



realtimepublishers.com[™]

The Definitive Guide[™] To

Controlling Malware, Spyware, Phishing, and Spam

McAfee[®]
Proven Security[™]

Dan Sullivan

Chapter 7 Technologies for Securing Information and IT Assets	129
Content-Specific Technologies for Information Security	129
Email Content Filtering	130
The Evolution of Email Functionality and Vulnerabilities	130
Harassment and Hostile Workplace Issues	132
Information Leakage and the Loss of Intellectual Property	133
Disclosure of Private Information	134
Email Content-Filtering Techniques	135
Multiple Techniques for Email Content Filtering	144
Filtering Spam	144
Policies and Actions for Filtered Content	145
Web Browsing and URL Blocking	145
Multi-Layered Security: Defense in Depth	147
Gateway Defenses	147
Network Security Measures	147
Desktop Security Measures	148
Summary	148

Copyright Statement

© 2006 Realtimepublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimepublishers.com, Inc. (the “Materials”) and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtimepublishers.com, Inc or its web site sponsors. In no event shall Realtimepublishers.com, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtimepublishers.com and the Realtimepublishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtimepublishers.com, please contact us via e-mail at info@realtimepublishers.com.

[**Editor's Note:** This eBook was downloaded from Realtime Nexus—The Digital Library. All leading technology guides from Realtimepublishers can be found at <http://nexus.realtimepublishers.com>.]

Chapter 7 Technologies for Securing Information and IT Assets

Throughout, this guide has have examined threats to an organization's ability to protect the integrity and confidentiality of its information. Some of the most troubling threats today include:

- Viruses, worms, and other forms of malware
- Spyware that monitors, gathers, and steals information about users
- Phishing scams and identity theft
- Spam that taxes network and email service resources
- Employee behavior with IT resources that violates regulations and company policies

Techniques and technologies for controlling these threats is the focus of this chapter. The chapter begins with a focus on content-specific measures for mitigating the impact of these threats. The chapter concludes with a discussion of how a multilayered defense strategy can effectively provide adequate levels of security for an organization while maintaining necessary levels of usability and performance.

Content-Specific Technologies for Information Security

Content-specific technologies for information security address the specific issues that arise when preventing external threats from inflicting damage using network-based communication services while addressing the threats from inside the organization. Three representative technologies are considered:

- Email content filtering
- Email spam filtering
- URL blocking

Email content filtering uses a variety of techniques to identify malicious or inappropriate content and prevent it from entering or leaving an organization's network. Email spam filtering is a specialized form of email content filtering that uses additional techniques specific to the practice of spamming. The final technology, URL blocking, is a relatively simple but highly useful technique to prevent access to Web sites and resources that are known to contain problematic content.

Email Content Filtering

Email content filtering has emerged as a necessary element of information security because of a number of inappropriate uses of email. Sometimes the inappropriate use originates from the outside, but such is not always the case. Inappropriate uses of email systems by employees, contractors, and other insiders have also contributed to the necessity of filtering email.

Four commonly recognized threats can be addressed with the use of email filtering:

- Malicious programs transmitted through email messages
- Offensive material sent through email, creating a hostile work environment
- Loss of intellectual property
- Disclosure of private information

These problems are not solely limited to email services, for example the problem of a hostile work place existed long before email. Email, however, has changed the ways in which these threats are realized and the speed and breadth with which they can damage an organization.

The Evolution of Email Functionality and Vulnerabilities

Email is a channel for moving staggering amounts of information in and out of an organization. In the early days of email adoption, few controls were needed. Users were primarily academics, researchers, students and government contractors and employees working with the early Internet or its predecessor, the Arpanet. Then email evolved, both technically and with respect to its user base.

On the technical side, advances such as the Multipurpose Internet Mail Extensions (MIME) standard allowed users to attach documents to their text messages. Users were no longer limited to sending basic text messages; documents, images, and executable files could be sent as well. Email clients also added functionality, supporting address books with email addresses of other users, supporting macros to automate routine tasks, and offering improved interface functions, such as the ability to display HTML.

Unfortunately, as is often the case with software, greater functionality leads to more vulnerabilities (see Figure 7.1). Hackers and malware developers have been able to exploit these vulnerabilities, transmitting viruses over email, which, from the virus writers' perspective, must have been a great improvement over transferring the virus by floppy disk, which was the old transmission method. Sometimes email functionality was combined with vulnerabilities in desktop applications, such as Microsoft Word, to create viruses such as the Melissa and I Love You viruses.

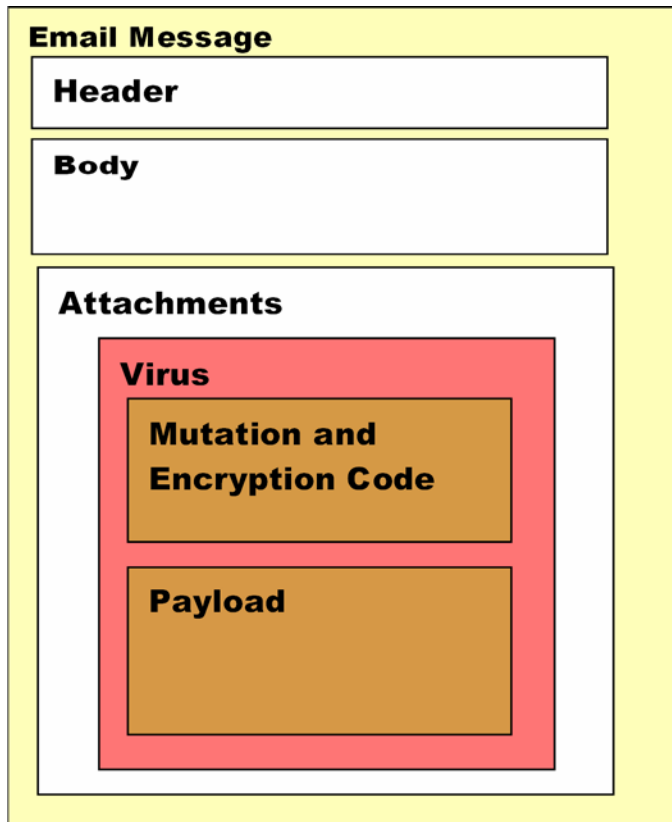


Figure 7.1: Improvements in email functionality, such as attachments, have become mechanisms for propagating malicious software.

