CHAPTER 16

FIGHTING SPAM DEFENSIVELY

The art of defending against unwanted e-mail goes beyond simply installing a few anti-spam programs and updating the profiles. Spam e-mail is a security threat, in that it denies availability to resources, and with a combined attack such as a spam message with a virus attachment, this threat can quickly become very serious and expensive. As with any information security-related issue, a defense-in-depth posture is required. All of the tools we covered in the previous 15 chapters of this book do a great job of managing your spam-fighting energies, but additional, basic information security techniques could exponentially increase your success.

In earlier chapters, we discussed e-mail management organization and policy, but what other network and IT management-level strategies reduce the threat spam poses to the organization? In this chapter, we discuss these strategies. Some are extreme, but most are up-and-coming spam-fighting techniques that preclude current tools and methods.

WIN BEFORE FIGHTING

In addition to implementing tools, policies, and management schemes to deal with the spam that's hitting your mail server, you can prevent spam from ever reaching the server in the first place, or even reaching your network, in many ways. Three "points of exploitation" are discussed in this section: e-mail addresses, challenge/response, and future spam fighting.

E-mail Addresses

First off, how do spammers get your e-mail address in the first place? We've discussed this briefly in Chapter 2, but here we cover both the techniques the spammers use and the strategies you can implement (as a systems administrator or user) to thwart these tactics.

Devious Methods

The price of spamming is cheap and their tools are proven in other applications. We discuss three common methods developed from other, well-established technologies: brute force, spiders, and mailing lists.

One of the most common techniques for a spammer to hit your e-mail address is similar to methods hackers use to crack passwords on user accounts: a brute-force dictionary attack. The spammer gets a domain name (yourorg.com) and begins sending mail to a "dictionary" of possible usernames. Thus, the dictionary includes: tom@yourorg.com, jane@yourorg.com, betty@yourorg.com, and so on. Never mind that tom@yourorg.com doesn't exist as a valid e-mail address; Jane and Betty just got spammed. Spammer dictionaries contain thousands upon thousands of iterations of common e-mail usernames, including all the standard aliases such as webmaster@yourorg.com, sales@yourorg.com, and more. To prove it, set up an account called bsmith@yourorg.com. We'll bet you the price of this book that within 24 hours that e-mail account will begin to receive credit, body part enlargement, and other less savory offers without anyone ever knowing the e-mail address exists.

Spammers also get e-mail addresses from spiders, automated programs that cull through thousands of web sites, newsgroups, and message boards, cataloging and storing valid e-mail addresses for later exploitation. These spiders are adapted from the same technology that allows search engines such as Google and AltaVista to catalog web pages by examining the content. These valid e-mail lists are then fed into the Big Spam Machine and the spammer is off and running. Oftentimes these lists are circulated to groups of spammers, giving them more bang for their buck.

NOTE According to a Center for Democracy and Technology study, e-mail spiders (often affectionately referred to as web page scrapers) are the e-mail harvester of choice, by 95 percent of spammers. Read more about the center's study and findings on its web site at *http://www.cdt.org/speech/spam/030319spamreport.shtml*.

Finally, spammers utilize the old, tried-and-true method of sending junk snail mail, the dreaded mailing list. Any time you sign up for "free offers," subscribe to magazines, or otherwise give your e-mail address out for any commercial transaction, there's a risk that your information will be sold and eventually end up in the hands of a spammer. Additionally, spammers themselves compile findings from their spider and brute-force methods into lists for sale to other spammers, completing the Hateful Circle of Spam.

Protecting Your Information

How do you protect your e-mail address and still utilize this almost free, unlimited communication method? It's a fine line, but next we discuss three complementary methods for keeping your e-mail address out of the hands of the forces of spam.

E-mail Aliasing One method that seems to work well is the use of e-mail aliasing. E-mail aliases are essentially pointers that front for real e-mail addresses. We discussed them briefly in Chapter 1, but here, the application is somewhat different. Instead of a functional alias for a group of people or a role (such as webmaster@yourorg.com or support@yourorg.com), you can utilize aliases for your own e-mail address and filter or abandon it if the spammers find it. For example, Tom's real e-mail address is tom@yourorg.com. He does not want this address to go out to the greater Internet, though it does appear on "offline" content, such as his business card and letterhead. Tom uses e-mail for three rather distinct purposes: work-related communication, such as communications with co-workers, vendors, and partners; personal communications, such as notes to a few friends and relatives that e-mail him at work; and finally professional groups, such as the chamber of commerce, local charities, and the like. Tom has his mail administrator set up three aliases: tomsmith@yourorg.com, tommysmith@yourorg.com, and tsmith@yourorg.com. All three of these aliases point back to tom-smith445576@yourorg.com. Mr. Smith might subscribe to a professional mailing list with the tsmith address, a family web site with tommysmith, and he might provide the tomsmith address to several vendors at an online conference. If, suddenly, Tom begins receiving spam that hits the tommysmith address, he can either

automatically filter all mail coming to that address to a specific folder for later review (provided his company doesn't use one of the fine tools outlined in this book), change the alias and communicate it to the few people that should know it, or remove the alias completely. Though somewhat cumbersome, changing e-mail aliases prevents changing or abandoning your *actual* e-mail address when spam becomes too much.

E-mail Obfuscation One common method on Usenet for thwarting spiders is e-mail obfuscation. This method entails changing how your e-mail address appears so that it fools a machine but not a human. Thus, if Elvis spends a lot of time posting to Usenet and web message boards, he would configure his e-mail address to appear as elvisREMOVE@graceMEland.com. A person reading this would know to take out *REMOVE* and *ME* if he or she wishes to send Elvis an e-mail, while a spam spider would catalog the address, as is. Though it wastes only miniscule resources of the spammer, and somewhat burdens a person wishing to send you mail, it also adequately protects your e-mail address when it's sitting out there on a web message board for months (or years). It's also a good idea never to put an e-mail address link anywhere on your web site. Use a web form with a backend Common Gateway Interface (CGI) for transmitting messages to specific groups, such as sales@yourorg.com, support@yourorg.com, and so on. This keeps your e-mail addresses protected while still allowing your customers to communicate with you.

TIP

One improvement on this method is being implemented on a few of the savvy web boards we visit. On the fly, the web board's posting engine converts all e-mail addresses into a JPEG image. Not only does this effectively thwart spiders, but it removes the somewhat cumbersome practice of obfuscating your e-mail address every time you post, configuring your client with an invalid e-mail address, or forgetting altogether.

