

Jean-Sébastien Guay-Leroux
jean-sebastien@guay-leroux.com

SMTP CONTENT FILTER SECURITY

<http://www.guay-leroux.com/>
April 2006

1. CONTENTS

1. CONTENTS	2
2. INTRODUCTION.....	3
3. THE CONTENT FILTERS	4
3.1. WHAT IS A CONTENT FILTER?	4
3.2. A COMPLEX CONTENT FILTER	5
3.3. THE PROBLEM.....	5
3.4. INTEGRATION SCHEME	6
3.5. EXTERNAL VULNERABLE COMPONENTS.....	7
4. PIRANA: A TOOL TO TEST CONTENT FILTER SECURITY.....	9
4.1. DESCRIPTION.....	9
4.2. TECHNIQUE USED TO MAKE EXPLOITATION MORE RELIABLE	10
4.3. DIFFERENT TECHNIQUES USED TO HIDE THE ATTACK	10
5. CONCLUSION	17

2. INTRODUCTION

Email has become an essential service for most people - who doesn't own an email address today? With time, it seemed obvious that numerous threats would come to light and propagate through this communication channel.

Some people, always seeking ways of making money, saw email as an excellent means of reaching a potential commercial market and solicitation by email (SPAM) was born. Virus writers also took advantage of this attack vector as a springboard for better infection. Thieves also got in on things, especially given the recent threat of phishing.

In order to protect themselves from attacks, system administrators implemented multiple technologies to protect their users, but are those programs as secure as the administrators would like them to be? And is the software ready to face all the malicious content found on the Internet today?

This paper looks at problems related to email filtering technology and presents a tool that will help security experts evaluate the security of a content filter.

3. THE CONTENT FILTERS

3.1. WHAT IS A CONTENT FILTER?

A content filter is a system that acts after the SMTP server receives email and applies various filtering policies defined by a network administrator. Once the scanning process is finished, the filter decides whether the message will be relayed or not.

The typical tasks a content filter performs on an email:

- Scan each attachment for virus-infected content
- Scan the email to see if it is SPAM
- Block dangerous content for the end user (MUA exploits, zip bombs, etc.)
- Block certain types of attachments (.EXE, .COM, PIF, etc.)
- Apply a whitelist and blacklist system

Figure 1 shows a simple schema of the path of a message in a content filter.

