Having a DLP in place might discourage users from committing data theft. If they know such a system is in place, they will also recognize an inherent risk for being detected when conducting data theft.

Preventing insider attacks is not necessarily something that can be done on the technological level. Disgruntled employees and employees who plan to leave the company are some of the most likely candidates to perform this type of attack. In environments where highly sensitive data is in use, it is important to focus on this human layer. The following points helps in addressing these social issues [24] [25]:

- Clear policies: Stating company policy clearly in a concise and easy to understand format will increase the likelihood that an employee will actually read and apply it when working. The policy should guide employees on what the expected behaviors and

requirements are, and also define prohibited activities. For DLP systems the policies play a fundamental role (see chapter 2.3.1, page 13).

- Good training: Training employees on security awareness as well as explaining the meaning behind the various company policies goes a long way in increasing employee understanding of the whole process, and how they can help improve it.
- Background checks: Performing background checks can assist in stopping untrustworthy individuals at an early stage.
- Physical security: Make sure that critical IT infrastructure and storage containing sensitive information is properly locked off. Theft can happen as soon as the opportunity opens itself, limiting this opportunity will go a long way in protecting business assets.
- Building trust: Treating employees fairly with trust is one of the simplest tools in combatting low morale and also goes a long way in building a loyal work force. How can you trust an employee that doesn't trust you?

It should still be emphasized that DLP works in addition to other security technologies, such as access control, IDS and data encryption, and does not replace any of these. Lack of company policies and controls regarding what can be brought to work, such as cameras and cell phones, falls outside the scope of DLP. Even with the highest security controls in place, nothing can prevent the attacker from just memorizing the sensitive information.

2.2.3 EXTERNAL ATTACKS

In the Verizon data breach investigations report for 2012, 98% of 855 breaches stemmed from external agents [26]. In comparison only 4% implicated insiders. The 2% overlap represents coordinated incidents between internal and external parties.

External attacks are what most people consider conventional hacker attacks. Basically, someone gains access to a system via a remote connection, such as the internet, and uses this access to steal data, create botnets or cause disruptions. The motives behind these attacks are mainly of financial nature (Figure 4 below). According to the report [26], being a target of opportunity (79% of cases), was more common than suffering a targeted attack. The reasoning is that the companies involved had vulnerable software exposed to the internet that is easy to exploit.

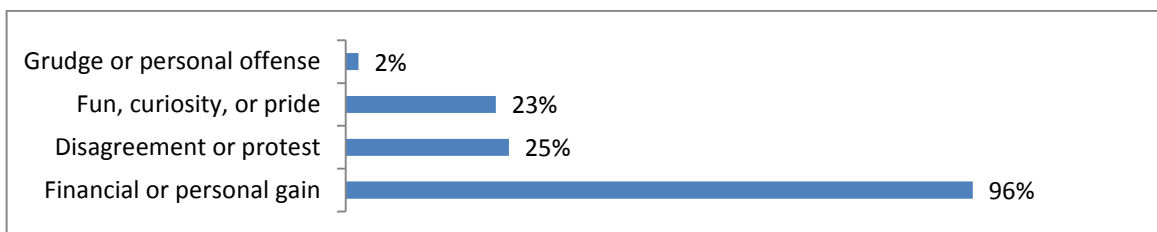


Figure 4: Motives behind external attacks [26].

Data Loss Prevention Systems and Their Weaknesses

The effectiveness of a DLP in these types of attacks depends mainly on the attacker's knowledge of said system. Some systems, such as Trend Micro DLP, use special patterns for detecting and stopping data-stealing malware. If the attacker then installs one of these malwares or accesses files monitored by the DLP, this might be enough to trigger an alert and get discovered.

If the attacker knows a DLP system is in place and gains administration privileges to a workstation, there won't be much in the way of suspending, or even uninstalling the DLP endpoint protection. If done properly, the management server will think the endpoint agent is offline (e.g. machine is turned off) and not issue alerts as a result of this behavior. With administrative privileges data encryption software can be also installed and used to avoid detection by the network DLP during data exfiltration.

Even after a breach has occurred the forensic evidence found in DLP access logs can be helpful in determining what happen and the impact of the breach. A problem with external attacks is that even with evidence you might not have the jurisdiction to investigate further. For example if the attack originated from China and the victim is in the US, the Chinese government might refuse to cooperate in the investigation and prosecution of the attacker. There is also the problem of connecting an individual or organization responsible for the crime to the IP-address(es) where the breach originated, especially when proxies have been used.

To summarize, the addition of a DLP might have some effect on detecting and stopping remote attacks. Still, having a firewall, IDS, anti-virus, conducting employee security awareness training and applying good security practices goes a lot further in remediating this threat, than just installing a DLP system.

The effectiveness and shortcomings of DLP are summarized in chapter 2.3.6.

2.3 THE DLP TECHNOLOGY

In chapter 2.1 and 2.2 we defined what DLP is and why there is a need for it. This chapter deals with how DLP works. Figure 5 below gives a basic overview of the physical parts common to a DLP system. The Endpoint DLP is installed directly on a workstation and keeps track of how data is stored (data at rest) and used (data in use). The network DLP is often placed between the LAN and WAN as a proxy which monitors network traffic (data in motion). The DLP server manages both these components and is mainly responsible for policy deployment (illustrated with green arrows) and logging policy violations. Endpoint DLP, Network DLP and the management server will be discussed in detail in chapters 2.3.2.2, 2.3.2.3 and 2.3.2.4 respectively.

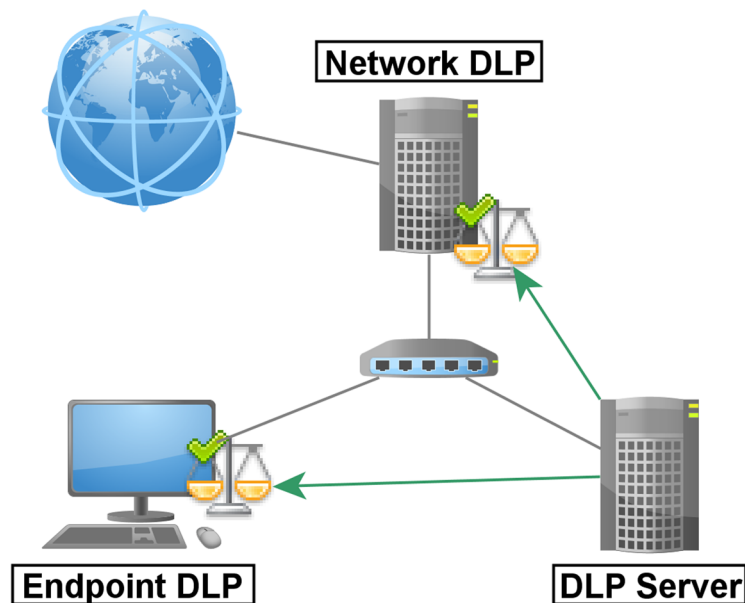


Figure 5: A basic DLP implementation

2.3.1 POLICIES

In the heart of DLP are the policies. Without them there would be no differentiation between public and sensitive data. Policies can be based on the organizations own specifications, but also external requirements, such as PCI DSS and similar.

DLP policy creation is one of the few tasks in a DLP deployment that involves the whole company and not just the IT department. At this stage it is important to look at existing policies and discuss with the people handling company data on how to properly classify, identify and protect this data. These policies are then converted into rules that the DLP can enforce during operation. As an example: A policy classifies java source code as an important company asset and therefore sensitive. Because of this, code should only be stored in the code repository and on the machines of the java developers. If a developer tries to save source code to any other location the DLP enforces policy and blocks the request.

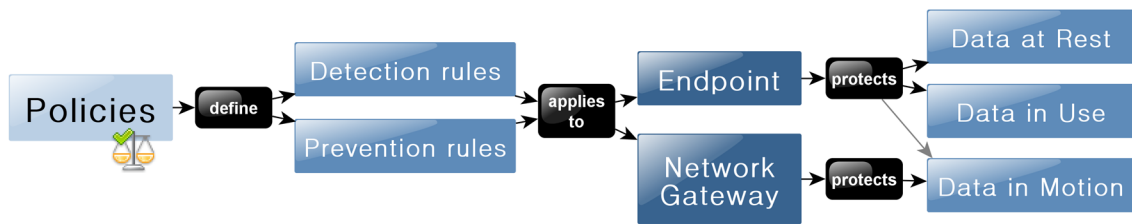


Figure 6: Policy overview

Figure 6 shows how regular policies are converted into rules the DLP software can use to enforce said policies. The detection rules specifies how to detect sensitive content, while the prevention rules specify how the detected content should be treated. These rules are then deployed to endpoint agents and the network DLP so they can be used when monitoring the different data states.

Multiple best practices guides exist on how DLP policy creation should be done. The RSA suggest asking the following questions [6]:

- Who is the policy going to apply to and how does it impact them?
- What type of information are you trying to protect?
- Why are you protecting it?
- Where should you protect it? Is the data in motion or in a datacenter? Is it being used at endpoints? Strategize which data state needs protecting first!
- When should you trigger a violation?
- How should you protect the data? Audits, encryption, blocking, etc. Choices should be made depending on the type of information.

In chapter 2.3.3 and 2.3.4 we will explain the different methods used by DLP to discovery sensitive data coupled with examples of how policies are converted into rules that can be interpreted and acted upon by the DLP. Before this an overview of a typical DLP architecture will be given.

2.3.2 CORE TECHNOLOGIES

DLP is applied differently depending on what state the data is in. For data at rest, the content of stored data is scanned; for data in use, the DLP interacts with input given to programs and OS; and for data in motion each data packet is analyzed. To effectively protect an organization, all these channels have to be monitored and managed, which makes DLP a bit more complex than the average firewall or anti-virus solution.

2.3.2.1 CONTENT DISCOVERY

Before a DLP system is implemented, organizations often face a situation where their digital data is spread over multiple locations with no control of where sensitive data is located. In DLP terms, content discovery is applying policy to discover where sensitive files are located. This can be in databases, on files shares, the local storage on laptops and workstation etc. The discovery works similarly to an anti-virus scan, but instead of looking for virus and malware, it

looks for sensitive documents and logs their location. From the results administrators and management can decide how to consolidate the files.

Content discovery deals with data at rest. The scan is done with the help of multiple methods, including simple file crawlers that can be installed on servers and workstation, remote file scans where the management server scans network shares, and endpoint scans where the DLP agents check the local storage of the machines they are installed on.

Truly advanced DLP systems can take file discovery one step further and automatically assign sensitivity file scores, encrypt data, and consolidate data to a more secure location when encountered. Alternatively, sensitive information can be redacted from a document with or without user interaction [27]. Because of the high risk of false positive when initially deploying an untested DLP policy, it is not recommended to take an active role in the discovery process, but instead just log it. This choice can be reconsidered after seeing how the policy behaves in practice.

Although content discovery is commonly used when initially deploying a DLP, running the scan regularly is not uncommon. Beware that scans can take up system resources and takes time to complete, so it should not be run during work hours [7].

2.3.2.2 ENDPOINT PROTECTION

DLP endpoint protection is installed on workstations and other devices in the form of an agent. The agent enforces policies by monitoring all data activity and scanning all locally stored files. Usually the agent also allows physical input to be controlled. This means an administrator can centrally disable USB, FireWire and other interfaces with ease. Also, the act of burning CDs or DVDs can be prevented. Figure 7 illustrates how sensitive documents are allowed to be stored on the network share, the local hard drive and in a database, but not in e-mails or on removable storage devices.

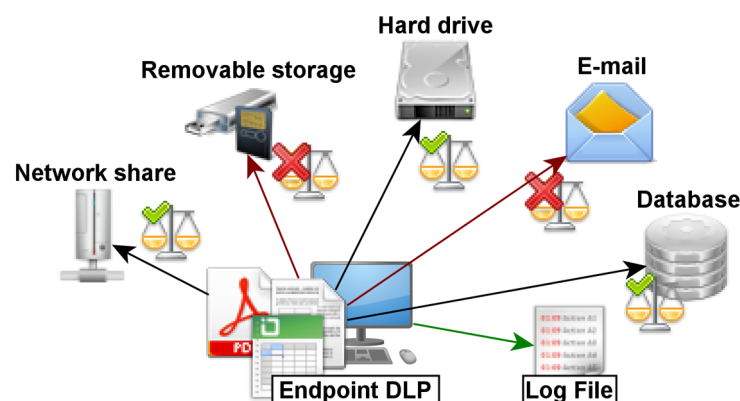


Figure 7: Endpoint DLP - Enforcing policy [10].

Traditional endpoint protection software allows similar blocking, but lack the content-aware component that is central to DLP. With both products the print screen functionality can be disabled, but with DLP the software can detect if no sensitive documents are open and re-enable print screen until sensitive content is detected again. The same principle can also be

applied to the copy-paste functionality, e.g. the DLP endpoint only blocks the action when copying sensitive data, but still allows other non-sensitive data to be copied.

In certain implementations it is also possible to only permit devices that encrypt all data stored on them or have the DLP automatically encrypt everything that is saved to an external storage device [28].

DLP endpoints feature different types of protection that can be categorized as follows: [7]

1. File system protection: Monitors all file operations similarly to real-time protection found in anti-virus software. This is to make sure files are not copied to unauthorized locations or that encryption is automatically applied by the DLP when saving data to said locations. Scans can also be performed on stored data to discover policy violations.
2. Network protection: Monitors data being transmitted over the network when the endpoint is away from the corporate network. Otherwise, the network DLP is responsible for this functionality.
3. Application/Kernel protection: DLP integrated in the OS and applications to prevent actions such as copying to the clipboard, taking screenshot or typing sensitive data into chat programs.

A common concern with end-point DLP is the resources needed to actively do content filtering. For an engineer working on CAD drawings, having the DLP constantly analyze data can eat up all the resources needed to render said drawing, which in the end hinders productivity. Most DLP products do allow some tweaking and can be configured to exclude certain types of data, storage locations and activities [28]. It is important to make sure any tweaks done are still conforming to policy.

If the endpoint does network DLP, this will be an additional strain on resources. By detecting if the machine is connected to the company network, the functionality can be disabled and offloaded to the company's dedicated network DLP. As soon as one is on the road again this feature can be automatically re-enabled to ensure full DLP coverage [7].

For an attacker, disabling the endpoint DLP will help in stealing data, but at the same time considerations have to be taken not to trigger any suspicion. Depending on the DLP product, uninstalling an endpoint agent might get reported to management server, but if the attacker is able to block such communication the server will think the endpoint is offline.

2.3.2.3 NETWORK MONITORING

The network DLP can have two different modes: A passive and an active mode. In passive mode the DLP inspects network traffic and logs any policy violation, while in active mode the DLP can also block any packets associated with the policy violation. What mode to use, depends on the requirements of the organization. For example if current policies results in lots of false positives being blocked, it might be better to run it in passive mode, since the data collected can still be useful in detecting real policy violations.

The placement of the DLP is usually a location in the network where it can intercept data leaving the local network. This can be data going to other less secure networks within the same company, remote sites connected with VPN, or most likely, the internet. The choices here consists of running the DLP as a gateway proxy or connecting it to a network SPAN or mirror port, much like an intrusion detection system (IDS). The latter option is only relevant when the DLP runs in passive mode [29].

Even though blocking e-mails and file uploads can be done from the endpoint agent, a network DLP enforces the policies even on devices without a locally installed agent. Setting up a network DLP also requires less work, because unlike endpoint DLP and file crawlers, every company server and workstation does not need to be touched for the protection to be in effect. Additionally, the DLP also applies to hired contractors and guests using the network [7].

The channels inspected by the network DLP vary greatly from vendor to vendor. HTTP, FTP and e-mail services are the most common. Additionally, instant messaging protocols, HTTPS and various file-sharing services are also often monitored in these products. In many cases e-mail is controlled by its own DLP component, either as a dedicated solution interacting with the e-mail server or as an endpoint component monitoring the local e-mail client [7]. The general examples presented here do not necessarily reflect all the DLP products out there.

MyDLP, which is evaluated in this paper, as well as McAfee DLP and other large vendors use ICAP (Internet Content Adaption Protocol) for content monitoring [30]. In this case, all packets sent through a network DLP proxy are forwarded to the ICAP service. For example, consider a user attempting to upload a sensitive document via the proxy. The proxy, acting as an ICAP client, asks the ICAP server to check the contents of this document before allowing it to pass out of the network. Since this would be detected as sensitive, ICAP can either modify the request so that it points to a page containing an error, or return an encapsulated HTTP response that indicates an HTTP error. In both these cases the content has been “adapted”. Further details regarding ICAP can be found in RFC3507 [31].

A closer look at how encrypted communication channels can be handled is found in chapter 2.3.4.5 on page 29.

2.3.2.4 CENTRAL MANAGEMENT

The responsibility of the DLP management server can be many. Some of the core functionality includes:

- Endpoint agent deployment and agent management tasks including policy deployment, software patching and log collection.
- Update software, common definitions and licensing information from the vendor’s server.
- Collect logs from other services, such as file crawlers and network DLPs, and keep these services updated with policies and software patches.
- Forward critical alerts to system administrators.
- Generate reports based on collected data.
- Provide tools to create and manage DLP policies.

Many of the points above can usually be accessed and managed through an administration interface. This is commonly a web application accessible from any browser, although command-line interfaces also exist.

The DLP server usually comes in the form of a hardware or virtual appliance. Depending on the vendor, the network monitoring and management is often done on the same device. Therefore, having a dedicated device is usually recommended because of the large amounts of data that will need to be processed at any given time.

All active agents report their activity to the management server, including policy violations. This is usually done over a special communication port open on both the endpoint and server. When deploying the endpoint it is important to define the address of the management server if the endpoint doesn't discover this automatically. If an endpoint agent is located outside the local network, all logging events will be stored locally on the endpoint until it is connected to the local network again. The management server usually displays which endpoints are up and running, and which are offline (outside the LAN or turned off). Reporting and maintaining communication in near-real time ensures the agents are always up to date with the latest software and policy updates [7].

2.3.3 DATA CLASSIFICATION

Various methods to analyze and discover sensitive data exist. The most common methods are the use of keyword matching, regular expressions and data fingerprinting/hashing. Although not widely implemented, statistical methods - similar to those used for spam mail - can also be used to identify sensitive data and is an area of high interest to the DLP industry [5]. This chapter will take a closer look at the different ways of identifying sensitive content.

2.3.3.1 KEYWORD MATCHING

Keyword matching is the most basic of all content analysis methods. Based on a pre-determined list of keywords, the scanner will go through the file system looking for plaintext strings that match any of the pre-determined keywords. As with any of the content analysis methods; when a match occurs, the DLP will enforce the configured policy.

This method is fast and can be effective when implemented in an environment where sensitive documents are identified by certain words or text strings. Now, in most real world scenarios it is not the case that every sensitive document can be identified that easily. Businesses that deal with social security numbers, credit card numbers and other ID's will have problems using keyword matching as these types of number are non-static values.

The use of keyword matching is only recommended for simple documents containing common static keywords; luckily, DLP products are not limited to only this type of content analysis [4].

2.3.3.2 REGULAR EXPRESSIONS

With keyword matching it is still possible to fill the blacklist with every conceivable credit card or social security number, but a far more effective way of doing this is with regular expressions. As an example:

```
^\d{3}-\d{2}-\d{4}$
```

The regular expression above will match any US hyphen-separated social security number in the format NNN-NN-NNNN. As an example, 123-23-3456 would produce a match, or even 892-45-1243. The expression looks for three digits followed by a hyphen, then two digits followed by a hyphen, and lastly the four digits completing the structure of a standard US social security number. In other countries additional regular expression will have to be defined accordingly to compensate for individual national ID numbers. For example in Norway where the equivalent ID number is called “Birth number” the regular expression `^[0-3]\d[01]\d{3}[]?\d{5}$` would be used. The number in this case is 11 digits long with a 6 digit date number followed by a 5 digit person number which is sometimes divided by a space.