Server-Side Encoding A method that's gaining popularity within server-side programming languages to include PHP, Perl, ColdFusion, and ASP is the server-side encoding of e-mail addresses, especially those that appear on web pages, such as <a mailto:> web links, message boards, and newsgroups. The basic character conversion method is similar to e-mail obfuscation, but the e-mail address actually functions without the sender having to type the address manually into their mail client. The characters of both the `mailto` tag and the e-mail address are converted into decimal encoding, rather than plain text. Thus, the hyperlink *elvis@trailerpark.com* becomes this:

```
<a href=mailto:&#101;&#108;&#118;&#105;&#115;&#64;&#116;&#114;&#97;&#105;&#108;
&#101;&#114;&#112;&#97;&#114;&#107;&#46;&#99;&#111;&#109;>&#101;&#108;&#118;
&#105;&#115;&#64;&#116;&#114;&#97;&#105;&#108;&#101;&#114;&#112;&#97;&#114;&
#107;&#46;&#99;&#111;&#109;</a>
```

The bad news is that this simplified character conversion technique fools only the least advanced e-mail harvesters. Most spammer tools actually already use these sneaky

methods to conceal the spammer's e-mail address, web site, and other contact information from blacklists, spam filters, and other anti-spam tools.

The advanced version of the preceding example is to wrap that encoded e-mail address in a server-side encoder that generates the e-mail address on the fly, rather than display it on the web page or within the HTML code. For example, if Elvis wanted to be contacted from his web site, but he didn't want his e-mail address on every spam CD from here to Seoul, he'd use a server-side script to produce something like this:

```
<script type="text/javascript">
//<![CDATA[
function hiveware_enkoder(){var i,j,x,y,x=
"x=\"783d22793e23344f613e3132793c363e5d233733356c37624a6d6a3d5b79792f3a386d" +
"3735373637643e34385d5d5d23666f3735687537376935363434656536353567373c6a3438" +
"2c3e3664333733336738363436373765 2a7c37367a2c37373e3339383764363837366638 76" +
"6f353366746635643333313138383634373838627138336629373828393136383664333835" +
"3a38262835372c7936332f333466313837363638333974763333637434377531643836643 7" +
"3733336437732932336a2d313a333338343138356635373739 2a2a38333c7e37357a5d2367" +
"353c6b31363e663637776237346d296534792f363164693532627336644275333329313735" +
"2a2a65383c7935383e7965372f74323776633a33747537327329373a322a37643c7a38353e" +
"283737283c643767703638732967346a3e6237313c36336a3d3831792f37326d6637646f68" +
"3837756937383c6a33372c3e633332313a382a7c34357a2c31663e7937342f743767766338" +
"3574753833732937376a2d6534362a64333c7e323767703a37732964676a3e3732363c3466" +
"6a3d3736792f38336d6638336f68333375693a343c6a31372c3e3238323133632a7c34317a" +
"2c34633e7937632f7433667663375d5d74755d233c73297a3e6a2d2834362a37673c7e3765" +
"7a3e283c7a2f6770747673346374673875733438296b296a2a3c233c6b3e6677626d29792f" +
"64696273427529312a2a3c793e792f74766374757329322a3c7a3e28283c677073296a3e31" +
"3c6a3d792f6d666f6875693c6a2c3e352a7c7a2c3e792f747663747573296a2d332a3c7e67" +
"7073296a3e333c6a3d792f6d666f6875693c6a2c3e352a7c7a2c3e792f747663747573296a" +
"2d332a3c7e7a3e7a2f747663747573296b2a3c223b793d27273b783d756e65736361706528" +
"78293b666f7228693d303b693c782e6c656e6774683b692b2b297b6a3d782e63686172436f" +
"646541742869292d313b6966286a3c3332296a2b3d39343b792b3d537472696e672e66726f" +
"6d43686172436f6465286a297d79\";y='';for(i=0;i<x.length;i+=2){y+=unescape('" +
"%'+x.substr(i,2));}y";
while(x=eval(x));}hiveware_enkoder();
//]]>
</script>
```

The rendered HTML from all that coding looks like this:

```
Test your link: elvisNO@SPAMmcgrawhill.com
```

Server-side encoding provides you with the ability to, or not to, obfuscate the display e-mail address on the page, while still protecting your address with script-wrapped encoding on the backend. See the sidebar "Server-side E-mail Encoding" for more information on server-side encoding.

Server-side E-mail Encoding

Some fine spamfighters out there have taken it upon themselves to keep the anti-spam side of the spam war one step ahead. Here are a few sites that might be of use to you if you wish to implement server-side e-mail address encoding on your web site:

- *http://jamesthornton.com/software/redirect-mailto.html* James Thorton has several script examples on his web site, including one in Perl, OpenACS, Cold Fusion, PHP, and ASP scripting languages.
- *http://www.hiveware.com/enkoder_form.php* We used the encoding application on the Hiveware web site to generate the JavaScript example in this chapter. Go to this link and input an e-mail address to see more of this example. Hiveware also distributes a Macintosh OS X application that does the same thing (called the Hiveware Enkoder).
- *http://www.tdavisconsulting.com/characterset.html* and *http://www.tdavisconsulting.com/spam.html* Tony Davis maintains a web site with a partial decimal character set (the first link) and a simple e-mail link encoding form that does not wrap the encoded link in a script (the second link).

And if you doubted for a second that the spam war was an escalating arms race, we offer the following web link. It's a handy e-mail address encoder that e-mails you the encoded address and a newsletter of "the weekly Internet and e-commerce news." You'll find it at *http://www.siteup.com/encoder.html*. *Do not put your real e-mail address in this form!* Though we cannot claim for certain that this is a spammer's site, the products offered are of that "dual-use" variety that tends to set off our spam warning bells.

Nonalphabetic E-mail Addresses The final method for protecting your e-mail address is meant to thwart the brute-force dictionary spammers and is definitely an extreme measure. Although most e-mail accounts are derived from a person's name to make the address easy to remember, some organizations and individuals are opting to change their addresses to nonsensical letter combinations or even numbers. Older netizens may remember that the online service CompuServe did this with usernames from its inception. While this may prevent legitimate senders from reaching you, in this era of smart address books and e-mail aliasing, one extreme measure may significantly reduce the amount of spam that ever touches your network.

