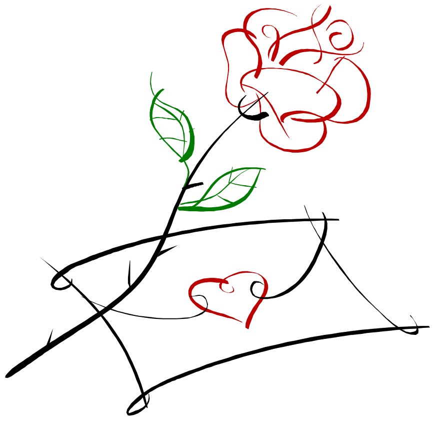
Spam – The Evolution



By Aseem Jakhar

null@null.co.in
aseemjakhar (aa tt) **gmail** d_o_t **co**m

Introduction

What is Spam?

No, it’s not “**SP**iced h**AM**” - the Hormel product. Now to define spam in a single line - “Spam in fact has no standard definition”☺. It differs on an individual basis. A message could be spam for one individual and not for the other for someone may not be a spam for someone else. There are different names by which spam messages are known, UBE – Unsolicited Bulk Email, UCE – Unsolicited Commercial email, Junk mails etc.

In short a message that is sent to a recipient without his consent and sent in bulk with commercial/harmful intent can safely be assumed to be spam.

History

There have been many spamming incidents even before the term spam was coined. The affected communities were newsgroups, chat rooms, e-mail users etc. On USENET for example there were spam posts like *“Green card lottery”* (rise of commercial spamming), *“MAKE.MONEY.FAST”* and the ARMM incident where a bug in the program, developed to cancel abusive posts, caused it to recursively send posts to the *news.admin.policy* newsgroup. The term *Spam* is believed to have been derived from “Monty Python's Flying Circus” show.

Nuances of Spam

Causing annoyance to the reader is not the only side effect. Spamming actually causes financial loss to organizations. The major issues are:

1. *Bandwidth Overload:* A lot of bandwidth is consumed when messages are sent or received in bulk.
2. *Storage overload:* Even though a message is detected as spam it is not deleted or rejected automatically. It consumes storage on the server till the time the actual recipient takes some action on it.
3. *Loss of End user productivity:* People take certain amount of time in reading messages or taking certain action on them. Top it up with bulk messages in the inbox/spam folder per day which people read/cleanup. Multiply that with the total number of employees in an organization and you’ll see what I mean.

Difficult problem to solve

There are various reasons that make spam a difficult problem to solve.

1. *Human Factor:* It affects the human user, by causing either annoyance or excitement but does not harm the machine where it is stored. Virus, Trojan etc on the other hand harm the host/target machine and not the user directly. In other words Spam is content driven whereas virus, worms and other malware are logic driven. (There is no patch for human stupidity☺).
2. *Dynamic nature:* Spam messages can be tweaked or changed to evade filtering but still preserve the intent of the message.
3. *Coming from valid but compromised source i.e. zombies.*
4. *Best of buddies:* virus, worms, Trojans use spamming techniques to spread themselves and in turn help spammers in sending large amount of spam emails.

Messaging Primer

Emails enter into the messaging system and are transferred from one server to another using SMTP protocol till they reach the recipient's message store. While reading the mails POP and IMAP protocols are used.