Regular expressions are suited for detecting variable, well-structured data. This includes source code and identification tags. It is important for companies to add these expressions according to their needs. In the case of an international organization, entries for each type of national ID numbers related to the countries the company has a presence in can be added. Regular expressions should be formulated specific enough so as to minimize false positives.

2.3.3.3 FINGERPRINTING

Another method for discovering sensitive documents is by comparing it to a group of files that one considers sensitive. If a match occurs, you are dealing with a sensitive file. To check if two files are identical you can compare them bit by bit, but an even more effective way of doing this is by first calculating a hash of both files and then compare the hashes bit by bit. The benefits of using cryptographic hash functions are:

- One-way: A hash function is a one-way function, so if an attacker gets hold of the hash it will not reveal any secrets from the document it was calculated from.
- Size: A hash requires considerable less space than the file it was calculated from and always results in the same bit length value (usually 128 or 160 bits depending on algorithm). This means transferring the hash over the network is fast and produces very little overhead. Additionally, storing the hash is easy and it replaces the need for storing all the sensitive files on the DLP server, just to be able to perform file comparisons.
- Unique: A hash is unique to the file it was calculated from. Even though two files could theoretical produce identical hashes, this has little practical effect on a DLP system.
- Performance: Modern hash algorithms are highly optimized and as results, work well, even on small mobile devices with limited resources.

In DLP the term fingerprinting is often used when talking about file hashing, but is not limited to the hash value itself. File attributes, such as picture EXIF information, file size, filename and other metadata can also be added to the file fingerprint. When used in combination with file hashing, DLP detection of sensitive files can be improved even further.

The simplest form of fingerprinting is performing the hash function on a whole file. All the generated fingerprints are then kept in a table in a database (similar to the keyword list mention earlier), and as such needs to be constantly updated to cover newly created or changed sensitive files. On the positive side, automated approaches exist to populate a hash table. For example you can tell the DLP to monitor a directory and fingerprint all files added to it.

The reason for using hashing over regular expression and wordlists is that not all files contain text strings. Images, audio and video can also be sensitive, but for these formats it is hard to apply other methods than fingerprinting. To increase the detailed of a fingerprint partial files hashing can be applied. The file is analyzed and divided it into several smaller chunks and for each chunk a hash value is calculated. This means that even if some parts of the file changes, it can still be considered sensitive as other parts still remain the same and matches the stored hash values [4].

Although partial files hashing is applicable to all files types, it excels at text detection. Even when sensitive texts have been converted to different file types the DLP is still effective. This is achieved by extracting all text to plain text - disregarding markup, typefaces, and other formatting - before generating a fingerprint and performing a comparison [32]. How chunks are divided when performing partial file hashing on text can be based on common sentence delimiters (.,:;) or a certain amount of words (example: 10 words per hash calculation). There is often a bit of sentence overlap between the chunks to improve accuracy. As with binary files a sensitive text document can still be detected, even if parts of it have changed. For example, you could edit a text document by removing some chapters, add some text and rewrite some parts, but many of the original paragraphs will still be intact. If one hash matches one of these intact paragraphs the document will still match the associated fingerprint.

A problem, especially when fingerprinting text, is that some sentence may be used in both sensitive and non-sensitive texts. As an example the same template for meeting notes might be used for both a public and secret meetings. This can result in false positives because the notes from the public meeting are considered sensitive since the introductory text and other elements are identical to the notes of the secret meeting. A scoring system, such as the one implemented in Trend Micro DLP, can be used to counter this. Depending on the number of partial fingerprints that the files have in common, DLP assigns a match level of high, medium, or low. The more fingerprints in common, the higher the level. When a certain threshold specified by the policy is reached, the file is considered sensitive [32]. Having a threshold in place will also help in countering the very unlikely hash collision. Given a situation where the average length of a text document is 10 000 randomly generated words, and a SHA-1 (160 bits) hash is calculated for 10 words at a time, you will still need, based on the

cryptanalysis from a French scholar [33], over $\sim 2.25 \times 10^{12}$ text documents for a hash collision to even be realistically possible.

One research paper suggests an additional way that can help reduce the amount of false positives and increase performance [3]; by stripping out common non-sensitive phrases from the texts, only the unique ones will be go through the fingerprinting process. In the research paper a common sentence is for example “the following is a summary of our meeting” because Google returns 15 000 results on this entry. To weed out additional neutral phrases from the fingerprinting process, the company can provide a set of public documents to train the system in what is considered sensitive. The performance increase is achieved as less hashing and comparisons are performed on each file [3].

Drawbacks of fingerprinting relates to the reformatting of binary files. For example converting a JPEG files to PNG or using the content in a different context [4]. See also chapter 2.3.5 on how text files can easily be obscured.

2.3.3.4 MACHINE LEARNING ALGORITHMS

For many organizations it is not unrealistic that gigabytes of sensitive data are created each month. In such scenarios adding keywords, regular expressions, and fingerprints can quickly become time consuming and counter-productive for the IT administration.

Machine Learning Algorithms seek to automate the identification of sensitive contents through training. An example of a system based on the same concept is automated spam filters which work as follows: First there is a learning phase where the filter is told which e-mail messages are considered spam and which are consider legitimate. As more and more messages are processed the algorithm learns and improves its accuracy. When one feels confident in the filter’s detection of spam messages, it is switched over from the learning phase to a managing phase that does not required constant supervision. Even though the filter operates in managing mode it still accepts feedback in case of false positives, false negatives or if it needs to expand its filtering capabilities to cover new types of spam (for example in a different language).

DLP systems using machine learning algorithms works similar to a spam filter. The algorithm is fed two document sets. One public set that represent what you would consider a representation of ordinary files, and a second “secret” set representing sensitive files that the DLP should protect. A good example of such an implementation is described in the paper [4], which demonstrates the use of machine learning algorithms to classify text documents. This was accomplished by providing a public training set based on a complete archive of Wikipedia and other publicly available documents, plus a set of sensitive documents.

A problem with a system like this is that it only works with text documents, and can give a high rate of false positive and false negatives if not trained properly. One also has to consider different types of languages when training this system and what to do with unrecognized text. For example: What if the text is in a language that has not been trained? Should it be considered secret? If not, what if the secret document is in another language? For international

organization is important to identify which languages sensitive documents are written in and train the system accordingly.

2.3.3.5 CONCEPTUAL/LEXICON

The conceptual/lexicon method of detection combines word usage patterns commonly associated with specific concepts. As an example, a concept can be criticizing co-workers or doing online shopping. When a high degree of words and expressions commonly associated with such a concept is detected, an alert is triggered [5].

The rules don't necessarily have to be complex. The *Invisible Witness* tool [34] developed by Onur Polatcan, Sumita Mishra, and Yin Pan, analyzes outgoing e-mail for one of the following conditions: Message is not written in English, spelling mistakes makes up more than 25% of the content, or attachment size is significantly larger than average for the given sender. If any of these conditions are met an administrator is alerted [34].

A second example is a study done on 289 695 e-mails from Enron [35]. The purpose of the study was to rank each e-mail according to a level of deception. This was done by analyzing the word usage of first person words, exclusive words (but, without, except), negative emotion words and action verbs. The messages that deviated most from the average could be categorized in 3 distinct groups:

1. E-mails with a large number of exclusive words. These tended to be emotionally-charged messages to coworkers, family and friends.
2. E-mails with many first person pronouns. The main content of these messages were non-business recreational activities.
3. E-mails with many action verbs. These were the deceptive messages as they are less cognitive complex and contain many action verbs as opposed to exclusive words and first person pronouns.

An interesting finding by the study was that [35] "attributes associated with the deception model capture emails that reflect a variety of potential problems with an organization, for example complaining, conveying information improperly, or spending organizational resources and employee time on non-work-related issues. Analysis by such a signature can therefore be useful for detecting both potential organizational dysfunctions and criminal behavior such as fraud".¹

For DLP, such a tool could be used to identify malicious activity before a data breach or similar occurs. The content of these messages or documents are often not based on presently known data, thus making this method superior to the previously mentioned detection methods in regard to behavior detection.

As of today no vendor has implemented such functionality yet. Apart from the implementation costs (e.g. designing templates for different types of concepts), it is also prone

¹ Copyright © 2005 P.S. Keila and D.B. Skillicorn.

to false positives and false negatives [5]. Additionally, the ethical concerns regarding employee profiling of this kind is highly controversial.

2.3.3.6 CATEGORIES

Categories, also known as compliance templates, use any combination of the methods already mentioned (mainly wordlist and regular expressions) to detect certain types of content. A good example is the PCI-DSS category which – in the case of Trend Micro DLP – is triggered when one following conditions are met [32]:

- 5 credit card numbers.
- 1 credit card number + 1 name.
- 1 credit card number + 1 partial date + expiry date keywords.

If one of the above conditions hold true the file or message is considered sensitive. Templates for other compliance requirements, such as HIPAA, are also usually included in DLP products. HIPAA for example looks for personal information found in American health insurance documents. This includes common names and personal ID numbers [32]. Although the compliance templates mentioned so far relates to American regulations, templates for detecting IBAN, SWIFT/BIC and other more international types of sensitive data are also to be found. Having these templates can save a lot of time when getting the system initially set up.

2.3.4 DATA PROTECTION

This chapter will go into depth on some of the DLP features that protect data, and have only been mentioned briefly so far.

2.3.4.1 ENDPOINT: SCANNING AND DETECTION

Scanning the contents of files can be done in different ways. As explained in chapter 2.3.3.3, for regular offices documents it is often common to extract the text strings before applying any detection methods. This ensures that the detection is applied on text content and not its formatting, which means sensitive documents that were originally in a plaintext format when fingerprinted is still detected even if it has later been converted into a Microsoft Word document or similar [32].

To easily parse documents and remove any formatting the DLP software often comes with different plugins for each file format. This is not limited to office documents, but also includes compressed archives, disk images, installers, system swap files etc. A good example of a tool that accomplishes this task is KeyView Filter, which, as quoted from their webpage [36], “extracts text, metadata and other relevant properties from over 1000 file formats on numerous platforms, leading to complete indexing of the entire enterprise corpus in a single index.” This makes it easy to apply wordlist, regular expression, fingerprint and machine-learning language rules to identify sensitive documents, and is perhaps the reason why Trend Micro and Symantec includes KeyView with their endpoint DLP software [37].

Scanning files like this makes it harder to circumvent the DLP by simply renaming or saving the document in another format. It also reduces overhead by only scanning and fingerprinting text content while ignoring formatting.

If more accuracy is wanted the partial fingerprinting can be done twice; once for the original file, and once for the version where all text content is parsed to plaintext. This ensures that sensitive content can still be detected, even after document conversion.

2.3.4.2 ENDPOINT: KERNEL INTEGRATION

As explained in chapter 2.3.2.2 the DLP endpoint integrates with the OS kernel and program functionality. This section will give some examples of this functionality.

a. BLOCKING COPY FUNCTIONALITY

Although blocking the ability to copy data to the clipboard is present in traditional endpoints, the DLP version is more intelligent. With DLP, when you copy data, the action is only blocked when sensitive data is copied. If non-sensitive data is being handled, the DLP will not interfere and as a result be less of an annoyance. For users commonly working with sensitive data, this functionality can be disabled and at the same time enabled for other groups of users.

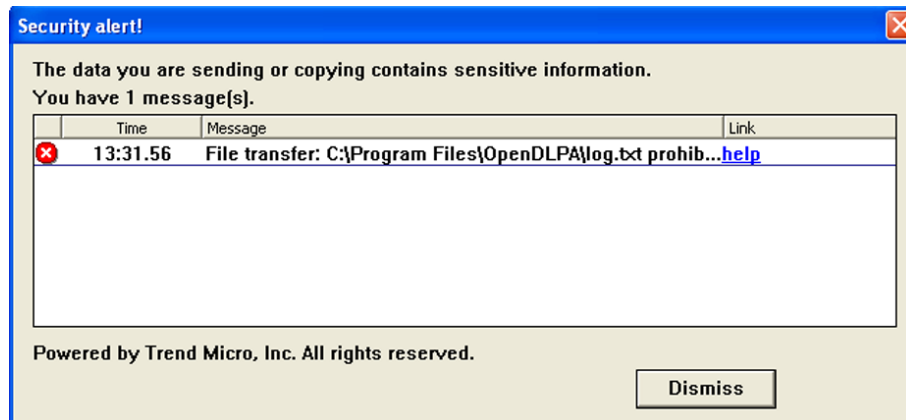


Figure 8: Trend Micro DLP blocking copying of the sensitive word "masteroppgave".

On a Windows system the clipboard process saves whatever is copied into memory [38]. A DLP endpoint agent can hook into this functionality and analyze all copied data that is sent to the clipboard. After removing text formatting, its detection methods are applied. Should the result violate any policies, the clipboard will be cleared and a warning issued (as illustrated in Figure 8 above). The warning in this case provides a link (here seen with the placeholder text "help"), which an administrator can configure to provide further information to the user in form of an educational webpage.

b. BLOCKING INPUTS AND DEVICES

To disable USB inputs various methods are used. One way of doing this is to disable the devices in Windows' device manager. But if more functionality is needed, such as allowing employees to use company issued USB-drives, disabling the USB controller is not a viable solution. A more practical solution is to hook into the USB mass storage library and as soon as a new device is discovered, the ID can be compared against a list of allowed devices (see Figure 9, next page). The device is allowed to mount if it matches the ID of a company issued USB drive. If not, the device is either ejected or mounted as read-only.

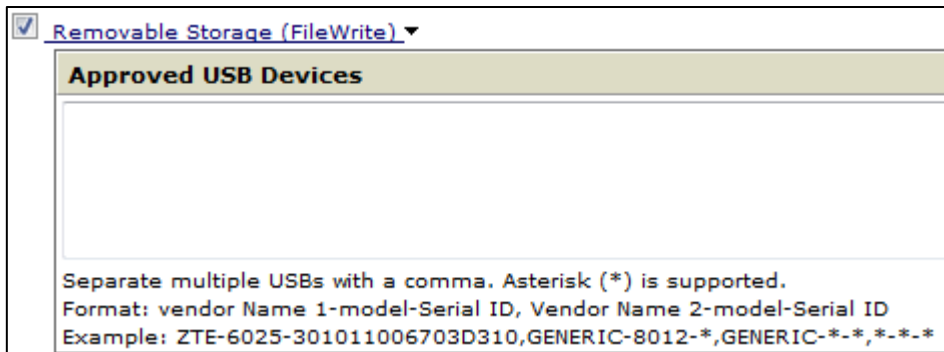


Figure 9: Trend Micro USB device configuration.

The same principles can also be applied to other I/O interfaces.

c. BLOCKING APPLICATION INPUT

Many DLP products advertise protection of data channels provided by popular communication applications such as Skype or Yahoo Messenger. In this case the DLP monitors all files the given process (e.g. skype.exe) accesses, and when sensitive files are involved, all read access is blocked (see Figure 10). Depending on the DLP product, this does not necessarily affect the text input of the program, meaning that sensitive data can still be sent by manually typing it in. On the other hand, network traffic containing sensitive data can still be intercepted by the network DLP and get blocked at this point. This is of course only relevant when the traffic is unencrypted, which does not apply to Skype traffic.

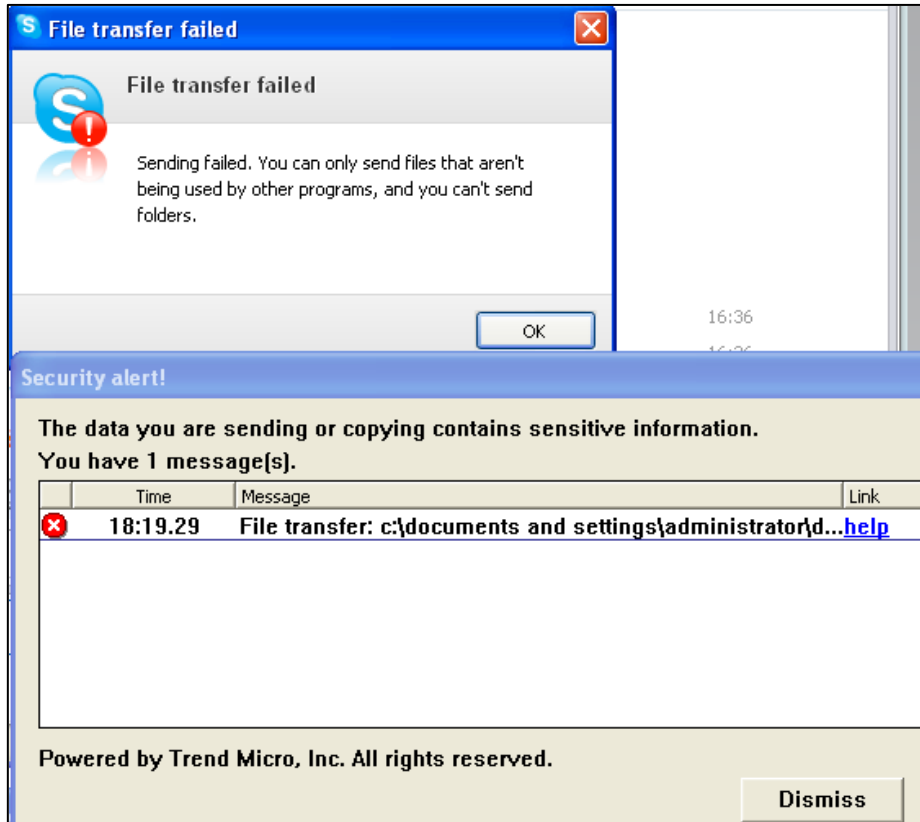


Figure 10: The DLP blocks Skype from loading a sensitive file.

d. PROTECTING E-MAIL

DLP offers two ways of protecting e-mail communication; integration with the mail server and endpoint integration with the local e-mail client. Figure 11 below illustrates a typical scenario where a local user sends the e-mail (marked as 1) to the mail transfer agent (MTA), using for example SMTP. Before sending the message out of the network, it is first forwarded to the DLP server for inspection (marked as 2). The DLP can then block or allow the message depending on the sensitivity of the content. How this integration is done varies between vendors. In many cases the DLP can even encrypt or mark messages for encryption as this is a requirement by many businesses [39].

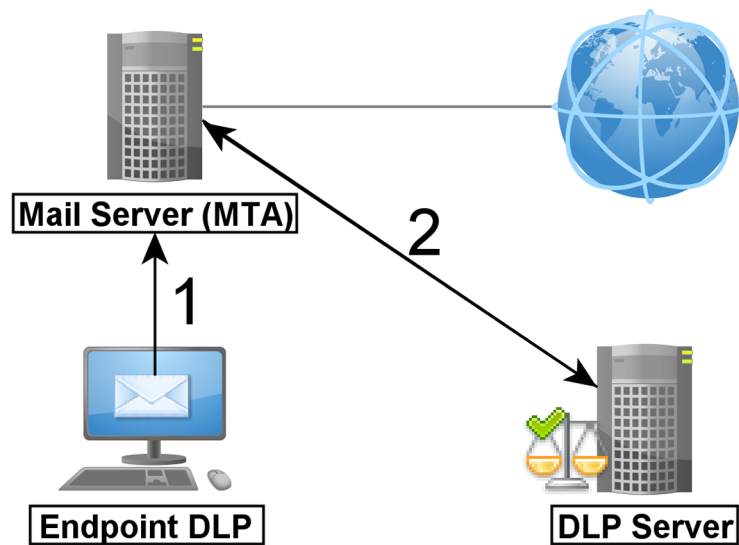


Figure 11: Analyzing e-mail communication.