Extreme E-mail Methods

Several somewhat extreme but common sense methods are available for protecting your e-mail address. The first is *never* to give out you address to anyone from which you do not want to receive e-mail. This advice is difficult to follow, and it does not necessarily protect your address from brute-force spamming, but it is effective, at least for a while. Alternatively, you may maintain a "pre-email" presence on one of the many free e-mail services available through such places as Yahoo! and Hotmail. These throwaway accounts can either be used as masking accounts, whereby you give out that address freely until a sender becomes "trusted," and then give out your real address, or simply used as a valid e-mail address for web and UseNet posting, registrations for web services, and the like.

Challenge/Response: The Next Weird Thing

One of the methods discussed seriously in Internet mail development circles is the use of a challenge/response system for authenticating e-mail. In Chapters 3 and 10, we discussed a product that incorporates this system. While this system can be exceedingly cumbersome to new senders, a challenge/response mail system (or some variation on the theme) may become the best and only solution to the growing spam problem, at least until the e-mail system as a whole is re-architected.

The Problem It Solves

We've spoken about how broken the e-mail system is today. Anyone can send anyone anything by simply obtaining (or guessing) someone's e-mail address. While in a snail-mail world, this is rarely dangerous and usually just annoying, the "free and open" e-mail system has cost organizations hundreds of millions of dollars in lost time and equipment to spam, viruses, worms, and other such attacks. To close this loop, the challenge/response system was developed.

Why Challenge/Response?

Going on the notion that spam comes from spammers and good e-mail only comes from legitimate senders, the challenge/response system requires initial interaction by an unknown sender with an automated system before the sender's e-mail is "verified." The simplified process works similar to this example:

1. An old school chum, Jerry, sends a mail to Tom at his e-mail address tom@yourorg.tld.
2. Tom's e-mail server receives the mail, compares the From field of the message to a list of known-good senders (essentially a whitelist).

3. Jerry's e-mail address does not appear in the whitelist, so the system sends an automated reply to Jerry directing him to a web site.
4. Jerry receives the message, goes to the web site, and has to validate that he is an actual human and not a Gigantic SpamBot Machine of Doom. Usually this entails looking at an image of numbers or a word and typing what he sees into an entry field.
5. Once confirmed, Jerry's message goes on to Tom's Inbox and Jerry is added to the whitelist.

If this were a spammer, the reply message from the server would either never register or drift off to whatever bit-bucket the spammer claims as a real e-mail address. Messages that aren't replied to are either dumped, the e-mail addresses added to a blacklist, or both. With this system, you are almost guaranteed to never receive bulk unsolicited spam—although there is one glaring hole and a few annoyances with challenge/response.

The first glaring hole is the spoofed known-good address in the From field. Often, spammers put a real address in the From field, and quite often it's a known good address. Either they spoof another address internal to your mail system (root@yourorg.com) or they actually put *your own* e-mail address in the From field. In these cases (without further configuration options), challenge/response fails. Most often, these systems are configured to whitelist all internal addresses (and possibly whole domains), as well as domains and addresses from common vendors and mail or web services (such as yahoo.com, msn.com, and others). Though most can be configured however the systems administrator sees fit, this adds to the system's overhead, considerably.

The minor annoyances are numerous. First of all, you're adding complication to a process that has been easy for several years (decades, for some of us). You enter someone's e-mail address into the To field, add a subject, type a message, and off the mail goes to its destination. Add to that the need to verify yourself, and some people won't bother (especially sales prospects, customers coming from unknown addresses, and the like). Second, as all complex systems have their hiccups, challenge/response adds another failure point to an e-mail system that is far from bug-free. Imagine a favored customer receiving hundreds of verification messages because your challenge/response system experienced a glitch. It's not something that would happen often, but is definitely a consideration. Likewise, imagine two companies that have implemented such a system trying to communicate with each other for the first time. Tom at The Weird Company sends mail to Jerry at The Happy Company. Happy's mail server sends a challenge back to Weird's that responds with a challenge message. Though most of the systems we've seen automatically add outgoing To addresses to the whitelist, this could be a possibility. Finally, an issue that has not seen a valid solution: accessibility for the disabled. Many programs for the sight-impaired user read text aloud to the user but do not interpret images. If a sight-impaired sender receives a challenge message to a sent mail, that person wouldn't be able to be verified as a legitimate sender without assistance.

While challenge/response shows some promise for a robust system, frankly Internet users aren't ready for it yet. The main problem is that most users don't understand the overall problems that spam presents to the e-mail world. Sure, they receive spam to their

inboxes and it's annoying, but they don't see the big picture of billions of unwanted messages soaking up resources worldwide (and costing lots of money to control). For users on a well-developed filtered network, such as SpamBayes or SpamAssassin, they may see only a fraction of the spam that's sent to them, further adding to the perception that spam is not a problem for them and that challenge/response is a draconian measure.

Future Spam-Fighting

To address the problem of the open mail system, engineers are hard at work to develop a fully verifiable, trusted system that not only legitimizes incoming senders, but also makes bulk spammers a thing of the past. Though this seems pie-in-the-sky to those of us that are in the trenches every day tweaking filter rules, tracking down and reporting spammers, the future of spam-fighting is trusted authentication. We have high hopes for IPv6 and other efforts that fundamentally change the way the underlying network operates. A common refrain with all of them is security, first and foremost.

KEEPING YOUR OWN HOUSE CLEAN

While the spam-fighting tools presented in this book are powerful and feature-rich, the first step to fighting spam is ensuring that you are not adding to the problem. In this section we cover the four simple rules for preventing your resources from being used by spammers: closing your open relays, hardening all hosts and servers, restricting access, and sniffing out spyware.

Open Relays

The main problem to the ongoing spam-fighting campaign is the open e-mail relay. In earlier chapters, we discussed DNSBLs that track and block mail from open relays, but how do you close your mail exchanger to spammers? In this section, we discuss how to secure mail relaying for Sendmail 8.12 and Microsoft Exchange 2000. Both of these mail servers deny relaying by default; however, certain configuration options allow some flexibility for mail exchange that may be required in your environment.