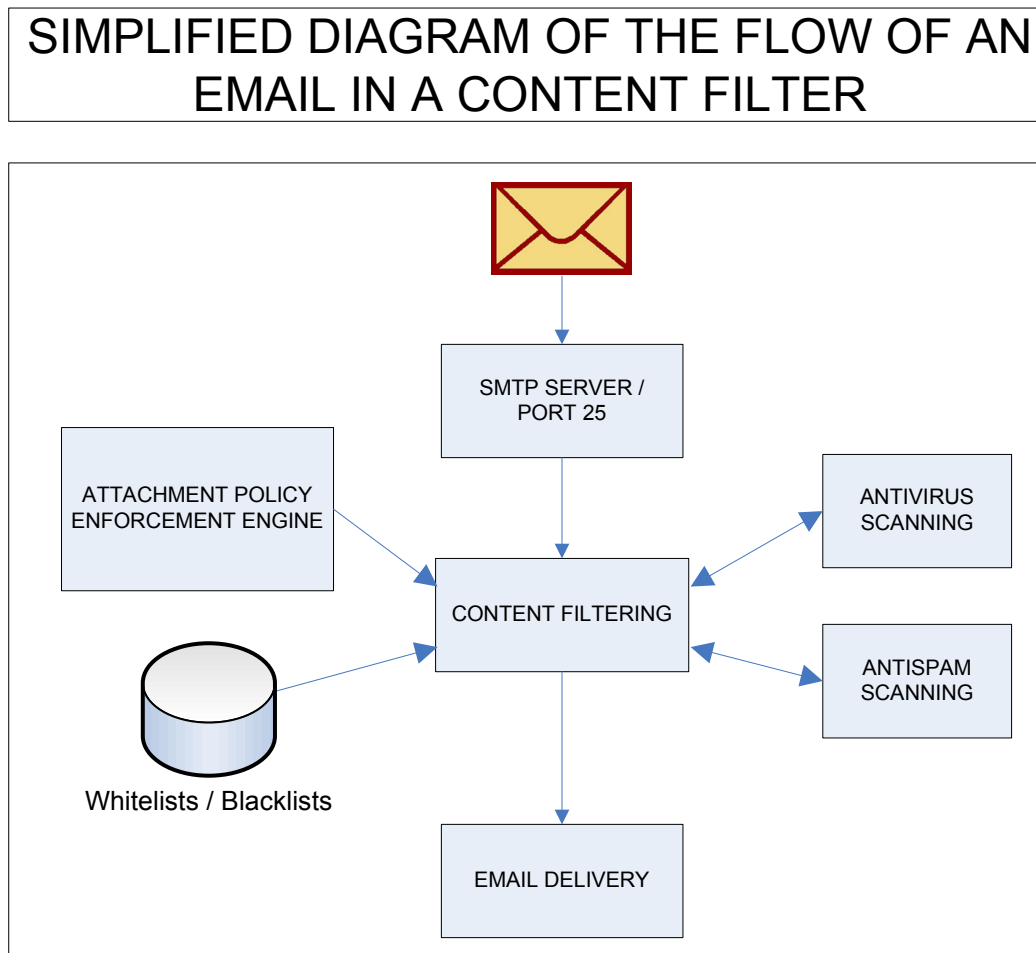


Figure 1: Simplified diagram of the flow of an email in a content filter

3.2. A COMPLEX CONTENT FILTER

Filtering email to eliminate viruses or SPAM, to prevent online fraud, or to establish an email policy, requires complex technology.

Any given solution must integrate decompression and decoding functions considering the many different message formats that now exist. Those functions are generally performed by optimized modules and libraries, which take care of the wide variety of possible situations. Usage of those libraries facilitates the development process of commercial solutions.

Often, developers have to use what is already available. They will integrate various components into a system in order to carry out a particular task. They will integrate software and libraries that will help them decompress, classify, decode, and sweep the email and its content.

The content filter, which at the beginning appeared to be a very simple tool, becomes a package of several software components in which the quality of the code, maintenance history, and safety can sometimes vary a great deal.

3.3. THE PROBLEM

The presence of many components around the core of the content filter helps the programmer concentrate on developing the main code of its filter; however, for many reasons the quality of the generic code around the filter may be dubious.

These tools were not necessarily created to be integrated in packages intended to improve the network safety. They were programmed with the goal of offering only basic functionality, often without integrating good secure coding practices. Today their robustness and safety is quickly put to the test by hackers, security experts and researchers.

Moreover, the threats and the need for new functionality in a content filtering device change and evolve quickly. Developers must quickly adapt and integrate various technologies, which are not necessarily mature in terms of safety. More often than not, the various tools and libraries, especially with older components, will not have safety or functional updates, which does not necessarily mean that they are free of any flaws!

In summary, the content filter's administrator must rely on unsafe libraries and generic modules. The safety of the solution will be dependent on the robustness of the filtering device's various components. In other words, the weakest link in the chain will probably be found in one of these components.

3.4. INTEGRATION SCHEME

The following diagram demonstrates at which stage in the scanning process each potentially vulnerable component is integrated. Figure 2 illustrates the wide variety of generic software components generally found in this kind of content filter. Any security vulnerability in one of these components can be fatal to the integrity of the system.