For more information about these viruses, see the McAfee Virus Information Library. Details of the Melissa virus are available at http://vil.nai.com/vil/content/v_10132.htm; for details about the I Love You virus, see http://vil.nai.com/vil/content/v_98617.htm.

Today, the simple fact is that email content cannot be trusted. This reality is unfortunate because the vast majority of users have no intention of inflicting harm by propagating malware or overwhelming email servers with spam. At the same time, the evolution of email to this point is not surprising. The potential for malicious behavior in cyberspace mirrors the same potential in other areas. Brick-and-mortar stores deploy anti-theft devices (euphemistically called “inventory control devices”) at exits to deter shoplifting; airports employ complex screening procedures to reduce risks to travelers. As email cannot be trusted, it must be examined and, if found to be malicious or problematic in some way, dealt with according to a well-defined procedure.

Common Email-Based Threats

Common forms of malicious software that can be transmitted through email include:

- Viruses—Malicious programs that propagate by using other programs. Some viruses purposefully change characteristics with each generation to avoid detection by antivirus software; antivirus systems, however, have changed detection techniques to counter those mutating schemes. Viruses may also encrypt themselves to avoid detection. This practice leaves them vulnerable to detection by looking for decryption routines with the body of the virus unless the decryption code is changed using a mutating technique.
- Worms—Malicious programs that propagate on their own. These programs typically take advantage of a vulnerability in an application, such as an email client. Mutating and encryption techniques used in viruses may also be used with worms.
- Trojan horses—These programs appear to do one thing, such as add a toolbar to a Web browser, and in fact also carry out malicious activities such as recording keystrokes and capturing passwords.
- Keyloggers—These programs intercept keyboard events and record keys typed by a user. The information collected by keyloggers may be copied to a server controlled by a hacker who then uses the information to compromise the computer, conduct identity theft, or perform some other form of theft or fraud.

Email systems allow outsiders to introduce malicious software into an organization; it is also used by insiders in ways contrary to the intention of the organizations that provide the system.


Harassment and Hostile Workplace Issues

A hostile workplace is defined as an environment in which the harassment fundamentally alters the conditions of a workplace. The concept of a hostile workplace arose out of litigation related to sexual harassment in the United States Supreme Court case, *Meritor Savings Bank v. Vinson*, which found sexual harassment prohibited by Title VII of the Civil Rights Act of 1964. As technology in the workplace has changed, it is not surprising that the considerations for a hostile workplace have changed to include technology, especially email.

Emails have been central to cases of litigation, dismissal, and employee discipline for well over a decade:

- In 1995, Chevron Corporation was ordered to pay \$2.2 million to female employees to settle a case involving inappropriate emails.
- In 2002, six employees of the State of Washington Labor Department were terminated for excessive use of the state-provided email system for personal use. One former employee was chided for emails that contained “shockingly explicit, vulgar, and very offensive” language (Source: <http://www.gigalaw.com/articles/2003-all/towns-2003-03-all.html>).
- In 2006, a Chesterfield Township, Michigan detective was suspended and demoted for sending pornographic material through email to coworkers (Source: <http://www.freep.com/apps/pbcs.dll/article?AID=/20060208/NEWS04/602080469/1001/NEWS>).

Preventing the distribution of offensive material is a key driver to the adoption of content-filtering systems.


 For more information about email and its role in hostile workplace considerations, see Douglas M. Towns, “E-Harassment in the Workplace” at <http://www.gigalaw.com/articles/2003-all/towns-2003-03-all.html>.

Another issue facing commercial organizations is the loss of intellectual property and other confidential or sensitive information through information leakage.

Information Leakage and the Loss of Intellectual Property

Information leakage occurs when proprietary information is intentionally or inadvertently released to individuals outside the organization. The intentional theft of information is on the rise. According to the 2005 Computer Security Institute/FBI Computer Crime and Security Survey, unauthorized access to information and theft of proprietary information rose significantly over previous years.


The motivations for intentional theft can range from financial incentives, such as stealing a competitor’s customer list, to revenge by a disgruntled employee. In one recent case, a former Honeywell employee released payroll and other personal information about 19,000 Honeywell employees on the Internet; how the perpetrator procured the information has not been disclosed (Source: <http://www.computerworld.com/securitytopics/security/story/0,10801,108434,00.html>).

 For the full report on the CSI/FBI 2005 Computer Crime and Security Survey, see http://www.gocsi.com/forms/fbi/csi_fbi_survey.jhtml. For a disturbingly long list of incidents involving the release of confidential and private information, see ComputerWorld’s Special Coverage: Data Security Breaches at <http://www.computerworld.com/news/special/pages/0,10911,3888,00.html>.

Unintentional disclosures can be damaging as well. Typically, these disclosures result from one of several root causes:

- Not understanding the full range of information stored in a document or database
- Ignorance of organization policy
- Not following security procedures

For example, many users of Microsoft Office might not be aware that metadata about the author and file location may be stored in the document. In addition, “deleted” information may remain in the file but not displayed by Word.

 AntiWord, a free Microsoft Word reader, maps Word formatted documents to text and includes a feature to include information hidden in Word. This tool can be used to determine whether sensitive information is inadvertently stored in a Word document. AntiWord is available at <http://www.winfield.demon.nl/>.