Sendmail

We chose Sendmail 8.12 for this chapter because this release was the first to deny relaying by default. All versions since 8.9 also deny relaying out-of-the-box. Although previous releases could be hacked to deny relays, the process was cumbersome and a bit arcane. For more information on Sendmail versions and anti-spam features, refer to *http://www.sendmail.org*.

Sendmail added several relaying features to its configuration files and rulesets to allow a mail administrator very specific or very general relaying practices. We cover a generic setup here, but for complex setups, such as for an ISP or other site where multiple offsite or off-domain relaying occurs, consult the sendmail web site or documentation.

With sendmail, you can relay an entire domain, relay based on mail exchanger records, relay messages based on the local or mail From field or based on an access-approved database that you configure. We cover the access-approved database in this chapter.

CAUTION Note that implementing custom anti-relaying rules could tie up resources both on your network and mail server as well as others. Many of these rules rely on domain or user lookups and may create additional communications traffic between mail servers, DNS servers, firewalls, and the like. Be sure you understand your own configuration and the configurations of other mail servers before you implement these types of rules.

For our purposes, we're going to stick with the blanket default deny relay feature for all networks outside ours. To micro-manage the senders, hosts, domains, and networks that we allow or deny to relay through our exchanger, we used the access_db feature. Though the access_db is akin to a standard white/blacklist feature available on most anti-spam tools, the importance of sendmail's implementation is that messages are accepted or rejected *before* the mail exchanger processes the mail. Sendmail checks the information provided by the sender during the connection session, applies its access database rules to the envelope and connection information, and then relays, denies, or otherwise dispositions the mail as directed by the access database.

Setting Up Your Access Database To set up an access database for sendmail, you must create a flat text file in your favorite editor listing the users, hosts, e-mail addresses, domains, and/or networks you wish to check and the actions you wish the access_db feature to take when it finds a match. Then you must use makemap to create a database map used by the keyed map lookups, essentially allowing sendmail to interpret the data in the database. Keep in mind that, by default, all users, hosts, domains, and networks outside your own are denied relay, unless you've implemented other features or configuration options.

First create the text file of triggers or keys and actions. The format for each line is *KEY<whitespace>ACTION*. We've created an example file here to illustrate the formatting and permutations of the various options and actions:

```
elvis@bigcheeseman.tld RELAY
fatheadspammer.tld REJECT
niceguy.fatheadspammer.tld OK
IPv6:4:5:6:7:8:9:10 OK
192.168.200 DISCARD
192.168.150.22 ERROR:"550 Get out of here you UBERSPAMMER"
```

In this example, the following occurs:

- Any mail from elvis@bigcheeseman.tld is automatically relayed.
- Any mail from the fatheadspammer.tld domain is rejected and sendmail replies to the sender with a general-purpose message.

- Any mail from the host niceguy.fatheadspammer.tld is deemed okay, even if the host is rejected by other rules in the running ruleset. Thus, this rule excepts the previous rule to reject all mail from fatheadspammer.tld.
- All messages from the IPv6 network 4:5:6:7:8:9:10:* are OK.
- All e-mail traffic for the network 192.168.200.* is not only denied relay, it's discarded without a reply.
- Mail from the host 192.168.150.22 is denied relay, and a reply with the indicated message is sent.

Once you've created your access database, you must create a map file using `makemap`. Using our example, this command creates the map_access file in /etc/mail/:

```
makemap hash /etc/mail/access < /etc/mail/access
```

When your access_db file is ready, you're next step is to configure sendmail to use it.

Configuring Sendmail To set up the access_db feature, edit the sendmail.mc file (or the .mc file for your sendmail implementation) and add the following line before the MAILER section:

```
FEATURE(`access_db', `has -T<TMPF> /etc/mail/access_map')
```

Note that the single quote before `access_db` is the grave accent or backward apostrophe (`) located in the upper-left corner of your keyboard (it shares the tilde key).

Once this line is added, make your sendmail.cf file by typing:

```
# m4 sendmail.mc > sendmail.cf
```

If this command is successful, you'll be returned to the command prompt with no messages. Restart sendmail using the new config file and you're off and running. For more information about using m4 and other sendmail features, check out *http://www.sendmail.org*.

Microsoft Exchange

Microsoft Exchange 5.0 and earlier versions have no provision for preventing unauthorized relay of e-mail through the SMTP gateway. Starting with version 5.5, Exchange allows you to allow or deny networks and hosts (by domain name or IP) to relay mail through the gateway. We cover Exchange 2000 in this section, but these methods also work for Exchange 2003. For more information about securing mail relays on Microsoft Exchange, refer to the Microsoft TechNet site at *http://www.microsoft.com/technet/*.

NOTE Though we don't cover Exchange 5.5 here, an excellent article giving step-by-step instructions for configuring and testing Exchange mail relays by Joseph Neubauer is on the Windows Network web site at *http://www.winnetmag.com/MicrosoftExchangeOutlook/Article/ArticleID/7696/MicrosoftExchangeOutlook_7696.html*.

Securing Mail Relays on Microsoft Exchange 2000 Exchange 2000 allows for a flexible SMTP mail relaying configuration, accepting or denying e-mail based on IP address, network, and/or domain name. Authenticated users can override these restrictions, reducing the effects of a misconfiguration.

To access the Relay Restrictions configuration, open the Exchange System Manager and navigate to your SMTP virtual server in the Administrative Groups/*AdminGroup*, where *AdminGroup* is your administrative group. Right-click the virtual server and choose Properties.

Under the Access tab, you may allow connections to the SMTP virtual server by clicking the Connections button or restrict connections to the mail server by clicking the Relaying button.

The Relay Restrictions window allows you to allow or deny relay access to networks or hosts by IP address or domain name. Simply click the Only The List Below radio button, and then click the Add button to add allowed hosts. Click the Allow All Computers That Successfully Authenticate radio button to ensure that all known-good hosts on your network are allowed relay access, regardless of the restrictions configured.

Securing Your Resources

Any computer connected to the Internet is a potential spam target, and not just on the receiving end. Even if you've secured your open e-mail relays, individual hosts on your network pose an inviting and free resource for potential spammers. With the advanced payloads currently carried by virus-spam combinations and spyware, it would not take much innovation to embed a web or mail server kit, allowing the spammer to serve up illicit web pages or exchange a flood of spam bound for victims elsewhere. In this section we discuss three areas to consider when securing your resources from spammer exploitation.

