Common Spammer Tricks

PROCESS™ SOFTWARE

As spam filters evolve to become better and better at identifying and discarding spam, they force the spammers to develop new tricks to circumvent the filters. This whitepaper examines some of the most common tricks employed by spammers to sneak messages through today's spam filtering solutions. The majority of these tricks only work on one spam filtering technique. Anti-spam filters that make use of multiple techniques will correctly filter messages that use most of these tricks.

Hiding Text

One of the most obvious ways to sneak a spam message past a filter is to "hide" the part of the message that makes it spam. If the spam filter can't find the offending text, it can't use it to determine if the message is spam or not. Spammers have developed several creative techniques to hide text so a human can find it, but spam filters cannot. The spammers are greatly aided by the widespread support of HTML in most email client software.

Splitting Words

A trick in very common usage is to break up "interesting" words with HTML comments. For example, the HTML:

```
Fr<!-- 63 -->ee mor<!-- adf -->tgage fin<!-- sdf -->anci<!-- e -->ng
```

would be rendered as:

```
Free mortgage financing
```

in an email client. Just ignoring HTML comments isn't enough, though. Most HTML rendering engines ignore any tag they don't recognize, so spammers can make up their own nonsense HTML tags and use them to split words.

JavaScript Messages

A very clever (but rarely used) trick is to place the entire contents of the spam message inside a JavaScript snippet that is activated when the message is opened. When the email client software executes the JavaScript, the text of the message is written out to the message display area. Because JavaScript is usually contained inside an HTML comment block, simple filters that ignore HTML comments will let the message through.

Some spammers go the extra mile and even encode the text that is written out to the message display area. The same JavaScript that writes out the message also decodes it. Encoding the message text lets it sneak by even filters that are smart enough to understand JavaScript.

Since there's almost never a legitimate reason to include JavaScript in an email message, the best defense against this trick is to classify as spam any message that contains JavaScript actions.

Pattern Recognition

The human eye is amazingly talented at pattern recognition, while computers are notoriously bad at it (at least without using an unacceptable amount of system resources). Spammers take advantage

of this by trying to disguise words in such a way that the human eye can still pick them out, while the same words look like nonsense to a computer.

Separating every character in a word with the same non-alphabetic character wreaks havoc with anti-spam filters that don't have the programmed intelligence to realize what's happening. Examples you've probably seen include:

```
V I A G R A       X'A'N'A'X   F*R*E*E     M\O\R\T\G\A\G\E
```

A twist on this is to replace characters in "bad" words with an accented version of the same character. While it looks pretty much the same to the human eye, it's as different as night and day to a computer. Examples include:

```
Vïágrä       Xàȟāx       FŔĘĘ        Mŏrtgãĝė
```

Some spammers go to the other extreme and remove all whitespace from their message and replace it with a common character, similar to the way that old teletype machines used to work. A common example is:

```
HotXgirlsXareXwaitingXtoXhearXfromXyou!
```

A technique that is used on occasion involves substituting numbers for alphabetic characters that have a similar appearance to hide words. This technique is referred to as "l33t 5p3ak" (that's "leet speak").

Dyslexia

Not only is the human eye good at pattern recognition, but it tries to force visual data into a pattern it recognizes. A popular optical illusion demonstrates that as long as the first and last letters of a word are in the correct place, the remaining letters can be in any order and the word can still be recognized. Adding extra letters or using phonetic spellings can also drastically change a word's spelling without making it unreadable. Spammers take advantage of this to create an almost infinite number of spellings for words and phrases that are commonly used in spam messages. Some examples are:

```
FWREE HRBAL VGRAIA SAMPULS
Nekad scoohlgurl photos!!
```

Tiny Nonsense

If a spammer can keep a spam filter from recognizing a "spammy" word for what it really is, a big part of his job is done. Some spam filters perform rudimentary HTML parsing by simply stripping out HTML tags, and then analyzing the text that's left. Spammers exploit those systems by inserting random letters into "spammy" words, but using HTML `<FONT>` tags to make them so small they're unnoticeable to users reading the message.

For example, the HTML code

```
Mo<FONT SIZE="1">sdf</FONT>rtga<FONT SIZE="1">lax</FONT>ge
```

would be rendered as

Mo$_{sdf}$rtga$_{lax}$ge

but it would look like

```
mosdfrtgalaxge
```

to a spam filter that doesn't understand the meaning of the <FONT> tags.

A variation of this trick is to use very small (one pixel wide) images to break up words. The images don't even have to really exist - they're so small that the email client's "broken image" replacement won't be visible.