In other cases, employees and contractors might not be aware of company policies governing sensitive information. For example, a marketing company making a proposal to a prospective client might be asked to provide references and examples of prior work. The salesperson may decide to send a sample from another client without first asking permission from the client, violating company policy. This act could leave the marketing firm liable to the customer whose information has been disclosed.

In the worst case, employees and contractors may be aware of company policies but violate them anyway. For example, an employee may be working with a vendor to create a demonstration of a new document management system proposed for the research and development department. The vendor asks for sample documents in order to create a realistic demonstration. Although the employee knows such documents cannot be sent outside the company, he or she emails them anyway, assuming they will just be used for the demonstration. Of course, once the documents leave the boundaries of the company's network, the company will have no control over their distribution and use.

In some cases, the disclosure of information does not involve proprietary information about a company but private information about its customers.

Disclosure of Private Information

Even when employees know the rules, there are times the rules are violated. Consider an employee rushing to meet a deadline to deliver a list of customer contacts for a marketing campaign for a bank. The contact information includes sensitive demographic data, including household income as well as account numbers. According to company policy, the data must be encrypted before being transmitted outside the corporate network, but the employee decides to “cut a corner” to save time and sends unencrypted data outside the company. If this data were intercepted, the company could be liable for violating privacy regulations.

Privacy regulations have been created by a number of government bodies and cover a range of topics:

- The Health Insurance Portability and Accountability Act (HIPAA) of 1996 specifies regulations for maintaining the privacy of protected healthcare information.
- The Gramm-Leach-Bliley Act (GLBA), formally known as the Financial Modernization Act of 1999, includes provisions to protect consumers' private financial information.
- California law, State Bill 1386 (SB 1386), addresses the threat of identity theft and fraud by requiring notification when a California resident's identifying information (such as Social Security number or driver's license number) is disclosed.
- The Safe Harbor framework, negotiated by the United States Department of Commerce and the European Commission, meets requirements of the European Commission's Directive on Data Protection, which defines standards for privacy protection for European Union citizens' data.

The intentional or unintentional disclosure of private information as well as other risks—such as receiving malicious programs through email, the exchange of offensive material on email, and the loss of intellectual property—constitutes compelling reasons to filter content. The next section examines several methods for addressing these risks.

Email Content-Filtering Techniques

Email content-filtering techniques range from the relatively simple to the complex. The simplest techniques use word matching to determine whether a message contains inappropriate content. At the other end of the spectrum, sophisticated antivirus detection techniques can scan for large numbers of known viruses using efficient pattern-matching methods.

Email content-filtering techniques can be categorized into several groups:

- Term matching
- Regular expression matching
- Matching in context
- Malware scanning

Each of these techniques provides varying levels of protection at equally varying levels of complexity. (That is, with respect to the way these programs work, not necessarily with the complexity of installation and administration). They all, however, fit into the stream of email processing in similar ways—following the pattern that Figure 7.2 shows. (Malware scanning does not use a dictionary, but a database of patterns; the overall structure is the same, however.)

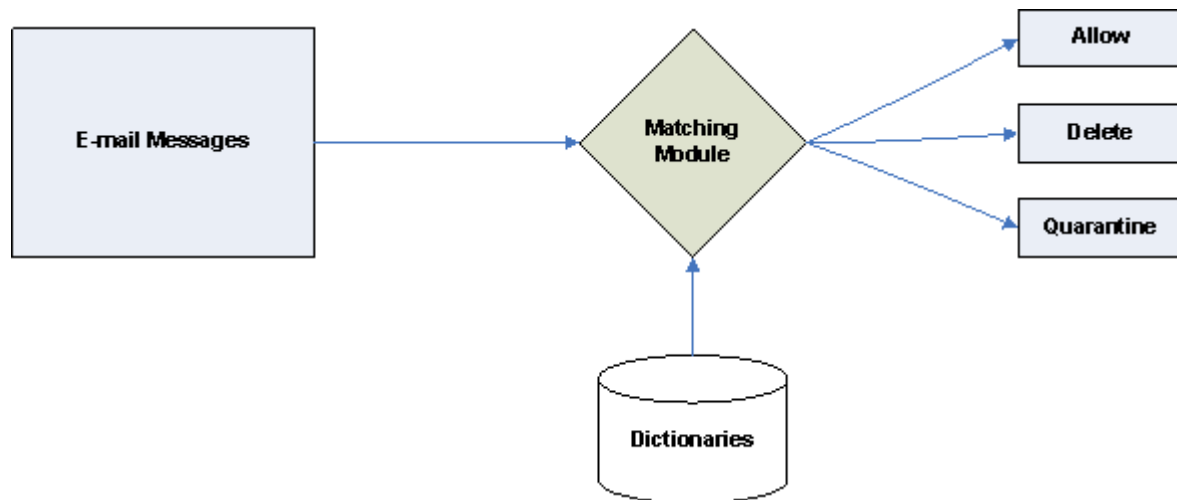


Figure 7.2: Email content scanning analyzes each message for particular patterns and carries out particular actions based on policies defined by email administrators.

Term Matching

Term matching is a process of scanning the contents of a message for particular words or terms that are deemed inappropriate. Terms are maintained in one or more dictionaries. The contents of a message are compared with the contents of the dictionaries to determine whether the message should be blocked. Although the process is relatively simple, there are a number of variations that vendors may use to improve the speed and accuracy of term matching. Because the methods used in commercial programs are proprietary, this discussion will focus on general techniques that may or may not be used in any particular system.

