# Table of Contents

1. Overview ......................................................... 1
   What is Content Filtering? ........................................ 2
   Content Filtering Overview ..................................... 3
   Content Filtering Limitations ................................... 4

2. Building a HIPAA Compliant Content Filtering System. .......... 5
   What is a Content Filtering System? ............................. 6
   What does HIPAA require? ....................................... 6
   Creating a Simple HIPAA Content Filter That Scans for Protected Health Information (PHI) .... 6
   Testing the New Content Filter ................................ 11
   Conclusion ......................................................... 14

3. Combining Content Filters Within a Rule ......................... 15
   Adhering to the HIPAA Guidelines .............................. 16
   Creating Multiple Content Filters Within a Rule .............. 19
   Testing the New Combined Content Filtering System ........ 22
      Sending a Test Message Through the Content Filter ....... 22
      Understanding the CSV File Contents ....................... 24
   Conclusion ......................................................... 27

4. Chaining Multiple Content Filters Using RuleSets .............. 29
   Limitations of Combining Filters Together in a Rule .......... 30
   RuleSet chaining Overview ..................................... 31
   Chaining a Content Filter to a Ruleset – Appliance Example. 35
   Another Content Filter Chaining Scenario .................... 38
   Examining the Results .......................................... 41
Conclusion ................................................................. 42

5. How Content Filtering Works .............................................. 43
   Building Blocks of an Email Message ................................. 44
      What is the MIME Format? ............................................ 44
      What is a MIME Part? .................................................. 44
   Scanning for Words, Phrases, or Patterns in a MIME Formatted Message ....................................................... 45
   What is a Content Type in the MIME Format? .......................... 46
   Scanning for Content Types in a MIME Formatted Message .......... 47
      Combining Lexicon, Regex, and Content Type Scanning to Form a Content Filter ............................................. 47

6. Creating Content Filter Files ............................................... 49
   Content Filter Files ...................................................... 50
      Lexicon Files ................................................................ 50
      Regular Expression Files .............................................. 54
      Content-Type Files ...................................................... 54

7. Applying the Knowledge ..................................................... 59
   Initial Design ............................................................... 60
   Final Tuning ................................................................. 63

A. Content Filter Configuration Parameters ............................... 65

B. Content Filter CSV Report File Structures .............................. 67
   Content Filter Files ...................................................... 68
      Lexicon Files ................................................................ 68
      Regular Expression Files .............................................. 72
      Content-Type Files ...................................................... 73

Index ................................................................. 77
Overview

This chapter contains the following sections:

- “What is Content Filtering?” on page 2
- “Content Filtering Overview” on page 3
- “Content Filtering Limitations” on page 4
WHAT IS CONTENT FILTERING?

Most people are familiar with web content filters that restrict access to “inappropriate” web sites based on web page content. An email content filter is similar, but rather than scanning web content, it scans email messages looking for specific words, patterns and content-types. Email content filters are often used like their web content equivalents, searching for malicious, inappropriate or spam containing content of various types. Because the IronPort Encryption appliance is used mainly for security, the Content Filter is primarily designed to scan for content indicating that a message should be encrypted—frequently as part of an effort to implement HIPAA, GLBA or similar regulatory controls. For example, messages can be selected for encrypted delivery if they contain specific health care words like “diagnosis” or “diabetes” or patterns that appear to be social security numbers.
CONTENT FILTERING OVERVIEW

A Content Filter enables you to inspect the content of an email, and then route, notify, encrypt or take some other action based on the results of the inspection. The Content Filter looks for at least one of three basic elements in an email message: specific words or phrases, patterns, or content-types. For the first two types (words/phrases and patterns), the Content Filter scans the message line by line, character by character, and searches for instances of the phrases and patterns in the content of the message.

While words, phrases and patterns are familiar to everyone, content-types may be a new concept. Content-type inspection allows you to look at the content types (for example, images or Word documents) found in a message rather than the textual or binary content of the message itself. To clarify, this allows you to make policy decisions based on the presence of content with specific content-types.

The following general steps are taken in analyzing the message and attachments:

- The lists of words, phrases, patterns and content-types that the Content Filter will scan for are configured in the Content Filter. For more information on compiling and using these lists please see Chapter 6, Creating the Content Filter Files.

- Scoring “weights” are assigned to the different words, phrases, patterns and content types. Each message has a score associated with it in the Content Filter that starts at zero. The message score is affected each time one of the identified words, phrases, patterns or content-types is encountered by the Content Filter.

  Note: Each instance of a word, phrase, etc. increases the cumulative score for a particular message. Therefore, if a particular word is encountered more than once the message score is affected more than once.

- When the score of a message meets or exceeds the Content Filter’s configured maximum score the message is considered a match and is routed accordingly.

To learn more about how the specifics of the Content Filter mechanism works, please see Section 2, Advanced Content Filter Configuration.
CONTENT FILTERING LIMITATIONS

HIPAA, GLBA and other privacy statutes generally require that you protect non-public information that can be associated with an individual. For example, a message from a stock broker might contain information about a stock trade you executed, including your customer ID, your address and the details of the trade. On reading the messages a human being can immediately determine whether the content falls under the GLBA guidelines. However, this is a more difficult task for a computer.

The Content Filter does a lexical analysis of each message but it is largely insensitive to context or semantics. As a result it is very important to tune your Content Filter. In this context ‘tuning’ is the process of applying your local knowledge of context and semantics to the content filtering process. For example, if you know that all patient communications include the patient ID, and that ID has a pattern that is readily distinguishable, it can be easy to identify messages that should be secured.

Content filtering is not perfect – even after tuning your Content Filter will suffer from some level of false positives and/or false negatives. For example, consider the phrase “diagnostic test”. It is reasonable to assume that this generally corresponds to the message containing health information if it comes from a health care worker. However, on some occasions the worker may be sending a personal email about problems with his car. With tuning you can reduce, but not eliminate, such filtering inaccuracies.
Building a HIPAA Compliant Content Filtering System

This chapter contains the following sections:

- “What is a Content Filtering System?” on page 6
- “What does HIPAA require?” on page 6
- “Creating a Simple HIPAA Content Filter That Scans for Protected Health Information (PHI)” on page 6
- “Testing the New Content Filter” on page 11
- “Conclusion” on page 14
What is a Content Filtering System?

The Content Filter is designed to be part of a complete Content Filtering System rather than a stand-alone tool. Using the architecture available in the IronPort Encryption appliance policy engine (rules, rulesets, and applications), multiple Content Filters can be combined or chained together to implement more complex content filtering, reporting and notification logic. In the following sections and chapters we will build a HIPAA Content FilterS, starting with a simple single-filter test and progressively adding capabilities.

What does HIPAA require?

“Protected health information” is regulated: The HIPAA privacy rule requires covered entities to protect the privacy of “protected health information” (PHI). PHI is defined as information held or disclosed by the covered entity in any form (electronic, paper records, oral communications) that:

1. Identifies an individual, and
2. Relates to:
   - the individual's past, present, or future physical or mental health or condition;
   - the provision of health care to the individual; or
   - the past, present, or future payment for the provision of health care to the individual.

Creating a Simple HIPAA Content Filter That Scans for Protected Health Information (PHI)

Adding and Configuring the Content Filter Test

A Content Filter is set up within the IronPort Encryption appliance configuration as a test inside of a rule. For more information about the relationship between rules and tests, see the IronPort Encryption Appliance Configuration Manual. The Content Filter test determines the basic behavior of the Content Filter and is also where you specify the name and location of the lexicon, regular expression and content type files.
This flowchart shows policy engine processing with a Content Filter configured:

1. **SMTP Message enters Policy engine**
2. **Check Policies in Root RuleSet**
3. **Does message contain enough Medical Terms to trigger encryption?**
   - Yes: **Encrypt Message**
   - No: **Continue with root policy tests**
4. **Encryption Decision**
   - Yes: **Encrypt Message**
   - No: **Deliver Message**
To demonstrate the configuration and capabilities of the Content Filter, the following example shows the creation of a new rule.

To add a Content Filter rule and test:

1. Log in to the Administration Console. Click the Configuration tab and navigate to Configuration > SMTP Adaptor > Router Rulesets > root.

2. Add a new rule named Check for HIPAA Content.

   **Note** — The order in which rules and rulesets appear impacts the behavior of the system. For example, if a new rule/ruleset is placed at the end of the list, the message may not use that rule/ruleset as intended if a rule/ruleset higher up on the list selects the message. Typically, the Content Filter rules are placed near the end of a ruleset. See the *IronPort Encryption Appliance Configuration Manual* for information about using rules and router rulesets.

3. Click the plus sign ( + ) next to the new rule to open the rule configuration.

4. In the Test section, select the Content Filter matcher from the Test Class drop-down list and click the Add Test button.
You add a new Content Filter test in the HIPAA Content Filter rule as shown here.

5. Click plus sign ( + ) next to the newly created Content Filter test to edit the configuration parameters for the content filter.
Here you can edit the parameters used to set up a Content Filter. For more information, see Appendix A, “Content Filter Configuration Parameters,” on page 65.

6. Change the following configuration parameters values:
   - Lexicon File = HIPAA.lex
   - Regular Expression File = HIPAA.res
   - Content Filter File = HIPAA.con
     These three files govern what words, phrases, patterns, and content types the Content Filter will look for in a given message.
   - CSV Directory = /<install_dir>/samples/content_filter

7. In the Actions section, set On Match ‘Send to Application’ to ‘Registered Envelope Enrolled’.

8. Click the **Deploy Changes** button.

9. Restart the SMTP Adaptor when prompted.

You have now configured a basic HIPAA content filter and are ready to start content filtering with a HIPAA Content Filters.
Testing the New Content Filter

Sending a Test Message Through the Content Filter

In order to test the newly added HIPAA Content Filter and examine the results we will need to send a test message through the IronPort Encryption appliance. You can send the message through the IronPort Encryption appliance using your favorite email client. The following shows a message composed in Outlook.