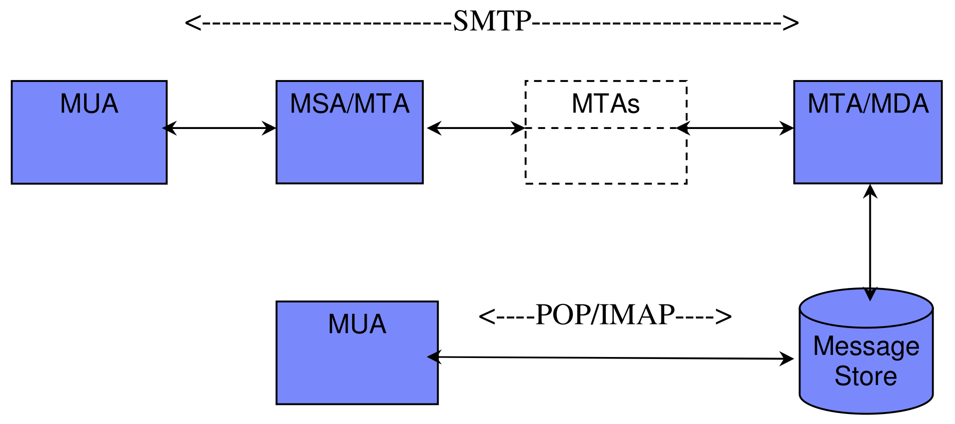
Sending emails

SMTP – Simple Mail Transfer Protocol (used for transferring messages)
MUA – Mail User agent (SMTP clients: outlook, thunderbird, evolution)
MSA – Mail Submission agent (local SMTP server for enforcing local policies)
MTA – Mail transfer agent (SMTP servers)
MDA – Mail Delivery agent (SMTP server/Message store)

Retrieving emails

MUA – Mail User Agent
POP – Post Office Protocol (simple retrieving of messages from inbox)
IMAP – Internet Message Access Protocol (Advanced features for manipulation of messages in the message store and multiple folders in mailboxes)

Path of a message



1. *SMTP is a store and forward protocol.*
2. *The message may pass through many mail routers before finally getting delivered to the recipient's mailbox.*
3. *Every SMTP server that receives a mail from another server (client for that server actually) takes the responsibility of delivering the message or sending a failure notice back. This transfer of responsibility happens after the SMTP client receives a 2XY response, from the server, for the single dot (<CRLF>.<CRLF>, where <CRLF> = \r\n, carriage return and line feed) sent after the message body.*
4. *At the final SMTP server (MDA) the message is stored in the message store for later access by the recipient.*
5. *The recipient uses POP/IMAP (or web access) to retrieve the mail and read/delete/forward or do whatever with it.*

Email Format

Every email has an envelope and a Message.

1. *Envelope:* It contains the sender and the recipients email addresses and is used for transferring the mail.
2. *Message:* It is divided into two parts – Headers and body.
   - *Headers contain tracing information, metadata etc for a message.*
   - *The body contains the actual data that is read/used by the recipient of the message.*
   - *Headers are of the form "headerName: headerValue". Every header has its own headerValue format. User specified (non-standard) headers should follow the syntax "X-HeaderName: HeaderValue"*
   - *Each line in the message ends with a <CRLF> pair. Empty lines will only have <CRLF>.*
   - *Headers and body are separated by two <CRLF> pairs.*

MIME (Multipurpose Internet Mail Extensions)

1. MIME was developed to enable the transfer of non text data with the mail like binary files, documents, images etc.
2. MIME forms a kind of a tree structure of MIME parts in the message.
3. Every MIME part inside a message has its own headers and body and can be thought of as an embedded message object in itself.

Getting inside a spammer's mind

Why do people send spam? What do they get out of it?
*T*he early virus writers usually wrote viruses for fun (at least *they* thought it was fun☺), but the idea behind spamming is totally different. The motive is profit/benefit.

How can spammers derive profit by sending spam?
Spam messages usually include marketing emails, phishing attacks, malware, etc.

The following section describes how spammers may operate. They need to have a motive/intent and also need hardware/software for executing their plan.

Intent

The type of spam messages created and sent are defined by the spammer's intentions:

1. *Marketing:* They need to market their products or services (legal or illegal) quickly using the cheapest media and because of the way SMTP works it costs nothing to send bulk mails anonymously. There is less risk of getting caught, less investment for sending mails. Products and services that spammers try to sell can be (but not limited to) medicines/drugs, Mortgage services, pirated software, imitation products (Rolex watches etc), pornographic content, malware etc.

2. *Phishing/Fraud:* They steal confidential information from innocent people by sending forged mails and impersonating others. (banks, financial organizations, Govt. etc).Stolen information can be credit card numbers, login credentials for e-banking, SSN and other confidential data. The emails contain logos and forged links to the spammers' websites designed to look as genuine banks, credit card companies, government sites etc.

3. *Malware:* Malware authors have found a new medium to spread their infection by using emails. Their intent is to create zombies/Botnets which they can control and further sell their services

to either spammers or anyone who wants extra CPU power for doing illegal work anonymously, use the machines as a storage medium or corrupt the data on machines for fun etc.