These techniques all determine whether a particular set of terms is present in a message. The term-matching program would then apply a calculation to determine the relative weight of the banned terms prior to making a decision to allow the message through or to handle the message in some other way. For example, if the term “gamble” is included in a dictionary of controlled terms, one occurrence of the term may not warrant blocking the message. One can easily imagine a message legitimately using that term in a metaphorical way:

```
From: Mary Jones <mjones@mycompany.com>
To: Kevin Johnson <kjohnson@mycompany.com>
CC:
Subject: Next Sales Call with Acme
```

Kevin,

I'm concerned about your approach on the next sales call with Acme. We can't gamble with this one - we need it to make this quarter's numbers. Call me to discuss.

Mary

However, in the context of other related terms also found in the dictionary, the cumulative effective of the presence of multiple terms can indicate inappropriate content:

```
From: gamble-while-at-work@onlinecasinoatwork.com
To: Kevin Johnson <kjohnson@mycompany.com>
CC:
Subject: 24-hour Online Casino

Blackjack! 30 kinds of Poker! American and European Roulette!
Slots and more!

Don't wait - Gamble while at work. Join us at
www.onlinecasinoatwork.com
```

Matching large numbers of terms against high volumes of emails requires efficient processing. Some techniques are:

- Dictionary lookups
- String matching
- Word stemming

Dictionary Matching

Dictionary lookups are simple but can be inefficient. This technique requires that a message be divided into lexical tokens as Figure 7.3 shows.

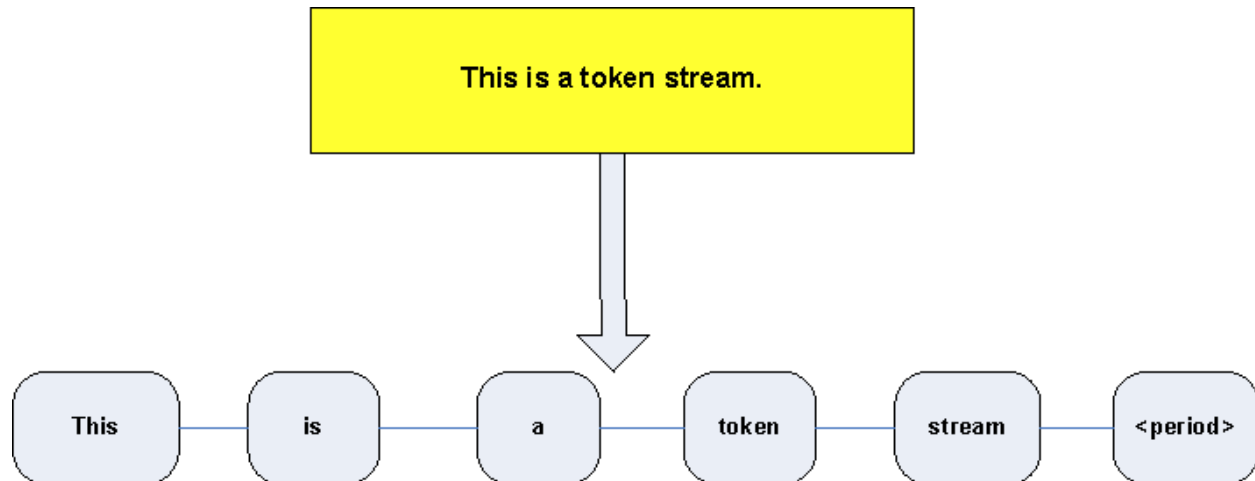


Figure 7.3: Lexical scanning divides a stream of text into individual tokens consisting of single words or punctuation marks that are further analyzed by pattern-matching techniques.

Each token is then compared with the contents of a dictionary and, if the word is present, the process takes that term into account in the cumulative measure of the message.

The process of looking up terms sounds simple, and it is, but when working with the volumes of text found even in small and mid-sized organizations, the time required to perform so many lookups can quickly add up. Without delving into too many technical details, consider three lookup methods.

- **Linear lookup**—Compare each term in the message with each entry in the dictionary. On average, the time to lookup a term in the dictionary is proportional to the length of the dictionary. This method is highly inefficient and impractical.
- **Binary lookup**—This approach uses a divide-and-conquer technique that allows terms to be found in logarithmic time. For example, if it takes 1 unit of time to search a list of 32 words, it will take 2 units of time to search 64 words, 3 units of time to search 128 words, 4 units of time to search 256 words, and so on. Although much better than a linear lookup, this approach can be impractical for large dictionaries.
- **Hash lookup**—This technique calculates a numeric value for each term that is used as an index to the dictionary. This technique requires about the same amount of time to lookup terms regardless of the size of the list. (Actually, the time can increase at some points when the dictionaries get very large, but in many cases, it is safe to assume a constant amount of time to perform the lookup).

The binary lookup works well with short lists and hash lookups work well with moderate-sized and large dictionaries. However, even the fastest of these techniques may not be optimal in all situations. Other techniques, based on string matching, may be more appropriate.

String Matching

In many cases, problems can be solved more efficiently by exploiting patterns in data. For example, if “gamble” and “gambling” are both words in a content-filtering dictionary, you can take advantage of the fact that five letters, “gambl,” appear in both and in the same order. One way to do so is to treat the content of the email message not as separate words but as a string of characters.

Rather than look up the work “gamble” in the dictionary, a content-scanning program could use the following rule:

Look for a “g” followed by an “a” followed by an “m” followed by a “b” followed by an “l” followed by either an “e” or an “i,” which is then followed by an “n” followed by a “g.”

This type of character-by-character pattern matching is quite an undertaking, so there are a number of highly efficient algorithms for pattern matching. One advantage of this approach is that it saves space in dictionary storage as well as provides a rapid method for matching terms. As Figure 7.4 shows, the terms “gamble” and “gambling” can be represented in a combined representation using a directed graph.