Hardening Your Hosts

Individual computers present an inviting target for computer criminals, especially spammers. Not only do they offer ready resources to further the criminal's processing needs, they also present a wealth of information about the user, the organization, and other, more attractive resources, on the organization's network. All of this information represents real value to the spammer. In the following sections, we present a few common sense methods to prevent or deter spammers from using your host computers as spam machines.

Secure A host computer runs a myriad of programs and services, each with its own associated vulnerabilities. Chief among these is the standard operating system services and kernel. While we leave debates about the most secure operating system to other, more flammable forums, the following common sense hardening methods should apply to any computer running any operating system.

- **Disable unneeded services** All operating systems and many applications run services in the background that the user never sees. While many of them are

legitimate, such as those controlling local sharing protocols and the like, most are unneeded and often insecure. Review the OS configuration of your host computers and disable or remove unneeded services. This includes any and all "server" services, such as personal web servers, file transfer services, and any service that could be available to the Internet.

- **Control application installation** Users want to install applications. Often there is a productivity need to have an application, but sometimes the user installs an application on his local host to test something he's working on, to evaluate the program, or just for fun. Create definitive policies concerning installation of applications that may allow outside users to exploit vulnerabilities. Back up these policies with access control tools on the host itself and through your network security posture.
- **Scan for open ports** Every network available service requires a port to facilitate communication between the server application and the client connecting to the service. For example, a web server commonly listens for web browser connections on port 80 and 443 (HTTPS.) Individual client computers should rarely (if ever) need to run any application that requires a listening port for communications (exceptions to this rule include Microsoft networking). All computers that host such services should be secured in your network DMZ. Use port-scanning tools such as Nmap to check your host computers regularly for unauthorized listening services.
- **Store data securely** Data is the lifeblood of anyone using a computer, whether it's pictures of your grandchildren or a report for work. Your personal and organizational data has value to you and to potential intruders, including spammers. While spammers rarely go to the trouble of hacking into systems for e-mail addresses, virus trojan payloads, spyware, and the like are specifically created to shuttle your data to criminals that use it for their own ends. As a best-practices policy, store all data on a file server within a secure network. In this manner, when a trojan or similar exploit gains access to your host, there is nothing to steal. If storing data on a server is not possible, consider encrypting all data or sensitive data on your hard drive, accessible only by password. While this can be a cumbersome process, it is the best way to thwart a potential breach. Also in support of this objective of availability, keep regular backups. Identify what data is important, back it up to archive media and circulate that to an offsite facility in the event of a catastrophe. The old saw "no backup, no restore" is now more critical than ever.
- **Keep a base configuration** Finally, when you've built a secure installation of an operating system, install this base configuration to all hosts on your network. You can accomplish this in many ways, from step-by-step installation guides to a single host operating system image (using Ghost or DriveImage or similar utilities) used on all host computers.

- **Remove any programs you aren't using** Especially sample CGI programs that come with web servers. The sample form mail script that came with your web server might include a bug that allows spammers to use your web server to send their mail.

Monitor As any security expert will tell you, you can't protect anything if you don't know where it is, what it's supposed to do, and what it's actually doing. Monitoring network and host activity is a broad subject, quite beyond the scope of this spam book, but these few simple guidelines (and a lot more reading) can give you the edge on spammers.

- **Intrusion detection/prevention systems** Many network and host intrusion detection/prevention systems are available to combat not only general attacks to your resources, but also specific threats, such as those used by spammers. While every individual host does not require its own monitoring, consider a network intrusion system that constantly monitors incoming and outgoing traffic. All of these systems provide configuration for alerts based on anomalous outgoing network traffic. If you know that all of your mail services should originate from your DMZ, a network intrusion detection system could be your first line of reporting when a host has been compromised and has an illicit mail server running on it. Host-based intrusion detection (usually in concert with a centralized reporting system) can alert you to unauthorized access, services, and other compromises, usually in real-time. Snort is an example of a network intrusion detection system, and Tripwire is an example of a host-based intrusion detection system.
- **Firewalls** Network and host firewalls prevent and allow network traffic to and from your networks and hosts. A proper firewall configuration, along with definitive network segmenting, can prevent illicit services, such as a spammer's rootkit, from turning your resources into free processing for computer criminals.
- **The human factor** While often disregarded, the person who sits in front of the computer for eight (or more!) hours a day is usually the best indicator that all is not well. Implement an organization-wide incident response procedure and advertise it to all users. When a user knows that her computer is not "acting right," she can do the monitoring work for you.

Respond The most successful exploits are those that are allowed to happen, even with all the best prevention and monitoring procedures and tools in place. Develop and use a rigorous response procedure for all incidents on your computing resources, including automated responses available with many information security tools and a manual procedure for stopping exploits in their tracks. There is no quicker way to find your network on a real-time black hole list than to allow a spammer a few hours of intimate control of your information resources.

SPYWARE: ANOTHER SPAM PATHWAY

Spyware consists of programs that are installed on a computer without the user's explicit permission that monitor and report on activity or other data on the computer. Spyware is also known as adware, trojan horses, and *$%#$$#@* web pop-ups. While many spyware programs are truly illicit, several legitimate programs, such as Microsoft Media Player, Real Media Player, and others, also collect information on user actions (DVD movies watched, MP3 music files opened, and so on) and surreptitiously report this information to a remote server. While this may be an innocuous exchange of information for the benefit of the user, it is still network communication that the user did not specifically allow and the vendor did not explicitly disclose (or perhaps it did, hidden in the fine print of the license agreement you clicked past). In most security profiles, this is a trojan horse program. In this section we cover the illicit version of these programs, their operation, potential exploits, and methods for preventing them from contributing to the spam problem.

Pop-ups: The New Spam

In the early days of the web, considerable energy was paid to developing "push" technology. The power of the web at that time was that a user saw only information that was specifically clicked. Those golden days of the web are far behind us, of course. Now when you open a web site, even one as legitimate as CNN, you can expect at least one web pop-up to appear. The best of them appear behind your main web browser. The nightmare pop-ups continue ad infinitum and may just install a handy trojan on your system so that you can receive these "important messages" even when you aren't browsing the web.

Several programs exist to stop pop-ups of all kinds and range from free host-based squashers to quite expensive enterprise-level applications. Most modern browsers have features that can be turned on to squelch those noisy pop-ups. While most virus-scanning programs do not stop "adware" programs such as Gator from installing on your system, you can configure your web browser to deny downloading automatically the Java-based cookies these trojans use to install on your system.