![Message example](image)

Dr. Jane Doe,

I have read the x-rays you forwarded and I am convinced they indicate a broken heart. While the fracture does not appear to be complete, I’ve forwarded the lab results and x-rays to Dr. Strangelove for a second opinion.

My office assistant informs me that we can submit the bill to medicare for payment. I’ve asked Ruth to call you in the morning to verify the information.

Please feel free to call me if my office if you want to discuss further.

Thank you.

Dr. Dre

Copy and paste the following text into your message:

Subject:

Preliminary diagnosis

Body:

Dr. Jane Doe,

I have read the x-rays you forwarded and I am convinced they indicate a broken heart. While the fracture does not appear to be complete, I’ve forwarded the lab results and x-rays to Dr. Strangelove for a second opinion.

My office assistant informs me that we can submit the bill to medicare for payment. I’ve asked Ruth to call you in the morning to verify the information.
Please feel free to call me if my office if you want to discuss further.

Thank you,

Dr. Dre

If you do not have an email client to use, you can send the message directly from the Administration Console. Click the Tools tab and select Test Messages. The Compose Message page is displayed. Use the text above to compose a message. Click the **Send** at the bottom of the form. The message ‘Message sent successfully’ should display above the From label. The message will be sent to the HIPAA Content Filter for filtering. To confirm, click on the Home tab in the Administration Console.

**Examining the Results**

Each Content Filter test creates and continuously updates two files that contain the ongoing results of the Content Filter’s scanning. You have already configured your HIPAA Content Filter to log to these files by setting the CSV Directory in the Content Filter test configuration. The following files will be generated in the specified directory:

- Summary File – cfsummary.csv
- Detail File – cfdetails.csv

**Summary File (cfsummary.csv)**

The cfsummary.csv file contains summary data returned by the Content Filter. It will contain one entry that corresponds to each message that has gone through the Content Filter. You can view the file in Excel.

**Detail File (cfdetails.csv)**

The cfdetails.csv file contains detailed data on every single match found by the content filter. This means that there can be several entries corresponding to a single message that has gone through the content filter. There also can be zero entries for a given message if that message had no potentially sensitive content. You can view the file in Excel.

**Understanding the CSV File Contents**

Once a message has gone through the Content Filter and its summary and detail records have been written, it is easy to determine exactly why a particular message was or was not selected by the Content Filter. Note the following points:

- Search the summary file for the subject of your message. Make sure that the subject is unique in the summary file (to make sure you have found the correct message). If it is not, you should check the time/date sent as well as the sender and recipients in order to track
down the appropriate message. You can look at all the per-message information here, such as the score and number of matching terms.

- The first field in the summary file will contain a unique message ID associated with that message. Copy this ID and open the details file to see information on individual matches made within that message by the content filter.

- Once in the details file, do a search for the message ID copied out of the summary file. You may not find any instances of this ID in the details file if the Content Filter found no matches in that message (the summary file has entries for every message going through the Content Filter while the details file may not).

- If there are one or more entries found for the message ID that means that matches were made for the message by the content filter. Each line in the details file containing the message ID found in the summary file describes one matched phrase, expression, or content-type found by the content filter. See the ‘Match Type’ entry in the section above to determine the type of match found.

When you are familiar with reading the summary and details CSV files, they can be instrumental in helping you fine tune the Content Filter.
Conclusion

You have now completed configuration of a basic content filter. You have also learned to use the CSV output files to analyze the Content Filter output and assist in tuning your Content Filter. For more advanced setup of the HIPAA Content FilterS continue on to the following chapter.
Combining Content Filters Within a Rule

In Chapter 2 you created a basic single filter HIPAA content filtering system. The advantage of the single filter approach is simplicity. The Content Filter test references a single set of Content Filter configuration files and logs to a single set of CSV files. However, this approach is extremely limited and implements very little contextual analysis. This chapter will give you the basic building blocks for adding some simple context to our HIPAA Content Filter analysis.

This chapter contains the following sections:

• “Adhering to the HIPAA Guidelines” on page 16
• “Creating Multiple Content Filters Within a Rule” on page 19
• “Testing the New Combined Content Filtering System” on page 22
• “Conclusion” on page 27
ADHERING TO THE HIPAA GUIDELINES

Referring back to Chapter 2, Building a HIPAA Compliant Content Filtering System, note that HIPAA regulations state that for a message to be considered PHI it must have both personal identifying information and health/medical information pertaining to the person identified. The requirement that both information types exist can be mapped to a content filtering system by creating two individual Content Filters, one each for medical terms and identifying information, and intersecting them. Only if both Content Filters match the message will it be selected by the Content Filter rule for routing.
Again, refer to the following flowchart for the general steps involved in this more accurate Content Filter system.

The message you sent through the IronPort Encryption appliance near the end of Chapter 2 is a good example of the limitations of a simple Content Filter when attempting to follow HIPAA
That message contained only health/medical type information but was still selected by our Content Filter. There was no HIPAA-related reason to secure this message. By using two separate Content Filters and combining them together we can ensure that only messages containing both types of PHI information get secured.
CREATING MULTIPLE CONTENT FILTERS WITHIN A RULE

You do not need to start over to build your new multi-filter rule. Open your existing HIPAA Content Filter rule. You’ll need to make a couple of configuration changes to the existing test and configure a second Content Filter test within the rule. Use the following steps to update the configuration:

1. Create the following directories for use in these sections:
   - HIPPA_Med
   - HIPPA_Priv
   - HIPPA_SSN

2. First, change the Match drop-down selection from “Any test” to “All tests”. The logic will now require that both the new Content Filter tests match and that both medical terms and individually identifying information are present, before a message is selected for encryption.

3. You’ll use two new pairs of lexicon and regular expression files in building the new multi-filter Content Filter rule: one pair for HIPAA medical terms and one for HIPAA individual identifying terms. Open your existing Content Filter test and make the following changes:
   - Change the Name parameter value to “HIPAA Medical Terms”.
   - Change the Lexicon File parameter to: `<Install_Dir>/samples/content_filter/HIPAA_Med.lex`
   - Change the Regular Expression File parameter to leave the parameter blank.
   - Change the maximum score to 30. You are only checking for the existence of medical terms in this test, so a single high value term should cause this Content Filter to match.
Change the CSV Directory to point to the HIPAA_Med directory you created in the previous chapter.

4. Next you will create a second test to scan for HIPAA personal identifying information. Keep in mind that the IronPort Encryption appliance logic will never get to this second test if the first HIPAA medical terms test does not match.

5. In the Tests section, select the Content Filter matcher from the Test Class drop-down and click the **Add Test** button. You will have a second Content Filter test in the HIPAA Content Filter rule as shown below. Click the ‘+’ symbol next to the newly created Content Filter test to edit the configuration parameters for the content filter. You will edit the parameters to configure this second test to scan for HIPAA privacy terms. Please see Appendix A, *Content Filter Configuration Parameters* for detailed information about all of the available Content Filter configuration parameters. Make the following changes to the test:

- Change the Name parameter value to “HIPAA Privacy Terms”
- Change the Lexicon File parameter to: `<Install_Dir>/samples/content_filter/HIPAA_Priv.lex`
- Change the Regular Expression File parameter to: `<Install_Dir>/samples/content_filter/HIPAA_Priv.res`
- Change the maximum score to 30. You have already determined that the message has HIPAA medical content in the prior test – now you’re checking to see if the message also contains Privacy information. If it contains any of the significant personal identifiers like
Social Security Number or an SSN numeric pattern the message very likely meets the HIPAA PHI requirements and should be secured.

- Change the CSV Directory to point to the HIPAA_Med directory you created in the previous chapter.
TESTING THE NEW COMBINED CONTENT FILTERING SYSTEM

Sending a Test Message Through the Content Filter

In order to test our new multi-filter HIPAA Content Filter and examine the results you will need to send a couple of test messages through IronPort Encryption appliance.

1. Send the same message you sent in the last chapter. This message contains only medical terms and should not match with your new filter. Once again, you can send the message through the IronPort Encryption appliance using an email client or through the Compose a Text Message feature accessed via the Tools tab of the Administration Console. The following image shows the message composed in Outlook.

   ![Message Composition Image]

   Dr. Jane Doe,

   I have read the x-rays you forwarded and I am convinced they indicate a broken heart. While the fracture does not appear to be complete, I’ve forwarded the lab results and x-rays to Dr. Strangelove for a second opinion.

   My office assistant informs me that we can submit the bill to medicare for payment. I’ve asked Ruth to call you in the morning to verify the information.

   Please feel free to call me if any office if you want to discuss further.

   Thank you,
   Dr. Doe

2. Copy and paste the following text into your message and send:

   Subject:
   Preliminary diagnosis

   Body:
   Dr. Jane Doe,

   I have read the x-rays you forwarded and I am convinced they indicate a broken heart. While the fracture does not appear to be complete, I’ve forwarded the lab results and x-rays to Dr. Strangelove for a second opinion.
My office assistant informs me that we can submit the bill to medicare for payment. I’ve asked Ruth to call you in the morning to verify the information.

Please feel free to call me if my office if you want to discuss further.