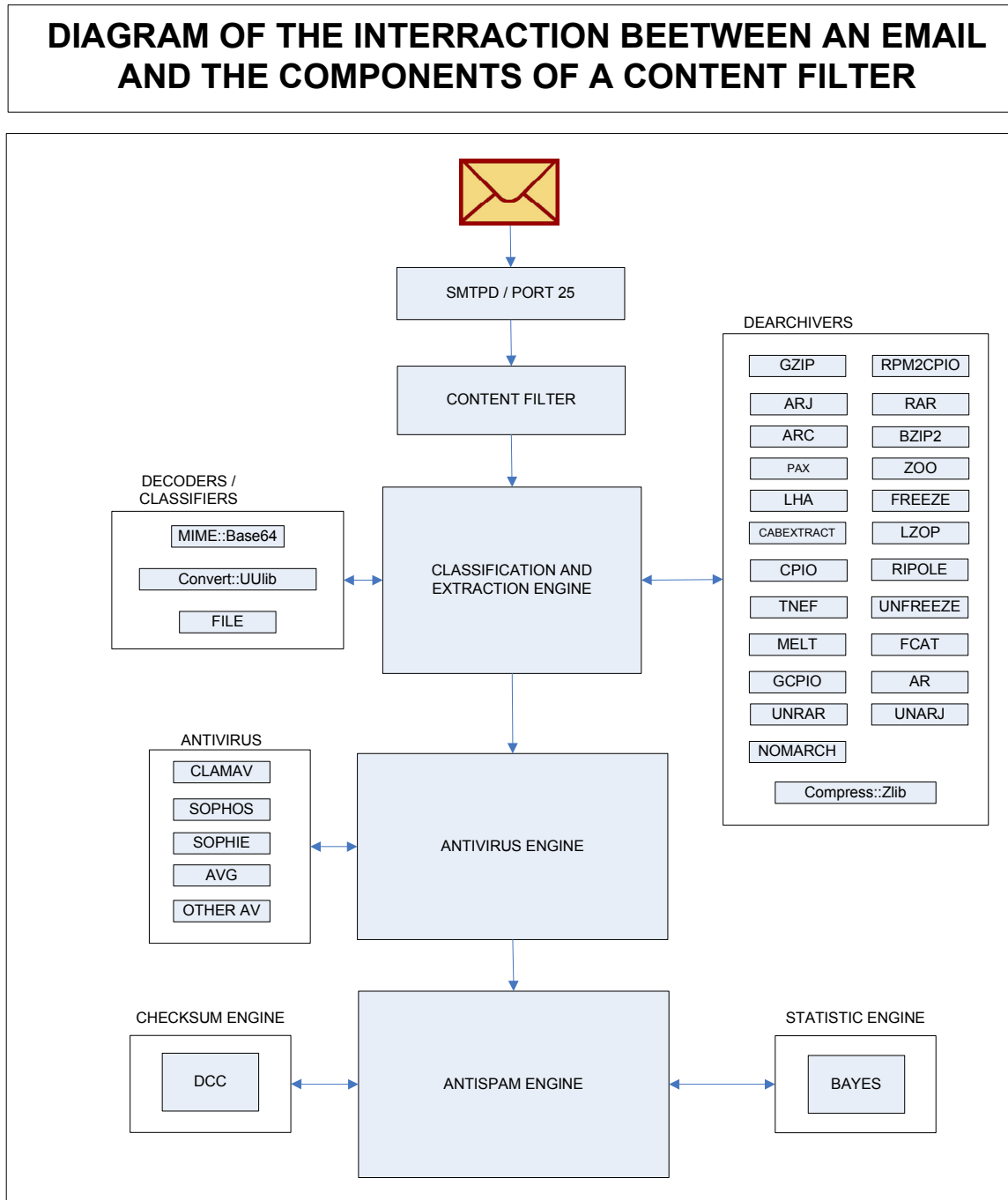


Figure 2

3.5. EXTERNAL VULNERABLE COMPONENTS

The following lists the five categories of external components that a content filter could integrate and which could contain security vulnerabilities.

3.5.1 Archivers

Virus writers quickly understood that they could avoid antivirus scan by encapsulating their virus in an archive. By using social engineering to convince the end user to open the archive, they obtained a virus, invisible to the antivirus system, that was, in fact, dangerous for the end user. Integration of various decompression components in the solution became necessary.

To achieve this, the content filter classifies each attachment by associating it with a file type - this task is carried out by the popular "file" tool available on many UNIX systems. Once the classification is done, the filter looks at the file type to see if it is a known archive type, such as ZIP, ARJ, LHA, etc. and tries to associate it with an unarchiver for the identified file. If all of those conditions are verified, the filter calls the unarchiver with the filename as an argument and continues recursively sweeping the content of the archive.

3.5.2 Decoders

In order to properly carry out its tasks, the content filter must be able to extract various attachments from the email. Sometimes, content can be incorporated in email directly uuencoded, or possibly as a MIME piece. There are many different ways of attaching these parts, and most email clients will not do it in exactly the same way. The content filter must be able to recognize and accept all different forms of encoded content, as a normal email client would do.

Therefore, the filters must, as accurately as possible, be able to emulate the behaviour of email software in order to protect them.

3.5.3 Classification program

A classification program identifies the file type by examining the first few bytes of a file and comparing them with a database containing several accepted definitions of types. Therefore, it is capable of determining if an accompanying document is an executable, an image, a PDF document, etc.

Content filters will usually not trust the extension of the accompanying document to make a judgment because it is a much less reliable technique. It simply calls the classification program and passes the file as an argument, and the software returns an exact interpretation of its true nature.

This type of utility is also useful in the decompression of archives. It allows confirmation of the file type before launching the decompression process, saving invaluable computing resources.

3.5.4 Antivirus

Of course, one of the main tasks of a content filter is protection against virulent content. An army of antivirus, commercial, and open source products are available for integration in the content filter.

3.5.5 Other software components

Other types of software components such as Bayesian statistical classifiers, content signature databases (DCC, RAZOR2), etc. exist. These components can be coded in many languages - some are, by nature, more secure than others. It is important to understand the mode of operation of all the implied constituents to estimate their exposure, as well as the risks associated with their use.