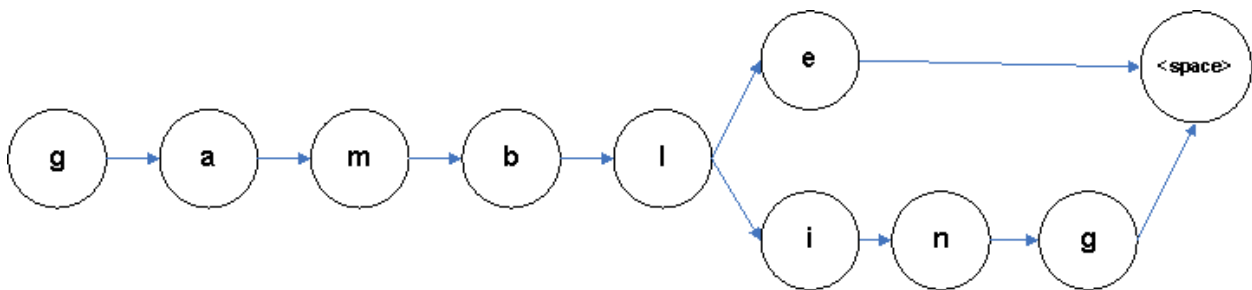



Figure 7.4: Multiple words can be efficiently stored and matched using a set of nodes and arcs.

 The formal term for this representation is a finite state machines (FSM). For more information about FSMs and string matching see <http://www.cs.princeton.edu/introcs/73fsa/>.

Word Stemming

Even with efficient storage and processing, the number of words in a dictionary can grow rapidly. For example, if an email administrator wants to block content related to gambling, the dictionary should include “gamble,” “gamble,” “gambled,” and “gambling.” A better approach is to use word stems, or roots of words, for matching. By using stemming, a single dictionary entry can be used to detect multiple forms of a word.

Limits of Term Matching

Term matching in its various forms is the most basic method for analyzing content. Although this technique is relatively simple to implement, there considerable drawbacks. Term matching does not work well with misspelled words. In the examples earlier, the word “Gamlbe” would not be detected as a controlled term. Similarly, spammers commonly use punctuation marks to disguise words that readily trigger detection. Instead of using “Low Mortgage Rates” in a subject line, spammers might use “Low Mort!gagge Rat-es.” Trying to capture all possible variations and misspellings is impractical.

One method of addressing this problem is to use a *soundex* or similar function. The soundex function computes a string of characters for a word based on the sounds of the characters that make up the word.

How to Compute a Soundex

The soundex for a word is calculated as follows:

1. Capitalize all letters in a word and remove punctuation marks.
2. Keep the first letter of the word.
3. Remove all occurrences of A, E, I, O, U, H, W, and Y.
4. Replace each of the following letters with the corresponding number:
 - 1 = B, F, P, V
 - 2 = C, G, J, K, Q, S, X, Z
 - 3 = D, T
 - 4 = L
 - 5 = M, N
 - 6 = R
5. Remove all pairs of digits that appear next to each other.
6. Pad the string with 0s if necessary to create a four-character string, and return the first four characters, which should be the first letter of the original word followed by three digits. For example, the word “gamble” and “gambllle” would both yield “G514” but the misspelling “gamlbe” yields “G541.”

A number of variations to the soundex algorithm have been developed as well as related algorithms.

Source: Soundex, Wikipedia, <http://en.wikipedia.org/wiki/Soundex>. For variations on the soundex algorithm, see “How To: Understanding Classic Soundex Algorithms” at <http://www.creativyst.com/Doc/Articles/SoundEx1/SoundEx1.htm>.

Even with the soundex algorithm, term matching as a content filter can generate a high number of false positives; that is, messages are identified as banned when in fact they are legitimate messages. To counter this problem, weighting schemes must be carefully crafted to include enough occurrences of controlled words to minimize false positives. This, however, introduces the chance of an unacceptable number of false negatives (messages that should be blocked but are categorized as legitimate). Alternatives to simple term matching, including regular expression matching and matching in context, seek to minimize these problems.


Regular Expression Matching

Regular expressions are patterns that can match multiple strings. Patterns are made up of letters, digits, and special pattern characters. The “*” character is a special pattern character that matches any number of characters. For example, rather than having a dictionary with the words “gamble,” “gambles,” “gambling,” and “gambled,” a regular expression pattern such as “gambl*” could be used to match them all.

Other commonly used special patterns include:

- [0-9] matches a digit
- [a-zA-Z] matches a letter
- \s matches a whitespace character, such as spaces and tabs
- \b matches a word boundary
- {n,m} used to modify another pattern, specifying that the pattern must occur at least *n* times but not more than *m* times

There are many other special patterns used in regular expressions that allow developers to craft precisely targeted patterns.

 For examples of regular expressions for matching strings, numbers, dates, email addresses, and other useful patterns, see The Regular Expression Library at <http://www.regexp.com/Default.aspx>.

An advantage of regular expressions over term matching is that regular expressions allow for matching more complex patterns of multiple terms. For example, a simple pattern such as

```
"Texas\s[hH]old*\s[pP]oker"
```

would match against “Texas Holdem Poker,” “Texas Hold’m Poker,” and similar variations. There would be no risk of having a legitimate message mistakenly categorized as a blocked message because it had both the terms “Texas” and “hold” in it, which might occur with a simple term-matching approach. Regular expressions are also useful for detecting well-defined numeric patterns, such as Social Security numbers, credit card numbers, bank routing codes, and so on.

A drawback of regular expressions is that, as the earlier example indicates, they can be cryptic and difficult to write. There is also additional processing time required to match regular expressions, and care must be taken when writing regular expressions to ensure that they are crafted efficiently.

Another limitation is that efficient regular expressions do not try to span too much text or capture too many patterns in a single expression. Thus, they work well for capturing the local context of a term (for example “Texas” near “poker”) but not for capturing the global context of a message. Additional techniques are needed for that.

Matching in Context

If a human were given the task of categorizing messages, the person would not be tripped up by minor misspellings, added punctuation marks, or controlled terms used in appropriate context. One of the reasons is that a human reader evaluates messages based on the broad context of the message, not just a limited string of characters within the message.