Thank you,

Dr. Dre

3. To confirm, click the Home tab in the Administration Console and verify that the counter for SMTP Delivery was incriminated, but the counter for RegisteredPxMail was not. This indicates that the message was sent unencrypted.

4. Next, send a second message as shown below through the new content filtering system.

5. Copy and paste the following text into your message and send:

Subject:

Preliminary diagnosis with patient information

Body:

Dr. Jane Doe,

I have read the x-rays you forwarded and I am convinced they indicate that your patient George Smith has a broken heart. While the fracture does not appear to be complete, I’ve forwarded the lab results and x-rays to Dr. Strangelove for a second opinion.

My office assistant informs me that we can submit the bill to medicare for payment. I’ve asked Ruth to call you in the morning to verify that George’s SSN is 123-45-6789.

Please feel free to call me if my office if you want to discuss further.

Thank you,

Dr. Dre
results and x-rays to Dr. Strangelove for a second opinion. My office assistant informs me that we can submit the bill to medicare for payment. I’ve asked Ruth to call you in the morning to verify that George’s SSN is 123-45-6789.

Please feel free to call me if my office if you want to discuss further.

Thank you,

Dr. Dre

Understanding the CSV File Contents

As described in the previous chapter you can now examine the contents of the CSV files to review the scan results for the two test documents. Remember that there are now two CSV file locations, the HIPAA_Med directory for the medical term Content Filter and the HIPAA_Priv directory for the privacy Content Filter.

First examine the CSV files in the HIPAA_Med directory. The two test messages used in this chapter intentionally contain the same medical terms matches. As a result, the CSV entries for both test messages are identical (except for message Id) and both messages match in the Medical Terms Content Filter.

1. Search the summary file for the subject of your message. Look at all the per-message information here, including the score and number of matching terms.
2. Look at the details file and check for matched phrases, expressions, or content types found by the content filter. Again, notice that the matches are identical in the two messages.

3. Note that both test messages matched in the Medical Terms Content Filter. In the new combined Content Filter system that is not enough to select the message for encryption. To meet the conditions for encryption in the new rule, the message also has to match for individually identifying content. Open the CSV files in the HIPAA_Priv directory. As you can see, the first test message did not match in the HIPAA Privacy Terms Content Filter and as a result wasn’t selected for encryption. The identical message did match the simple Content Filter in the last chapter. The additional context provided by combining the two filters allowed a more accurate determination and prevented a “false positive” in this case.

4. Now look at the results for the second message. This test, containing additional individually identifying information like the term SSN and a social security number, did match the HIPAA Privacy Terms Content Filter and as a result was selected for encryption.

Privacy CSV Summary
Privacy CSV Details

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;001901c7049a83751542003002a6c0@qq:stx.com&gt;</td>
<td>text/html</td>
<td>HPAA_Priv</td>
<td>body</td>
<td>W</td>
<td>name</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>&lt;001a01c7049a44749653003002a6c0@qq:stx.com&gt;</td>
<td>multipart/alternative</td>
<td>HPAA_Priv</td>
<td>subject</td>
<td>W</td>
<td>patient</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>&lt;001a01c7049a44749653003002a6c0@qq:stx.com&gt;</td>
<td>text/plain</td>
<td>HPAA_Priv</td>
<td>body</td>
<td>W</td>
<td>dr</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>&lt;001a01c7049a44749653003002a6c0@qq:stx.com&gt;</td>
<td>text/plain</td>
<td>HPAA_Priv</td>
<td>body</td>
<td>W</td>
<td>patient</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>&lt;001a01c7049a44749653003002a6c0@qq:stx.com&gt;</td>
<td>text/plain</td>
<td>HPAA_Priv</td>
<td>body</td>
<td>W</td>
<td>dr</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>&lt;001a01c7049a44749653003002a6c0@qq:stx.com&gt;</td>
<td>text/plain</td>
<td>HPAA_Priv</td>
<td>body</td>
<td>W</td>
<td>ssn</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>&lt;001a01c7049a44749653003002a6c0@qq:stx.com&gt;</td>
<td>text/plain</td>
<td>HPAA_Priv</td>
<td>body</td>
<td>W</td>
<td>dr</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>&lt;001a01c7049a44749653003002a6c0@qq:stx.com&gt;</td>
<td>text/plain</td>
<td>HPAA_Priv</td>
<td>body</td>
<td>W</td>
<td>ssn</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>&lt;001a01c7049a44749653003002a6c0@qq:stx.com&gt;</td>
<td>text/plain</td>
<td>HPAA_Priv</td>
<td>body</td>
<td>R</td>
<td>123-45-6789</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>&lt;001a01c7049a44749653003002a6c0@qq:stx.com&gt;</td>
<td>text/html</td>
<td>HPAA_Priv</td>
<td>body</td>
<td>W</td>
<td>name</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>&lt;001a01c7049a44749653003002a6c0@qq:stx.com&gt;</td>
<td>text/html</td>
<td>HPAA_Priv</td>
<td>body</td>
<td>W</td>
<td>patient</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>&lt;001a01c7049a44749653003002a6c0@qq:stx.com&gt;</td>
<td>text/html</td>
<td>HPAA_Priv</td>
<td>body</td>
<td>W</td>
<td>dr</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>&lt;001a01c7049a44749653003002a6c0@qq:stx.com&gt;</td>
<td>text/html</td>
<td>HPAA_Priv</td>
<td>body</td>
<td>W</td>
<td>ssn</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>&lt;001a01c7049a44749653003002a6c0@qq:stx.com&gt;</td>
<td>text/html</td>
<td>HPAA_Priv</td>
<td>body</td>
<td>W</td>
<td>dr</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>&lt;001a01c7049a44749653003002a6c0@qq:stx.com&gt;</td>
<td>text/html</td>
<td>HPAA_Priv</td>
<td>body</td>
<td>W</td>
<td>ssn</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>&lt;001a01c7049a44749653003002a6c0@qq:stx.com&gt;</td>
<td>text/html</td>
<td>HPAA_Priv</td>
<td>body</td>
<td>R</td>
<td>123-45-6789</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>&lt;001a01c7049a44749653003002a6c0@qq:stx.com&gt;</td>
<td>text/html</td>
<td>HPAA_Priv</td>
<td>body</td>
<td>W</td>
<td>name</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>
CONCLUSION

You have now completed configuration of an advanced Content Filtering System. This new system no longer selects messages for securing based solely on medical or personal information and instead requires that both types of information be present. For even more advanced setup of the HIPAA Content Filters continue on to the following chapter.
Chaining Multiple Content Filters Using RuleSets

In the preceding chapter, you created a more complex multi-filter HIPAA content filtering system. This system combined multiple Content Filters together creating a notion of context that the original simple Content Filter lacked. This is a major step towards creating an intelligent content filtering system. However, you can further utilize the IronPort Encryption appliance policy architecture to make even more accurate content filtering decisions.

This chapter contains the following sections:

- “Limitations of Combining Filters Together in a Rule” on page 30
- “RuleSet chaining Overview” on page 31
- “Chaining a Content Filter to a Ruleset – Appliance Example” on page 35
- “Another Content Filter Chaining Scenario” on page 38
- “Examining the Results” on page 41
- “Conclusion” on page 42
LIMITATIONS OF COMBINING FILTERS TOGETHER IN A RULE

The major limitation of combining multiple Content Filters within a single rule is that there can only be one intersection of sensitive message content types. The encryption server rule logic makes it possible to test for medical information and personal information in one rule, as configured in Chapter 3. However, the logic does not support something more complex like medical information AND (Social Security Numbers OR Zip Codes OR Patient ID’s or Other Privacy Information). The boolean logic is relegated to ANDing or ORing every test together. There is no concept of a complex AND/OR relationship between groups of tests.

In most organizations these more complex relationships between different sensitive information types can and do exist. Consider most types of personal information (phone numbers, social security numbers, patient ID numbers, etc.). These identifying numbers are very rarely found without an accompanying descriptive label, and vice versa. If a nine digit number is found in a message, but no instance of the phrases ‘Social Security Number’ or “Soc. Sec.” or the word ‘ssn’ is found, it is highly unlikely the number is an SSN. As a result, the number can be ignored by the content filter. Conversely, it is also very unlikely that the presence of the word ‘ssn’ without an accompanying nine digit number should be considered as an instance of personal information in a message. The sentence “Please call me with your ssn so I can complete your account setup” should not add to the Content Filter score of the message according to HIPAA guidelines.
RULESET CHAINING OVERVIEW

The IronPort Encryption appliance provides a simple and powerful method for chaining Content Filter tests, using RuleSets. You can use RuleSet chaining to direct messages through a sequence of content filter Content Filter rules. The default Content Filter configuration included on the appliance uses the following logic:

1. Rule 1 – look for Medical Terms
   a. Test 1 – use medical lexicon and regex to identify if the Medical Terms portion of the HIPAA requirement is met

If the message matches, go to the HIPAA content filter RuleSet

AND

2. Rule 2 – Look for Zip Code terms and numbers. If both exist, encrypt.
   a. Test 1 – use Zip lexicon (ex. “zip code”)
   AND
   b. Test 2 – use Zip regex to identify (ex. “90210”)

OR

3. Rule 3 – Look for Phone Number terms and numbers. If both exist, encrypt.
   a. Test 1 – use Phone lexicon
   AND
   b. Test 2 – use Phone regex to identify

OR

4. Rule 4 – Look for Social Security terms and numbers. If both exist, encrypt.
   a. Test 1 – use SSN lexicon
   AND
   b. Test 2 – use SSN regex to identify

OR

5. Rule 5 – Look for Email terms and numbers. If both exist, encrypt.
   a. Test 1 – use Email lexicon
   AND
   b. Test 2 – use Email regex to identify

OR
6. Rule 6 – Look for Privacy Code terms and numbers. If both exist, encrypt.  
    a. Test 1 – use Privacy lexicon  
    AND  
    b. Test 2 – use Privacy regex to identify
The following diagram shows the same logic in flowchart format:
This configuration allows the Content Filter system to first identify that HIPAA medical terms are present, then through a sequence of tests identify terms like labeled zip codes, social security numbers and other privacy terms. This ensures that both facets of the HIPAA guidelines are met while avoiding false positive matches.
The following screen shows the chained Content Filter system implemented on the IronPort Encryption appliance. As shown in the first screen, the Content Filter system starts with a Content Filter test that searches the message for all lexicon medical terms. This test does not include a regex scan because there are no medical terms that require pattern matching. Note that the matching score is low – 20 points. All this test is trying to determine is whether the first HIPAA requirement, existence of medical terms, is met in the message.