4. PIRANA: A TOOL TO TEST CONTENT FILTER SECURITY

4.1. DESCRIPTION

PIRANA is an exploitation framework that tests the security of a content filter. By means of a vulnerability database, the content filter to be tested will be bombarded by various emails containing a malicious payload intended to compromise the computing platform. PIRANA's goal is to test whether or not any vulnerability exists on the content filtering platform.

The tool is a PERL program, which builds email and attaches malicious payloads generated by various exploitation codes, then sends it to the target. Several techniques were developed to improve reliability and add discretion. The tool is modular and it is possible to add support for new vulnerabilities that could emerge in the future.

Figure 3 shows numerous vulnerabilities found in the generic components of a content filter. This list does not claim to be exhaustive and shows only a limited number of vulnerabilities. The availability of an exploitation code for each of these vulnerabilities is also shown.

VULNERABILITY TABLE OF SOFTWARE COMPONENTS USED IN A CONTENT FILTER				
Vulnerable component	OSVDB #	Vulnerable versions	Actual version (01/02/2006)	Exploit available in PIRANA ?
LHA	5753	<= 1.17	1.17*	YES
	5754	<= 1.17		YES
Zlib	17827	<= 1.2.2	1.2.3*	NO
Unarj	11695	<= 2.65	2.65*	YES
File	6456	<= 3.38	4.16	YES
	12255	<= 4.11		NO
Convert ::UUlib	15867	<= 1.05	1.06	NO
Zoo	Xxx	2.10	2.10*	YES
Clamav	20482	<= 0.87	0.88	NO

Figure 3: Vulnerability table of the components

* Patch provided by the vendors

4.2. TECHNIQUE USED TO MAKE EXPLOITATION MORE RELIABLE

4.2.1 Possible brute force attack

Exploitation of a buffer overflow or format string vulnerability involves many unknown but required variables to successfully achieve the attack.

In the past, many techniques were developed to facilitate exploitation, and reduce the range of addresses of the unknown variables. One of these methods consists in prepending the "shellcode" with a series of "NOP" to facilitate exploitation.

Brute force is a well-known technique of maximizing the chances of exploiting a program. If the context allows, it is possible to sequentially attempt potential valid values and allow successful exploitation. For example, during a buffer overflow attack, where to set the instruction pointer once it is controlled is unknown. With successive attempts, the correct value will eventually be reached and the program will be hijacked from its primary task.

For this technique to be applicable to a particular context, the environment must be able to reset itself to its initial state if the attack fails. It is not a problem in the case of a content filter because the filter never dies - even if the different called components fail. An example of a non-context resetting environment would be the case of the "Bind" DNS server because once it dies, the service does not restart.

The brute force technique is implanted in PIRANA by joining several attachments with various offsets in a single email. The content filter will analyze each of the attachments in the email. This technique is much more difficult to detect because, instead of sending 50 emails with 50 different malicious files, only one email will be sent containing the 50 malicious files.

4.3. DIFFERENT TECHNIQUES USED TO HIDE THE ATTACK

Stealth is usually a desired attribute in an attack. In this case an obvious problem arises: there are two players who may notice the attack. These players are the administrator of the content filter and the user who receives a copy of the email.

Methods of hiding the attack and masking the identity of the source are discussed in this section.

4.3.1 IDS evasion by encoding attachments

The SMTP protocol essentially supports only a set of characters represented in 7 bits. At first, this limitation seemed to raise problems for the transmission of binary content or extended character sets.

To mitigate this limitation, it is possible to use the MIME specification. MIME allows content encoding represented under 8 bits with 7 bits.

This inherent limitation to SMTP becomes interesting for PIRANA because, by encoding the malicious content, the evasion of IDS is possible. These systems analyze the network traffic and search for obvious signs of attacks, for example a "shellcode" or a series of "NOP". If the malicious content is encoded it will go unnoticed with the majority of IDS.

Therefore, it is possible to realize two tasks at once with the MIME specification: allow the transfer of binary files and hide the attack from IDS.

4.3.2 Joining a virus

Today, the majority of the content filters include one or several antiviruses in their architecture, allowing them to intercept content and prevent an infection early in the process of delivery.

Sometimes, the content filter will handle the entire message before deciding what to do with the email. It will sweep the message with an antispam and an antivirus, it will decode the content, uncompress the archives and associate a file type to each attachment. These tasks will be executed before making a decision about what to do with the message.

If the content filter detects a virus, odds are the message will never reach its final destination. On the other hand, the content filter will still process the entire message before taking action on what to do with it. The malicious code, intended to take control of the filter, is then processed by the vulnerable components.