Execution Process

1. Firstly spammers require a list of address to send e-mails to, for either infecting or marketing etc. Various email spider software can be used to crawl the web and extract email addresses. There even exist people who sell email address databases from various sources illegally.
2. Now they need web space to advertise their products/services or create fake web sites (of banks etc). They purchase the web space using stolen credit cards or host it from a place, where there are easy or no cyber laws.
3. The next step is to find software/hardware for sending mails. They might use their own machines to send mails directly over the internet or through open relays or buy *Botnets* from crackers or infect machines themselves and then use them for sending mails.

Fighting Spam

Layered security

The AntiSpam engines can be deployed at strategic places based on the requirements/needs of the customers. Following are the layers where it can be applied:

Server Layer
1. *Network boundary/Gateways:* These are the servers that are listed in the DNS MX records for the recipient's/sender's domain. These are the most preferred places to install Antispam/Antivirus filters.
2. *Mail routers:* These can be any of the internal mail routers/relays. However it is not a common practice to keep the antispam filters as part of the internal mail routers.
3. *Message Store:* The last/first server which holds the mailboxes for recipients/senders of an organization. These are usually kept inside the private network.

Client Layer
1. *POP/IMAP/SMTP Proxies:* These act as transparent/opaque proxies for user for sending/ receiving mails. These filter the mail and optionally tag the email or store it in a different location (ex. SPAM mailbox in case of POP/IMAP proxies and quarantined in case of SMTP proxy)
2. *Plugins to MUAs:* These are plugins to message sending/receiving clients and filter the mails (before sending - for SMTP and after receiving for POP/IMAP)

NOTE: Unfortunately most organizations use antispam filters only on receiving mails and ignore the outgoing mails. This may lead to a bad reputation of the organization's sending MTAs incase there is an outburst of spam messages from the internal network's zombies. This coupled with no authentication in an SMTP session, outgoing port 25 open from the firewall for internal network is a bliss for an attacker/spammer.

AntiSpam Technologies

Whitelists/Blocklists

They are used to allow/deny/filter the mails coming from sources listed in these type lists. These lists contain IP / domain / user information which can be queried to check whether a sender's (or recipient in some cases) IP, domain or user name is present in the list and appropriate action can be taken based on the information retrieved. Some actions
can be (but not limited to):

1. Whitelists – Accept connection from the SMTP client IP, bypass the all/selected filters for the listed SMTP client IP or HELO/EHLO domain, MAIL FROM domain/user or RCPT domain/user.

2. Blocklists – Reject SMTP connection for listed IP, quarantine mail for listed IP, HELO/EHLO domain, MAIL FROM domain/user, RCPT domain/user.

These lists can be divided into two types based on their location:

Application/Local lists

These are internal to the application (SMTP server / AntiSpam software) and are maintained by the administrator or the application itself.

DNS based/External Lists

These are list servers present on the Internet and provide service to any client trying to query them. These list various type of sources such as open relays, virus sending machines, spam sending machines etc. There query mechanism is through DNS and list IPs / domains / email addresses (not widely used). The DNS query is for an A type record and the response is an IP address (127.0.0.x where x depends on the policy of the list server) if the query type is listed in the list server. Following is the mechanism to query them (assuming the DNS list server name is bl.example.com):

- *IP address query: If one wants to query for and IP address say 192.168.0.1 the format is similar to Reverse DNS query – 1.0.168.192.bl.example.com, DNS response is 127.0.0.x (if listed) or nothing (if not listed).*
- *Domain name query: Say you want to query for domain foo.bar.blackhat.com, DNS query – foo.bar.blackhat.com.bl.example.com and the response type is the same as for IP query.*
- *User name query: Query for user@foo.blackhat.com, DNS query – user.foo.blackhat.com.bl.example.com and the response type is the same as for IP query.*

NOTE: Please choose wisely as to which DNS based list you want to use and whether you want to block the client based on it's presence in any blocklist (not a very good idea).

Greylisting

It is something between whitelisting/blocklisting in the sense that the message is temporarily rejected (with 4yz reply code from the server). This technique exploits the RFC specification for a good reason (but has its own side effects). The RFC specifies temporary error reply code and states that the SMTP client should queue the message and retry later in case of an error in transmission. The idea behind it is that the Spammers use simple SMTP implementations which do not queue the message on failure and will not likely retry whereas an RFC 2821 compliant SMTP implementation will queue the message on temp failure and retry at a later time. The basic information required for greylisting is usually a 3 tuple:

- The SMTP client's IP
- The envelope sender address
- The envelope recipient address

Though variation of the above can be used in greylisting implementations based on their requirements and problems with current SMTP implementations like using a network block instead of the full IP in some cases(retrying MTA is different from the original sending MTA) , ignoring sender/recipient address etc.