Another feature of this Content Filter rule is that, rather than pointing directly to a delivery application such as Registered Envelope, it directs selected messages to a Router RuleSet, called ‘HIPAA Content Filter’. As described in the previous section, this allows the system to test for various content combinations in context.

The HIPAA content filter Router RuleSet is predefined on the appliance. To configure a new ruleset, click the Configuration tab and then navigate to Configuration > SMTPModule > Router RuleSets. Add a unique name and optionally a description as shown in the next example, and click the Add Router Ruleset button.
A new ruleset appears under Router RuleSets. For more information, see the IronPort Encryption Appliance Configuration Manual.

The next two screens show the contents of the HIPAA content filter RuleSet, with a series of very specific filter tests and the configuration of the Content Filter rule for identifying SSN’s. The tests in the HIPAA content filter RuleSet include scans for social security numbers, zip codes and other privacy information. Looking at the SSN Content Filter rule logic, the configuration requires that all tests match, so both an SSN term and an SSN pattern must be found for the message to match. As a result, once a message matches the SSN rule you are now able to determine that:

1. HIPAA medical terms are present (from the earlier medical terms scan),
2. the term SSN or social security number (for example) is present, and
3. a pattern that matches a social security number is also present.
Since all the pieces required to meet the HIPAA guidelines for PHI are present, the message should be encrypted.
ANOTHER CONTENT FILTER CHAINING SCENARIO

To further illustrate the power of using ruleset chaining with multiple Content Filters consider the following example. This example uses ruleset chaining in a slightly different way to solve a different logical problem. In the previous sample configuration the goal was to combine several distinct AND and OR operations together to try to identify HIPAA content intelligently – in this case we will utilize ruleset chaining to make more than a simple TRUE or FALSE match decision about a message.

Normally, a match within any rule can only initiate two actions. Either the message is passed through to the next rule (a non-match scenario) or it is sent to a specific application for processing (a match scenario). This is useful in situations where you want to either encrypt a message or not, but falls short if you want to put other conditions into the decision making process. There is no way to deliver the message to different processors depending on how the message matched. For instance, the IronPort Encryption appliance allows you to configure the authentication strength, or “sensitivity”, of an encrypted message. When a stricter authentication method is configured a password must be entered each and every time the envelope is opened. A less sensitive method allows for authentication using cookieed (cached) credentials. In a frequent scenario a customer could elect to send messages containing very sensitive terms or larger lists of sensitive data using the envelope with more stringent authentication requirements while sending less sensitive messages via the less restrictive Envelope. Trying to do this in a single rule Content Filter system is impossible as the message either matches or doesn’t. All matches are delivered to the same place.

To get around this limitation, matched messages can once again be sent to a ruleset instead of an encryption application. The general logic for this type of match follows:

1. Rule 1 – look for Medical Terms
   a. Test 1 – use medical lexicon and regex to identify if the Medical Terms portion of the HIPAA requirement is met

   **If the message matches, go to the HIPAA content filter RuleSet**

   AND

   a. Test 1 – use SSN lexicon
      AND
   b. Test 2 – use SSN regex to identify

   **If the message matches, go to the new Sensitive Delivery RuleSet**

   AND

3. Rule 3 – Look for messages that matched with very high scores
   a. Test 1 – Were very sensitive medical terms matched?

   OR
b. Test 2 – Were large numbers of Social Security numbers matched?
4. IF Yes, send the message encrypted and require a password for each opening.
5. IF No, send the message encrypted but allow the recipient to authenticate using secure credentials stored in a cookie on the desktop.
The following diagram shows the same logic in flowchart format.
EXAMINING THE RESULTS

After sending messages through your new chained content filter, use the same process you followed in Chapters 2 and 3 to examine the results. The only difference is that you now have a few directories to check for CSV files, one for each Content Filter rule. All CSV files will reference the same message ID, so it is possible to cross-reference the CSV match information in the medical terms Content Filter rule with the information in the HIPAA content filter rule where the message matched. See Appendix B for additional information on the content of the CSV files.

This ruleset can then do further testing on the matched message. For example, a test can be run on the actual score that the message received from the content filter. The message can then be routed to different sensitivity encryption applications depending on the results of this testing.
CONCLUSION

You now understand all of the component parts of a comprehensive Content Filter System. This Content Filter System allows you to add some real context to your scanning process only selecting messages based on criteria that closely match the HIPAA requirements. The following chapters and Appendices will help provide additional background and detail on these Content Filter capabilities, but you have all the basic tools to implement a successful Content Filter system.
How Content Filtering Works

This chapter contains the following sections:

- “Building Blocks of an Email Message” on page 44
- “Scanning for Words, Phrases, or Patterns in a MIME Formatted Message” on page 45
- “What is a Content Type in the MIME Format?” on page 46
- “Scanning for Content Types in a MIME Formatted Message” on page 47
BUILDING BLOCKS OF AN EMAIL MESSAGE

In order to fully understand how content filtering works, it is necessary to understand the way email messages are constructed.

What is the MIME Format?

The MIME (Multipurpose Internet Mail Extensions) format is the worldwide standard used by mail servers and clients available today. This format is used to encode messages in a uniform way so that viewing and transmission of messages can be done on multiple competing platforms. For an overview of the MIME RFC please look here: http://www.apps.ietf.org/rfc/rfc2045.html

What is a MIME Part?

A MIME message is composed of one or more MIME ‘parts’. A part is a combination of a data section (the content) and a metadata section (the MIME headers). These headers describes the type, format, encoding, etc. of the associated data section. Without these headers a mail client would have no idea how to display the content in the part.

Here is an example of a very simple mime message, including its headers.

```plaintext
Message-ID: <20363353.1166134931457.JavaMail.jsmith@jsmith-737>
From: test_sender@ironport.com
To: test_recipient@ironport.com
Subject: example MIME message
MIME-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
Dear TestRecipient,
Example MIME Message contents section.
Sincerely,
TestSender
```

The first seven lines, up to and including the line beginning with “Content-Transfer-Encoding”, are MIME headers describing the contents of the email message and how they should be displayed. The actual contents start with “Dear TestRecipient,”. This is a single part MIME message (i.e., there are no attachments or alternative content).
SCANNING FOR WORDS, PHRASES, OR PATTERNS IN A MIME FORMATTED MESSAGE

Part of the functionality of the Content Filter is to search for words, phrases, and patterns in an incoming message. Words and phrases are searched for by the ‘lexicon’ portion of the Content Filter while patterns are searched for by the ‘regex’ portion of the Content Filter (regex stands for regular expression; you can readily find more information on regular expressions on the internet). For example, the lexicon portion of the Content Filter will search for exact words like “healthcare” or exact phrases like “primary physician”. The regex portion will search for patterns of numbers and other characters. For example, a regex Content Filter can be set up to search for 5 consecutive numbers surrounded by word boundaries (a zip code). Obviously this would be very difficult to do as a lexicon (specific word) type Content Filter.

Additionally, lexicon and regex Content Filters can be set up to only scan certain portions of a MIME message. You can include or exclude the headers, attachments, body, and various other sections for each word, phrase, or pattern your Content Filter looks for. For example, scanning the headers of a MIME message usually does not have a lot of real world value so excluding those headers can save scanning time as well as headaches from false positive matches.
WHAT IS A CONTENT TYPE IN THE MIME FORMAT?

One of the headers that is important to the Content Filter is the Content-Type header. This header describes the content type of the part associated with the header. In our very simple example message above there is a content type of text/plain. This tells us that the message is, as expected, a plain text document. Most common file types have pre-defined MIME content types associated with them. For instance, a .jpg image file attachment will have a Content-Type header of image/jpeg, HTML documents will have a Content-Type header of text/html, etc.

The multipart type is another Content-Type that allows a MIME message to be split up into multiple parts. This allows MIME messages to include attachments as well as alternative content sections. For example, a common MIME message (such as a text message with a .jpg file attached) would have a main part of content type multipart/mixed. This main part would have two child parts. The first would be of type text/plain (and would be labelled as the main body) and the second would be of type image/jpeg. Multipart parts can also have multipart type children which results in a MIME message having a multilevel tree-like structure.