It is possible to take advantage of this content filter behaviour to better hide the attack.

TABLE OF ACTIONS AND POSSIBLE CONSEQUENCES ON AN INFECTED EMAIL	
Action	Visibility of the attack
Message destruction without regard to its content	Low
Message put in quarantine	Moderate
Attachment destruction	Moderate

Figure 4

4.3.3 Sending from a free email account

It is also possible to exploit the anonymous nature of several sites offering free email addresses. By generating the various malicious files and by sending them from the web interface of a freemailer, it is possible to launch a completely anonymous attack.

The only forensic data left on the target server will be the IP address and the email address used during the attack - both anonymous.

4.3.4 Compression of all the malicious files into one archive

Sometimes, PIRANA needs to send many files simultaneously. For example, when using a brute force attack, the program is required to send several files, each containing a different offset.

This method suggests compressing all the files in only one archive. As a result, the content filter will take care of the decompression of the archive, and will process each file.

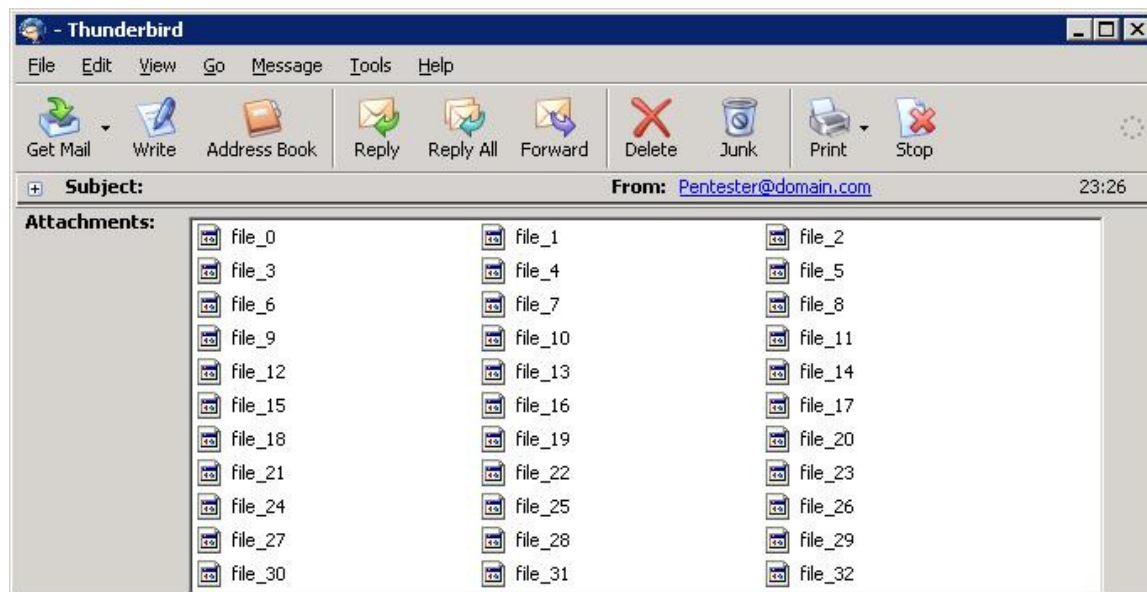


Figure 5: An email that did not undergo the compression technique

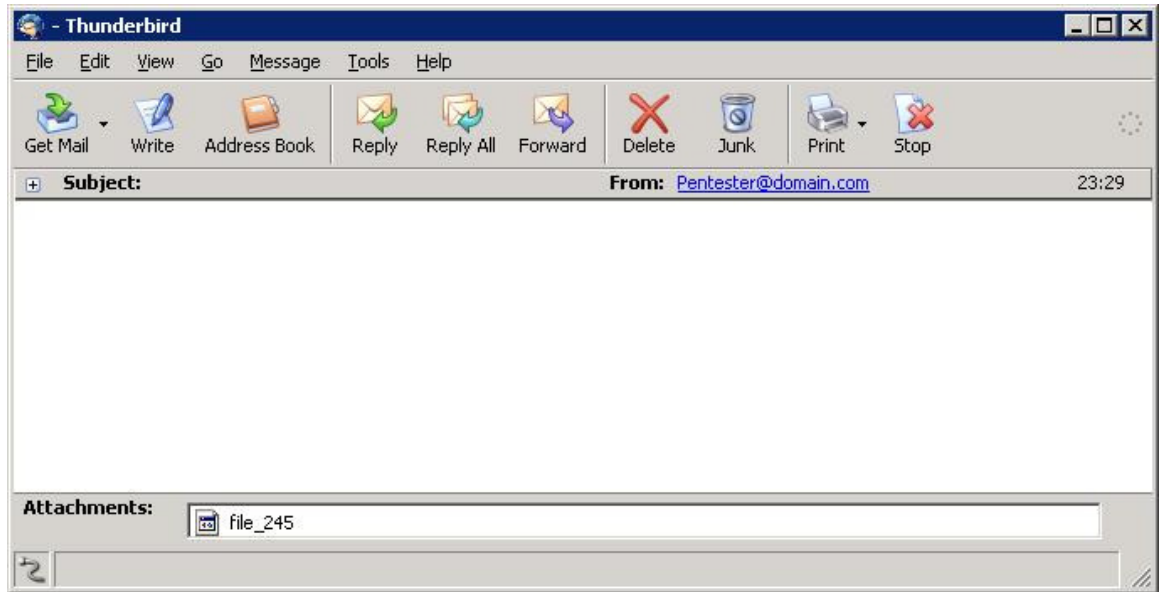


Figure 6: An email that underwent the compression technique.

A message with a single attachment remains more discreet and generates fewer log entries.

4.3.5 Malware removal from the attachment list

Of course, with only a single attached document, contrary to several hundreds, the hostile message is stealthier. To reach a superior level of discretion, removing all the attachments likely to rouse the email recipient's attention is recommended.

PIRANA implements two techniques to eliminate malicious files appearing in the attachments list in the email client.

Invisible picture technique

When a message is sent in HTML format, the email client shows a list of all the attachments. When the HTML code directly references one of those attachments, it disappears from the attached file list. PIRANA uses this characteristic to conceal malicious files.

To achieve this, it only needs to attach, in MIME pieces, the various files that PIRANA will use to test the security of the content filter. To access the attachments later, a Content-ID field will be assigned to them (with which referencing of the malicious file in the HTML is possible.) Once this is done, it will be necessary to reference all the MIME pieces previously attached by using an appropriate HTML directive.

The following directive integrates the attachment file-01 into the final HTML email, using the keyword "cid" :

```

```

multipart/alternative technique

By definition, the MIME type "multipart / alternative" defines alternative versions of the same information. When the email client sees the MIME directive, it is going to choose the best version of the attachments that it can display on a screen.

For example, if an e-mail is "multipart / alternative" type it could contain a text and a HTML version of the same document. The email client will then decide, according to its parameters, the best version to display.

PIRANA uses this feature. All malicious attachments will be specified as an alternative version of the final HTML email displayed to the client. Obviously, the email client will never be able to display the binary attachments, so it will fall back on the last version it can display, which is the HTML document.