Another form of pop-up spam has recently been plaguing Windows users who connect their systems directly to the Internet. This one uses the Windows Messenger Service that's a standard part of Windows Networking. (Note that this is not the same thing as Windows Messenger or MSN Messenger instant messaging clients.) On most Windows networks, this text-based pop-up service is of benefit because it lets you know that your print job is done or that backups have been completed, or it lets a systems administrator send a network-wide notice (for example, that a server is about to go offline). Unfortunately, it also requires no authentication and is anonymous over the Internet, so spammers have started using it to get their messages across.

To prevent abuse of the Messenger Service, you can either use a firewall that blocks the port used by Windows Messenger Service (port 135), disable the service completely,

or do both. We recommend both, unless you're on a corporate network and disabling the service would impact the support of your network in a negative way, in which case you should block the port on your perimeter. In fact, given the vulnerabilities exploited by the Blaster worm and others, we recommend that you block all Windows Networking ports (ports 135 through 139) at your perimeter. You shouldn't need to use Windows Networking over the Internet. If you do, use a virtual private network (VPN) instead.

Disabling the Messenger Service in Windows 2000 and Windows XP

You can learn how to disable the Messenger service in Windows 2000 by consulting the following web site: *http://www.microsoft.com/windows2000/techinfo/administration/communications/msgrspam.asp.*

And in Windows XP, consult *http://www.microsoft.com/windowsxp/pro/using/howto/communicate/stopspam.asp*.

If you're running Windows 98 or ME, you're out of luck: The Messenger Service cannot be disabled on these versions of Windows.

Using Internet Connection Firewall to Stop Messenger Service Spam

Windows XP comes with Internet Connection Firewall (ICF), which can be used to prevent incoming Messenger Service spam or attacks. Starting with Windows XP Service Pack 1, ICF blocks all incoming unicast, multicast, and UDP traffic on specific network connections that have ICF turned on.

Various implementations and considerations are covered in the following article on the Microsoft support web site: *http://support.microsoft.com/?kbid=330904.*

True Spyware

Actual spyware is used as a marketing tool for those Internet leech companies that can't figure out a more legitimate way to make a buck. While you surf the Web, these programs monitor the sites you visit and report back to a central server what pages you viewed and how long you spent there. Many, like Gator, also pop up mini-advertisements related to the sites you visit, as you visit them. While no confirmed spyware programs actually pick up your e-mail address, once such a program is on your system, it doesn't take a huge jump in functionality to snag both your address and all those others in your address book. Most anti–pop-up programs disable or remove spyware, as well, and most anti-virus programs detect and quarantine or delete true trojan horse programs (spyware's more evil twin). If you have become infected with these programs, the only other solution is to wipe your hard drive clean and reinstall from scratch.

Anti-Spyware Tools

Anti-spyware tools scan your browser cache, memory, system registry, hard drives, and removable drives for various spyware. Most of these programs also block web and spyware pop-up adds that we all love so much. Most also function similar to anti-virus packages that should be familiar, operating either in a passive/manual mode, whereby you initiate a scan on your system, or in "active" mode, where the software actually

detects the spyware in-flight, stops it, and asks you to decide whether to install it or not. After detecting the spyware components, all anti-spyware systems delete or quarantine the offending programs.

Ad-aware 6.0 from Lavasoft

We cover only one of a sea of anti-spyware systems out there: Ad-aware from Lavasoft. This program is available from the Lavasoft web site for free (noncommercial use), though the anti–pop-up feature is available only if you pay for it. Download the current version of Ad-aware from *http://www.lavasoftusa.com*.

Installing Ad-aware 6.0 Perform these steps for Ad-aware installation success:

1. Double-click aaw6.exe. The Ad-aware 6 Personal window appears.
2. Click Next, read and agree to the license agreement, and click Next, again. The Destination Location window appears.
3. Choose a destination on your hard drive and click the Next button. The Start Installation window appears.
4. Click the Next button, and Ad-aware begins installation.
5. Once installation is done, the Ad-aware main window appears, as shown in Figure 16-1.

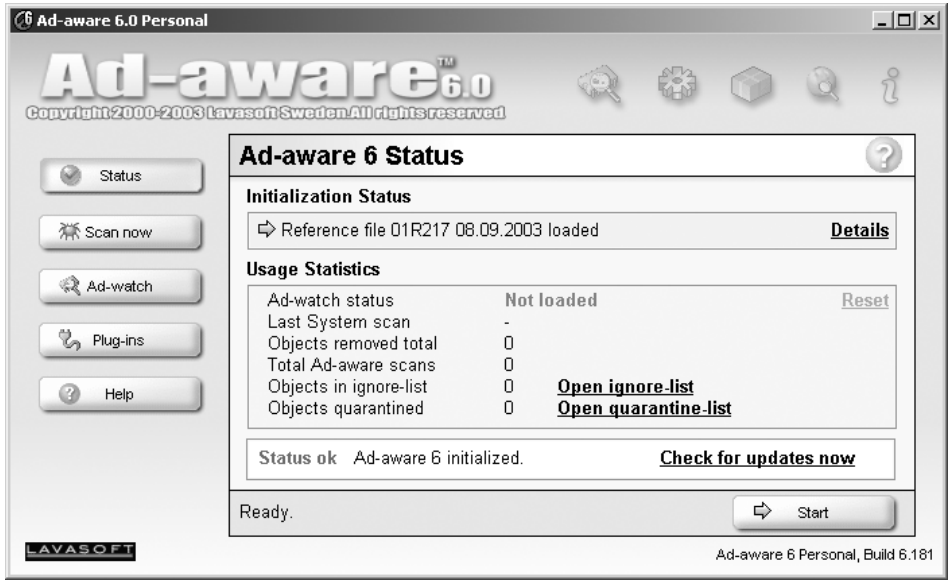
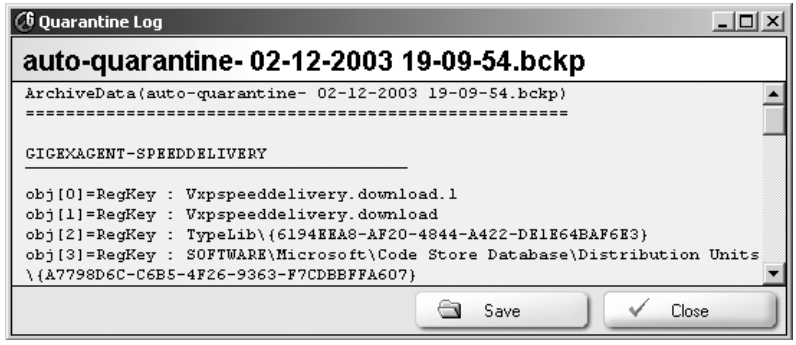