Note: Even messages with no parts other than a single simple text/plain part as in the example above will still have a headers section. All proper MIME messages do.
SCANNING FOR CONTENT TYPES IN A MIME FORMATTED MESSAGE

The Content Filter can also execute scanning of MIME parts based on the Content Type of that part. This scanning overrides any other scanning that would be done by the lexicon or regex portions of the Content Filter on a particular Content Type. For example, a given Content Filter can be configured to give image type parts of a MIME message a blanket value of zero. A score of zero indicates that the content of an attachment of this type should be ignored. This is valuable since generally all ‘matches’ found within the binary information in an image are going to be false positives. The Content Filter sees the binary attachment as a sequence of numbers and letters rather than an image. The Content Filter will interpret the numbers as they are represented and may interpret the numbers as zip codes, phone numbers, or Social Security Numbers. Knowing that the attachment type is of a particular type, the scanning may be over-ridden and a score assigned to the entire attachment instead. This type of matching is very valuable for all Content Types a message may have that are not intelligently understood by the Content Filter.

Combining Lexicon, Regex, and Content Type Scanning to Form a Content Filter

To create a complex, fully functional Content Filter it is often necessary to combine two or even all three filtering types in a single Content Filter (lexicon, regex, and content type). The lexicon portion will search for words and phrases, the regex portion will search for patterns, and the content type section will search for MIME parts with particular content types. The scores from all three sections will get added together to form a total score for a message. The Content Filter will then match or not depending on whether or not the total score for the message meets the threshold value in the Content Filter configuration.
Creating Content Filter Files

This chapter contains the following section:

• “Content Filter Files” on page 50
CONTENT FILTER FILES

Three filter file types can be associated with a Content Filter. As expected, these match up with the three types of filtering the Content Filter can do: lexicon, regex, and content type filtering. For a Content Filter to be equipped with a lexicon filter it must have an associated lexicon filter file that includes all words and phrases the Content Filter will scan for, the value associated with each of the words and phrases, and, lastly, the parts of the message (subject, body, attachments, etc.) to scan for those words and phrases. This same general structure and login also applies to patterns in a regex filter file and, for the most part, content types in the content type filter file. The one exception is that the content type filter file is not constrained to searching inside of particular subsections of a MIME part; it searches the whole message for particular parts.

The rest of this chapter will go over the architecture of each filter file type in detail. To see how the Content Filter is actually configured to use filter files please refer back to Chapter 2 of this document, which goes through the steps of configuring a Content Filter to use the filter files that you create.

Lexicon Files

The Lexicon file contains the words or phrases that you want to search for, a score that will be applied each time an instance of that word or phrase is found and the parts of the message that you want to scan for these words and phrases. You must configure this file prior to performing content filtering. The following is a working example of a Content Filter lexicon file that also describes the file format via comments.

```plaintext
# $Id: samples.lex,v 1.6 2006/11/16 22:54:04 ehoward Exp $
# This sample lexicon file demonstrates all the features of a lexicon file.
# Lexicon files can contain comments. Anything on a line after a pound sign (hash, octothorpe) is ignored.
# Lexicon files can also contain blank lines, which are ignored:
# Starting and trailing whitespace is also ignored. The next line consists of just whitespace:
# Most lines in a lexicon file specify a match term, and consist of a word or phrase, an optional score, an optional part specifier, and an optional score number (score the match every time it is found or only once). The fields are separated by one or more tabs or two or more spaces. The simplest term is just a word. The word is searched for in all parts of a message and if found, adds the default score to the total.
hydrogen
```
# Comments can appear on the same line as a term.

boron     # Use the default score

# A term can also have an explicit score (see below for the various ways a # score can be specified).

helium     10

# A term can have an optional part specifier (the default is all)

lithium    all

# A term can also have a score number (the default is scoreAll)

beryllium  scoreOnce     # score only the first occurrence of this match
oxygen     scoreAll      # score all occurrences of this match

# The match term is required to be listed first, but the three optional # fields can occur in any order, and case doesn't matter

carbon     sCoreOnCe     10     all
nitrogen    10             ALL    scoreall

# The score can be fairly complex. Here are examples of all the forms of the # score and their final value, assuming the default score specified in the # configuration UI is 25.

neon       0              # 0     no affect on total score
sodium     20             # 20    a specific score, increases total # score on match
magnesium  +20            # 20    a specific score, increases total # score on match
aluminum   -20            # -20   a negative score, decreases total # score on match
silicon    -2147483648    # -2147483648 the smallest negative score
phosphorus 2147483647     # 2147483647 the largest positive score
sulfur     def            # 25    the default score
chlorine   Def            # 25    the default score, the score is # case-insensitive
argon      +def           # 25    the default score
potassium  -def           # -25   the negative of the default score
calcium    def + 10       # 35    a positive offset from the default # score
scandium   def - 10       # 15    a negative offset from the default # score
titanium   def * 2        # 50    a multiple of the default score
vanadium  def x 2  # 50  a multiple of the default score
chromium  def / 2  # 12  a divisor of the default score
            #    (integer only)
manganese +def + 30  # 55  a positive offset from the default score
            #    (integer only)
iron      -def + 30  # 5   a positive offset from the negative score
                      #    (integer only)
cobalt    inf            # positive infinity, the total score is
                      #    always greater than any maximum score set (or equal if the maximum score is inf)
nickel    INF            # positive infinity, the score is case-insensitive
copper   +inf           # positive infinity
zinc      -inf           # negative infinity, the total score is
                      #    always less than any maximum score set (or equal if the maximum score is inf)

# If this file is saved as UTF-8 or UTF-16, the Unicode infinity character
# (U+221E) may also be used for infinity, and Unicode mathematical operators
# may be used in place of ASCII mathematical operators.

# The part specifiers can also be complex. These terms show all the
# available part specifiers and how they may be combined. The comment shows
# the number the part specifier resolves to internally (for comparison of
# the examples only, the number isn't normally seen external to the content
# filter).

niobium   def     none           # 0     doesn't match anywhere
molybdenum def     headers        # 1
technetium def     body           # 2
ruthenium  def     signature      # 4
rhodium    def     attachments    # 8
palladium  def     subject        # 16
silver     def     message        # 30   subject, body, signature,
                      #    attachments
cadmium    def     all            # 31   headers, subject, body,
                      #    signature, attachments
indium     def                    # 15   default is all
tin        def     Headers        # 1   part specifiers are
                      #    case-insensitive
antinomy   def     headers body   # 3   headers and body
tellurium  def     headers, body   # 3   headers and body
iodine     def     headers + body # 3   headers and body
xenon      def     headers | body # 3   headers and body
cesium     def     headers & body # 3   headers and body
# A term can match a word or a phrase. A phrase is multiple words separated
# by a single space. The longest possible match is always used and any
# matches to the suffix of a matched phrase are not used. For example, given
# the following terms:

tungsten 10
tungsten rhenium 20
rhenium 30

# If the phrase "tungsten rhenium" appears anywhere in a message, it matches
# the second term only, adding 20 to the total score.

# Phrases may be arbitrarily long, here's one with six words.

plutonium curium einsteinium fermium mendelevium nobelium

# ContentFilter backtracks when matching phrases. For example, given the
# following terms:

lawrencium rutherfordium dubnium 10
rutherfordium seaborgium 20

# "lawrencium rutherfordium seaborgium" will match the second term for a
# score of 20. Without backtracking, by the time ContentFilter sees
# "seaborgium" in the input, it's too late to match "rutherfordium
# seaborgium".

# When writing a CSV file, the name of the match group (the lexicon or list
# of regular expressions) containing a matching word or phrase is also
# written. A match group name is normally just the basename of the file (the
# file name without any suffix, this lexicon's name is "samples"). The name
# can be changed using the +group directive:

+group example

# The group name is separated from the +group directive by a space. Using
# +group allows multiple lexicons to be placed in a single file, but appear
# separate in the CSV file. The name of a match group has no effect on the
# actual filtering process.
Regular Expression Files

The regular expression file is very similar to the lexicon file except that a regular expression compliant to the Perl 5.0.0.3 standard should be substituted for the word or phrase used in the lexicon file. You must set this file up prior to performing regular expression content filtering.

The following is a working example of a Content Filter regex file that also describes the Content Filter regex filter file format via comments.

```
# $Id: samples.res,v 1.2 2004/01/15 01:32:39 cvista Exp $
# This sample regular expression file demonstrates regular expressions. The
# regular expression file is the same as the lexicon file, except instead of
# having a simple word or phrase as the first field in a term, the first
# field is a Perl 5.003 compatible regular expression. Thus this file just
# demonstrates a few regular expressions, not the full complement of score
# and part specifiers. See sample.lex for that.
#
\d\d\d-?\d\d-?\d\d\d# A Social Security Number
\d\d\d[ -] [A-Z][A-Z][A-Z]# A Washington state car license plate
\d[A-Z][A-Z][A-Z]\d\d\d# A California car license plate
\\b(\\w+)\\s+(\\1)+\\b# A sequence of two or more repeated words
\\b(M\{0,3\}|D?C\{0,3\}|X[LC]|L?X\{0,3\}|I[XV]|V?I\{0,3\})\\b
# All Roman Numerals from 1 to 3999. Larger
# numbers could be matched using U+2181 and # U+2182.
```

Content-Type Files

The content-type file contains the types of content you want to search for as well as a score that will be applied each time an instance of that content type is found in the parts of the message being scanned. You must set this file up prior to performing content type filtering.

```
# This sample content-type file demonstrates all the features of a content-type
# file. The content-type file closely follows both the lexicon and
# regular expression file formats. The major difference is that the content-type
# file has a Content-Type as the first field in a term. Additionally
# there
# should not be a part specifier as part of a content-type filter file term.
#
# --- MIME Message structure ---
#
# In order to understand how content-type filtering works it is essential to
```
# first understand how email messages are constructed. Email messages follow # the MIME format and are constructed of at least one MIME 'part'. Each part # represents a piece of content (email body, attachment, alternative content, # etc.).