To improve the quality of email filtering, software must take into account the global context of a message. This is done in a number of ways:

- Rule-based analysis
- Categorization based on terms
- Categorization based on n-grams

Rule-based analysis complements categorization schemes and may be used in conjunction to improve performance.

Rule-Based Analysis

Rule-based analysis uses a series of IF-THEN rules that can examine both the structural elements and content of a message. Structural elements are characteristics of the message that include:

- The sender of the message
- The number of recipients
- The size of the message
- The number and types of attachments

These elements can be used to craft rules to allow or block particular messages. For example:

- If the sender is a member of the legal department, a message is allowed regardless of the content.
- If the number of recipients is greater than 10, block the message.
- If the file extension of a file does not match the MIME type detected by examining the contents of the file, quarantine the message.

Rules can also be based on the contents of a message. These rules could use a series of terms and Boolean operators to detect patterns indicative of inappropriate content. The Boolean operators could include AND, OR, NOT, and NEAR. Example rules in this category are:

- If “Texas” NEAR “poker,” block message
- If “breast” AND NOT “cancer,” block message
- If “gamble” AND “slot” AND “online,” block message

Rules that operate on the content of the message suffer from the same drawbacks as term-based matching; that is, misspellings and added punctuation marks can result in mistakes in categorizing messages. However, structural rules provide a mechanism for controlling filtering based on global properties and other characteristics not tied to terms used in the body of messages. To avoid the limits of targeted rule-based systems, email content can also be categorized by looking at the entire set of terms used in a message.

Categorization Based on Terms

Earlier, this chapter discussed the use of terms and dictionary lookups to determine whether a message should be allowed or managed in some other way (for example, deleted, quarantined, and so on). In that scheme, the scanning system was examining individual terms and determining whether the collective set of controlled terms in the message warranted blocking the message. Although limitations of this approach are well known, the use of terms for categorization is not without merit, you simply need to adjust how you use the terms.

To improve on simple term lookups, you can use categorization techniques that use *all* terms in a message to determine the classification of a message. The object is to measure both the frequency with which terms are used and the context in which they are used.

Suppose when an email message is scanned, it can be categorized into two groups: allowed or blocked. Also assume you have a large number of samples of messages that should be allowed as well as a large sample of messages that should be blocked. These messages can be used with algorithms known as *supervised learning algorithms* to train a scanning program with statistical measures for categorizing email messages.

A well-known and widely used method for categorizing email content, as well as other categorization tasks, is Naïve Bayes, also referred to as Bayesian categorization (see Figure 7.5). This algorithm uses examples to determine the frequency with which certain characteristics (such as terms) appear in relation to the likelihood that the message is in a particular category (such as allowed or blocked). For example, if the term “gamble” appears frequently in messages that are categorized as blocked and infrequently in example messages that are categorized as allowed, then a new message with that term is more likely to be classified as blocked.

 The details of Naïve Bayes are beyond the scope of this chapter. For more information about this approach, see http://en.wikipedia.org/wiki/Naive_Bayes.



Figure 7.5: Categorization algorithms, such as Naive Bayes, group messages based on shared characteristics with previously categorized messages.

Classification algorithms, such as Naïve Bayes, can use just about any characteristic of the objects they are classifying. (There are some restrictions related to the independence of these characteristics, but we will assume the restrictions are met for our purposes). When working with email messages, you can, of course, use terms, but then you run into the same problems: misspellings and added punctuation marks within words can throw off your measures. Fortunately, when working with messages with high error rates, you can use a simple variation of the algorithm to compensate for the problem.

Categorization Based on N-Grams

An n-gram is a sequence of characters extracted from another set of characters. The “n” in n-gram indicates the number of characters extracted. For example, “cat,” “ate,” and “teg” are all 3-grams. To determine the set of n-grams for a piece of text:

1. Start with the first letter of the text.
2. Select n characters, and output as an n-gram.
3. Move to the next character.
4. Repeat steps 2 and 3 until the end of the text is reached.


For example, the text

Don't wait - Gamble while at work.

Yields the following 3-grams:

"Don"	"on'"	"n't"	"t "	"t w"
" wa"	"wai"	"ait"	"it "	"t -"
" - "	"- G"	" Ga"	"Gam"	"amb"
"mbl"	"ble"	"le "	"e w"	" wh"
"whi"	"hil"	"ile"	"le "	"e a"
" at"	"at "	"t w"	" wo"	"wor"
"ork"				


The advantage of this approach is that the characteristics used are smaller than terms, so a minor difference in a term, such as a transposed pair of characters, will change some n-grams generated from that word, but not all of them. Similarly, with the use of punctuation marks in words, some n-grams will be different from those generated from the correct spelling, but many will be the same. N-grams help to dilute the influence of errors such as misspellings because much of the information about the word or term is reflected in a number of n-grams.

 For more information about how N-grams are used for categorization, see William B. Cavnar, John M. Trenkle “N-Gram-Based Text Categorization” at http://citeseer.ist.psu.edu/rd/43754390%2C68861%2C1%2C0.25%2CDownload/http://citeseer.ist.psu.edu/cache/papers/cs/810/http:zSzzSzwwww.info.unicaen.frzSz%7EgiquetzSzclassifzSzcavnar_trenkle_ngram.pdf/n-gram-based-text.pdf.

Filtering emails based on the content of messages can use a combination of term matching, structure-based rules, and categorization methods. When the task turns to detecting malware, specialized approaches are required.

Malware Scanning

Detecting malicious software in content streams, such as email, is particularly challenging. Certainly, spammers try to mask the contents of their messages in such a way as to avoid detection, but virus and worm writers have created far more sophisticated techniques for avoiding detection. Conventional text-scanning technologies, such as term matching and Naïve Bayes categorization, are not sufficient to combat the latest generation of malware. Two general approaches are used to detect malware: signature-based detection and behavior-based detection.