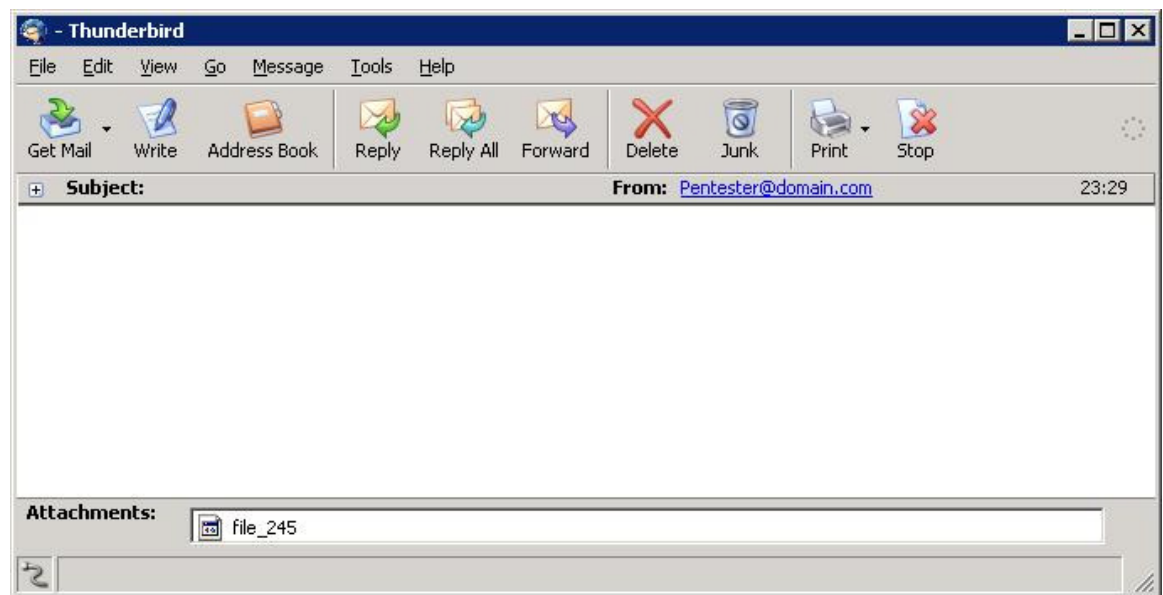


Figure 7: Email sent without the attachment removal technique

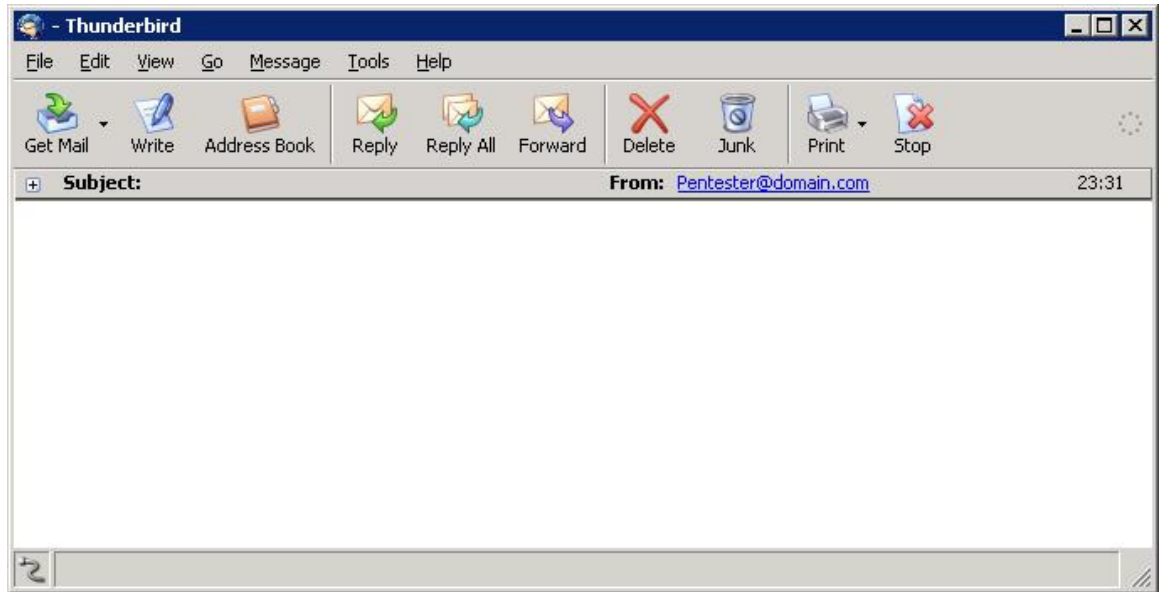


Figure 8: The same email sent with the attachment removal technique

Figures 7 and 8 present the visible content for the same payload. The email without the visible attachment goes unnoticed.

4.3.6 Disguise the message to make it less suspicious

It is also possible to use social engineering to be stealthier. If a user receives a suspect email, he could contact his network administrator who would then be able to analyze the details of the attack and identify the source.

On the other hand, if the hostile email is disguised as a harmless email the user will not consider contacting his network administrator necessary and will probably just delete the email.

Looking at both emails below, it is extremely difficult to believe they contain malicious code intended to compromise the security of the content filter.