Endpoint DLP have many ways of enforcing policy in regards to e-mail. It can integrate with the local MTA and enforce policy on messages here, it can block the mail application from attaching sensitive documents (in this case sensitive message content is not blocked), or better, it can be integrated with a mail application plugin [40].

Figure 12 on the following page shows additional options to e-mail DLP. If a company wishes, all e-mail to certain domains can be blocked (blacklisted) or allowed (whitelisted). By whitelisting domains the DLP content scanner is bypassed. In this case one has to be careful when different levels of sensitivity within the same domain exists.

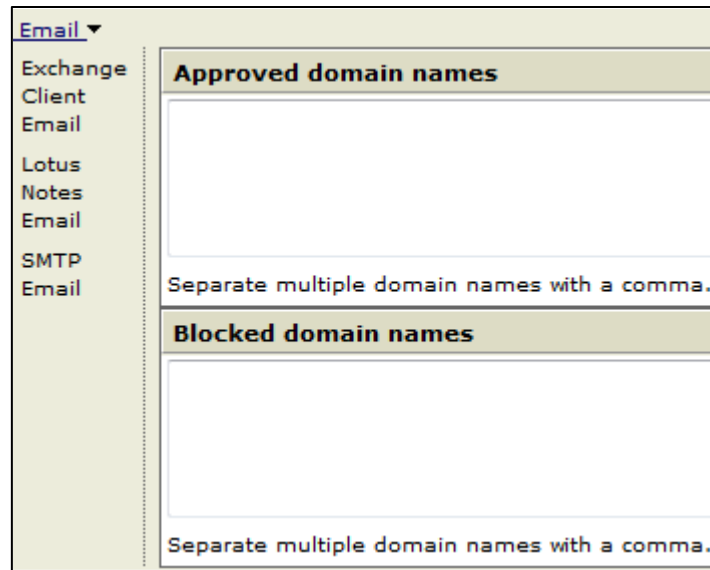


Figure 12: If desired, sensitive e-mails can be allowed to specific domains.

2.3.4.3 AUTOMATIC DATA REMOVAL

When defining a policy in the DLP, you first define a detection rule, and then an action that is to be taken if the detection rule is triggered. This action depends on the DLP product, but some common ones are listed here:

- Block: The operation related to the sensitive data is blocked.
- Allow: The operation related to the sensitive data is allowed.
- Alert: An alert dialog is displayed when the user triggers the DLP.
- Encrypt: The data is automatically encrypted when it is for example saved to a USB drive or transferred to a specific location.
- Justify: The user has to give a reason when the DLP is triggered. If it was accidental the user can cancel the original operation.
- Log: Logs the incident.

All of the above actions can be combined so that when the DLP triggers, multiple actions are executed. Having these actions makes it easier for users to work where a DLP is in place. In a situation where you trust the user, having the “justify” option enabled, means accidental leaks can be avoided, while still allowing sensitive documents to be legitimately sent and false positives to be bypassed. In large complex environments, the actions can be set up on a per group basis, so as to be as little intrusive to the employees as possible while still providing data protection.

The article from infosecurity (used with permission) on the next page is a good example of how a DLP system can help security, while not interfering with company workflow [41].

British Waterways: Keeping data privacy watertight [41]

British Waterways is responsible for the canal and river network in Britain, looking after a 2,200-mile network used by 13 million people each year. British Waterways also issues licenses for 40,000 boats and moorings, generating an income of \$37 million (£23m).

This licensing role means that the organization is handling both sensitive personal data – such as boat-owners’ identities – and the credit card and banking data it needs to process payments.

Boat owners still complete license applications using paper forms, and British Waterways uses scanning technology to convert the paper forms to computer records, and a PDF-based workflow to manage documents.

From a data protection standpoint, however, this raises a number of issues. Scanning in whole forms and storing them electronically is efficient, but it could place license holders’ data at risk by increasing the number of employees who can view it. It would also require British Waterways to treat the entire application form as sensitive data.

To reduce this risk, British Waterways has a multi-step approach to data loss prevention. Firstly, when the scanning system, supplied by vendor Kofax, captures a license form, it recognizes personal details such as addresses, and financial information.

This data is then extracted for separate processing, and redacted – or blanked out – from the electronic facsimile of the forms.

“We are very proactive in removing sensitive information from customer documents as they are processed, and before they are stored”, says project manager Keith Lester.

“The software is clever enough so that, with our own forms, bank or credit card information is redacted before the forms are processed...that is converted to a PDF that also shows only the redacted image.”

A very small team of staff at British Waterways is responsible for keying in credit card details to a commercial credit card system and for setting up direct debits (recurring transactions) from banks.

Document security procedures go hand in hand with strict policies governing other use of confidential data. Data protection policies are documented, with non-compliance viewed as a disciplinary offense.

Emails with sensitive data are encrypted, as are communications with other organizations or institutions. Staff laptops are also encrypted. “Our regulations for information security and data protection are part of our employment terms”, says Lester.

However, by automating security measures – such as redacting sensitive data – and encrypting laptop hard drives by default, the organization aims to cut down on human error too. “Security is largely automated and hidden from the user, so it’s not something they have to think about every time they need to do something”, he says.

2.3.4.4 LOGGING AND FORENSIC REPOSITORIES

The incident database logs when policies have been violated and when any other configured logging events occur. The amount of information saved per log entry varies across products. The most common details includes the location of where the event happened, time and date, what happened and what action was taken (block, allow etc.). If the system is connected to a directory service such as LDAP or Active Directory, usernames of the logged in person at a given machine can be linked to each violation. Other logs, such as system events, are usually stored separately from DLP incidents.

The generation of reports can be done with the logs gathered. In many cases the DLP systems comes with their own account that only have access to this functionality. This means access can be given to managers, HR representatives and others, while still restricting policy and endpoint management to certified personnel.

Many DLPs also come with a forensic repository. This is an encrypted database separate from the incident database which logs and stores all sensitive data related to a policy violation. This includes complete e-mail messages with file attachments, clipboard contents or the whole file which was transferred. The context in which the violation occurred (e.g. username, source and destination) is also recorded. For DLP endpoints the forensic data is usually captured first then uploaded to the management server. Somewhat problematic is the storage of all this information, especially if high amounts of false positive are detected. An attacker could fill up the DLP server storage by triggering enough policy violation and if the oldest entries are deleted to make room for new ones, the attacker can fill the repository with “sensitive” dummy files to remove any traces of a real data breach.

2.3.4.5 DEALING WITH ENCRYPTION

To ensure secure communication over the internet, it is essential that companies take advantage of some form of encryption. A recent trend is that web services redirect users to their TLS protected page (HTTPS), which in return offers better protection of the data being transmitted. For companies, using HTTPS and other encrypted communication channels offers a secure way of transmitting sensitive data. From the viewpoint of DLP, encryption can be difficult to work with.

Dealing with encryption is especially problematic for the network DLP, which cannot analyze encrypted data in a comprehensible manner. If sensitive documents are leaked through an encrypted connection, the DLP will be unable to block it. For endpoint DLP the same applies to encrypted files. If a file is encrypted, the DLP will not be able to read the contents and enforce policy. The good news then is that loss of sensitive data via encryption is relatively rare in practice. In Trustwave’s global security report for 2010 and 2011 only a few of 420 incidents used encryption or some sort obfuscation for their data exfiltration. None of the victims involved had a DLP in place, which would most likely have picked up on this unencrypted data exfiltration and issued a warning [42] [43].

For secure communication it is possible to use proxy servers to intercept the communication. Basically, the proxy, or network DLP in our case, acts as a reverse proxy and

launches a man-in-the-middle attack as soon as a client wants to establish a connection with an HTTPS server [29]. Normally, the client's browser would issue a certificate warning, but since it is configured with the company's certificate authority (CA), the certificate is trusted. A drawback to this is that any browsers missing the CA in its configuration will still issue a warning. Additionally, the DLP will still not be able to interpret any files that were encrypted or obfuscated before transmission.

With endpoint DLP the action can be blocked before the document is encrypted and transmitted. Basically, any file access to an untrusted program (e.g. encryption tool) can be denied. Of course, not giving local users permission to install and use third party tools will go a long way in combatting this problem. As the next sub-chapter demonstrates, users can still obfuscate documents if they have write privileges to them, so being able to encrypt files or the communication is not necessarily required.

We have on multiple occasions mentioned how DLP can automatically file encryption if it is detected as sensitive. There are two common ways of doing this. The first one is where the DLP is responsible for the encryption. The encryption keys used are provided by the DLP or administrator and distributed to the endpoints. When a file is stored somewhere where policy requires encryption, the DLP encrypts with the key provided. The other way is for the DLP to mark a file for encryption and delegate the task to a third party program. An enterprise Information Rights Management (IRM) system is a typical example of such a program. Basically, the DLP discovers sensitive data and instructs the third party IRM to encrypt it for extra protection [44]. If the data is stolen, it will be worthless to the attacker if the encryption keys are unknown. This of course is an additional incentive for the attacker to attack the DLP system directly, as the encryption keys might be uncovered.

2.3.5 FEATURE EXAMPLES FROM BYPASSING A DLP

In chapter 2.2 we concluded that DLP is highly effective at preventing accidental data loss. For internal and remote attacks a DLP is only effective if the attacker lacks administrator rights, or the documents involved are non-writable. Let's see why.

1. Attacker has administration rights: With these rights the attacker can either uninstall the endpoint agent, or install an application that encrypts the contents before it is sent (or both).
2. The attacker has file-write access: To send a document that bypasses all DLP rules, the attacker opens the document and uses the find and replace ability in the text editor to replace every instance of the letter "e" with strings such as "a#()i". Doing the same with a number, such as "2" ensures that any SSN or CC number rules are not triggered. To de-obfuscate the document, the attacker reverses the process. In case of image files, the attacker can convert them to another format, e.g. jpeg to png. The drawback of this method is the considerable amount of time it takes when dealing with larger amounts of files, even though office macros and similar tools can help in automating it.

The risk of the first attack is not considered high, as it is best practice to limit user privileges on a local workstation. The second attack has a much higher risk involved as write access is fundamental to doing one's work. Reducing the risk is possible, but requires restricting the workflow of the user. For example, a document can be tagged as confidential when it matches a DLP policy and keep this tag for its remaining lifetime, even if the sensitive parts are later removed or obfuscated. Still, it should be said that a DLP will never stop a truly dedicated and technical inclined person from performing data theft. The risks must therefore be calculated accordingly.

2.3.6 TECHNOLOGICAL SHORTCOMINGS

Many of the limitations mentioned so far can be summarized as follows:

- Monitoring encrypted data and communication is only possible when the DLP has access to the key or controls the key exchange in some way. If the DLP does neither, it will be unable to inspect encrypted data.
- To effectively protect different files the DLP needs to be able to parse them. If an unsupported file format containing sensitive content is used for data theft, the DLP will not be as reliable in detecting this compared to when common file formats are used.
- For gateway DLPs, the ability to detect different types of network traffic depends on what the product supports. Popular protocols, such as HTTP, HTTPS, SMTP and FTP can usually be monitored, but other more "exotic" protocols can be used to circumvent the DLP.
- DLP does not stop people from photographing the display, memorizing information or stealing the hard drive. To prevent this, other types of security procedures must be implemented.

2.3.7 ALTERNATIVE USES

With the functionality included in DLP products, more unorthodox uses are also available. We have summarized a few of them here:

a. AUDIT OF SENSITIVE ASSETS

By defining and setting up DLP policies, the system can be used to scan file shares, databases and other storage mediums on a corporate system to assess where sensitive files are located. The results can be used to evaluate if the organization's current treatment of sensitive data is in accordance with company policy and/or external regulation [10]. When sold as a service, the DLP can be set up to perform the assessment, and then be removed again when the job is complete.

b. PERSONAL ANTI-VIRUS

A DLP doesn't necessarily need to be used exclusively for detecting sensitive content. In situations where malware is discovered, but not detected by anti-virus software, the malicious files can be fingerprinted by the DLP and used together with a "quarantine on discovery"-policy. File names ending in .jpg.exe or .pdf.bat, can also be added to such a policy for better

protection. Keep in mind that a DLP will never be as effective (or ineffective) as commercial anti-virus software, and should not be considered as a replacement. If an employee mass renames files to one of the unwanted file endings, the consequences could be dramatic. Still, the same employee could also delete the files directly, which means the DLP system does not necessarily make things worse.

c. KEEPING TRACK OF EMPLOYEES

With its comprehensive logging functionality and directory integration, DLP can also be used in keeping track of productivity and general behavior of employees. Management can secretly be alerted when an employee accesses certain resources or sends e-mail containing swear words or similar. Additionally, timestamps can be used to map the productivity of workers as they access files. Since this is not the main idea behind DLP, such logging would not be very reliable for this use.

d. STEALING SENSITIVE ASSETS

By turning the table, an attacker can use the same method as used when auditing for sensitive assets, to steal sensitive data. Because the DLP automates file scanning task, much time will be saved in discovering where sensitive data is located. With a bit of work the attacker can configure file-crawling agents to also transfer all sensitive data it comes across to a given location.

Using a DLP this way is not unheard of [8] and is one of the main problems that the next chapter will explore.

3 VULNERABILITIES IN DLP SYSTEMS

Chapter 2 established the responsibilities and inner workings of a DLP system. From this chapter on the focus will change from how DLP works and protects sensitive assets, to how an attacker can take advantage of the DLP system and use it to speed up data exfiltration. You might ask yourself, why would an attacker want to do this in the first place? Well, let's take a look at some of the things a DLP does:

- Keeps a database of common patterns found in sensitive text documents, as well as hashes pertaining to sensitive files. In some cases copies of sensitive data are stored on the DLP management server.
- All DLP endpoints are managed by the server through definition and software updates.

Now, let's review these same points from the perspective of someone malicious:

- By gaining access to **one** server I can gain the knowledge of how to effectively identify sensitive files, their location, and if I'm lucky, the files themselves.
- By changing the definitions I could get the endpoints to delete critical system files or instruct them to transfer sensitive files to the server or a remote location. By tricking the software update mechanism I could replace the endpoint software with malware containing key loggers, bot-functionality, remote shell etc.

Clearly, the DLP system itself must be hardened enough to withstand such a threat, but is this applied in practice? Chapter 3 is composed of a chapter 3.1, which describes a threat scenario concerning an external attack against the DLP system of an imaginary corporation. Chapter 3.2, gives some background info about the evaluation of MyDLP and Trend Micro DLP, which is presented in chapters 3.3 and 3.4, respectively. Chapter 3.5 demonstrates how a remote attacker could gain access to the DLP management server and perform an attack based on the threat scenario given in chapter 3.1. All results are summarized in chapter 3.6.

3.1 THREAT SCENARIO

Definition:

A threat scenario is a chain or series of events that is initiated by a threat and that may lead to an unwanted incident [45].

3.1.1 BACKGROUND

The ACME Corporation has for the last decade increasingly moved more of its research and development over to a digital environment. Blueprints are no longer scattered across the floor, and documents are not carried back and forward between participants for minor edits and corrections. You could say ACME is now part of the 21st century and reaps the benefits provide with information technology in the form of collaboration and communication tools. Of course, the transition to the digital age does not come without its drawbacks.

Following news reports of cyber terrorism and the theft of intellectual property, the management has become more aware of the risks associated with keeping the company connected to the internet. Even with firewalls and anti-virus software in place the management feels something must be done to protect their precious data. What was once blueprints and office documents covering office floors are now digital CAD-drawings, PDFs and Excel spreadsheets scattered across file-shares, external hard drives, CDs, thumb drives, local machines, databases and so the list goes on.

To get control over the ever growing amount of digital files ACME hires a security consultant. After reviewing the situation it is decided to invest in a DLP solution. The DLP is configured to ensure that classified files are only stored at designated locations and that they are not accidentally attached to e-mail or uploaded to various websites. Employee training has also been conducted, and although some initial snags were met, they have now gotten used the more secure workflow as promised by the DLP product.

Somewhere in the Asia-Pacific region a well-financed group has received intel about ACME playing a great role in the development of a new type of military gadget. Getting their hands on this technology will greatly help them in some way. Of course sending a spy over would be a tedious tasks filled with ramifications if the operation should detected. Instead, a criminal hacker is hired to steal all of ACMEs sensitive research.

3.1.2 THE THREAT MATERIALIZED

The story presented so far gives an example of company using DLP. Having a DLP does not necessarily have any effect on reducing the potential threat, but might limit the impact of an attack. Since one of the goals of this thesis is to discover weaknesses in DLP software that can assists attackers in stealing data, and if possible create an exploit to demonstrate said weaknesses, we have developed a scenario that form the basis of the attack later demonstrated in chapter 3.5.

After some trial and error, the hacker finally manages to get access to a workstation on ACME's own network (as seen on Figure 13, next page). At this stage the attacker has already

gone through the steps *Intelligence Gathering*, *Vulnerability Research*, *Exploit* and *Control* in the attack process (explained in chapter 1.4) and installed a remote shell (illustrated with a skull head on Figure 13), but accidentally, the presence of an endpoint DLP client is discovered. The attacker finds this quite interesting and investigates how this can be of help to the operation. As a result a new attack process is created where the new target is the DLP system.

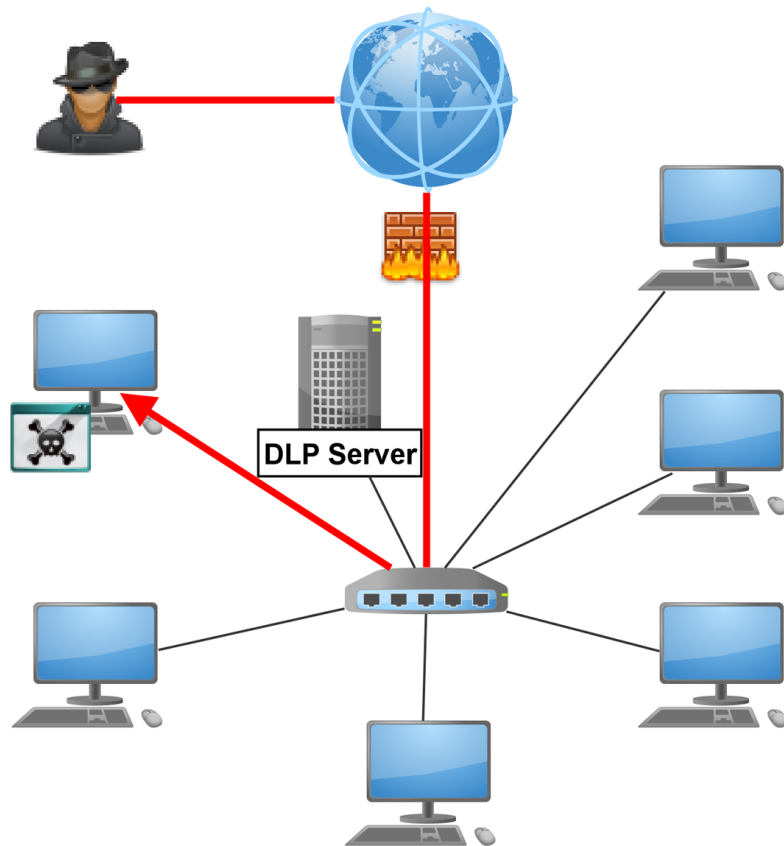


Figure 13: Step 1 - Company workstation compromised.

After some investigation the attacker concludes that the DLP should be disabled so as not to interfere with the attack. Additionally, the attacker wants to see if access to the system can be obtained. Having this might help in gaining more knowledge about the sensitive data of the corporation.