# Each part also has a Content-Type header that describes the content # contained # in the particular part. The Content-Type header is divided by a '/' into a # 'type' and 'subtype'. For example, in the Content-Type 'text/html', 'text' # is # the type while 'html' is the subtype.

# There are also part types that do not themselves represent content but # rather # allow the message to include more complex arrangements of content. These # are the # 'multipart' types. The most common of these is 'multipart/mixed' which can # have an arbitrary number of 'child parts', each of which can contain # content. In more complex MIME messages these child parts can also be of a # multipart type as well.

# The result is tree-like structure for complex MIME messages illustrated # below.

```
# multipart/mixed (1)
#                   ________|________________
#                   |                        |
#       multipart/alternative (2)     image/jpeg (5)
#        ________|________
#       |                |
#    text/plain (3)   text/html (4)
```

# Here the root part (1) has a content type of multipart/mixed allowing it to # have multiple, arbitrary-typed, children. Part 5 is an image/jpeg type (a # .jpeg file) and will likely be displayed in the message as an attachment. # Parts 2, 3, and 4 make up the main body of the message (the main content # displayed when the message is read). The multipart/alternative type # indicates # that all of its child types should be different representations of the same # data. An email client will attempt to display the more complex data, in # this # case the text/html part (4), first. If the client is unable to display the # more complex type it will revert to displaying the simple text/plain part (3).

# --- Content-Type terms --- #
A content type term is made up of at least a content-type hierarchy. A content-type hierarchy consists of one or more content types linked together with the ':' character to form a chain (hierarchy). The simplest form of this has only a single content type as below.

text/plain

A content-type hierarchy traces the content-types from root to leaf of the scanned message. For example, the content-type hierarchy of part (4) above is multipart/mixed followed by multipart/alternative, finally followed by text/html. In the content-type filter file it would be described as follows.

multipart/mixed:multipart/alternative:text/html100

As shown above, you can also add a score value to the term (as in the lexicon and regex filter files). This score overrides the default score configured in the content filter setup.

Note that content type matching does not apply to part specifiers (subject, headers, body, etc.). A part specifier is actually a subsection of a message part. If you are matching the entire part it does not make sense to specify subparts to match within.

--- Wildcards ---

There are two different wildcard symbols that can be used in content type terms, the '*' and the '?/?'. The '*' symbol can replace the content type and/or subtype of any given entry in a term. The '*' can take any value when matching.

multipart/*:text/plain

This term would match a text/plain part with a parent of multipart/mixed as well as multipart/alternative (or any other subtype of the multipart type) and give it the default value.

*/*:text/plain-inf

Here it makes no difference what type parent the text/plain part has as long as that parent is the root of the message (has no parent of its own). This *would not* match the message described above as the text/plain part there has a parent as well as a grandparent.
The '?\/' wildcard can only be used as both type *and* subtype of a part (i.e. must look like '?/\?', multipart/\? will not work). The '?/\?' wildcard matches similarly to the '*\*' wildcard except that instead of matching exactly one part type/subtype, it matches zero or more.

?/?:text/plain

Everything that would be matched by */*:text/plain is also matched by this term, but many others will match as well. This term would also match a text/plain part with no parent (a very simple message indeed) as well as one with an arbitrary number of ancestors in its hierarchy. For example, this content-type hierarchy *would* match part (3) of the message described above.

This wildcard is extremely valuable when you want to match all instances of a certain part (say, binary image data) regardless of how far down a part hierarchy it is.

?/?:image/*0

This term would capture any image type parts (regardless of subtype) and ignore it (give the entire part a value of zero). This prevents the lexicon and regex portions of the content filter from accidentally matching meaningless strings in binary data. On the other hand, if you know that the images being sent through your content filter often contain private information, you could give the above term a value of 'inf' to ensure that any message with an image attachment gets matched.

--- Matching ---

When a content-type in a message is matched by a content-type hierarchy in a content filter it means that, for that part, the lexicon and regex portions of the content filter will be overridden. No matching will be done on the content of that part. The only exception to this is that MIME 'header' data for the part will be matched if the lexicon or regex filters are configured to do so.
Applying the Knowledge

In the previous six chapters we have gone through various exercises designed to help you become familiar with setting up individual Content Filters as well as full fledged Content Filter systems. You should now know how to create Content Filter tests, rules, and rulesets, and also have an idea of how to combine them together to create complex logical flows in your Content Filter system. All of the building blocks are there. This chapter will discuss methods for dynamically combining those building blocks into a content filter system that works for your environment.

This chapter contains the following sections:

- “Initial Design” on page 60
- “Final Tuning” on page 63
INITIAL DESIGN

There are several critical questions that need answering before you start designing your Content Filter system. You may have additional criteria that you need to consider but generally you should ask the following questions when conceptualizing your Content Filter. We will refer back to the examples from Section 1 (where we designed a HIPAA compliant Content Filter system) to help in understanding how to answer these questions.

What types of information does the Content Filter system need to find?

In the example in Section 1 we found that HIPAA regulations require the protection of any information that:

1. Identifies an individual, and
2. Relates to:
   - the individual’s past, present, or future physical or mental health or condition;
   - the provision of health care to the individual; or
   - the past, present, or future payment for the provision of health care to the individual.

For different types of organizations (for example, law firms or banking institutions) the types of information that the Content Filter system needs to search for will be somewhat different. Identify the types of information that your Content Filter system will need to scan for.

Even within organization types there is no one perfect list of content and content types that meet everyone’s requirements. The sample lexicon and regex files provided with the IronPort Encryption appliance were designed to include most of the sensitive terms and patterns relevant to a HIPAA or Privacy Content Filter system. However, you will almost certainly need to add additional sensitive terms and patterns specific to your particular environment. Once added, your Content Filter system will provide more accurate HIPAA or Privacy coverage for your organization.

Fortunately it is very easy to add terms to the lexicon, regex, and content-type files. Skip back to Chapter 6 for a thorough overview of the structure of the files.

What relationships exist between the different types of information that the Content Filter system will look for?

The HIPAA regulations reveal some more obvious relationships between information types. Specifically, it calls for personally identifiable information and medical information relevant to that person. More subtle interactions between the pieces of HIPAA data were also discussed earlier. For example, a nine digit number without any associated identifier (“SSN, SSN#, social security number, etc.”) is not likely to be HIPAA relevant information. Look carefully at the different information types that you are looking for in order to find these more subtle relationships. They can help make your Content Filter system much more intelligent and accurate.
What are false negatives and false positives and why are they important?

False Negatives are messages that contain sensitive information but aren’t identified for encryption by the Content Filter system scanning them. This is obviously a major concern as the sensitive message leaves the organization unprotected. Most organizations set a goal of zero false negatives. The easiest way to achieve this goal is to encrypt everything. Although the Secure Envelope recipient’s decryption experience is among the simplest and most user friendly available, it is not completely transparent. As a result, many customers strive to encrypt only messages containing sensitive content. This is where control of False Positives, or messages that don’t contain truly sensitive content but are identified by the Content Filter system for encryption, becomes an issue. When planning your Content Filter system look at the kinds of data you will be working with and try to identify where false positives might come from. Additionally, once the Content Filter system has been set up, continue to use the CSV reporting files to isolate and eliminate scenarios that result in false positives. This may mean adding more complex logic and more Content Filter tests/rules/rulesets to your Content Filter system.

What kinds of data should be ignored by the Content Filter system?

The ideal answer to this question is probably obvious – you should ignore any data that isn’t sensitive. Implementing a system that accomplishes this goal requires several or more cycles of testing and tuning the underlying Content Filter logic and filter files. An example of a tuning decision that might ignore data is how you choose to treat image and binary files. Both types of files could contain sensitive information, but neither can be fully scanned by a Content Filter test. There are a number of ways to resolve this concern. You could use content-type matching to ignore messages with attached images and/or binaries or use the same method to assign high enough Content Filter values to guarantee encryption or create a policy that blocks delivery of these messages and notifies the sender of the policy violation. Another type of data that might be ignored is personal information in signature blocks. Signature blocks typically include personal information like phone numbers and email addresses, often including tags like Phone #: and Email Addr:. While this information appears to be “personally identifying information” to the Content Filter system, it identifies the sender instead of the recipient and as a result is not protected under HIPAA guidelines. These are just examples of the kinds of data you may want to ignore when tuning your Content Filter system. Use the content-type portion of your Content Filter to identify and operate on content-types you wish treat differently than the rest of the message.

Do I need to spend the time and effort identifying these organization-specific relationships?

It is not necessarily to start with a more complex Content Filter configuration. Many organizations choose to encrypt messages if there is any suspicion that the content is sensitive. If you do not mind some level of false positives, a simple single-rule content filter may fit your needs perfectly. For organizations that try to minimize both false positives and negatives, it is likely that you will find a chained Content Filter system like the one described in Chapter 4 to best suit your needs.
How can I utilize these relationships and my organization-specific information and requirements to create a more intelligent Content Filter system?

You should include your organization-specific information in the lexicon and regex files whether or not you choose to implement a more sophisticated Content Filter system. If you do choose to try to minimized false positives as well as negatives, see Chapters 3 and 4 for instructions on modifying the default Content Filter configuration or creating a new configuration to meet your unique requirements.

How should I implement my new Content Filter system?