Figure 16-1. The Ad-aware main window

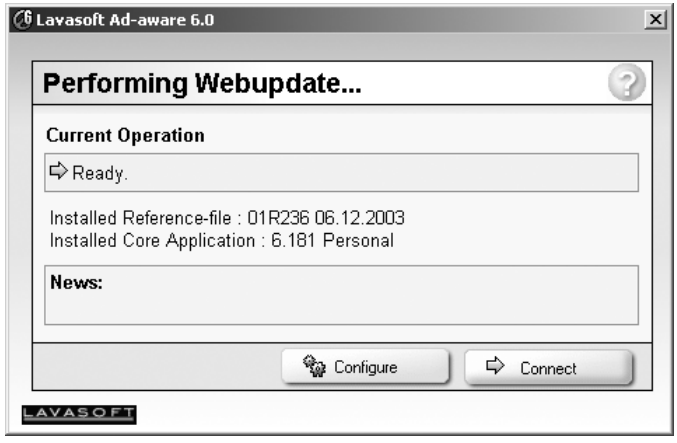
Using Ad-aware Ad-aware's use is pretty straightforward. The main window defaults to a status report, showing you the initialization status and usage statistics, including last system scan, objects removed total, total ad-aware scans, objects in ignore-list, and objects quarantined.

From the main screen, you can view and use the following features:

- **Initialization status details** Click the Details link to view the current reference file loaded (Figure 16-2). The reference file contains the profiles of spyware in the wild that Ad-aware detects and quarantines. This file can be updated from the main screen by clicking the Check For Updates Now link.
- **Open Quarantine List** Clicking this link allows you to view, delete, or restore any quarantined spyware (Figure 16-3). If you click an auto-quarantine file from the Quarantined Objects window, you can view the individual files in the quarantine group, as shown here.



- **Check for Updates Now** Clicking this link allows you to check Lavasoft for updated profiles for Ad-aware, as shown next. Click the Connect button, and Ad-aware alerts you if there are updates available and gives you the option to install them.



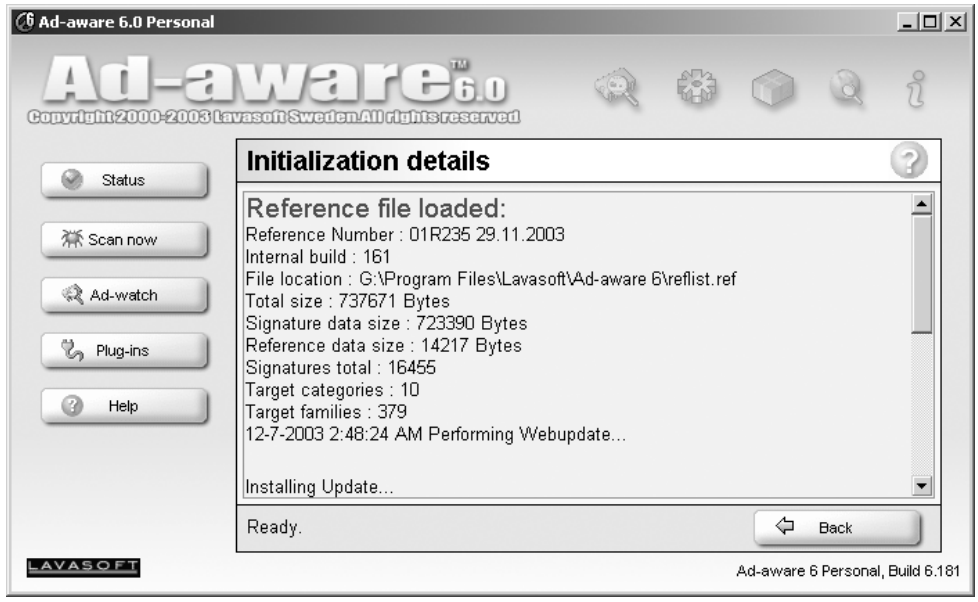


Figure 16-2. Details of the reference files installed on your software

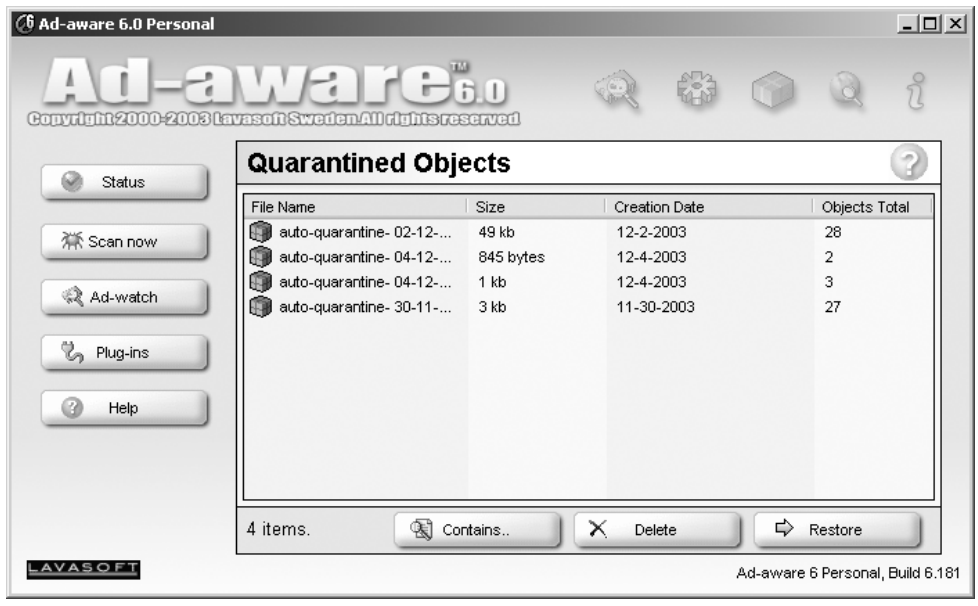


Figure 16-3. View the quarantined items and delete or restore them.