The attacker disables the endpoint DLP and moves on to target the DLP server. The web interface and other services running on the server are discovered. After probing for vulnerabilities a weakness is discovered. This is enough to gain access, escalate system privileges and compromise the server (Figure 14, next page).

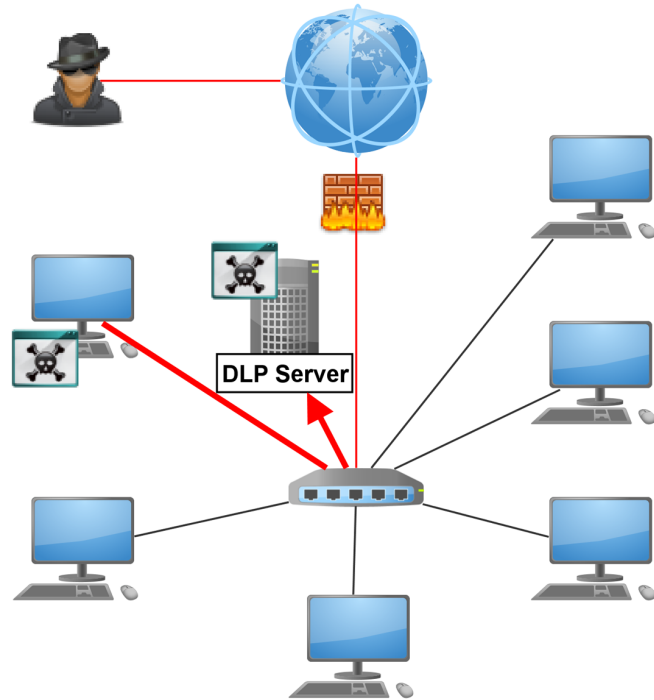


Figure 14: Step 2 - DLP management server compromised.

With full control of the DLP Server, the attacker targets the auto-update mechanism to deploy a malicious payload that is executed on all DLP endpoints found on the network (Figure 15). By using the search patterns defined by ACME's DLP policies, the attacker can now automate data gathering by retrieving any file matching these pattern. To lessen suspicion, the DLP endpoints upload this data to the DLP server before being encrypting and exfiltrating it to a remote destination.

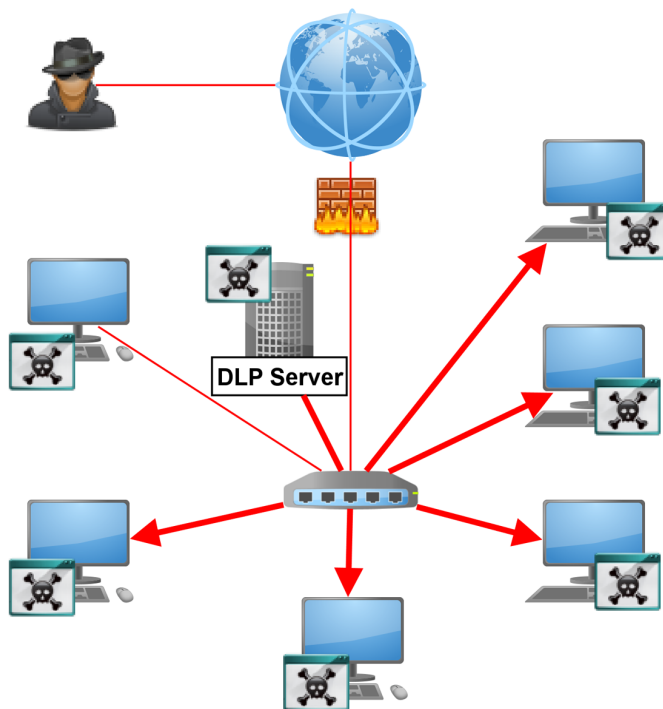


Figure 15: Step 3 - DLP endpoint agents compromised.

3.1.3 SUMMARY

In this threat scenario we assume no other security is in place to detect or mitigate such an attack, or the attacker is knowledgeable enough to circumvent such systems. The main point we are trying to make is that when a DLP system is present it is possible to accelerate the attack and discover even more sensitive documents.

The scenario can have a serious impact on the damage done. Not only are sensitive documents gathered and stolen, but it is done in such an automated and quick way, that any opportunity to interrupt the attack is limited by a small time frame. As such the risk associated with having a DLP must be calculated to see if the benefits outweigh any drawbacks. But before performing any risk evaluation, it is important to investigate and see if such an attack can be performed in practice.

3.2 FINDING THE FLAWS

In the next two chapters we have evaluated MyDLP (chapter 3.3) and Trend Micro DLP (chapter 3.4) to see how secure these products are. Some of the discovered vulnerabilities are used to demonstrate an attack against one of these DLP systems in chapter 3.5.

To gain access to a DLP system various attack vectors can be used:

Web-application: Most DLP systems are administered with a web application that provides a user interface for management controls. This application, as well as the technology used to serve it, makes for potential targets that can be exploited.

Maintenance channel: This is the channel where the DLP management system communicates with the endpoints agents to offer policy updates, software updates, log exchanges etc.

Misconfigured and outdated services: DLP servers have in many cases also additional services that are remotely accessible. A good example is the DLP proxy server that in some cases runs on the same server as where management is done. If this or other services are misconfigured or out of data, an attacker can use this to gain access. The same can also be said if any of these services are misconfigures or set up with default credentials. What OS is used also comes into play, as many popular server OS have multiple known vulnerabilities.

As was mentioned in chapter 1.4, the focus of this thesis is to evaluate the web application that is used to deliver the management console. However, other services critical to the DLP were also evaluated and in some cases exploited.

3.2.1 TEST ENVIRONMENT

In all evaluations, the products were installed in a virtual environment. The virtual environment consisted of a virtual network with all virtual machines (VMs) connected to it. The vendor install instructions were followed to setup the DLP management servers. Additionally, two Windows XP VMs were configured to test endpoint protection and find out how they interact with the management server.

Testing the web application was done from the host machine running Windows 7 and a separate VM running BackTrack 5R2 [46]. Various free security tools were used during the whole evaluation. These are mentioned throughout the next two chapters in the context of the results they produced.


References to server IP or domain is replaced with the placeholder text “{ServerAdr}”.

3.3 EVALUATING MYDLP

MyDLP is an open source DLP solution available in two versions; MyDLP Enterprise Edition and MyDLP Community edition. Of these versions, we evaluated the freely available Community Edition. While the Enterprise version receives updates first, no known security updates have been released to either version [47].

We installed MyDLP CE Virtual Appliance i386 (VMX) Version 1.0.0 RC 1, released March 17th, 2011; a pre-installed version meant for running in a virtual environment [48]. The server OS is Ubuntu 10.04.2 LTS released February 18th, 2011 [49].

After the VM was up and running in VMware, the applications and daemons as seen in Table 1 were reported by the “netstat” command to be listening on various network ports.



Tested version: 1.0.0 RC 1 (Community Edition) release January 2011

Source model: Open Source

License: GNU General Public License v3

Components: Endpoint protection, network monitoring

Detection: Wordlist, regular expressions, fingerprinting

Webpage: <http://www.mydlp.com/>

Name	Port	Version	URI	Known Vulnerabilities
Apache HTTP Server	80, 443	2.2.14 (Ubuntu)	http://apache.org/	17 ²
MySQL	3306	5.1.41-3ubuntu12.10	http://www.mysql.com/	38 ³
SMTP Proxy	10026	1.0 RC1		
OpenSSH	22	5.3p1	http://www.openssh.com/	2 ⁴
ICAP ⁵	1344	N/A	http://icap-forum.org/	
Squid	3128, 3129, 58039	3.0 STABLE19	http://www.squid-cache.org/	6 ⁶
EPMD	4369	N/A	http://www.erlang.org/	
MyDLP beacon	15151 (UDP)	1.0 RC1		
VMware vami-sfc	5488, 5489	N/A	http://www.vmware.com/	

Table 1: MyDLP - Running network applications and daemons

Apache HTTP Server was up and running to deliver the content of the web application. Even though port 80 is open, only port 443 is used for accessing the administration web application. MySQL listens to localhost only and is therefore not accessible remotely. Port 22 is open for remote administration over SSH.

² http://httpd.apache.org/security/vulnerabilities_22.html

³ http://www.cvedetails.com/vulnerability-list/vendor_id-185/product_id-316/version_id-91595/MySQL-MySQL-5.1.41.html

⁴ <http://www.openssh.com/security.html>

⁵ Squid module used for traffic monitoring.

⁶ <http://www.squid-cache.org/Advisories/>

The Squid proxy with its ICAP (Internet Content Adaption Protocol, see chapter 2.3.2.3) module for content inspection was also up and running. This means network monitoring is done on the same server as the DLP administrative functions. The MyDLP beacon is an auto discovery daemon that ensures zero-configuration of endpoint agents. Lastly, the VMware component is present since the server is running as a virtual machine.

The known vulnerabilities are of mostly medium and low risk. In the case of Apache HTTP Server, many of the critical vulnerabilities require modules, such as “mod_proxy” or “mod_isapi”, to be enabled to even function. In MyDLP these modules are disabled. For MySQL many of the vulnerabilities are undisclosed or require an authenticated session to be exploited. A large part of the vulnerabilities are denial of service (DoS) attacks, which are not ideal for gaining local privileges on the administration server. The impact of DoS attacks on the DLP infrastructure is further discussed in chapter 4.5.

All management is done from a web application programmed in Flash ActionScript 3. This application interacts with a PHP backend with the help of Adobe Flex (as of November 2011 renamed to Apache Flex [50]) and the Action Message Format (AMF) that it uses. Additionally, various forms written in PHP and JavaScript are used to upload content to the server. The web application uses the following frameworks and libraries:

Name	Installed version	Latest version	URI	Known Vulnerabilities
Doctrine	1.2.2	2.2.1	http://doctrine-project.org/	1 ⁷
PhpSysInfo	3.0.0	3.0.16	http://phpsysinfo.sourceforge.net/	
Symfony	1.3-1.4	2.0.12	http://symfony-project.org/	1 ⁸
Zend framework	1.10.6	1.11.11	http://framework.zend.com/	2 ⁹
adodb5	5.11	5.16	http://adodb.sourceforge.net/	2 ¹⁰

Table 2: MyDLP - Web technologies used

Doctrine is an object relational mapper (ORM) that manages all queries between PHP data objects (PDOs) and the MySQL database in a persistent manner. As Doctrine has been developed for several years it specifies a secure way of creating queries (prepared statements) in its own doctrine query language (DQL). As long as this is followed and the web application developer does not insert input directly in the SQL command, the application will be safe from SQL injection.

PhpSysInfo is a simple script for retrieving and displaying system information (disk space, memory usage etc.). This information can be extended with the help of plug-ins.

Symfony is a framework that according to their website “[speeds] up the creation and maintenance of your PHP web applications.” In MyDLP only the YAML component from

⁷ <http://www.doctrine-project.org/blog/doctrine-security-fix.html>

⁸ http://trac.symfony-project.org/browser/tags/RELEASE_1_4_12/CHANGELOG#L155

⁹ <http://framework.zend.com/security/advisories>

¹⁰ <http://phplens.com/lens/lensforum/msgs.php?id=19055>

Symfony is used. This component parses YAML strings to convert them to PHP arrays, or from PHP arrays to YAML strings [51].

The Zend framework offers the implementation of “web 2.0” technology into any web application that takes advantage of it. As an example the framework offers modules to interact with the APIs of multiple online services, which will save the developer a lot of work, since such integration does not have to be built from scratch.

ADOdb works as a database abstraction library for PHP and unlike Doctrine does not do any object relational mapping [52].

Although the number of vulnerabilities is much lower for these frameworks and libraries when compared to the applications in Table 1, they are easier to exploit. Most of the known exploits are cross-site script (XSS) vulnerability that can be used by an attacker for session fixation, session stealing or malware distribution. More of these vulnerabilities will be explained during this evaluation.

3.3.1 PENETRATING THE APPLICATION

Before evaluating the web application, we will first take a look at the security configuration of OS and applications.

3.3.1.1 IPTABLES

No configuration is present, exposing all services listening on a port. The manual does not instruct the administrator to configure iptables. There is also no other firewall software installed. It would be good practice to at least mention this in the documentation.

3.3.1.2 MYSQL AND THE DATABASE

The database used by MyDLP is MySQL version 5.1.41-3ubuntu12.10. This is the official Ubuntu build and any updates are offered through the Ubuntu software repository. Even though MySQL listens on port 3306 it only allows connections from localhost (127.0.0.1). As a result, it does not show up on port scans and can be considered secure even though iptables is not configured to block external communication to this port. The newest MySQL release is version 5.1.62, released March 21st, 2012 [53].

The database for MyDLP is called “mydpl” and is accessible with the username “root”, no password. For fingerprinting, no files are stored in the database, only the fingerprint and related file attributes. When a sensitive file is marked for fingerprinting, MyDLP generates a MD5 hash value from the full file; this means no partial file hashing is done. The database user account is the same across different installs of MyDLP. Users of the MyDLP administration interface are stored under “sh_user” with a user ID, user role, e-mail, username and the unsalted password hash.

If an attacker gets hold of the password hash, having the hash salted makes it impractical to perform a dictionary attack with a pre-computed rainbow table. This is the recommend approached when storing hashed credentials.

3.3.1.3 SQUID AND ICAP (INTERNET CONTENT ADAPTION PROTOCOL)

The proxy has two ports open; one for http and one for https communication. All traffic sent through these ports is automatically redirected to the ICAP content filtering server listening on port 1344 for inspection.

Depending on the configuration, Squid can also tunnel traffic to local services on the machine it is running with the CONNECT parameter. This would be a way of reaching the MySQL server, but in the case of MyDLP, such requests are rejected [54].

As a test we set up the proxy and accessed Gmail. When connecting to https over a proxy a certificate warning will be issued. In a real deployment you would want to add the proxy server's certificate to the company CA so it becomes trusted. If the proxy discovers sensitive data being sent (e.g. in an e-mail message sent with Gmail) the connection is blocked¹¹.

3.3.1.4 E-MAIL PROXY

As shown in Table 1 the SMTP proxy service listens on port 10026. This is used to redirect outgoing e-mail through the MyDLP content filter when used in combination with postfix for sending outgoing messages. As an example, the Postfix configuration file (`/etc/postfix/main.cf`), can be changed with the following command to add after-queue content filter functionality [55]:

```
sudo postconf -e "content_filter = dlp:[127.0.0.1]:10026"
```

When an e-mail is sent using the postfix SMTP server, the message is first redirected to the MyDLP server for content inspection. If no sensitive content is discovered, the message is forwarded to the original recipient.

3.3.1.5 CERTIFICATES AND ENCRYPTION KEYS

The secure certificates used by MyDLP when using HTTPS are store in the following locations:

```
/etc/mydlp/ssl/public.pem  
/etc/mydlp/ssl/private.pem
```

These are generated by OpenSSL at the same time as the HTTP server is set up. The presence of other certificates and keys is unknown.

3.3.1.6 APACHE HTTP SERVER

MyDLP runs Apache HTTP Server version 2.2.14 (Ubuntu) to deliver all web content. As of 21.01.2012 the newest version is 2.2.22 [56].

The PHP 5 module is running, as well as proxy and SSL modules. For PHP a SOAP extension is used to enable SOAP server functions. This enables the client to exchange information directly to the management engine, which is controlled by the web application.

¹¹ A video of how this work has been posted by the creators of MyDLP at <http://www.youtube.com/watch?v=yQTEW9RBMAY>

3.3.1.7 ENDPOINT AGENT

The endpoint connects to UDP port 15151 after install. To find the management server, a UDP broadcast is sent. By capturing the network traffic with Wireshark¹² while the endpoint agent is running, we found the contents of the server discovery messages. First the endpoint sends: “mydlp 1.0 client mydlpreq”. The server then responds through its MyDLP beacon service with the message “mydlp 1.0 server mydlpoffer {ServerAdr}”. The endpoint saves the server address to the registry so it can be further used as reference. Even when the IP is saved, the endpoint will continue sending the same broadcast every 6 minutes. Since no verification is done on the message exchange, an attacker can send out custom responses to the UDP broadcast and trick the endpoints into connecting to a malicious DLP server instead of the legitimate one.

Compared to Trend Micro DLP, MyDLP does not issue any descriptive errors when it blocks a user action. For example when copying a sensitive document to a USB-drive a standard Windows warning is displayed saying the file cannot be read. The user will not know that MyDLP blocked the request and might make other assumptions about what happened. If the user learns the DLP blocked their action, they might be more careful when handling sensitive data in the future.

The endpoint agent communicates with the server over HTTPS. All messages are sent as SOAP request with authentication. These messages include the exchange of policy updates and event logging. The address the endpoint agent uses was the one given after performing a UDP broadcast. The attack on this UDP broadcast can further be expanded to include a Man-in-the-middle (MITM) attack proxy, where all SOAP requests are sent to the attacker’s proxy before being redirected to the legitimate MyDLP server. This allows an attacker to eavesdrop and modify traffic during transmission.

Endpoint configuration files are found in the “C:\Program Files\MyDLP\conf” directory [57].

3.3.1.8 THE WEB APPLICATION

Chapter 1.4 presented the Web Hacker’s Methodology as a way of evaluating the security of web application. The checklist used is available in Appendix I. Following are the main highlights from this evaluation.

a. MAP THE APPLICATION’S CONTENT

With complete access to the source code, there was little that could be hidden from our view. Since the MyDLP source code and its compiled binaries are freely available for download, an attacker can search for vulnerabilities and test exploits in an emulated environment before attacking a real target. This can help in keeping the attacker hidden, as less malicious activity has to be generated when mapping the application.

There is little default content present. If you access the non-http site an “It Works!” page is shown. In this case the virtual host for port 80 has its document root set to the default

¹² Wireshark is a utility that captures all packets sent over a network interface [78].

“/var/www” directory. When accessing the server IP with HTTPS, the browser is redirected to https://{ServerAdr}/mydlp/ where the web console resides. For ADOdb, some test scripts were found under the “tests” directory. Although the scripts are not dangerous themselves, they do allow some interaction with the DB.

An automated scan with the open source web scanner w3af [58] revealed a small vulnerability disclosure. In any request that results in a 404 Not Found error, the server type, version and OS (Apache/2.2.14 (Ubuntu)) is displayed. Information disclosure like this can be useful to an attacker when searching for exploits. This information can be hidden by adding “ServerSignature Off” and “ServerTokens ProductOnly” to the apache configuration file.

Path	File	Parameter
\mydlp\file-upload\	upload.php	\$_REQUEST
\mydlp-web-manager\	class.FileUpload.php	\$_GET
\mydlp-web-manager\	create_pdf.php	\$_GET
\mydlp-web-manager\	documentation.php	\$_GET
\mydlp-web-manager\	js-fileuploader.php	\$_GET, \$_REQUEST
\mydlp-web-manager\	service.php	\$_GET
\mydlp-web-manager\adodb5\	adodb-pager.inc.php	\$_GET
\mydlp-web-manager\adodb5\	adodb-perf.inc.php	\$_GET, \$_POST, \$_REQUEST
\mydlp-web-manager\adodb5\	server.php	\$_REQUEST
\mydlp-web-manager\adodb5\perf\	perf-oci8.inc.php	\$_GET
\mydlp-web-manager\adodb5\tests\	test.php	\$_GET
\mydlp-web-manager\adodb5\tests\	test4.php	\$_GET
\mydlp-web-manager\adodb5\tests\	test-active-record.php	\$_GET
\mydlp-web-manager\adodb5\tests\	test-active-recs2.php	\$_GET
\mydlp-web-manager\adodb5\tests\	testdatabases.inc.php	\$_GET
\mydlp-web-manager\adodb5\tests\	test-perf.php	\$_GET
\mydlp-web-manager\adodb5\tests\	testsessions.php	\$_GET
\mydlp-web-manager\adodb5\tests\	tmssql.php	\$_GET
\mydlp-web-manager\file-upload\	upload.php	\$_REQUEST
\mydlp-web-manager\lib\sysinfo\	js.php	\$_GET
\mydlp-web-manager\lib\sysinfo\	xml.php	\$_GET
\mydlp-web-manager\lib\sysinfo\language\	language.php	\$_GET
\mydlp-web-manager\lib\Zend\	Session.php	\$_REQUEST
\mydlp-web-manager\lib\Zend\Auth\Adapter\	OpenId.php	\$_GET, \$_POST
\mydlp-web-manager\	bckprstr.php*	\$_REQUEST
\mydlp-web-manager\	install.php	\$_REQUEST
\mydlp-web-manager\	js-fileuploader.php	\$_REQUEST
\mydlp-web-manager\	quarantine.php*	\$_REQUEST
\mydlp-web-manager\	schema.php	\$_REQUEST
\mydlp-web-manager\	upload.php*	\$_REQUEST

Table 3: Files with PHP \$_GET, \$_REQUEST or \$_POST parameters.