 This section briefly describes highlights of malware scanning; for details about the evolution of malware and countermeasures, see Chapter 3.

Signature-based detection uses a set of patterns that act like fingerprints to uniquely identify malicious programs. Antivirus vendors are constantly adding to their signature databases to keep up with emerging viruses, worms, and related malware. Once a signature has been identified for a malicious program, the code can be readily blocked at the network or desktop levels. As described in Chapter 3, virus writers have used code-changing techniques to avoid detection by signature-based systems. Antivirus researchers and developers have responded with another type of detection method, known as behavior-based detection.

In behavior-based detection, an antivirus program looks for specific behaviors likely to be found in malicious code but not as likely in legitimate code. These include making particular types of system calls and changing certain registry keys. Behavior-based detection may also simulate an execution of a program to determine the kinds of actions carried out by the program. Like other areas of email content filtering, malware detection is best performed with a combination of complimentary techniques.

Multiple Techniques for Email Content Filtering

Filtering email presents two opposing challenges: the process must be comprehensive and it must be efficient. If the process is not comprehensive, email that should be blocked will be allowed through; however, if the scanning is too strict, legitimate emails may be blocked. Finding the right balance between blocking banned content and not blocking allowed messages can require a combination of techniques, each with distinct strengths. Regardless of which techniques are used, they must be efficient. Even small and midsized organizations might have to manage large volumes of emails, so performance is also a primary concern. A specialized version of email scanning that often combines multiple techniques is anti-spam systems.

Filtering Spam

Spam requires a special case of email filtering. Email administrators have to deal with large volumes of unwanted, unsolicited email. In addition, spammers are constantly trying to evade detection, so anti-spam developers are engaged in a similar cat-and-mouse game that antivirus developers find themselves in. Spam is also an example of where multiple techniques work better than any single technique, in most cases. Although a spammer may be able to avoid detection by a Naïve Bayes filter by including long strings of nonsense text at the end of a message, a rule-based detection system or regular expression pattern matching might detect such tricks.

Policies and Actions for Filtered Content

After a message has been categorized as spam, offensive material, or otherwise banned content, the question arises, what to do with it? This is where policies are needed. Policies are sets of rules that define the action taken by the scanning system in response to a event, such as the detection of spam. Responses may include:

- Deleting an outbound message deemed in violation of company policy
- Sending a notification to the sender that a message was deleted because it violated company policy
- Quarantining an inbound message categorized as spam
- Deleting an inbound message categorized as offensive or otherwise inappropriate
- Re-routing a message deemed offensive or otherwise inappropriate to an administrator's queue for review and further action

Ideally, policy enforcement would be supported with robust reporting to allow administrators to track trends in spam, offensive content, and the types of actions taken in response. Of course, not all potentially offensive and inappropriate material enters the organization through email systems. Comprehensive content security must also include the ability to block inappropriate browsing on the Web.

Web Browsing and URL Blocking

Most organizations do not concern themselves with employees who spend a few minutes checking weather forecasts, traffic conditions, or reading the news online during breaks. There are, however, plenty of time wasting, inappropriate, and patently offensive sites on the Internet that should never be accessed from company computers. For those cases, URL blocking is a reasonable mechanism for controlling access to those sites.

As Figure 7.6 shows, when a browser makes a request to retrieve a URL, a URL blocking program first checks the URL against a database of banned sites. If the requested URL is in the database, the request is not sent on and, typically, a message is displayed notifying the user that the site requested is not allowed. If the requested URL is not on the list, it continues to be processed.