Chapters 3 and 4 illustrated how information relationship types can be exploited to make a Content Filter system more intelligent. Achieving a good balance that minimizes both false positives and negatives requires up-front planning and ongoing tuning of the lexicon and regex file content. Once the sensitive information is identified, make it the centerpiece of a comprehensive compliance plan. The first and most critical step is to make everyone in your organization aware of the regulations you must legally follow, as well as the content that must be protected to meet your responsibilities. It is much easier to train people not to send sensitive information than it is to catch all sent info. Often a good next step is to run the content filter in “audit” mode for some period of time, notifying the sender, a regulatory administrator or both when a sensitive email is sent. While in audit mode you will not encrypt the messages – only indicate that they met the Content Filter criteria.
FINAL TUNING

With even the most diligent care the first attempt at creating a new Content Filter system will likely not yield a perfect result. A tuning process is generally needed to get the Content Filter into a production-ready state. The most valuable tool for tuning the Content Filter is the CSV reporting system described first in Chapter 2 of this document. A more detailed description of the fields displayed in the CSV files is found in Appendix B: Content Filter CSV Report File Structures. The two CSV files produced by each Content Filter test are the summary and details file. The summary file provides an overview of the scan of each message, showing the message ID, message header information like To: and From: addresses, the score of the message and whether the test matched or not. The details file includes an entry for each term, pattern or content-type that matched in the message, with entries for the message ID, the term, pattern, etc. that matched and the score assigned for that match.

Using the CSV files it is possible to identify exactly which matches contributed to the per-message score. It can be very valuable to send a known corpus of email through the Content Filter system, using the CSV files to identify the matches and comparing the results against expectations. Using this approach it is possible to eliminate terms or patterns that don’t apply to your organization, add specific terms that are missing or alter the per-instance score of specific items in the filter files. The process of fine tuning doesn’t happen overnight. It frequently takes weeks to settle on the best set of terms, patterns and content types and to adjust the associated scores. Fortunately this isn’t a particularly time-intensive activity, with several days of message flow typically followed by an hour or two of analyzing CSV output and altering filter files of Content Filter test configuration.
Content Filter Configuration Parameters

Chapter 2 explains how to add a basic Content Filter test to your IronPort Encryption appliance configuration. This appendix goes into more detail describing what each parameter in the Content Filter configuration does.

Name – Name of this specific content filter. This field is not important for the operation of the Content Filter and is only meant to help distinguish one Content Filter from another in the Administration Console of the IronPort Encryption appliance.

Lexicon File – Path to the file containing the list of terms used for lexical content filtering.

Lexicon File Encoding – Specifies the file encoding type for the lexicon file.

Regular Expression File – Path to the file containing the list of terms used for regex content filtering.

Regular Expression File Encoding – Specifies the file encoding type for the regular expressions file

Content Type File – Path to the file containing the list of terms used for content-type content filtering.

Content Type File Encoding – Specifies the file encoding type for the content type file.

Default Score – Score value used for words in lexicon, regex, and content-type files that do not have Content Filter scores associated with them

Maximum Score – Per-message score threshold for the Content Filter to consider the message a match

Encrypted Score – Score to add when encrypted PDF, jar entry or zip entry is found. Since the Content Filter cannot look inside of these types of attachments this generic will be associated with them.

Damaged Score – Score to add when damaged encrypted PDF, jar entry or zip entry is found. Since the Content Filter cannot look inside of damaged archive attachments this generic value will be associated with them.

Archive File Recursion Depth – If an archive type attachment has any archives within it, how many levels deep should the Content Filter scan these embedded archive files. Use a value of zero for no limit.
Add Score Header – If checked, the final score is added to the message as a MIME X header. The name will be X-PostX-CF-Score and the value will be the Content Filter score of the message. That enables the system to perform further processing later based upon the score of the message.

Stop When Maximum Score Reached – Specifies whether the Content Filter stops processing and considers it a match when the maximum score is reached. This is a “short circuiting” mechanism. As mentioned earlier, the Content Filter scans the message through one MIME part at a time. After the message has reached the maximum score, the system will stop scanning the message since the threshold has been reached. Although this saves processing time, it is important to note that the message could have ended up with a lower score if negative values are being used.

Multipart/alternative Score is Sum of Parts' Scores – If checked, score multipart/alternative parts as the sum of the scores of their parts, instead of the maximum. Note: multipart/alternative parts are supposed to have some number of child parts all representing the same information. If left unchecked, the Content Filter will only take the highest scoring child part into account when scoring the message. This is useful as all subparts of a Multipart/Alternative MIME part should share the same general content. It is often unnecessary to count the score of that content for every subpart.

CSV Directory – If set, a pair of Comma Separated Value files containing match information is written in this directory. These files contain comprehensive logging information on the results of the content filtering done on all messages coming through the content filter. This should be a path to a directory located on the same file system as the encryption server instance.

Write Domain Names Only – If checked, just the domain name portions of email addresses are written to the CVS files.

Log File Reading – Specifies whether to log to the IronPort Encryption appliance log files when it reads the lexicon file, regular expression file, or content type file.

Log File Contents – Specifies whether to log to the IronPort Encryption appliance log files the entries in the lexicon file or regular expression file when the files are read.

Log Total Score – Specifies whether to log to the IronPort Encryption appliance log files the total score of each email.

Log Individual Scores – Specifies whether to log to the IronPort Encryption appliance log files the individual scores of each match within each email.

Log Nonmatches – Specifies whether to log to the IronPort Encryption appliance log files the nonmatches in each email.

Use Bitform – Specifies whether Microsoft Office attachments will be processed by using Bitform to extract the text. This is recommended.
Content Filter CSV Report File Structures

This appendix contains the following sections:

• “Content Filter Files” on page 68
• “Lexicon Files” on page 68
• “Regular Expression Files” on page 72
• “Content-Type Files” on page 73
CONTENT FILTER FILES

Three filter file types can be associated with a Content Filter. As expected, these match up with the three types of filtering the Content Filter can do: lexicon, regex, and content type filtering. For a Content Filter to be equipped with a lexicon filter it must have an associated lexicon filter file that includes all words and phrases the Content Filter will scan for, the value associated with each of the words and phrases, and, lastly, the parts of the message (subject, body, attachments, etc.) to scan for those words and phrases. This same general structure and login also applies to patterns in a regex file and, for the most part, content types in the content type filter file. The one exception is that the content type filter file is not constrained to searching inside of particular subsections of a MIME part; it searches the whole message for particular parts.

The rest of this chapter will go over the architecture of each filter file type in detail. To see how the Content Filter is actually configured to use filter files please refer back to Chapter 2 of this document, which goes through the steps of configuring a Content Filter to use the filter files that you create.

Lexicon Files

The Lexicon file contains the words or phrases that you want to search for, a score that will be applied each time an instance of that word or phrase is found and the parts of the message that you want to scan for these words and phrases. You must configure this file prior to performing content filtering. The following is a working example of a Content Filter lexicon file that also describes the file format via comments.

```plaintext
#
# This sample lexicon file demonstrates all the features of a lexicon file.
# Lexicon files can contain comments. Anything on a line after a pound sign
# (hash, octothorpe) is ignored.
# Lexicon files can also contain blank lines, which are ignored:
#
# Starting and trailing whitespace is also ignored. The next line consists
# of just whitespace:

# Most lines in a lexicon file specify a match term, and consist of a word
# or phrase, an optional score, an optional part specifier, an optional
# score number (score the match every time it is found or only once), and an
# optional edit flag. The fields are separated by one or more tabs or two or
# more spaces. The simplest term is just a word. The word is searched for in
# all parts of a message and if found, adds the default score to the total.

hydrogen
```
# Comments can appear on the same line as a term.

boron    # Use the default score

# A term can also have an explicit score (see below for the various ways a # score can be specified).

helium  10

# A term can have an optional part specifier (the default is all)

lithium   all

# A term can also have a score number (the default is scoreAll)

beryllium  scoreOnce    # Score only the first occurrence of this match
oxygen    scoreAll     # Score all occurrences of this match

# The match term is required to be listed first, but the four optional # fields can occur in any order, and case doesn't matter

carbon    sCoreOnCe  10    all    Edit
nitrogen    edit       10    ALL    scoreall

# The score can be fairly complex. Here are examples of all the forms of the # score and their final values, assuming the default score specified in the # configuration UI is 25.

neon       0        # 0    no affect on total score
sodium    20        # 20    a specific score, increases total #    score on match
magnesium +20        # 20    a specific score, increases total #    score on match
aluminum  -20        # -20    a negative score, decreases total #    score on match
silicon   -2147483648    # -2147483648    the smallest negative score
phosphorus  2147483647    # 2147483647    the largest positive score
sulfur    def       # 25    the default score
chlorine    Def     # 25    the default score, the score is #    case-insensitive
argon    +def       # 25    the default score
potassium  -def      # -25    the negative of the default score
calcium    def + 10   # 35    a positive offset from the default #    score
scandium    def - 10   # 15    a negative offset from the default #    score
titanium  def * 2        # 50    a multiple of the default score
vanadium  def x 2        # 50    a multiple of the default score
chromium def / 2        # 12    a divisor of the default score
                    #    (integer only)
manganese +def + 30    # 55    a positive offset from the default
                    #    score
iron -def + 30         # 5     a positive offset from the negative
                    #    score of the default score
cobalt inf            # positive infinity, the total score is
                    #    always greater than any maximum
                    #    score set (or equal if the maximum
                    #    score is inf)
nickel INF             # positive infinity, the score is case-
                    #    insensitive
copper +inf            # positive infinity
zinc -inf              # negative infinity, the total score is
                    #    always less than any maximum score
                    #    set (or equal if the maximum score
                    #    is inf)

# If this file is saved as UTF-8 or UTF-16, the Unicode infinity character
# (U+221E) may also be used for infinity, and Unicode mathematical operators
# may be used in place of ASCII mathematical operators.