As most of the backend content is written in PHP, it is of great interest to the attacker to find out where any form of user input can be given. The files in Table 3, previous page, accept user defined input through the PHP `$_GET`, `$_REQUEST` or `$_POST` parameters.

For many of the files marked with * in Table 3, the user must have an authenticated session established in advance before any input is processed. An example of this is found in the file “bckprstr.php”, which is used to import and restore MyDLP backups:

```
if(!$GLOBALS['DEBUG']      &&      (!isset($_SESSION['user_role'])      ||  
$_SESSION['user_role'] != 0)) {  
  
    error_log('Unauthorized access - Backup - Restore');  
  
    exit();  
  
}
```

As can be seen, authentication can be bypassed if MyDLP is configured with `register_globals` enabled in the PHP preference file (`php.ini`). All an attacker has to do is add the GET parameter “`?DEBUG=1`” to the end of the URI before loading the page. In its standard configuration, MyDLP has “`register_globals`” set to disabled, which means debug mode cannot be enabled.

Some of the files listed in Table 3 will be mentioned again as we work through this evaluation.

b. ANALYZE THE APPLICATION

In the main web directory (locally `/usr/share/mydpl/`) for HTTPS connections, the MyDLP web console can be found. There are two sub-directories here: “`mydpl`” and “`mydpl-web-manager`”. The main application “`index.swf`” is found in the first directory, while libraries and frameworks are located in the latter. The management application itself is written in ActionScript 3 and runs as a Flash 10 applet. For calls to the various libraries Action Message Format (AMF) is used together with a Zend AMF Server. AMF is a special wrapper for all requests sent from the flash applet running in the browser to the server. The format is loosely based on SOAP and uses HTTP to send the request in a binary format. All these messages are then interpreted by the Zend AMF server and forwarded to the various PHP classes.

After analyzing the directory structure and technologies used, the attack surface of the MyDLP web application looks as follows:

- Vulnerabilities found in various components used by the management console (see Table 2 for a complete list).
- Files that allow user input (see Table 3), but do not require an authenticated session for input to be processed.
- The flash based management application as it handles input for user login and account recovery.

c. TEST CLIENT-SIDE CONTROLS

No client side test is done on the username input when authenticating. Since the field specifies the username as “e-mail” it would be natural to validate that input is in fact an e-mail address. In the case of passwords, a client side check is done to control the length. When password length is less than 4 characters or more than 32 characters, a client side warning is displayed. Even with a warning displayed you are still allowed to submit the input, and as a result you also have to wait for the web server to respond before you can continue to use the application.

For account retrieval the e-mail submitted is controlled for correct formatting. If invalid, it is not possible to submit the e-mail and no requests is sent to the server.

The flash applet was further decompiled with HP’s SwfScan v1.0; a tool that scans the source code of flash applications to look for vulnerable patterns [59]. Examples of such vulnerabilities include storing credentials directly in the source code, left over debug code and incorrect handling of user input. In our case SwfScan gave only false positives when analyzing the MyDLP web console flash application. The de-compiler also allows manual analysis of the source code, but in regards to application functionality (including authentication, account creation) all requests are forwarded and processed by the server side PHP backend. The applet only acts as a GUI.

d. TEST AUTHENTICATION MECHANISM

The MyDLP management application is only accessible over TLS, which establishes a secure connection for client-server communication. When authenticating, the password submitted is hashed with SHA-1 client side before transfer to the server. No salting occurs. SHA-1 hashed versions of passwords are stored in the database, and with the help of Doctrine, the relevant SHA-1 value is retrieved and matched during user authentication.

In regard to username enumeration, the same error is displayed when trying various username/password combinations at the login screen. The account recovery function is more verbose and displays a different message if a valid e-mail is submitted. This makes user enumeration possible. If the user has its username set to the same e-mail address as can be enumerated, only the password needs to be guessed. It should also be noted that accounts are not blocked after x amounts of failed logins. This makes it possible to perform brute force attacks.

Another interesting observation is that usernames are not checked for uniqueness. This means two users with the same username can be created on the same server. This is a design flaw that is especially vulnerable if the web application allows user self-registration. In case the two users were to use the same password, either accidental or through a brute-force attempt, the second user might gain extra privileges originally reserve for the other. In MyDLP only the administrator can register new user and it is therefore highly unlikely that this will be a practical problem. As for new user account creation, passwords are required to be between 6 and 32 characters long. Even though the title for the username login dialog input is “e-mail”, the username is not required to be an e-mail address.

e. TEST THE SESSION MANAGEMENT MECHANISMS

User authentication is controlled by PHPSESSID, which is set in a cookie as soon as index.swf has loaded. Because of the nature of flash applets, should you accidentally reload the management console, you will need to log in again. In this case the previous session will not be invalidated (e.g. direct access to .php-files can still be obtained). The session is only linked to the PHPSESSID cookie, which after a successful authentication has the following user specific values associated with it: user ID, username, role type and customer ID. If an attacker steals the session, there is no security controls to check if the source of the request has changed (e.g. recording the IP address when first establishing the session and comparing). There is no specific timeout set in the cookie, and the session timeout relies on the default PHP settings, meaning the session will be cleared if it is removed by the garbage collector [60]. If an administrator leaves the web console running, the session will never time out as the Flash applet send requests to the server at a given interval, thereby renewing the session's age.

The file "Login.php" found under "{webroot}\mydlp-web-manager\controller" is responsible for authentication. When authenticated the \$_SESSION variables user_id, user_name, user_type and customer_id are set and used throughout the rest of the session. All session creation happens over a TLS encrypted connection.

Session fixation and session stealing is possible when using an XSS vulnerability. The attacker will in this case construct a malicious link that the victim clicks on that either sets the session cookie or steals it depending on the attacker's preference. To construct a malicious link, a JavaScript is embedded in the URI path and will be executed when clicked by the user. If done correctly the user will not become suspicious and perform an authentication that the attacker can use to access the restricted parts of the web application.

As soon as index.swf is loaded, the file "{webroot}\mydlp-web-manager\index.php" is also initialized. This establishes a Zend AMF Server session which is responsible for managing the Messages sent between the client-side Flash application and the server side PHP classes. This session is not authenticated, but the PHPSESSID cookie is forwarded with each request. If the cookie is missing, the request will be ignored.

Session termination is done in Login.php with the userLogout function. This clears the \$_SESSION variables and also terminates the Zend AMF session.

f. TEST ACCESS CONTROLS

MyDLP uses role-based access control with 3 different types of user roles: The admin, manager and tracker. The admin has access to all features, the manager has access to policy and log management and the tracker has access to only the logging facilities. When querying the database, an account value is returned. This is 0 for admin, 1 for manger and 2 for tracker. This value is controlled for each function. This means a manager cannot add a new user to the system since such functionality is reserved only for the admin user with id 0. All unauthorized attempts are recorded in the error log.

Even without authentication, access control to certain functionality can be obtained. The file “{webroot}\mydpl\upload.php” is used to upload sensitive files for fingerprinting. Access control is decided by the GET parameter “?type=”. If this parameter is set to 1 or 2 the user can access the functionality of this file. This vulnerability allows the attacker to upload a file that will reside in the server’s /var/tmp/ folder. The attributes of this file, such as name, upload date, description tags and hash value is inserted into the MyDLP database. If the attacker adds a fingerprint of a critical system file commonly found on endpoints (e.g. c:\windows\system32\hal.dll), the results might be disastrous depending on how MyDLP is setup to deal with fingerprinted files. An additional parameter, “?import”, can also be set, but in this case the upload fails as the target directory is non-writable.

A feature of the web console is that it displays system statistics of the server. This can be useful to ensure that there is enough free drive space and that the system is not overloaded. To deliver the content of these statistics phpsysinfo is used. In its default install the script retrieves system statistics and displays them on a webpage. In the case of MyDLP this webpage (/sysinfo/index.php) has been deleted and instead all the statistics are parsed from an xml file that can be retrieved from “https://{ServerAdr}/mydpl-web-manager/lib/sysinfo/xml.php?plugin=complete”. Since this file is publically viewable, an attacker can gather detailed statistics about the webserver by adding the correct get parameters.

No other vulnerabilities in regard to access control were discovered.

g. TEST FOR INPUT-BASED VULNERABILITIES

Testing MyDLP for SQL injection attacks is hard with automated tools. Even if the query results in an error, nothing is displayed to the user. This makes the use of automated tools difficult since blind SQL injection does not give any output that these tools can use to determine if an attack was successful or not. To determine how the input was received by MySQL, the query log can be monitored and database dumps analyzed.

By running all queries through Doctrine, MyDLP ensures that all interaction with the database is properly sanitized. The “upload.php” that allows anyone to add file entries to the database is still safe from SQL injection. Our own tests on this file only resulted in the database being populated with lots of useless data. Even then, the operation of the web application was not affected. The use of Doctrine also applies to all SOAP PHP object classes.

3.4 EVALUATING TREND MICRO DLP

Trend Micro acquired Provilla in 2007 [15]. The target of the acquisition was the LeakProof DLP product that Provilla had been developing. Since then the product has been rebranded and updated several times, but multiple references to Provilla are still present in certificates, source code and installers.

Today the product goes under the name Trend Micro DLP. The version we evaluated is version 5.5.1359 from November 2011 running as a CentOS 4.6 virtual appliance. For the rest of the chapter Trend Micro will be abbreviated to TM.

After completing the virtual machine install, the applications and daemons as seen in Table 4 were reported by the “netstat” command to be listening on various network ports.



Tested version: 5.5.1359 released November 2011

Source model: Closed Source

License: Proprietary commercial software

Components: Endpoint protection, network monitoring (requires appliance)

Detection: Wordlist, regular expressions, fingerprinting

Webpage: <http://www.trendmicro.com/us/enterprise/data-protection/>

Name	Port	Version	URI	Known Vulnerabilities
Apache Tomcat	8080, 8443 (Secure)	5.5.7.0	http://apache.org/	42 ¹³
MySQL	3306	5.0.68	http://www.mysql.com/	22 ¹⁴
OpenSSH	22	3.9p1	http://www.openssh.com/	13 ¹⁵
TM DLP Control	8804, 8904 (Secure)	5.5.1353		

Table 4: Trend Micro DLP - Running network applications and daemons

If the number of known vulnerabilities is scary, keep in mind that in addition to this the Java virtual machine (1.6.0_10-b33) that Tomcat and other modules run in, have 132 known vulnerabilities.¹⁶ Of course, not all of these vulnerabilities apply to this server configuration or are of critical nature, but they do make the job easier for an attacker to gain access.

Apache Tomcat is running to deliver the content of the web application. This is done over port 8080 (HTTP) and 8443 (HTTPS). Although the application is available through both the secure and insecure port, the browser is redirected to the TLS-secured login page when entering the application root (<http://{ServerAdr}:8080/dsc/>). As listed, the DLP uses a MySQL database, and since this is a Linux box, port 22 is open for remote administration with the help of OpenSSH.

¹³ <http://tomcat.apache.org/security-5.html>

¹⁴ <http://www.cvedetails.com/version/71374/MySQL-MySQL-5.0.75.html>

¹⁵ <http://www.openssh.com/security.html>

¹⁶ <http://www.cvedetails.com/version/82994/SUN-JDK-1.6.0.html> and <http://www.cvedetails.com/version/82972/SUN-JDK-1.6.0.html>

The most critical vulnerabilities of the Apache Tomcat server relates to directory traversal. This will be explored in section *g* on page 58. Apart from that there a several critical DoS vulnerabilities present and a couple of XSS vulnerabilities can be found in the included sample files and manager application. Compared to MyDLP the MySQL database contains less vulnerabilities as it is based on another development branch. The nature of the vulnerabilities is similar, with DoS attacks leading the way. In many cases remote authentication is required, which is already obtained if an SQL injection can be found. In this case the best approach would be to limit the permissions of the account used to communicate with the database on behalf of the web application.

Last on the list is OpenSSH. The amount of vulnerabilities here is quite high as the installed version 3.9p1 was released August 18th, 2004 [61]. The p in the version number stands for portable. A couple of the vulnerabilities only apply to the portable version, which makes this install more vulnerable than the non-portable version. Still, the vulnerabilities do not permit authentication bypass on a CentOS system and are of little use to us.

All management is done from the java-based DLP web application. This application interacts with the Struts framework to deliver content to the user. The web server is also used to deliver content to the endpoints. The modules used on the web server can be seen in Table 5 below.

Name	Installed version	Latest version	URI	Known Vulnerabilities
Spring Framework	1.2	3.1.1	http://www.springsource.org/	2 ¹⁷
Acegi Security ¹⁸	0.8.2	3.1.0	http://www.springsource.org/spring-security	
Dojo Toolkit	1.1.1	1.4.1	http://dojotoolkit.org/	5 ¹⁹
Apache Struts Framework	1.2.4	2.3.1.1	http://struts.apache.org/	6 ²⁰

Table 5: Trend Micro DLP - Web technologies used

The Spring framework is an application development framework for enterprise Java applications and is maintained by VMware [62]. In TM DLP only the security module (Acegi Security) from this framework is used. It should be noted that in more recent versions of the Spring Framework, Acegi Security has been renamed to Spring Security. The known vulnerabilities in Spring do not apply to the security module, as such TM DLP is not vulnerable from the known vulnerabilities as it only implements Acegi Security.

Dojo Toolkit helps in building rich internet applications. The toolkit focuses on delivering dynamic user interfaces that can easily be linked with the backend functionality of the application [63]. Most of the known vulnerabilities are XSS vulnerabilities included with

¹⁷ <http://www.cvedetails.com/product/20168/Vmware-Springsource-Spring-Security.html>

¹⁸ Part of Spring Framework – Now called Spring Security.

¹⁹ <http://spotthevuln.com/category/all-software/dojo/>

²⁰ <http://www.cvedetails.com/version-list/45/6117/1/Apache-Struts.html>

the sample and test files of this framework. Since TM DLP ships with these sample files, such an attack is feasible. More on these vulnerabilities will be mentioned later in the evaluation.

The Apache Struts framework aims to provide the missing pieces needed to create enterprise-grade applications that are easy to maintain over time [64]. Like many other frameworks Struts tries to simplify development by providing commonly used functionality that developers can tap into instead of developing themselves. Since Struts 2 is based on a product called WebWork 2, the architecture is different from Struts 1. The number of known vulnerabilities listed in Table 5 only applies to version 1, which is what TM DLP uses. This includes 4 XSS, 1 DoS and 1 bypass vulnerability. The bypass vulnerability allows validation to be circumvented when given a specific parameter that the application does not check for. This can allow unauthorized access to application functionality. Still, as will be explained in this evaluation, this and other vulnerabilities are only relevant after user authenticated through Acegi Security.

3.4.1 PENETRATING THE APPLICATION

Before evaluating the web application, we will first take a look at the security configuration of OS and applications.

3.4.1.1 IPTABLES

A configuration is present that allows incoming communication to the following ports:

- TCP port 22: used by OpenSSL for remote administration. Can also be used to administer the DLP through a CLI interface.
- TCP port 8804 and 8904. Used between DLP endpoint agents and server to communicate. The former port is insecure, while the latter uses SSL.
- TCP port 8080 and 8443. HTTP and HTTPS server ports used by Apache Tomcat.
- The UDP port 1558 is also open for remote connections. The netstat utility does not reveal anything listening on this port.

These open ports and their underlying services represent the attack surface of the DLP management server.

3.4.1.2 MYSQL AND THE DATABASE

The MySQL listens on port 3306. Even though the configuration allows for external connections, the port has been closed with iptables. The main database used by Trend Micro DLP is called “dgedb” and is accessible with the username “provilla” and password “Let!U@V#4728” in its default configuration. Changing the password is probably not a good idea as it is hardcoded several places in the software. For fingerprinting, no files are stored in the database. Users of the TM DLP administration interface are stored in table “dg_aa_user” with a user ID, e-mail and password. Instead of hashing and salting the password it has been encrypted. This means, with the correct key the original password can be recovered, a problem we will explore further later in this evaluation.

Unlike MyDLP, TM DLP does not save user passwords as hashes in the database. Instead they are stored in the format shown on Figure 16 below.

```
-- Dumping data for table `dg_aa_user`
--
LOCK TABLES `dg_aa_user` WRITE;
/*!40000 ALTER TABLE `dg_aa_user` DISABLE KEYS */;
INSERT INTO `dg_aa_user` VALUES (1,'admin','Administrator for LeakProof','bRPdfv
NQXkkTWDDiacdhsQ==','xxx','xxx','admin@dgatetech.com',NULL,5,'2012-01-31 10:33:3
6','2012-02-01 14:51:24',1);
/*!40000 ALTER TABLE `dg_aa_user` ENABLE KEYS */;
UNLOCK TABLES;
```

Figure 16: Database user table

The string is clearly base64 encoded as it ends with “==”; a form of padding that ensures the whole string is divisible by 3. When decoding the base64 encoded password string, the output is gibberish. If it was a base64 encoded hash, we would have expected a readable string output, as hashes are often represented as common printable characters when generated. Another option is that we are dealing with an encrypted string that was base64 encoded before being transferred to the DB. Representing a string in base64 ensures illegal characters that might lead to processing errors are not present.

To investigate further the java classes responsible for encrypting passwords were analyzed. By doing this we hoped to uncover either the encryption key, or the hashing algorithm used. After some digging the following method was discovered:

```
private static String[] getDefaults()
{
    String k1 = "NwRdIFb0IGV4aXQgdG8=";
    char[] kchars = k1.toCharArray();
    kchars[3] = 'o';
    kchars[1] = '3';
    kchars[6] = 'N';
    kchars[10] = 'V';

    return new String[] { "Blowfish", new String(kchars) };
}
```

As can be seen, the string `k1` represents the base64 encoded encryption key. Because the word “Blowfish” is mentioned we assume this is the encryption algorithm used. In addition to this, some tricks have been used to obscure the original key `k1`. By replacing the relevant character specified in the array we ended up with the base64 string: `N3RoIFNoIGV4aXQgdG8=`

When decoded this string becomes: 7th St exit to

By using this as the key we could decrypt the administrator password stored in the database by using Blowfish in ECB mode. The result is the administrator password “masteroppgave” which is correct.

The original encryption key k_1 is hardcoded into the software and remains the same across different installations of TM DLP. Even if it was generated on install the entropy of the key would be quite low since English words are used. One of the problems with using the same encryption key across installation, is that if the attacker is able to read the database, all encrypted data can be decrypted instantly without running resource intensive brute-force attacks or other cracking techniques. Basically, since we have revealed the encryption key here, any passwords stored in TM DLP databases are no safer than when stored in plaintext.