URL One-Liner

Some spammers have discovered that the best way to hide message text from spam filters is to avoid putting any text at all in the message. These spammers send messages that contain only a brief, generic sentence and a URL. If an email user clicks on the URL, they are taken to a web page where the spammer makes his product pitch.

Because the message only contains a small amount of generic text and a URL, there's virtually nothing a spam filter can use to determine if the message is spam or not. When spammers started using this trick, they would host their sites on a relatively small number of systems. Spam filters could simply block messages that contained a URL referencing one of these systems. But the spammers have increased their level of sophistication, and now they use numerous redirects and domain aliases to disguise the true identity of the systems hosting their sites.

The only way to reliably identify this sort of message as spam is for an anti-spam filter to actually visit the website whose URL appears in the message. Based on the contents of the website, the filter could determine if the message was spam. The problem with this is that it significantly slows down the flow of email. Many corporations, academic institutions, and ISPs aren't going to be willing to purchase the additional email servers required to keep email flowing smoothly.

One Big Image

Some spammers don't want to deal with the hassle of setting up a deliberately confusing system of web sites and redirects to sneak their message past spam filters. As an alternative, they craft messages that consist solely of an image file. Most email clients will display the image, but anti-spam filters will treat the image as an opaque block of binary data. As long as it's contained in the image, the text of the spam message can be as obvious as the spammer wants it to be.

Software that can identify text inside an image has been available for several years, but it requires a large amount of memory and CPU time to work. Building the ability to recognize image text into an anti-spam filter would severely impact the flow of messages because of the load it would place on the

email gateway. Additionally, the use of unusual fonts can confuse the image text recognition to the point where it can't identify the text.

The problem with this trick is that the image makes the email message much larger than a similar message that doesn't contain the image. Spammers don't like this, because it cuts down on the number of messages they can send in a given time period. The more messages they send, the more likely they are to get a response. To solve this problem, the spammers place the image on a website and just include an HTML reference to it in the email message.

Anti-spam vendors may flag this type of message as spam by filtering messages that contain a reference to an image on an external website. This works in almost all cases, but some legitimate e-commerce and webmail sites such as Amazon and Yahoo use this method. Email users tend to become unhappy when they can't receive messages from these sites, so filters can't use this to identify spam.

Encodings

Email wasn't designed to transmit binary attachments, so some low-level trickery is required to accomplish this even for legitimate messages. Usually an encoding scheme such as Base64 is used to turn a binary attachment into plain text characters so email systems can handle it. Many spammers use this method to encode their message text, in the hope that the anti-spam filter on the remote email server isn't smart enough to decode messages before filtering them.

In addition to encoding the message text itself, many spammers go one step further and use HTML entity encodings to hide each individual character of their message. For example, the HTML encoding

```
&#70;&#82;&#69;&#69;
```

would be rendered as

```
FREE
```

in an HTML-enabled email client.

Anti-spam filters have to be smart enough to know about every form of email and HTML encoding that email user agents support, and be able to perform multiple decoding operations on the same message.

ASCII "Art"

Software that generates large letter glyphs out of standard letters has been pre-empted by the spammers. While it's difficult to represent large amounts of text with this trick, it can be used to disguise words that a spam filter might use to determine if a message is spam. The below example looks like a large number of pound signs to most anti-spam filters.

```
######   #####    ######   ######
```

```
#       #    #  #       #
#####   #    #  #####   #####
#       #####   #       #
#       #   #   #       #
#       #    #  ######  ######

#####   #    #   ####    #####   ####    ####
#    #  #    #  #    #     #    #    #  #
#    #  ######  #    #     #    #    #   ####
#####   #    #  #    #     #    #    #       #
#       #    #  #    #     #    #    #  #    #
#       #    #   ####      #     ####    ####
```

Spam filters can be programmed to recognize the trick, as long as nonsense characters are used. Some very creative spammers use actual words to create the letters, increasing the chance that their message will pass through a spam filter unchallenged.

Vertical Horizon

A combination of HTML tables and a fixed-width font lets spammers break a message into what looks like random characters to spam filters that don't thoroughly understand HTML. The words or phrases to be disguised are arranged into table columns, with the first letter of each word in the first column, the second letter of each word in the second column, and so on. A simple example looks like: (the borders are added for clarity - they would be invisible if the message was viewed in an email client)

F	R	E	E			
N	A	U	G	H	T	Y
P	H	O	T	O	S	

If the HTML used to generate the table is stripped out, the text looks like this:

```
FNP RAH EUO EGT HO TS Y
```

The really sneaky part of this trick is that because the nonsense words are created from real words, filters that analyze the ratio of characters to determine if a sentence is random garbage don't work.