# The part specifiers can also be complex. These terms show all the
# available part specifiers and how they may be combined. The comment shows
# the number the part specifier resolves to internally (for comparison of
# the examples only, the number isn't normally seen external to the content
# filter).

niobium def none        # 0     doesn't match anywhere
molybdenum def headers  # 1
technetium def body     # 2
ruthenium def signature # 4
rhodium def attachments # 8
palladium def subject   # 16
silver def message      # 30    subject, body, signature,
                    #    attachments
cadmium def all          # 31   headers, subject, body,
                    #    signature, attachments
indium def               # 31    default is all
tin def Headers          # 1     part specifiers are
                    #    case-insensitive
antinomy def headers body # 3     headers and body
tellurium def headers, body # 3    headers and body
iodine def headers + body # 3     headers and body
xenon def headers | body  # 3     headers and body
cesium def headers & body # 3     headers and body
barium def message - attachments  # 22 subject, body, signature
lanthanum def all - signature  # 27 headers, subject, body,
                 # attachments

# If editing is turned on in the configuration, lines with edit on them will
# replace the matched term with an X per letter (or number or underscore).

cerium edit

# Each time cerium is encountered in the subject, body, or signature it will
# be replaced with XXXXXX. Editing is independent of score number, so with:

praseodymium edit  scoreonce

# praseodymium will be replaced with XXXXXXXXXXXX every time it occurs, but
# will only contribute the default score once to the total score.

# A term can match a word or a phrase. A phrase is multiple words separated
# by a single space. The longest possible match is always used and any
# matches to the suffix of a matched phrase are not used. For example, given
# the following terms:

tungsten  10
tungsten rhenium  20
rhenium     30

# If the phrase "tungsten rhenium" appears anywhere in a message, it matches
# the second term only, adding 20 to the total score.

# Phrases may be arbitrarily long, here's one with six words.

plutonium curium einsteinium fermium mendelevium nobelium

# Content Filter backtracks when matching phrases. For example, given the
# following terms:

lawrencium rutherfordium dubnium 10
rutherfordium seaborgium        20

# "lawrencium rutherfordium seaborgium" will match the second term for a
# score of 20. Without backtracking, by the time Content Filter sees
# "seaborgium" in the input, it's too late to match "rutherfordium
# seaborgium".

# When writing a CSV file, the name of the match group (the lexicon or list
# of regular expressions) containing a matching word or phrase is also
Regular Expression Files

The regular expression file is very similar to the lexicon file except that a regular expression compliant to the Perl 5.003 standard should be substituted for the word or phrase used in the lexicon file. You must set this file up prior to performing regular expression content filtering.

The following is a working example of a Content Filter regex file that also describes the Content Filter regex filter file format via comments.
Content-Type Files

The content-type file contains the types of content you want to search for as well as a score that will be applied each time an instance of that content type is found in the parts of the message being scanned. You must set this file up prior to performing content type filtering.

--- MIME Message structure ---

In order to understand how content-type filtering works it is essential to first understand how email messages are constructed. Email messages follow the MIME format and are constructed of at least one MIME 'part'. Each part represents a piece of content (email body, attachment, alternative content, etc.).

Each part also has a Content-Type header that describes the content contained in the particular part. The Content-Type header is divided by a '/' into a 'type' and 'subtype'. For example, in the Content-Type 'text/html', 'text' is the type while 'html' is the subtype.

There are also part types that do not themselves represent content but rather allow for more complex arrangements of content to be achieved. These are the 'multipart' types. The most common of these is 'multipart/mixed' which can have an arbitrary number of 'child parts' that each themselves can contain content. In more complex MIME messages these child parts can also be of a multipart type as well.

The result is a tree-like structure for complex MIME messages, an example of which is illustrated here.
Here the root part (1) has a content type of multipart/mixed, allowing it to have multiple, arbitrarily typed, children. Part 6 is an image/jpeg type (a JPEG file) and will likely be displayed as an attachment to the message. Parts 2-5 make up the main body of the message (the main content displayed when the message is read). The multipart/alternative type indicates that all of its children parts should be different representations of the same data. An email client will display the most complex type it recognizes, in many cases the text/html part (4). If the client is unable to display the more complex type it will revert to displaying the simple text/plain part (3).

--- Content-Type terms ---

A content type term is made up of at least a content-type hierarchy. A content-type hierarchy consists of one or more content types linked together with the ':' character to form a chain (hierarchy). The simplest form of this has only a single content type as below.

text/rtf

A content-type hierarchy traces the content-types from the root to a leaf of a message. For example, the content-type hierarchy of part (5) above is multipart/mixed followed by multipart/alternative, finally followed by text/rtf. In the content-type filter file it would be described as follows.

multipart/mixed:multipart/alternative:text/rtf 100

As shown, you can also add a score value to the term. The score has the same form and same range of possible values as in the lexicon and regex files (see samples.lex for the forms the score can take). This score overrides the default score configured in the content filter setup.

Note that part specifiers (subject, headers, body, etc.) and the edit specifier are not applicable to content type matching. A part specifier is actually a subsection of a message part. If you are matching the entire part it does not make sense to specify subparts to match within.

--- Wildcards ---
# There are two different wildcard symbols that can be used in content type terms, '>*</p>
# and '?/?'. '*' symbol can replace the content type and/or subtype of any given entry in a term. The '*' matches any value when # matching.

multipart/*/text/rtf

# This term would match a text/rtf part with a parent of multipart/mixed as well as multipart/alternative (or any other subtype of the multipart type) and give it the default score.

*/text/rtf  -inf

# Here it makes no different what type parent the text/rtf part has as long as that parent is the root of the message (has no parent of its own). This *would not* match the message illustrated above as the text/rtf part has a parent as well as a grandparent.

?/?:*#text/rtf

# The '?' wildcard can only be used as both type *and* subtype of a part (i.e. must look like '?/?', multipart/? will not work). The '?/?' wildcard matches similarly to the '*' wildcard except that instead of matching exactly one part type/subtype, it matches zero or more.

?/?:text/rtf

# Everything that would be matched by */*:text/plain is also matched by this term, but many others will match as well. This term also matches a text/plain part with no parent (a very simple message indeed) as well as one with an arbitrary number of ancestors in its hierarchy. For example, this content-type hierarchy *would* match part (5) of the message described above.

?/?:text/plain

# This wildcard is extremely valuable when you want to match all instances of a certain part (say, binary image data) regardless of how far down a part hierarchy it is.

?/?:image/*  0

# This term matches any image type parts (regardless of subtype) and ignores them (by giving the entire part a score of zero). This prevents the lexicon and regex portions of the content filter from accidentally matching meaningless strings in binary data. On the other hand, if you know that the images being sent through your content filter often contain private information, you could give the above term a value of 'inf' to ensure that any message with an image attachment gets matched.
# --- Matching --- #

# When a content-type in a message is matched by a content-type hierarchy in
# a content filter it means that, for that part, the lexicon and regex
# portions of the content filter will be overridden. No matching will be
# done on the content of that part. The only exception to this is that MIME
# 'header' data for the part will be matched if the lexicon or regex filters
# are configured to do so.
Index

A
Add Score Header 66
AND 30, 38
Archive File Recursion Depth 65

C
cfdetails.csv 12
cfsummary.csv 12
Chaining 35
combined Content Filter system
testing 22
combined types 47
Compose a Text Message 22
Content Filter Chaining 38
Content Filter logic
tuning 61
Content Filter system
implementing 62
Content Filter Test
adding 6
content filtering
definitions 2
how does it work 3
steps 3
Content Filtering System 6
content filters
combining multiple filters 19
Content Type 47
Content Type File 65
Content Type File Encoding 65
content type filtering 50, 68
content-type file 54, 73
content-types 3
CSV Directory 66
CSV file contents 12, 24
CSV output files 14

D
Damaged Score 65
data
types to ignore 61
Default Score 65
Detail File 12

E
e-mail messages
building blocks 44
Encrypted Score 65

F
false negatives 61
false positives 61
filter files
tuning 61
final tuning 63

G
GLBA 4

H
HIPAA 2
requirements 6
HIPAA guidelines 16
HIPAA Privacy Terms Content Filter 25
HIPAA regulations 60
HIPAA_Med directory 24
HIPAA_Priv 24, 25

I
initial design 60
IronPort Encryption appliance 35

L
Lexicon 47
lexicon 50, 68
Lexicon File 65
Lexicon file 50, 68
Lexicon File Encoding 65
limitations 4
Log File Contents 66
Log File Reading 66
Log Individual Scores 66
Log Nonmatches 66
Log Total Score 66

M
Maximum Score 65
MIME (Multipurpose Internet Mail Extensions) 44
MIME Format 46
about 44
mime message
eample 44
MIME part 44
multipart 46
Multipart/alternative Score is Sum of Parts' Scores 66

N
Name 65

O
OR 30, 38
overview 1

P
Privacy Content Filter system 60
Privacy CSV Summary 25
privacy statutes 4
protected health information (PHI) 6

R
Regex 47
regex 50, 68
Regular Expression File 65
regular expression file 54, 72
Regular Expression File Encoding 65
rules 31
RuleSet chaining 31
RuleSets 30

S
Scanning
words, phrases, patterns in MIME message 45
Scoring 3
sending a text message via the content filter 11
Stop When Maximum Score Reached 66
Summary File 12

test messages
sending 22
tests 36

U
Use Bitform 66

W
Write Domain Names Only 66

Z
Privacy CSV Details 26