3.4.1.3 CERTIFICATES AND ENCRYPTION KEYS

The certificates and encryption keys used for HTTPS communication are stored in a hidden keystore found at the following path: `/home/dgate/prod/manager/conf/.keystore`

During certificate generation the password “changeit” is set. The generation is done with the key and certificate management utility `keytool`. The reason for using this instead of OpenSSL is that it creates Java compatible keystores [65]. During evaluation the key was imported into Wireshark to allow all SSL traffic to be decrypted. This worked well for all HTTPS traffic, but failed for endpoint communication as another channel with a different key pair is used.

3.4.1.4 APACHE TOMCAT

Version 5.5.7.0 of Tomcat was released on 30th January 2005 and is scheduled to reach end of life 30th September 2012 [66]. A common exploit performed against Tomcat server relates to the application manager; a web application that allows other web applications to be deployed, either from the server itself or from a user uploaded web archive (WAR). If weak administrator credentials are used, an attacker can deploy malicious java web applications, ultimately, leading to server compromise.

Many of the vulnerabilities found in this version of Apache Tomcat are DoS attacks. This type of attack will bring down the management server, but might not bring down DLP as a whole. See further discussion in chapter 4.5.

3.4.1.5 ENDPOINT AGENT

The endpoint DLP agent can be deployed over Active Directory, SCCM or installed manually on each computer with a .bat-file. We choose the latter option for our evaluation. The bat file gives the options to protect the installation with a password and to hide the installation folder when done. The agent is installed to “\WINDOWS\system32\dgagent”, but when hidden is not visible even when Windows’ “show hidden files” and “show system protected files” options are enabled. By hooking into the kernel, API calls to hidden files, processes, or registry keys can be filtered out and ensure the DLP endpoint is not detected. This means you need to know the exact path and use the Windows run prompt to open the application’s directory.

If an attacker has gained administrator access to a workstation they can stop the endpoint protection with the command “net stop dsasvc”.

3.4.1.6 SERVICE CHANNEL

The endpoint connects to the management server through port 8904. This channel is used for updating the DLP definitions and report on policy violations. All message exchanges use the xmlBlaster messaging middleware to transmit policy updates, logs and forensic data. This is done over a TLS encrypted channel, but because of the proprietary nature of the source code we didn't investigate this any further.

In case the agent needs to be patched to a newer version, the management server instructs it to start its auto-update mechanism. The auto updater then connects to the regular web server on the following addresses to get the appropriate patch:

```
http://{ServerAdr}:8080/dsc/au/env_0/dlp_agent_config.zip  
http://{ServerAdr}:8080/dsc/au/env_0/server.ini  
http://{ServerAdr}:8080/dsc/au/env_0/_binary/pb1321_agent_{version}.zip
```

If the auto-update mechanism is intercepted or compromised, great damage can be done. For example a software update can be modified to include a malicious payload that allows the attacker remote control of the system. An attack like this is presented in chapter 3.5. Auto-update mechanisms are further discussed in chapter 4.1.

3.4.1.7 THE WEB APPLICATION

Chapter 1.4 presented the Web Hacker's Methodology as a way of evaluating the security of web application. The checklist used is available in Appendix I. Following are the main highlights from this evaluation.

a. MAP THE APPLICATION'S CONTENT

TM DLP's source code for the endpoint agent and message service is compiled and hard to reverse engineer. The web app, on the other hand is quite different. The websites can be viewed without any special tools and there exist plenty of Java de-compilers for the occasional .jar and .class files. For an attacker the product can still be downloaded, installed and analyzed for vulnerability before attacking a real target. As with MyDLP, this can help in keeping the malicious activity hidden, as mapping the application does not have to be performed on the target network.

There is little default content present. However the Tomcat documentation located in "/manager/html-manager-howto.html" and "manager-howto.html" are discovered with automated scanners, such as the one included in w3af [58]. Apart from this the Dojo toolkit test files - many of whom contain XSS vulnerabilities - are directly accessible under "https://{ServerIP}:8443/dsc/HIE/lib/dojo" and "https://{ServerIP}:8443/dsc/HIE/lib/dojox". An attacker can use these vulnerabilities to gain access to the web application by fixating or stealing the session cookie. This is done by getting an authenticated user to click link with a malicious script as was explained in the MyDLP evaluation.

As mentioned in chapter 3.4.1.4, a technique for compromising a Tomcat server is through the application manager. This web application is accessible from the URI "http://{ServerIP}:8080/manager". For TM DLP the manager ships in its default configuration

with no default administrator user. This means any access to for example “http://{ServerIP}:8080/manager/http” will be unsuccessful as no account with the required access privileges exist.

A previous admin interface login can be found under “http://{ServerIP}:8080/dsc/dscLogin.jsp” and is accessible without TLS. Any failed login here results in a redirect to the newer login page: login.jsp.

b. ANALYZE THE APPLICATION

When connecting over HTTPS connections, the TM DLP management console can be found by adding dsc to the server address (http://{ServerAdr}:8443/dsc/). The main application is accessible from login.jsp and redirects users to the web application files found in the “/pages/” directory after successful authentication. All requests to different pages are contained within POST requests without any type of message wrapper.

The main configuration of the web application is controlled by the file “web.xml”. This file controls application information, file redirects, protected resources, user accounts for basic and digest HTTP authentication, and much more.

After analyzing the directory structure and technologies used, the attack surface of the TM DLP web application looks as follows:

- Vulnerabilities found in various components used by the management console (see Table 5 for a complete list).
- Files that allow user input, but do not require an authenticated session for input to be processed.

c. TEST CLIENT-SIDE CONTROLS

There are no client-side controls on the username and password fields. Basically, any input, even blank fields, gets sent to the server for processing.

For new users there is a requirement for password length to be between 8 and 32 characters. The control for this is only issued client-side and can be bypassed by modifying the request.

Even though the web app runs on Java, no applets are loaded client-side. However, a SWF-object located in the Dojo Toolkit demo files contains vulnerabilities. This client side attack is called Cross Site Flashing (XSF) and occurs when one flash object can load another one and access its sandbox functionality [67]. In this case the value “Security.allowDomain” was set to “*”, which means this file can be loaded from a third party flash objects located at another domain. Such an attack can be devastating if the vulnerable flash applet has access to critical security functions that works without authentication.

d. TEST AUTHENTICATION MECHANISM

During authentication the password and username are sent in plain text over HTTPS as a post request in the fields “j_username” and “j_password” as seen below:

```
Content-Type: application/x-www-form-urlencoded
j_username={username}?j_password={password}&Submit=Log+On
```

The post request is directed to “/dsc/j_acegi_security_check” which is the application component responsible for authenticating the user. Acegi security is set up in the web.xml configuration file by specifying filter-rules. The rules specify files and directories Acegi security should protect. When a request is made to these resources by a client, Acegi intercepts them and checks if the user has previously established and authenticated session that allows access to this resource. If not, the user is redirected to the login page [68].

Apart from this, the login page itself offers no “remember me” or account recovery functionality. Username enumeration is not possible as every failed login request results in the same non-verbose message.

The server also comes with an older login page, “dscLogin.jsp”, which also allows for successful user authentication and does not require a TLS connection to function. Not encrypting the transmission of plaintext credentials is dangerous as these can more easily be collected by an attacker. By getting an administrator to use this unsecured login page on an eavesdropped connection, credentials can be stolen. The POST request is the same as on the newer login page, but lack “&Submit=Log+On” at the end. Even, when successfully authenticated, the extra string value does not affect the web application functionality in any way.

e. TEST THE SESSION MANAGEMENT MECHANISMS

The session ID is established as soon as the login page is loaded. This is a cookie with the value “JSESSIONID=” followed by 32 character string. If the “JSESSIONID=” cookie is removed or tampered with the user is automatically redirected to the login page. Changing the IP does not affect the session in any way. By authenticating with one browser and accessing the same page with another browser we could bypass authentication by injecting the authenticated session ID using Burp Proxy [69]. If an attacker can steal or set the cookie of an authenticated session, there will be no other security controls preventing management console access.

Additionally, a cookie with the value “testcookie=enabled” is set; this value does not affect the use of the web application.

Termination is done from either user logout or through session timeout. The session timeout value is set to 30 minutes in web.xml.

f. TEST ACCESS CONTROLS

When installing Trend Micro DLP, 4 accounts have to be set up: admin, enabled, root and dgate. Of these only admin can authenticate to the web application. The user enabled is

used to administer the DLP over a custom cli which is accessible when connecting to the server over SSH. The root user allows direct access to the underlying operating system, in this case CentOS, and dgate is a service account used for running the various DLP applications and daemons on the server.

The management consoles allows for additional users with different roles to be added. The default roles available are:

Administrator – Role with permissions to perform all operations

Security Manager – Role with permissions to manage policies and monitor security activity

In the case an account with read-only access is wanted, new roles can be created with a granular set of access control setting (Figure 17).

Permission		Permission		
Access Areas		read	write	execute
Data Protection				
	Digital Assets	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Compliance Templates	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Company Policies	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Data Discovery	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	Device Control	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	Reports	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	Logs	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Update	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Administration				
	Server Configuration	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Agent Configuration	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Crawler Management	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	Data Management	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Agent Management	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Management Console	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	LDAP	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Product License	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 17: Trend Micro DLP - Role creation

When adding users, it is not possible to specify a username that already exists in the web application. System usernames such as root and enable can be registered, although this doesn't have any impact of overall server security.

g. TEST FOR INPUT-BASED VULNERABILITIES

TM DLP has a previously known input-based vulnerability which was patched in hot fix 1348 released by TM, September 2011 [70]. The vulnerability uses the Tomcat directory traversal vulnerability described in CVE-2008-2938 [71] to access files outside the web root directory, including “/etc/passwd/” [72]. This is done by replacing punctuation with their Unicode (UTF-8) equivalent values as punctuation is ignored by the server. In this case “%co%ae” equals one punctuation, and can be used to traverse to previous directories in the path (../) as seen in the example below:

```
https://{ServerAddr}:8443/dsc/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae/c0%ae/etc/passwd
```

When starting the evaluation of TM DLP, we wanted to test this vulnerability before applying the latest available patch. As would be expected, the “passwd” file was accessible, and after applying hot fix 1348, the vulnerability could not be exploited this way anymore.

The proper way to fix this issue is to update Java and Tomcat to a non-vulnerable version, which addresses this issue. Before and after applying this patch we controlled if either of these components had been updated. The result was disappointing. Neither the Tomcat server nor the Java SDK had been updated. Instead, the hot fix issued by TM only blacklists the Unicode characters that the original exploit uses. This gives us an incentive to investigate further. TM could have missed other characters that also can be used for directory traversal attacks.

To investigate further the directory traversal fuzzer DotDotPwn 3 was used [73]. This Perl script goes through multiple variations of character encodings in the hope of exposing system files outside the web root. The script was run with the following command:

```
perl dotdotpwn.pl -m http-url -u http://{ServerAddr}:8080/dsc/TRAVERSAL -f /etc/passwd -k "root:" -d 7 -t 200 -s
```

The results were positive. Even though some doors had been closed with the TM hot fix, others were open. Meaning files, such as “/etc/passwd”, can still be downloaded for offline brute-forcing. In total DotDotPwn reported 22 successful ways of accessing a system file outside the web directory.

After some experimenting we discovered that the files shown by this vulnerability are only those readable by the web service user “dgate”. For example, the database files are not accessible for this user and cannot be read, so the vulnerability has some limitations. In the next chapter we present how an attacker can use this vulnerability to gain root access to the DLP server.

3.5 ATTACKING THE DLP

To illustrate what is possible when attacking a DLP, we have taken a step further and compromised our Trend Micro (TM) DLP server. For this we used the directory traversal vulnerability presented in our evaluation to crack and expose the password for the root user. In the case where this password is very strong, it will be very hard to execute such an attack. Still, a determined and skillful attacker should have no problem finding another way into the system.

The password cracker John the Ripper²¹ handles cracking of Linux “passwd”-files without any problems. The program accepts dictionary and brute-force cracking and can work with salted values as is default for Linux “passwd”-files. Following is the output given by John the Ripper in our attempt to reveal the root password.

```
# john /tmp/passwd.export
Loaded 4 password (FreeBSD MD5 [32/32])
(After letting the tool run for some time, the status is checked)

# john -show /tmp/passwd.export
root:secret:0:0:root:/root:/bin/bash

1 password cracked, 3 left
```

As can be seen, the root password is “secret”. With this password the attacker can now gain access to SSH and interact with the DLP management server directly. The next goal is to get administrator credentials for the web interface. For this there are two options: either a user can be added directly to the database used by TM DLP, or the passwords of existing account can be decrypted.

Cracking the password of web console users is explained in chapter 3.4.1.2. To summarize, the account password is blowfish encrypted with the key “7th St exit to” and converted to base64 before inserted into the database. For an attacker, all that is needed is to export these base64 encoded password values and decrypt them. The easiest way of exporting the database is to perform a mysqldump and find the hashes in the output. Having more administrator passwords available is also helpful in gaining access to additional systems and services of the target company.

With the administrator account credentials the attacker can now access the TM DLP management console. The most interesting features here are the policies, endpoint administration and logging facilities. This is because the policies will tell the attacker what is considered sensitive by the company, and the endpoint administration will make data exfiltration simple. Having access to log administration makes it easier to cover ones tracks when ending the operation. Since the TM endpoint DLP agents don’t include data exfiltration

²¹ <http://www.openwall.com/john/>

capabilities, it is of interest to the attacker to replace it with something that does. For this demonstration OpenDLP was used.

OpenDLP is an open source DLP implementation that is mainly used for assessment of sensitive data and used during penetration testing. The tool allows agentless scanning of files shares, as well as agent-based scanning of windows installations. The tool uses regular expressions to identify sensitive content; this makes it easy for an attacker to copy already generated policies from other DLP systems into OpenDLP. The presentation *Gone in 60 Minutes: Stealing Sensitive Data from Thousands of Systems Simultaneously with OpenDLP* by Andrew Gavin outlines the exfiltration capabilities of the system when used for penetration testing [8]. The goal of the attack now is to take advantage of this functionality to automate and speed up the data theft.

Summary >Outdated Online Patches					
Endpoints With Outdated Patches					
Endpoints	IP Address	Current Version	New Version	Group	Last Modified
TORE-848B920B93	192.168.11.22	5.5.1263	5.5.1353	Production Environm...	4/25/12 8:42:34 PM CEST

1-1 of 1 page 1 of 1

Deploy Updates Back

Table 6: Trend Micro DLP - Endpoint update deployment.

The web console of TM DLP makes it easy to administer and deploy updates to all DLP endpoint agents. When deploying a software update the agents are instructed to download and apply the patch available from the DLP web server. This process is described in Chapter 3.4.1.6. Since the endpoint agent runs as administrator, the patch will be applied under the same privileges. With root access to the server distributing these patches, the attacker can replace the patch with a malicious version. In our case, an unattended installer was created. The installer turns off DLP protection, disables simple file sharing (so OpenDLP can deploy its own agents over SMB) and creates a new administrator account with known credentials. For this demonstration a regular Windows install was used, but in business environments where systems are administered with Active Directory, the installer will have to be modified accordingly.

The TM DLP auto-updater downloads the patch from the server and verifies the file size against the file size value found in the file “server.ini” (see Figure 18, next page). If this value does not match, the update will fail. It is therefore important to edit this before performing an agent patch deployment. Apart from this check, no other features have been implemented to prevent tampering of the agent patch process. When deployed, each endpoint system is now prepared to receive an OpenDLP agent.

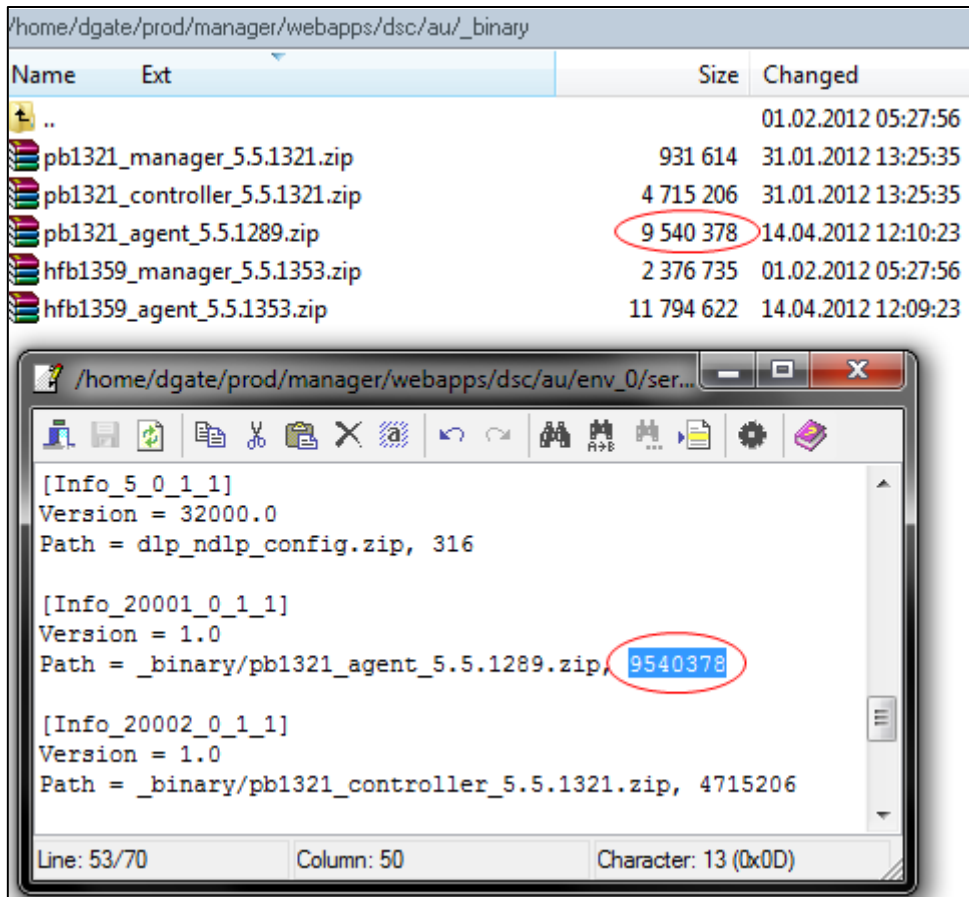


Figure 18: server.ini file size value.

The OpenDLP software can be installed manually or downloaded as a VM. During our testing we ran the system as a VM. An attacker can run this as a VM on the compromised workstation or install it directly on the DLP server. In the latter case additional modules, like Perl and Apache HTTP must also be installed, which might lead to complications. Still, having OpenDLP running on the TM DLP server will seem less suspicious as it is normal that this server interacts with all endpoints on the network.

Start a New Scan	
Scan name	Demo
Profile	Demo (or create a new profile)
Notes	<ul style="list-style-type: none"> • For Windows Share scans, enter systems in this format: \\1.2.3.4\Share • For MS SQL scans with different instances, enter systems in this format: server/instance • Otherwise, just list IP addresses
Systems to scan (newline-delimited)	192.168.11.22
Start	

Figure 19: Starting a new scan with OpenDLP.