See Attached

While it isn't a commonly used trick, some spammers create a Microsoft Office document containing their marketing message, and attach it to their spam messages. Anti-spam filters that don't know how to parse Office documents will let the message by, as long as the rest of the message appears to be innocent. Office documents are substantially larger than plain text, so this technique reduces the speed at which a spammer can send messages.

Bait And Switch

The Multipurpose Internet Mail Extension (MIME) standard is used to break an email message up into several logical parts. For example, a mail message might contain plain text in one part, HTML in another part, and an attachment in yet another part. Some mail clients don't support HTML, so most HTML mail clients send messages with both plain text and HTML parts that contain the same text.

Simple anti-spam filters usually only look at the plain text part of a multi-part email message, while HTML mail clients typically display only the HTML part of the message. Spammers take advantage of this by placing an innocuous message that will pass through an anti-spam filter in the plain text part of the message. At the same time, they place their spam message in the HTML part of the message. Their hope is that the anti-spam filter will only look at the innocent plain text message, while the recipient's HTML mail client will display the decidedly non-innocent HTML message.

To catch this trick, an anti-spam filter has to be able to understand and disassemble MIME-formatted messages. This isn't a trivial task, and most simple filters only examine plain text.

URL Hiding

The vast majority of spam messages include a link to a website where the spam recipient can purchase the advertised product or service. Spam filters can pull the URLs out of an email message and compare them against a known list of spam sites. If an email contains the URL of a spam site, odds are good that the email message is spam and should be filtered. To try to prevent this from happening, spammers use several tricks to hide URLS both from spam filters and from end users.

URL Encoding

The easiest way to hide a URL from a spam filter is to disguise the fact it's a URL (or at least try to disguise the site name in the URL). Most web browsers recognize URLs that contain hexadecimal character representations, as well as too many IP address encodings to list here. Some common encodings used by spammers are:

- "Dotless" IP address encoding: `http://3329460759/`
- Hexadecimal IP address encoding: `http://c6738a17/`
- Hexadecimal URL encoding: `http://%77%77%77.pro%63%45s%73%2ecom`
- HTML entity encoding: `http://www.&#112;ro&#67;ess&#46com`
- Bogus URL-encoded username: `http://840p28453@3329460759/`

Faking It

Some spammers believe that their product or service is so compelling that they can sell it to anyone who visits their site. Initially, they would try to trick people into visiting their site by claiming that the site was something other than what it appeared to be - a link to new book releases on Amazon.com, for example. Recipients of such messages quickly figured out that regardless of the surrounding text, the URL `http://www.jaiiisii.com` didn't lead to Amazon.

The spammers soon switched to using HTML to make the URL look like `http://www.amazon.com`, regardless of the site it actually linked to. Unfortunately for the spammers, most end users know that common HTML email clients display the actual link in the status bar when the mouse is placed over a hyperlink.

Once again, the spammers have risen to the challenge. JavaScript provides the ability to change the contents of the status bar on most web browsers and email clients. The spammers have begun adding JavaScript code to their messages that changes the status bar contents to something benign when the mouse is placed over a link. The more nefarious spammers use tricks like this to fool email recipients into disclosing personal information, such as credit card numbers, to the spammer.

Copy And Paste

Amazing as it may seem, some people actually want to receive spam. These people are the spammers' target audience, and this trick shows that they're willing to do a little work to view a spammer's site. In some email messages the spammers split the URL into two or more sections, and provide instructions for putting it back together in a web browser. An example of one such message is:

```
type www.cnn then the follow URL into your browser's address bar:
.24750.net/content.htm
```

Hash Busting

Checksum and signature matching anti-spam solutions depend on a spammer sending out a large number of messages that are all almost identical. The vendors of such solutions maintain large numbers of "trap" addresses at major ISPs, which they monitor in real-time. When a spam message reaches one of the trap addresses, the vendor quickly writes a rule to catch that message. All of the sites running that vendor's software check for rule updates at frequent intervals, so they start blocking that particular spam message soon after the spammer begins sending it. But if the spammer changes the text of the spam message substantially enough frequently enough, the anti-spam vendors will be forced to identify each variation and write a new rule for it.

If the message text changes often enough, then the spammer will always stay one step ahead of the anti-spam vendor and be virtually guaranteed that the messages will slip by. Below are some methods that spammers use to try to evade checksum and signature matching filters.