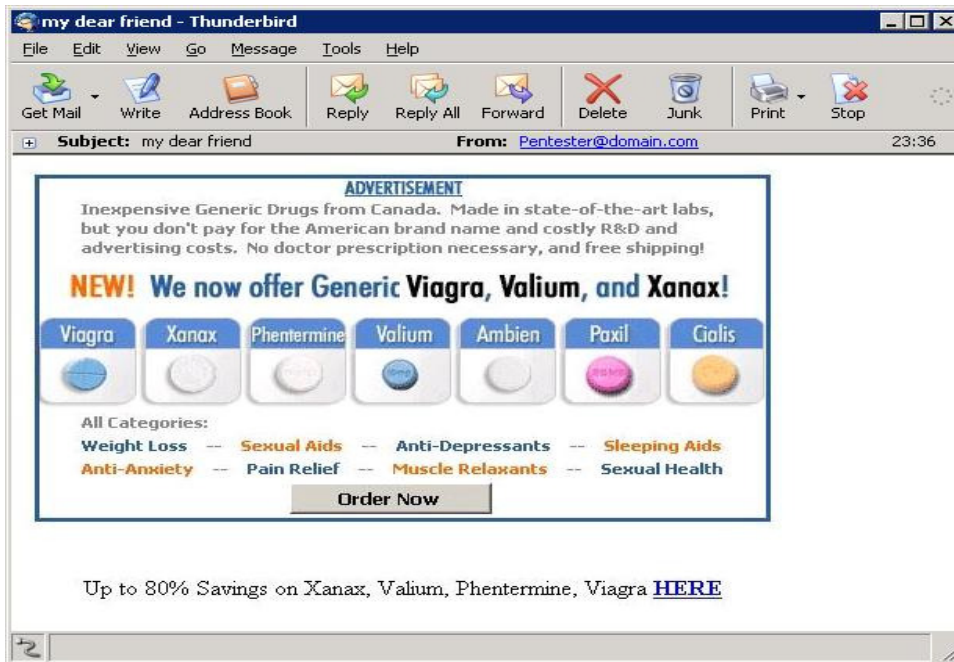


Figure 9: Malicious email disguised as a simple SPAM

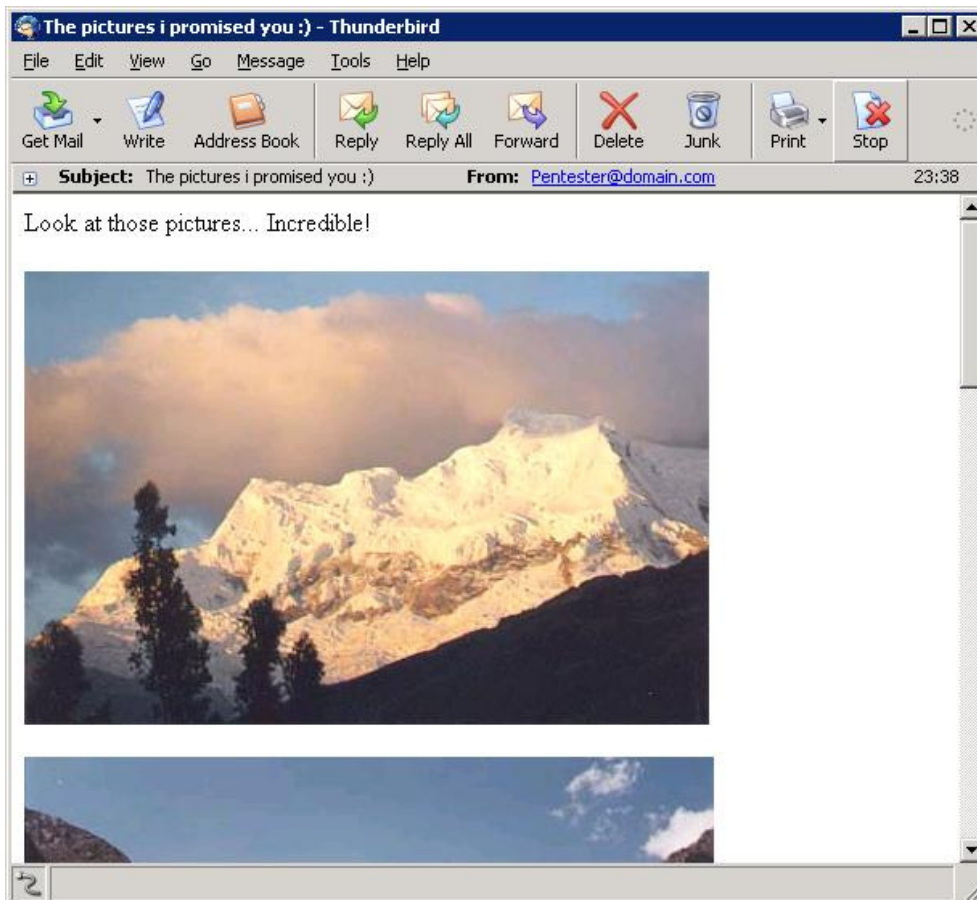


Figure 10: Malicious content concealed in a simple email

5. CONCLUSION

The main objective of this paper was discussion of content filters and their associated vulnerabilities. The different components of a content filter proved to be the weakest links in the chain.

The position of a SMTP content filter in a network, as well as the nature of the data that circulates through it, make it a target of choice. PIRANA was written to demonstrate that the risk was not hypothetical but very real. Techniques used by PIRANA to improve stealth and reliability were also introduced.

Remember that security is not a product you buy. It is impossible to be secure if the corrective measure being implemented is full of security holes. Think seriously about it the next time you buy a security product – you don't want your security product to be the Achilles' heel of your network.