- **Scan Now** This is the meat and potatoes of Ad-aware. Click the Scan Now button and you're presented with a few options. You can perform a smart scan, a custom scan, or select individual files or folders to scan. Click the Next button to start your scan, and you'll see a window showing you what Ad-aware is doing.

Handling Detected Objects Once Ad-aware finishes its scan, you have to tell it what to do with the evil it's detected on your system. The program displays a flashing red spider and the number of objects detected (see Figure 16-4).

Click the Next button to view the objects (see Figure 16-5). Each item lists the vendor, type, category, object, and a brief description if available. To quarantine the objects, place a check mark next to the desired objects and then click the Next button. All items are quarantined.

NOTE When Ad-aware finishes its scan, it emits a horrible noise when it has detected spyware on your system. If you have heart problems or are easily startled, be sure to disable the sound by clicking the gear icon (Settings) in the upper-right corner of the main window, and then click the Tweak settings button and the Misc. Settings tree. Click Play Sound If Scan Produced A Result to change the green checkmark to a red X. Then click the Proceed button.

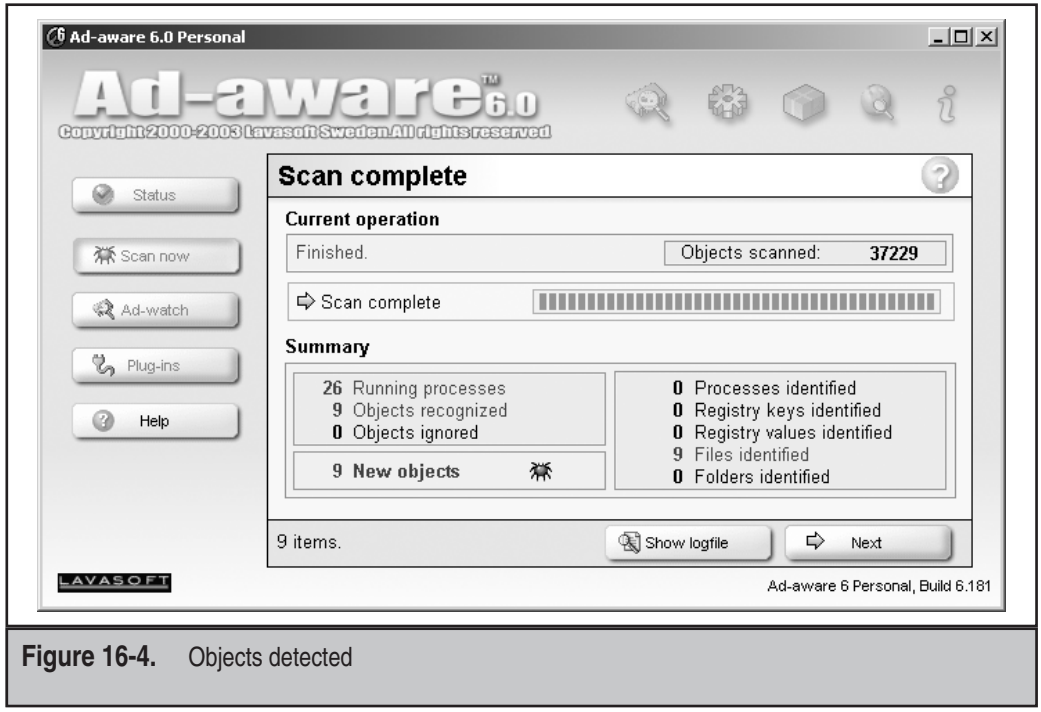


Figure 16-4. Objects detected

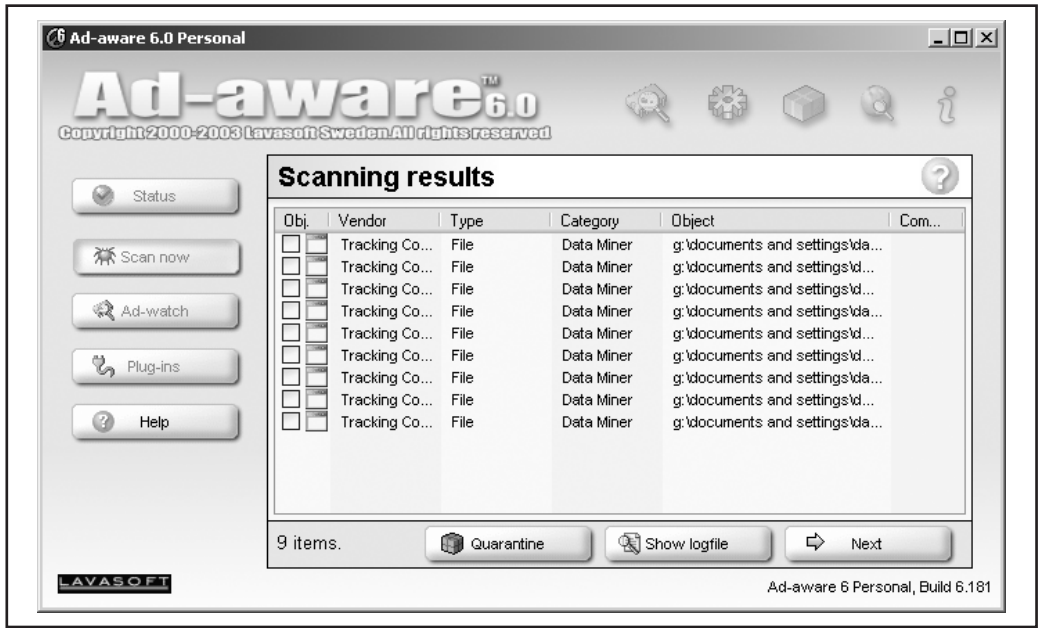


Figure 16-5. Scanning results

SUMMARY

In this chapter, we talked about various strategies, outside of your anti-spam tools, for combating spam before it actually reaches you, including protecting your e-mail addresses, keeping your own mail server resources out of the hands of the spammers, and protecting your hosts and other computers on your network from becoming zombie spam-spewers. Finally, we discussed spyware, its new convergence with spammer tools, and anti-spyware programs that protect you from that threat.