Mad-Lib

Long used by schoolchildren for amusement, the art of the mad-lib has been brought to new heights by the spammers. The basic idea behind a mad-lib is to modify a phrase or sentence by replacing one or more words with synonyms. Doing this to the text in an email message keeps checksum and signature-matching anti-spam filters from identifying the message as spam.

When this trick was first developed, spammers would send out several thousand messages that had the same text. Then they would replace a few words here or there with synonyms, and send another several thousand messages. Several pieces of software used by spammers to generate messages (called *spamware*) now automatically perform random substitutions. Some of the best spamware currently available allow the spammer to write several different versions of each paragraph of the spam message. When the spamware generates messages, it randomly chooses which version of each paragraph to use. The more advanced spamware even has a built-in thesaurus and randomly modifies certain words.

For example, a spammer sends a spam message that consists of four paragraphs. If the spammer writes five different versions of each, there are 625 unique combinations of paragraphs. If the spamware randomly swaps a couple words from each paragraph for synonyms, there are tens of millions of unique spam messages that can be generated, each of which essentially says the same thing.

Vendors of checksum and signature-matching filters are always modifying their matching algorithms to try to account for small changes in message text, but substantial changes like modifying an entire paragraph render them useless. Bayesian and heuristic filtering, which don't depend on large numbers of spam messages being virtually identical, are immune to this trick.

Entropy

A simpler variation of the mad-lib trick is to insert strings of randomly generated characters throughout the message. Most spamware packages in common usage can be set to do this automatically by the spammer. Because these character strings are randomly generated, they can vary widely from message to message. The resulting difference in messages prevents some signature checking systems from identifying the message as spam. If a Bayesian system doesn't take the effort to prevent random tokens from being stored in its token database, the databases can quickly become filled with random data. As the databases grow, performance will degrade.

Luckily, random character sequences can be easily identified by anti-spam filters with the programmed intelligence to do so. The random character strings inserted by spamware are typically much longer than actual words and have a very high ratio of consonants to vowels.

Bayesian Sneaking & Poisoning

Bayesian filters are often touted as the ultimate anti-spam solution: they "learn" spammer's tricks quickly, and are highly accurate. Of course, that makes them a prime target for spammers. There are two main methods to get around a Bayesian filter: write your spam message so it doesn't contain any words that are normally used in spam messages, or "poison" the Bayesian filter's database. The two most common methods used to poison a Bayesian database are filling the database full of nonsense tokens, and getting users to identify messages that contain non-spam words as spam.

Both methods involve inserting large blocks of either random characters or words found in a dictionary into a spam message. When the Bayesian filter is "taught" that the message is spam, it adds all of the words to its spam token database. If the spam message contains a large number of dictionary words, any future messages that contain them are more likely to be identified as spam. So far, this technique has been ineffective for two reasons:

- While there are over a 100,000 words in most English-language dictionaries, only a couple thousand are in common usage. Most of the random dictionary words that appear in spam messages are words that never appear in most people's email messages. There aren't very many email users who receive messages containing the words "recondite" or "variegated".
- Words that appear in non-spam messages will already have a sufficiently high non-spam value that their appearance in the occasional spam message won't compromise the filter's accuracy. The only way for spammers to force these words to have a high spam value is to know which words appear frequently in your legitimate email, and send a large number of spam messages that contain those words.

Including a large number of words that are composed entirely of random characters in spam messages can cause Bayesian databases to grow quickly. As the token databases become larger, they have a proportional impact on the filter's performance. If the Bayesian filter isn't smart enough to purge nonsense words from its databases on a regular basis, the databases will eventually consume so many system resources that email can't be delivered in a timely fashion.

Even though the following tricks to poison Bayesian databases have little (if any) success, they still appear very frequently in spam messages.

Word Salad

Many spammers believe that if they put enough non-spam text in their spam message as a decoy, it will make it past anti-spam filters. Usually, this takes the form of several lines of random dictionary words at the end of the message:

```
First time newbie amateur model strutting her stuff
http://ky3.hopano384.net/x502.html?8eqyef0j

coproduct cladophora aerosol countryman bedim
```

More advanced spammers use lines of random words arranged so they appear to be valid sentences:

```
Sound is drop. Line whether soft oxygen. Cross burn make
suggest, minute. Cover part reason. Why fresh wire. Notice, are
fact find hold. Move such light city, feet. Near hot, pick other
busy, book.
```

White-Out

To hide the non-spam text used to sneak the message by the filter from the recipient of the message, the spammer uses HTML font tags to make it the same color as the background of the message. The spam text itself is set to display in a color that contrasts with the message background.