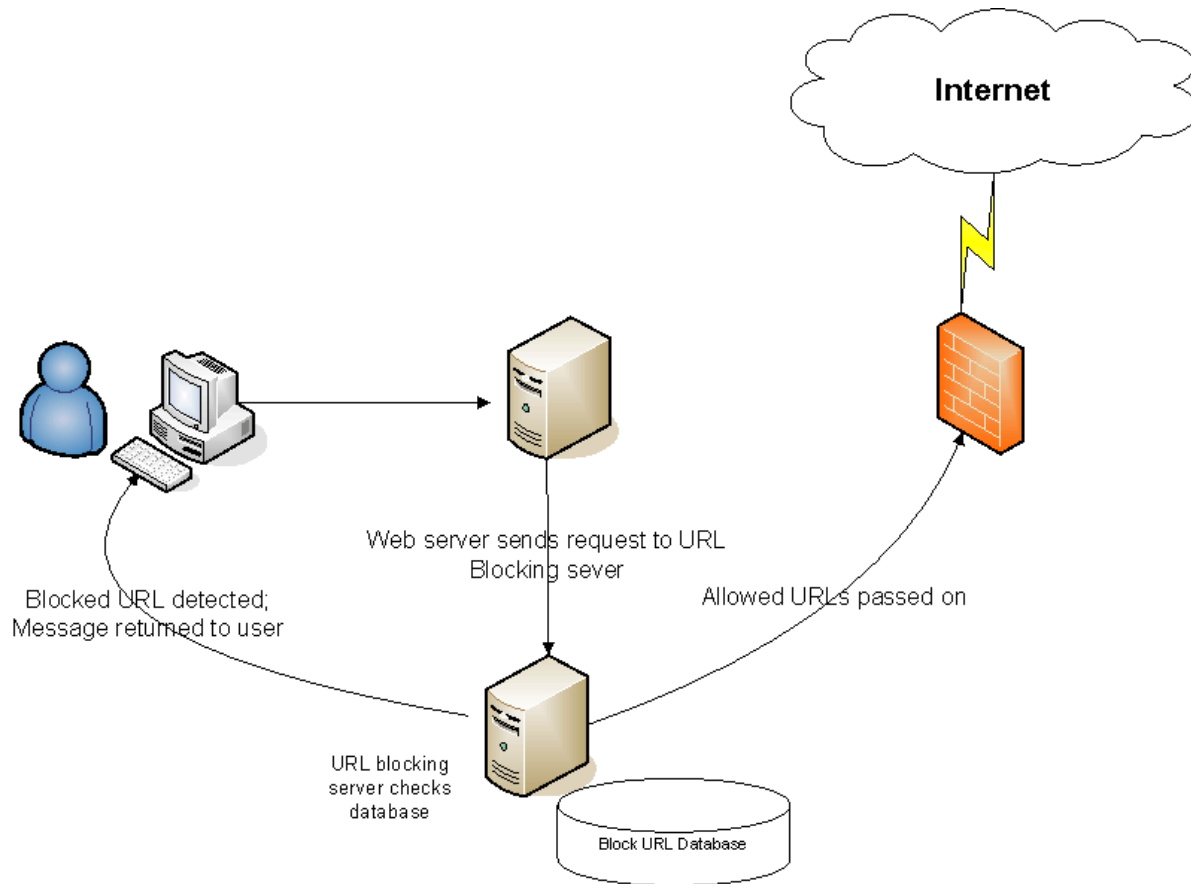


Figure 7.6: URL blocking intercepts requests for Internet resources and compares them with lists of banned sites.

The database of URLs may contain both white lists and black lists. White lists are known sites that users are allowed to access. The most restrictive form of URL blocking allows access only to sites on white lists. Black lists are compilations of URLs of known sites that are not allowed to users. Because the Web changes constantly and Web sites can easily move content from one domain to another, to be effective, URL blocking databases must be kept up to date in near real time. Vendors and some open-source projects maintain real-time black lists that are used in URL blocking systems. It is impractical for a single business to implement a full black list.

Securing content in an enterprise is a multifaceted challenge. Systems and network administrators must counter malicious software, inappropriate content, and the potential loss of proprietary and private information. Meeting these challenges is best done by implementing a general security practice known as defense in depth.

Multi-Layered Security: Defense in Depth

The defense-in-depth strategy acknowledges the fact that no single security technique or system can address all potential threats. For example, a properly configured firewall can block probes and attacks from outside the network, but someone with access to the internal servers and applications of a network will not be stopped by the firewall. You can broadly categorize security measures as applying at three levels:

- Gateway
- Network
- Desktop

Security at these three levels can provide a comprehensive, multi-layered defense.

Gateway Defenses

The purpose of gateway defenses is to prevent unwanted content from entering and to prevent controlled content from leaving a network. The first line of defense at this point is a network firewall. Firewalls perform packet filtering, monitor the state of communications between systems inside and outside the network, and, in some cases, provide basic access control and other security functions.

Just within the network perimeter is an ideal position for content filtering. At this point, inbound traffic has cleared the preliminary line of defense provided by the firewall; it is also the last point before the firewall to analyze outgoing traffic. A well-formed email message, for example, will be allowed through a firewall configured for SMTP traffic even though it might contain spam or malicious software. A content-filtering network appliance configured for anti-spam scanning or antivirus scanning could block the message before it reaches the email server.

Gateway defenses are barriers between the internal network and the Internet. Within the network, additional measures are required.

Network Security Measures

Within the network, security measures include two key activities—locking down servers and applications and detecting anomalous behavior within the network. Once an intruder has gained access to a network, either by breaching perimeter defenses or through legitimate access to the network for other purposes, the intruder can attack servers and the applications that run on them. Targets include:

- File servers containing proprietary documents
- Web servers that can be compromised and used for other purposes
- Database servers with customer information

Locking down these servers is essential. This task entails shutting down any unnecessary services, removing applications that are not essential (for example, production servers should never have compilers installed), ensuring that critical patches are installed, and ensuring that the operating system (OS) and applications are properly configured. Host intrusion detection systems should be used on critical servers to detect any unauthorized changes to system files or key data sources.


Identity management and access controls are also central to network security. Users should be given the minimum access required to do their jobs, and access to information should be granted on a need-to-know basis. Networks can be monitored with intrusion prevention systems to detect anomalous behavior, such as unusually high levels of SMTP traffic from several desktop devices—which could indicate compromised machines being used as bots for spamming.

Network security is a broad and deep discipline. Effectively securing content depends on well-defended network services.

Desktop Security Measures

The final layer of defense is at the desktop. Although gateway defenses and network security measures can provide countermeasures to many threats, some malicious code or damaging attack may reach the desktop. Such is especially the case for mobile devices that are not always connected to the network and therefore not continuously protected. At a minimum, desktops should be protected with

- Antivirus software
- Anti-spyware
- Personal firewall
- Automatic OS updates
- Vulnerability scanners for desktop systems, such as Microsoft Baseline Analyzer

 The Microsoft Baseline Analyzer can identify many known vulnerabilities, missing patches, and insecure configurations in Windows OSs. The tool is freely available at <http://www.microsoft.com/technet/security/tools/mbsahome.mspx>.

A multi-layered security scheme that includes gateway defenses, such as content scanning, as well as network and desktop security measures, is a critical element of effective security management practices.

Summary

There are a multitude of threats on the Internet. Businesses and organizations that leverage the benefits of the Internet must also contend with the risks. Effective content security requires the ability to scan content that enters and leaves a network to ensure that inappropriate, offensive, proprietary, or private information is not disseminated against organizational policies. Fortunately, a wide range of techniques, from basic term matching to sophisticated pattern matching, are available to systems administrators and managers responsible for securing an organization's content.

Download Additional eBooks from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this eBook to be informative, we encourage you to download more of our industry-leading technology eBooks and video guides at Realtime Nexus. Please visit <http://nexus.realtimepublishers.com>.