OpenDLP needs a profile specified before it can deploy its endpoint agents. This profile is populated with the scanner type (agent or non-agent based), the admin credentials that the malicious installer created and the regular expressions needed to detect sensitive files. When this profile is saved, it is time to start a scan. The scan dialog (Figure 19, previous page) lets one specify the target address where the OpenDLP agent should be deployed and executed. After pressing “Start” the following dialog appears (Figure 20):

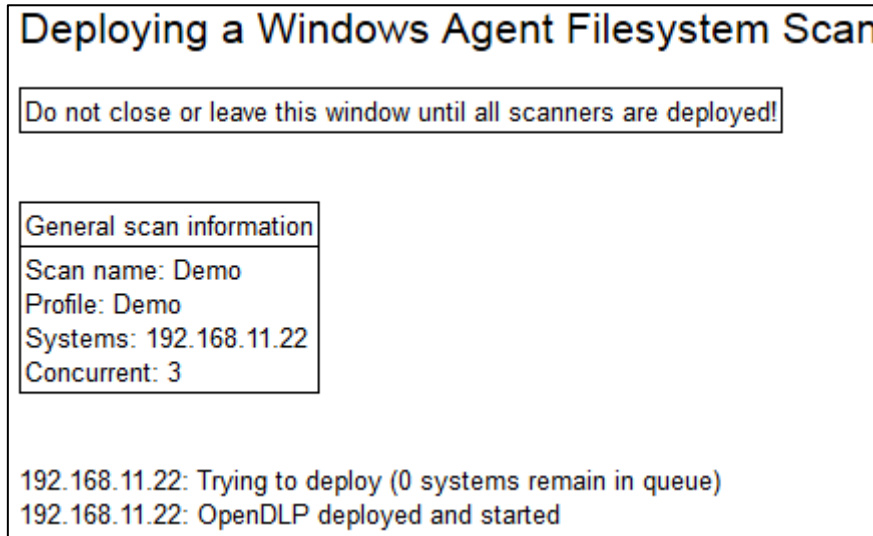


Figure 20: Deploying remote agents with OpenDLP

If the malicious installer did its job and the profile properly setup, the agent should be deployed almost immediately. In the case where multiple OpenDLP agents are deployed the “concurrent” setting specifies how many agents to deploy at a given time interval.

After deployment, the progress of one or more scans can be monitored (see Figure 21). This data is continuously uploaded to the OpenDLP server along with all sensitive files the agent comes across. If the attacker so chooses, scans can be stopped or uninstalled from the same page at any time.

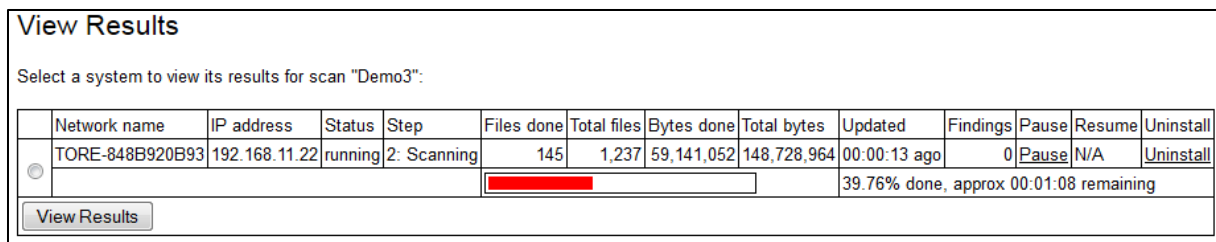



Figure 21: Overview of running agents.

When the scans complete the agent can be uninstalled by the administrator. The results of the scan can be viewed at any time along with any sensitive files discovered (see Figure 22, next page). The files can also be downloaded directly from the administration interface if desired.

Data Loss Prevention Systems and Their Weaknesses

Results for 192.168.11.22 (TORE-848B920B93):

Profile	Demo		
Status	finished		
Step	3: Done		
Files Done	1,237		
Files Total	1,237		
Bytes Done	148,099,545		
Bytes Total	148,728,964		
Progress			
Percentage	100%		
Completion Time			
Total Findings	657		
False Positives	0		
Valid Findings	657		
Updated	00:01:12 ago		
Pause	N/A		
Resume	N/A		
Stop and Uninstall	N/A		

#	Regex	Pattern	File (click to download)
1	ThisIsSecret	ThisIsSecret	C:\Documents and Settings\Administrator\Desktop\Sensitive document.txt

Figure 22: Results from running agent.

After collecting sensitive data it is time to transfer it from the target network. Encrypting the data and then transmitting it out through regular channels such as HTTP or HTTPS can be done. Or one can use SFTP and similar to perform this task. Since the data is encrypted beforehand, any gateway DLP listening in will not be able to read and analyze the contents.

During file transfer the attacker can use the time to clear his tracks. This means deleting or modifying any logs that clearly shows the incident occurred. If this is done well, and the endpoint DLPs are restored to their original state the network breach might remain unnoticed.

3.6 RESULTS

Both evaluated products contain out-of-date software components, many of whom carry a large amount of known vulnerabilities. The version of these components is usually reflected in when the feature was originally implemented by the DLP developers. Although patches to the DLP software exist, these do not focus on updating third party components, even when these components play a critical role in the workings of the DLP. Even without patching, the components can be made less dangerous by deleting included demo and test files that are not used by the DLP to begin with.

Because of the different web application architectures of the evaluated products, there was little overlap in the discovered vulnerabilities. The main vulnerabilities discovered in MyDLP deals with information disclosure of system statistics and access to restricted functionality. These can be used together to launch a DoS attack against the system. By continuously uploading files to the server and populating the database table the attacker could monitor the system statistics to see when drive space is nearly full and disrupt the DLP. Being able to upload a file that resides on the remote system is helpful if a remote command execution vulnerability that can execute local files is later discovered.

For Trend Micro DLP new variations of an old and patched directory traversal vulnerability can still be used to read local system files. This is critical since the Apache Tomcat user has read privileges to the `/etc/passwd` file among others. Bad security practice can be found regarding how passwords are encrypted and stored in the database. The same encryption key is used across installations and, as demonstrated, can easily be revealed. Basically, the security of stored user credentials in Trend Micro DLP is hardly better than storing the same information in plain text.

In both products the use of default credentials (e.g. they remain the same across multiple installations) is quite common. Only the certificates used by the HTTPS connection were found to be generated at install time. The best practice would be to generate all passwords and certificated at install time, except for those passwords the user is going to use.

The third party demo and test files found in both products contain multiple XSS vulnerabilities. Although we didn't focus on XSS attacks in this thesis, as they in our scenario are quite "noisy" (e.g. requires interaction with DLP administrators), they can still be of great usefulness to an attacker if no better alternative exist.

4 DISCUSSION

4.1 ATTACKING UPDATE MECHANISMS AND SERVICE CHANNELS

When researching how to perform the attack demonstrated in chapter 3.5, we were surprised how easy it was to control the patch deployment and replace the legitimate patch with a malicious one. Having a client server administration model where patches can be deployed to the client is nothing new. In fact it is used in multiple systems including centralized anti-virus solutions, backup software and directory services. Automatic software update systems are also susceptible to the same type of attack if the attacker can fool the client into thinking it is communicating with the legitimate update server. Tools, such as EvilGrade [74], automate the procedure and offers integration with the software update mechanism of multiple applications including Java, Skype, Apple Update and Winamp. All that is required by the attacker is to either launch a man-in-the-middle or a DNS spoofing attack against the client. The proper way of protecting against such attacks is to sign all updates with a digital signature and establishing a secure channel. Before installation the signature of the software update can be verified and if it is correct, installed.

By sending the right requests to a DLP endpoint agent an attacker can add new policies, disable protections or trigger software update mechanisms. It is therefore essential that the DLP can differentiate between requests from the legitimate DLP server and those from a malicious user. The best way to do this is by establishing a trusted secure communication channel between server and endpoint agents to ensure that the agent does not blindly trust whatever is being sent to it. The server should also have a way of identifying that the agents are legitimate, so it doesn't distribute all the DLP policies to an endpoint set up by an attacker. This can for example be done by generating and integrating a unique certificate in each endpoint deployment. If done properly the attacker would have to go through extreme lengths to obtain a valid certificate, at which point, finding another attack vector would be simpler.

4.2 SECURITY OF OTHER PRODUCTS

During our researched we only had access to products that the vendors themselves made freely available for testing. Most DLP products on the market require a hardware appliance, which are usually only available for potential buyers of said devices. Testing the appliance themselves is not much harder than their virtual counterpart. Appliances like this are usually regular computers running well-known operating systems, such as Red Hat Linux/CentOS or Windows Server.

From a statistical point of view, testing 2 solutions is not adequate enough to conclude that all DLP products contain critical security flaws. If we take the vendors analyzed in Gartner's magic quadrant report for 2011 and add MyDLP, we end up with a sample of 2 out of 14 DLP products. To add to this, Trend Micro themselves are lagging behind with their DLP solution, as the feature set is inferior to that of the competition [75]. For organization serious about DLP, Trend Micro's offering is hardly even considered [11]. MyDLP, which also was evaluated, only has a few developers with limited resources. In comparison to the market leaders both these products fall short.

In an ideal situation we would have the possibility and resources to evaluate more products to get a clearer picture over the situation. Since this is not the case, we have to look for other data related to the parties involved. The research done by Ben Williams and Daniel Compton has revealed over 70 security vulnerabilities in security gateway and network monitoring software [76]. As an example, one of the security gateway evaluated by Ben Williams was Websense Triton 7.6, which is also used to administer their DLP product. Although a patch has now been issued, one of the vulnerabilities found was quite alarming. With an unauthenticated command-injection a remote user could launch commands as SYSTEM on the Windows Server system. The whitepaper offers the following example for changing an administrator's password to "blah" [76]:

```
https://192.168.1.30:xxxx/xxxx?xxxx=echo .pdf%26net user administrator blah|
```

The whitepaper further goes on to discuss vulnerabilities in McAfee's web security gateway, which offers multiple ways of escalating privileges after authentication [76]. Although this is not McAfee's DLP offering, the company itself has quite a large presence in the DLP market, and it would not be quite unthinkable that web UI codebase is shared across their security portfolio.

Using this additional research one can see that flaws in security are not necessarily linked to how small a company or its product is. Instead, they concern security vendors of all sizes and are more widespread than what was originally believed.

4.3 SECURITY VENDORS AS FRONT FIGURES FOR COMPUTER SECURITY

If the companies we trust to secure our IT infrastructure cannot be bothered to evaluate their own products for security flaws, where does that leave us?

When investing in security and the systems that usually go with them, it is expected that the end result is a more secure IT infrastructure. When the investment leads to the implementation of vulnerable systems, this expectation is not met. The systems that are supposed to secure the infrastructure become a liability and the return value of the initial investment is next to nothing. Still, most businesses don't have the resources to check the quality of the products and services they buy. Instead, they have to rely on expertise of the company or middle-men they have contracted.

Vulnerabilities are discovered continuously and are found in all types of software. Compared to the software industry as a whole, security vendors don't necessarily look bad, but when one takes into account the fundamental value of the security field (e.g. making stuff secure), this comparison crumbles. Security vendors should be front figures by demonstrating good security practices at all levels. Rushing out products with vulnerabilities is not the way to do this. The irony in all this is that most security vendors also have their own vulnerability research lab. With the help of their blog, these labs often publish research on new malware, hacker attacks and other threats in great technical depth. Basically, the resources and knowledge to do a vulnerability assessment of one owns products already exist on the payroll.

From an economic perspective there are multiple factors that need to be considered. Firstly, conducting a proper security evaluation can be quite expensive. Even though security researchers exist on the payroll, this does not necessarily mean they have the skillset and experience required to conduct a security evaluation. In this case, outside consultants are needed, which costs quite a bit more.

Secondly, after a successful evaluation the program needs to be patched, and then tested before public release. Depending on what security vulnerabilities were discovered, such patching can be everything from smaller updates of some components to code rewrites of the entire security architecture.

Thirdly, if a vendor already spent millions acquiring a new security product, you naturally want to cut the corners you can to recover the losses incurred.

Lastly, costs play a major factor to the customer. If the customer is oblivious to the security of the product and cares more about the price, doing a security evaluation might not be cost effective. Instead the amount saved can be used to deliver a marketable product feature or lower the cost of the product so it can compete better. In the case of the vendor's security blogs, the brand building and marketing of these websites have a clear value to the decision makers. A security evaluation on the other hand, does not.

By putting light on these issues, customers, as well as security professional will become more aware of the fact that products produced by security vendors are not necessarily secure. The optimal outcome is more pressure put on vendors to take an active approach in finding and patching security vulnerabilities.

4.4 INTEGRATING DLP INTO BUSINESS WORKFLOW

Bypassing a DLP system does not necessarily stem from malicious activity. In general certain security solutions may be an annoyance and even hamper productivity. For a security researcher the anti-virus scanner must be disabled to be able to analyze a malicious script. For a user editing a lot of confidential data, the same DLP feature that prevents copy-paste, can quickly become an irritating productivity stopper. How then does one deal with this?

When starting the implementation of a DLP system it is important to recognize the different needs and workflows related to each department in an organization. Having the system work against the users is not an ideal state. Workarounds will be found. Take password policies for example:

A user is required to change his computer password every 3 months. The user is not allowed to use one of the previous 5 passwords. The workaround for the user is the increase the number in the end of the password by 1 or write it down on a post-it note.

The policy in this case was ineffective in increasing security as it was not properly conceived. In the case of DLP one has to be considerate of this as the system can in many cases interfere with the legitimate work of users. The following points explain various considerations to be taken to better integrate DLP into user workflow and ensure secure handling of sensitive data:

- **Educate:** When blocking an action, explain why and teach the user about the relevant policy. For example if the user tries to save a document with financial information on a public share, tell that the business has to comply with financial regulations, and because of this the document cannot be saved to this location. In this case it can also be good to list the allowed save locations.
- **Feedback:** Allow users to easily report cases of false positives and give general feedback regarding the DLP system. If feedback is acted on and quickly implemented the users might even take a liking to the system.
- **Trust:** If all or some of the users are considered to be loyal and knows how to properly threat sensitive data, allowing them to fully or partially bypass DLP blocks with a provided reason, can save headaches for both users and administrators.

4.5 DOS ATTACKS

Many of the vulnerabilities found in web server software are denial of service (DoS) attacks. Since DLP systems run web server software it is interesting to look at the implications of such an attack on these systems.

Endpoint DLP: These systems are made to work independent from the management server and network gateway. They also are not equipped with web server software, which makes them less vulnerable. In case a DoS vulnerability is found on a DLP endpoint, an attacker can target one or more of these vulnerable systems. If all endpoints in an organization become paralyzed as a result of such an attack, the impact will be severe.

Network DLP: Here two situations must be considered. Is the network DLP acting as a gateway proxy or passively connected to a network monitoring or SPAN port. If the former, it is possible that a DoS attack will bring down network connectivity, which unlike paralyzing all DLP endpoints, will also affect systems not running DLP endpoint agents. If the latter situation occurs, the network DLP will in the worst case stop reporting on policy violations. Since the systems monitors passively, bringing it down should not affect any other systems.

Management server: If this device suffers a DoS attack, it is unlikely that other components will suffer. The components of a DLP system as a whole work quite independently from each other and should continue to offer data protection even when a connection to the management server cannot be obtained. Because of the all the services remotely accessible on a management server, it is most likely this will be the victim of a DoS attack.

Even with its somewhat resilient architecture, having the DLP vulnerable to DoS attack is undesired. From our evaluation, these attacks are often found in older vulnerable HTTP server software, but misconfigured or badly coded DLP components can also be a source of additional vulnerabilities. For example, if no limits are set on what is being logged, the server could run out of disk space and stop responding. When designing software, it is important to have an attacker's mindset and identify these potential weaknesses. This will prevent shipping software that might negatively affect the availability of a customer's IT systems and thereby result in financial losses.

5 CONCLUSION

Today's DLP implementations rely heavily on pattern based and signature detection. As demonstrated any experienced attacker will have several ways to bypass the protection and the products should therefore not be seen as an end-all solution to data protection. Still, when used in combination with other security technologies, DLP will help in making it difficult to perform data theft. For example, sensitive data scattered across various locations within an organization's IT infrastructure can be assessed by the DLP and consolidated to more secure locations; thereby making it harder to get to. Much like an intrusion detection system, a DLP can give early warnings of data theft taking place through its various logging facilities. Even after an attack, the forensic evidence collected by a DLP can be valuable. DLP systems also excel at preventing accidental leaks and attacks from people lacking the technological understanding of the system.

Securing all digital assets from the ground up can be a daunting – if not impossible – task. Even though DLP combined with other security technologies is an attempt at doing this, the goal is still not within reach. Organizations considering DLP should also look at what are perhaps bigger underlying social problems. If employees cannot be trusted, a technological solution is probably not the right cure.

DLP was one of the security buzzword of both 2007 and 2008. As a result multiple products popped up, some of whom were acquired, relabeled and rushed out on the market to maximize sales. Many of these products are still actively developed, but as analysts predict, some have started to fall behind now that the show is over. This can be seen in the products we evaluated. Both are not as actively developed as before and are starting to fall behind the competition. As a result organizations face a risk when implementing these products as they run on outdated technology prone to multiple publicly known vulnerabilities. But even more alarming is that actively developed security products –both DLP and non-DLP – have also been found to contain critical security flaws [76].

Although only representing a small part of all security products, the products evaluated shows a lack of security best practice from an industry that specialize in security products. As demonstrated, the compromise of the DLP system will greatly assist an attacker in conducting data theft. This could have been prevented if the product was not implemented in the first place, but for many organizations the benefits outweighs the risks. The problem in this case is not DLP as a concept, but the various products trying to implement it.

More effort is needed to secure the products that secure our data, and a step in doing so is to bring more attention to this problem. Relying on third party researches to discover and report vulnerabilities is not the correct solution. Instead, customers dependent on these products have to demand that they be actively tested and patched by the vendors themselves.

5.1 FUTURE WORK

Based on the problems encountered during our research and the conclusion of this thesis, we would like to see more being done in the following areas:

The use of the agent/server model is not unique to DLP, but also applies to other products, such as managed anti-virus. If these systems are vulnerable to attacks where the agents can be tricked into communicating with a malicious server instead of the real one, the impact will be significant. Compared to enterprise anti-virus solutions, the install base of DLP products is a drop in the ocean. Evaluation of additional security products is therefore needed to see if such problems are taken seriously in various industries. For example, an attacker could take over backup agents from a centralized backup solution and use them for data theft

In addition to this, the research done by Infobyte Security Research sought to improve the security of software auto-update mechanisms; it would also be interesting to see how much better this has gotten over the past years in a wide range of products.

While best practices for building secure web applications exist²², little effort has been put on how third party libraries and frameworks should be kept up to date. By patching a library, a developer might get worried that configuration files get overwritten or other parts of the web application starts behaving wrong. This could lead to important security updates being skipped as this results in less work for the developer. Creating a standard that makes it as simple as possible to update third party libraries and frameworks will make it easier to maintain a product past its release date and as a consequence improve security. Additionally, providing demo and test files as a separate download will go a long way in delivering a secure-by-default, as most vulnerabilities are often found in these files.

Testing “exotic” protocols such as Action Message Format is quite hard without proper tools. Many of the tools encountered during our research were old and dysfunctional, even when the things they test for are still relevant today. Developing tools that can assist in the evaluation of niche web technology will be greatly appreciated by both us and the security community. This can be done by extending existing tools, such as the OWASP Zed Attack Proxy (ZAP) [77], or by starting from scratch.