If a filter is smart enough to identify this trick, it can immediately classify as spam a message that contains a large amount of text that is the same color as the background. Overly simplistic Bayesian filters can potentially be poisoned by this method, and signature-matching filters can be fooled if the text is randomly changed between messages.

Several advanced versions of this technique are currently being used by spammers in an attempt to get around smart filters that recognize the original trick. One variation is to display the non-spam text in a color that is very close to the background color, but not exactly the same. This tends to work very well, since most modern computer screens can display upwards of 16 million colors while the human eye can only distinguish approximately 10,000 colors. To catch this variation, filters have to contain a fuzzy logic model that can mimic the human eye's response to color.

Another popular variation is to display the non-spam text in the same color as the spam text, but in a miniscule font size. Several thousand words of non-spam text can be condensed into a couple inches of screen space if they're displayed in a 2 point font size.

HTML Abuse

Various HTML tags can be abused to hide text from a user while making sure a Bayesian filter sees it. Some common examples are:

- Placing random words in the `<TITLE>` tag of an HTML message. Most email clients don't display the contents of this tag.
- Placing a paragraph of random text or text from a news site into the `VALUE` attribute of a hidden HTML form field.
- Using a `<COMMENT>` tag to hide large blocks of random text.

Social Engineering

The best spam filter for any email account is the person who owns that account. Even as smart as most humans are, spammers try to trick them into reading spam messages that made it past the filter or into not reporting the spam to the proper authorities.

Faking Legitimacy

Many spammers try to convince email recipients that their actions are legal and protected by law. Huge legal disclaimers, usually citing non-existent or non-applicable laws, appear at the bottom of the message. While this won't allow the messages to sneak through spam filters (in fact, a lot of filters specifically look for these "legal" statements), it may make the recipients of the spam uneasy about reporting the spam to their system administrator or reporting the spammer to the proper authority.

Spammers depend heavily on their messages not being handed over to a system administrator, anti-spam vendor, or government authority. As soon as the message is handed over, it's much more likely that rules will be put in place to trap the message in the future and that the spammer may even be prosecuted.

Sender Unknown

As almost everyone who receives spam has noticed by now, you can't trust the name or email address that appears in the "From" header. If you see a message in your inbox that came from an address like `bigtool4u@hotmail.com`, you're not nearly as likely to read it as you are a message that came from somebody who works at your company or organization. Spammers know that, and they take advantage of it by correlating addresses from the same domain. Because of the way the email transmission protocols were designed, there's no good way for an email server to verify that a message is really being sent by the person whose email address appears in the "From" header.

Conclusion

As anti-spam filters evolve, spammers are keeping pace by developing sophisticated tricks in an attempt to avoid having their messages filtered. While not a trivial task, it is possible for spammers to construct a message that can bypass an anti-spam filter that only uses one filtering method. An anti-spam filter that uses multiple filtering methods increases the difficulty of sneaking messages through by several orders of magnitude.

PreciseMail Anti-Spam Gateway eliminates unwanted email at the Internet gateway or mail server without filtering legitimate email messages, producing a large potential cost savings to organizations of all sizes. Its highly accurate filtering engine uses multiple filtering technologies, making it difficult for spammers to circumvent. An extensive heuristic filtering system is used in parallel with a Bayesian artificial intelligence engine to identify spam. This combination of technologies has proven more effective at eliminating spam than some other spam-filtering techniques, such as signature matching or challenge/response.

About PreciseMail Anti-Spam Gateway

PreciseMail Anti-Spam Gateway is an enterprise software solution that eliminates spam, phishing and virus threats at the Internet gateway or mail server. It has a proven 98% spam detection accuracy rate out-of-the-box without filtering legitimate messages. PreciseMail Anti-Spam Gateway has a highly sophisticated filtering engine is based on a combination of proven heuristic, DNS blacklisting, and Bayesian artificial intelligence technologies, which automatically learn how to separate spam messages from legitimate email. As a result, PreciseMail Anti-Spam Gateway can determine whether email is spam instead of passively reacting to known spammers by creating rules that block them after a spam attack occurs.

About Process Software

Process Software has been a premier supplier of communications software solutions to mission critical environments for twenty years. We were early innovators of email software and anti-spam technology. Process Software has a proven track record of success with thousands of customers, including many Global 2000 and Fortune 1000 companies.

U.S.A.: (800) 722-7770 • International: (508 879-6994 • Fax: (508) 879-0042
E-mail: info@process.com • Web: http://www.process.com/