²² A good resource is the The Open Web Application Security Project (OWASP) and their website <https://www.owasp.org/>

6 BIBLIOGRAPHY

1. **Stuttard, Dafydd and Pinto, Marcus.** *The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws*. Indianapolis : Wiley Publishing, Inc., 2008. 978-0-470-17077-9.
2. **Ian Amit, Iftach.** Sexy Defense. *I Am Security*. [Online] April 18, 2012. [Cited: May 3, 2012.] <http://www.iamit.org/blog/2012/04/sexy-defense/>.
3. **Gessiou, Eleni, Vu, Quang Hieu and Ioannidis, Sotiris.** *IRILD: an Information Retrieval based method for Information Leak Detection*. Institute of Computer Science, FORTH, Greece and Etisalat BT Innovation Center, Khalifa University, UAE. 2010.
4. **Hart, Michael, Manadhata, Pratyusa and Johnson, Rob.** Text Classification for Data Loss Prevention. s.l. : Springer Berlin / Heidelberg, 2011. Vol. 6794, pp. 18-37. 10.1007/978-3-642-22263-4_2. 978-3-642-22262-7.
5. **Mogull, Rich.** Understanding and Selecting a DLP Solution: Part 2, Content Awareness. *Securosis.com*. [Online] Securosis, September 13, 2007. [Cited: March 28, 2012.] <https://securosis.com/blog/understanding-and-selecting-a-dlp-solution-part-2-content-awareness>.
6. **arieanna.** Policy Creation: Ask the Right Questions. *Absolute Software*. [Online] December 23, 2008. [Cited: January 12, 2012.] <http://blog.absolute.com/policy-creation-ask-the-right-questions/>.
7. **Mogull, Rich.** *Best Practices for Endpoint Data Loss Prevention*. s.l. : Securosis, L.L.C., 2009.
8. **Gavin, Andrew.** Gone in 60 Minutes: Stealing Sensitive Data from Thousands of Systems Simultaneously with OpenDLP. *YouTube*. [Online] February 3, 2011. [Cited: April 28, 2012.] <http://www.youtube.com/watch?v=kz3M--LhyBg>.
9. **Lawton, George.** New Technology Prevents Data Leakage. *Computer*. 1, 9 9, 2008, Vol. 41, 9, pp. 14-17.
10. **Quellet, Eric.** s.l. : Data Loss Prevention, 2009.
11. **DLPX.** Predictions: 2011 Gartner Magic Quadrant for Data Loss Prevention. *DLP experts*. [Online] June 29, 2011. [Cited: May 3, 2012.] <http://www.dlpxperts.com/dlpxblog/2011/6/29/predictions-2011-gartner-magic-quadrant-for-data-loss-preven.html>.
12. **SC Magazine.** Winners 2012: Symantec for Symantec Data Loss Prevention. *SC Magazine Awards*. [Online] 2011. [Cited: 04 04, 2012.] <http://awards.scmagazine.com/symantec-symantec-data-loss-prevention>.

13. **The Radicati Group, Inc.** Data Loss Prevention Market, 2010-2014. *The Radicati Group*. [Online] December 2010. [Cited: April 05, 2012.] <http://www.radicati.com/wp/wp-content/uploads/2010/12/Data-Loss-Prevention-Market-2010-2014-Brochure.pdf>.
14. **Stiennon, Richard.** McAfee acquires Onigma. *ZDNet*. [Online] October 15, 2006. [Cited: January 20, 2012.] <http://www.zdnet.com/blog/threatchaos/mcafee-acquires-onigma/421>.
15. **Naraine, Ryan.** Trend Micro makes DLP move, Symantec stands pat. *ZDNet*. [Online] October 25, 2007. [Cited: January 20, 2012.] <http://www.zdnet.com/blog/security/trend-micro-makes-dlp-move-symantec-stands-pat/611>.
16. **Firstbrook, Peter, et al.** *Hype Cycle for Data and Application Security*. s.l. : Gartner inc., 2008.
17. **Williams, Christopher.** Virgin Media collects customer banking details on CD, then loses it. *The Register*. [Online] June 20, 2008. [Cited: January 20, 2012.] http://www.theregister.co.uk/2008/06/20/virgin_media_banking_loss/.
18. **annualcreditreport.co.uk.** Historic Data Loss. *annualcreditreport.co.uk*. [Online] May 20, 2008. [Cited: January 20, 2012.] <https://www.annualcreditreport.co.uk/identity-theft/data-loss/490/virgin-media.htm>.
19. **Information Age.** Manpower escapes fine for employee data gaffe. *Information Age*. [Online] January 23, 2012. [Cited: January 29, 2012.] <http://www.information-age.com/channels/security-and-continuity/news/1686813/manpower-escapes-fine-for-employee-data-gaffe.thtml>.
20. —. Hays emails contractor pay rates to RBS staff. *Information Age*. [Online] September 29, 2011. [Cited: January 21, 2012.] <http://www.information-age.com/channels/information-management/news/1649928/hays-emails-contractor-pay-rates-to-rbs-staff.thtml>.
21. **Takebayashi, T., et al.** 1, s.l. : Data Loss Prevention Technologies. Fujitsu Ltd, 2010, Fujitsu Scientific and Technical Journal, Vol. 46, pp. 47--55.
22. **Webopedia.** Insider Attack. *Webopedia*. [Online] [Cited: January 13, 2012.] http://www.webopedia.com/TERM/I/insider_attack.html.
23. **CSO Magazine, U.S. Secret Service, Deloitte, US CERT.** *2011 Cybersecurity Watch Survey*. s.l. : CSO Magazine, 2011.
24. **Boe, Tammy A.** Gaining and/or Maintaining Employee Trust Within Service Organizations. *University of Wisconsin-Stout*. [Online] August 2002. [Cited: April 11, 2012.] <http://www2.uwstout.edu/content/lib/thesis/2002/2002boet.pdf>.
25. **Sawyer, John H.** How To Prevent Data Leaks From Happening To Your Organization. *Dark Reading*. [Online] April 9, 2012. [Cited: April 11, 2012.]

<http://www.darkreading.com/security/client-security/232800363/how-to-prevent-data-leaks-from-happening-to-your-organization.html>.

26. **Verizon; U.S. Secret Service; Dutch High Tech Crime Unit.** *Data Breach Investigations Report 2012*. s.l. : Verizon, 2012.

27. **Park, Y., et al.** s.l. : System for automatic estimation of data sensitivity with applications to access control and other applications, 2011. Proceedings of the 16th ACM symposium on Access control models and technologies. pp. 145--146.

28. **Blakely, Benjamin, Rabe, Mark and Duffy, Justin.** Data loss prevention comes of age. *IT World*. [Online] April 20, 2010. [Cited: January 25, 2012.] <http://www.itworld.com/software/105076/data-loss-prevention-comes-age>.

29. **Mogull, Rich.** Implementing DLP: Deploying Network DLP. *Securosis*. [Online] February 13, 2012. [Cited: May 9, 2012.] <https://securosis.com/blog/implementing-dlp-deploying-network-dlp>.

30. **ICAP Forum.** ICAP Forum. *ICAP Forum*. [Online] 2011. [Cited: March 28, 2012.] <http://www.icap-forum.org/>.

31. **Elson, J., Cerpa, A. and UCLA.** RFC3507 - Internet Content Adaptation Protocol (ICAP). *ICAP Forum*. [Online] April 2003. [Cited: March 28, 2011.] <http://www.icap-forum.org/documents/specification/rfc3507.txt>.

32. **Trend Micro.** *Trend Micro DLP Administrator's Guide*. s.l. : Trend Micro, 2010.

33. **Manuel, Stephane.** Classification and Generation of Disturbance Vector for Collision Attacks Against SHA-1. *iacr.org*. [Online] 2008. [Cited: May 4, 2012.] <http://eprint.iacr.org/2008/469.pdf>.

34. **Polatcan, Onur, Mishra, Sumita and Pan, Yin.** New York : E-mail Behavior Profiling based on Attachment Type and Language, 2011. 6th Annual Symposium on Information Assurance (ASIA '11). pp. 6-10.

35. **Keila, P.S. and Skillicorn, D.B.** s.l. : Detecting Unusual Email Communication. IBM, 2005. Proceedings of the 2005 conference of the Centre for Advanced Studies on Collaborative research (CASCON '05).

36. **Autonomy Corp.** KeyView IDOL & Connectors. *autonomy.com*. [Online] [Cited: March 28, 2012.] <http://www.autonomy.com/content/Products/idol-modules-connectors/index.en.html>.

37. **Symantec.** What does kvoop.exe do for Symantec DLP? *Symantec Connect*. [Online] August 18, 2010. [Cited: March 28, 2012.] <http://www.symantec.com/connect/forums/what-does-kvoopexe-do-symantec-dlp>.

38. **SuperKoko.** Clipboard copy/paste detection. *codeguru*. [Online] June 6, 2005. [Cited: March 28, 2012.] <http://forums.codeguru.com/showthread.php?t=343977>.

39. **M, Margaret.** MTA Integration for SMTP Prevent. *Symantec Connect*. [Online] October 13, 2010. [Cited: May 22, 2012.] <http://www.symantec.com/connect/forums/mta-integration-smtp-prevent>.
40. **McAfee.** Host Data Loss Prevention 9.x Outlook add-in. *McAfee.com*. [Online] March 26, 2012. [Cited: May 5, 2012.] <https://kc.mcafee.com/corporate/index?page=content&id=KB59774>.
41. **Pritchard, Stephen.** The Truth About DLP. *infosecurity*. July 2011, Vol. 8, 4, pp. 18-21.
42. **Trustwave.** *Global Security Report 2010*. s.l. : Trustwave, 2010.
43. —. *Global Security Report 2011*. s.l. : Trustwave, 2011.
44. **Thorpe, Simon.** Data loss prevention (DLP) solutions with document encryption . *Oracle IRM, the official blog*. [Online] Oracle, September 24, 2010. [Cited: May 11, 2012.] https://blogs.oracle.com/irm/entry/data_loss_prevention_dlp_solut.
45. **Lund, Mass Soldal, Solhaug, Bjørnar and Stølen, Kjetil.** *Model-Driven Risk Analysis: The Coras Approach*. Oslo : Springer, 2011. 978-3-1642-12322-1.
46. **BackTrack.** BackTrack Linux. *BackTrack Linux*. [Online] March 1, 2012. [Cited: May 11, 2012.] <http://www.backtrack-linux.org/>.
47. **GitHub.** MyDLP. *GitHub.com*. [Online] April 2012. [Cited: May 5, 2012.] <https://github.com/mydlp/mydlp>.
48. **MyDLP.** MyDLP CE Virtual Appliances . *MyDLP*. [Online] March 17, 2011. [Cited: May 9, 2012.] <http://www.mydlp.com/downloads/viewcategory/4>.
49. **Canonical.** Ubuntu 10.04.2 LTS released. *Ubuntu.com*. [Online] February 18, 2011. [Cited: April 28, 2012.] <https://lists.ubuntu.com/archives/ubuntu-announce/2011-February/000141.html>.
50. **Adobe Flex Team.** Your Questions About Flex. *The Official Flex Team Blog*. [Online] Adobe Systems Inc., November 12, 2011. [Cited: May 4, 2012.] <http://blogs.adobe.com/flex/2011/11>.
51. **Symfoni.** The YAML Component. *Symfoni*. [Online] [Cited: May 04, 2012.] <http://symfony.com/doc/master/components/yaml.html>.
52. **ADODB.** ADODB Database Abstraction Library for PHP (and Python). *adodb.sourceforge.net*. [Online] 2011. [Cited: May 4, 2012.] <http://adodb.sourceforge.net/>.
53. **Oracle.** D.1. Changes in Release 5.1.x (Production). *MySQL*. [Online] Oracle, April 2012. [Cited: April 14, 2012.] <http://dev.mysql.com/doc/refman/5.1/en/news-5-1-x.html>.
54. **Squid.** Feature: HTTPS (HTTP Secure or HTTP over SSL/TLS). *Squid Proxy*. [Online] December 22, 2011. [Cited: March 30, 2012.] <http://wiki.squid-cache.org/Features/HTTPS>.

55. **MyDLP**. Postfix Content Filter configuration for MyDLP. *MyDLP*. [Online] February 28, 2011. [Cited: April 28, 2012.] <http://www.mydlp.com/tutorial/postfix-content-filter-configuration-for-mydlp>.
56. **The Apache Software Foundation**. Apache HTTP Server Project. *Apache.org*. [Online] April 17, 2012. [Cited: April 24, 2012.] <http://httpd.apache.org/>.
57. **MyDLP**. MyDLP Endpoint Registry Configurations. *MyDLP*. [Online] January 30, 2011. [Cited: March 30, 2012.] <http://www.mydlp.com/tutorial/mydlp-endpoint-registry-configurations>.
58. **Riancho, Andrés**. Web Application Attack and Audit Framework. *w3af*. [Online] May 25, 2011. [Cited: May 9, 2012.] <http://w3af.sourceforge.net/>.
59. **Whit3Rabbit**. SWFScan - FREE Flash decompiler. *HP*. [Online] January 6, 2012. [Cited: May 4, 2012.] <http://h30499.www3.hp.com/t5/Following-the-White-Rabbit/SWFScan-FREE-Flash-decompiler/ba-p/5440167>.
60. **kiamlaluno**. How long will my session last? *stackoverflow*. [Online] August 7, 2010. [Cited: May 9, 2012.] <http://stackoverflow.com/questions/1516266/how-long-will-my-session-last>.
61. **Friedl, Markus**. OpenSSH 3.9 released. *marc.info*. [Online] August 18, 2004. [Cited: May 4, 2012.] <http://marc.info/?l=openbsd-misc&m=109282003209482&w=2>.
62. **VMware**. SpringSource. *SpringSource.org*. [Online] May 3, 2012. [Cited: May 4, 2012.] <http://www.springsource.org/>.
63. **dojo**. Sophisticated WebApps with Dojo. *dojotoolkit.org*. [Online] 2012. [Cited: May 4, 2012.] <http://dojotoolkit.org/features/desktop>.
64. **Apache Software Foundation**. Struts. *Apache.org*. [Online] Apache Software Foundation, May 4, 2012. [Cited: May 9, 2012.] <http://struts.apache.org>.
65. **Oracle**. keytool - Key and Certificate Management Tool. *Oracle Docs*. [Online] Oracle, 2010. [Cited: April 11, 2012.] <http://docs.oracle.com/javase/1.3/docs/tooldocs/win32/keytool.html>.
66. **The Apache Software Foundation**. Apache Tomcat. *Apache.org*. [Online] April 5, 2012. [Cited: April 25, 2012.] <http://tomcat.apache.org/>.
67. **OWASP**. Testing for Cross site flashing (OWASP-DV-004). *The Open Web Application Security Project*. [Online] August 2009, 20. [Cited: May 2012, 22.] https://www.owasp.org/index.php/Testing_for_Cross_site_flashing_%28OWASP-DV-004%29.
68. **Vashishtha, ShriKant**. Acegi Security in One Hour. *Javaworld*. [Online] October 18, 2007. [Cited: May 9, 2012.] <http://www.javaworld.com/javaworld/jw-10-2007/jw-10-acegi2.html>.
69. **PortSwigger Ltd**. Burp Proxy. *PortSwigger Web Security*. [Online] 2011. [Cited: May 9, 2012.] <http://www.portswigger.net/burp/proxy.html>.

70. **Trend Micro.** Readme Hot Fix - Build 1359. *Trend Micro Support China*. [Online] November 9, 2011. [Cited: May 9, 2012.] http://support.trendmicro.com.cn/TM-Product/Product/DLP/5.5/Hotfix/1359/Readme_LP_55_EN_hfb1359_11092011.txt.
71. **CVE.** CVE-2008-2938. *CVE*. [Online] June 30, 2008. [Cited: May 9, 2012.] <http://cve.mitre.org/cgi-bin/cvename.cgi?name=2008-2938>.
72. **White Hat Consultores.** Trend Micro Data Loss Prevention Virtual Appliance 5.5 Directory Traversal. *Exploit DB*. [Online] May 27, 2011. [Cited: May 9, 2012.] <http://www.exploit-db.com/exploits/17388/>.
73. **chrix.** DotDotPwn - The Directory Traversal Fuzzer. *sectester.net*. [Online] February 3, 2012. [Cited: May 9, 2012.] <http://dotdotpwn.sectester.net/>.
74. **Infobyte Security Research.** isr-evilgrade-Readme.txt. *Infobyte Security Research*. [Online] Oktober 29, 2010. [Cited: May 7, 2012.] <http://www.infobyte.com.ar/down/isr-evilgrade-Readme.txt>.
75. **Quellet, Eric and McMillan, Rob.** Magic Quadrant for Content-Aware Data Loss Prevention 2011. *Gartner*. [Online] August 10, 2011. [Cited: May 7, 2012.] <http://www.gartner.com/technology/reprints.do?id=1-16XQWWD&ct=110810&st=sb>.
76. **Williams, Ben.** They ought to know better: Exploiting Security Gateways via their Web Interfaces. *ngssecure*. [Online] March 2012. [Cited: May 7, 2012.] http://www.ngssecure.com/Libraries/White_Papers/ExploitingSecurityGatewaysViaWebInterfacesWhitepaper.sflb.ashx.
77. **OWASP.** OWASP Zed Attack Proxy Project. *The Open Web Application Security Project*. [Online] May 27, 2012. [Cited: May 28, 2012.] The Zed Attack Proxy (ZAP).
78. **Wireshark.** Wireshark. *Wireshark.org*. [Online] April 6, 2012. [Cited: May 9, 2012.] <http://www.wireshark.org/>.

APPENDIX I: WEB APPLICATION HACKING CHECKLIST

The following list was modified for our attack scenario. The original can be found in the book *The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws* written by Dafydd Stuttard and Marcus Pinto [1].

1. Map the Application's Content
 - 1.1. Explore Visible Content
 - 1.2. Consult Public Resources
 - 1.3. Discover Hidden Content
 - 1.4. Discover Default Content
 - 1.5. Enumerate Identifier-Specified Functions
 - 1.6. Test for Debug Parameters
2. Analyze the Application
 - 2.1. Identify Functionality
 - 2.2. Identify Data Entry Points
 - 2.3. Identify the Technologies Used
 - 2.4. Map the Attack Surface
3. Test Client-Side Controls
 - 3.1. Test Transmission of Data via the Client
 - 3.2. Test Client-Side Controls over User Input
 - 3.3. Test Thick-Client Components
 - 3.3.1. Test Java Applets
 - 3.3.2. Test ActiveX controls
 - 3.3.3. Test Shockwave Flash objects
4. Test the Authentication Mechanism
 - 4.1. Understand the Mechanism
 - 4.2. Test Password Quality
 - 4.3. Test for Username Enumeration
 - 4.4. Test Resilience to Password Guessing
 - 4.5. Test Any Account Recovery Function
 - 4.6. Test Any Remember Me Function
 - 4.7. Test Any Impersonation Function
 - 4.8. Test Username Uniqueness
 - 4.9. Test Predictability of Auto-Generated Credentials
 - 4.10. Check for Unsafe Transmission of Credential
 - 4.11. Check for Unsafe Distribution of Credentials
 - 4.12. Test for Logic Flaws
 - 4.12.1. Test for Fail-Open Conditions
 - 4.12.2. Test Any Multistage Mechanisms
 - 5.1. Exploit Any Vulnerabilities to Gain Unauthorized Access
6. Test the Session Management Mechanism
 - 6.1. Understand the Mechanism
 - 6.2. Test Tokens for Meaning
 - 6.3. Test Tokens for Predictability
 - 6.4. Check for Insecure Transmission of Tokens
 - 6.5. Check for Disclosure of Tokens in Logs
 - 6.6. Check Mapping of Tokens to Sessions
 - 6.7. Test Session Termination
 - 6.8. Check for Session Fixation
 - 6.9. Check for XSRF

- 6.10. Check Cookie Scope
- 7. Test Access Controls
 - 7.1. Understand the Access Control Requirements
 - 7.2. Testing with Multiple Accounts
 - 7.3. Testing with Limited Access
 - 7.4. Test for Insecure Access Control Methods
- 8. Test for Input-Based Vulnerabilities
 - 8.1. Fuzz All Request Parameters
 - 8.2. Test for SQL Injection
 - 8.3. Test for OS Command Injection
 - 8.4. Test for Path Traversal
 - 8.5. Test for Script Injection
 - 8.6. Test for File Inclusion