How it works

When an SMTP client connects and starts an SMTP session the recipient server checks the current three tuple information in its database, if it is not present (i.e. first connection/transaction, the IP, sender, recipient combo has never been seen before) the server issues a temp failure *"4yz"* to the client and adds the tuple info to it's database. If the client tries at a later point, the connection/message is accepted and may be filtered or sent to the recipients INBOX. Fortunately spammers don't waste time in queuing and retrying and this technique is still effective.

SPF (Sender policy framework)

This works in the exact opposite way how MX works, while MX records specify the authorized mail receivers, SPF records (TXT or SPF) specify the authorized senders for a particular domain. It is a common misconception among people that SPF is an AntiSpam measure. No, it's not, it actually is an Anti forgery/Phishing technique which may/may not help in fighting spam.

- *It is sender driven, in the sense that the sending domain needs to publish records in order to use the functionality of SPF*
- *It uses DNS TXT (also specifies SPF RR Type in the rfc) records to publish the SPF protocol string.*
- *It specifies certain senders as authorized for a particular domain.*

How it works

A domain wishing to publish authorized sender MTAs for it's domain just needs to add SPF protocol compliant string in the TXT record for that domain. At the receiving end , the client's IP, MAIL FROM address and the HELO/EHLO domain is taken and a query for TXT record is sent to the MAIL FROM domain or HELO/EHLO domain and based on the SPF protocol string and the above information it is verified whether the SMTP client is authorized to send mail on behalf of the sender domain or not.

DKIM (Domain Keys Identified Mail)

It is a technique to sign the message. It also helps in fighting forgery as the signing domain claims the responsibility for injecting the message into the message transport system. This technique is also not exactly an AntiSpam technique. An excerpt from the DKIM rfc "Protection of email identity **may** assist in the global control of spam and phishing". Not sure when that "*may*" will actually happen☺.

- *It is sender driven, in the sense that the signing domain is responsible for signing the message and publishing the public key (for decrypting the hash by the receiver).*
- *As of now it uses DNS TXT records to publish the public key.*
- *The signing domain claims responsibility of injecting the message and the integrity of the message is also preserved during the transport.*

How it works

The signing domain publishes a public key in the TXT records of one of it's sub domains (having *"_domainkey"* as a sub domain component in the domain name, for Ex: *_domainkey.example.com*, *subdomain._domainkey.example.com*) which is DKIM protocol specific. The process for creating the signature is:

- *The hash of the body is created using hashing algorithm (sha256 as of now) and inserted into the DKIM-Signature header.*
- *The hash of chosen headers (DKIM-signature header is also appended which has the hash of the body) is created using the same hashing algorithm.*
- *Finally the signature is created by signing the header hash with the private key of the signing domain and inserted into the DKIM-Signature header.*

The process for verifying is almost similar to the process of creating the signature with the exception of signing with private key. In the latter case, the final step includes decrypting the signature with the public key obtained from the *DNS TXT record* lookup and matching it with the header hash just created. If it matches all is well, else we've got a problem somewhere.

Challenge/Response Systems

In these types of systems the receiving MTA sends an online/offline challenge to the sender/sending MTA and waits for a response (offline). If it gets the required response it forwards the mail to the recipient else acts on the message based on what policy is set for these type of messages like deleting/tagging/putting in Spam mailbox etc.

The challenge response systems require human intervention to solve the challenge and cause delay in delivery. They also unnecessary send challenge messages to unknown senders, since spammers use forged address, the challenge may go to a person who owns the email address spoofed by the spammer. This behavior in effect causes unintentional spamming by the receiver (challenger). The challenges can be:

- *Sent as a bounce message.*
- *A Temp Error SMTP reply code text.*
- *A URL which is required to be clicked to solve the challenge.*
- *A Captcha on a web page to be solved.*
- *A reply to the bounce message.*

HashCash

As the name suggest it uses hashes. The cash here refers to the time the CPU is consumed to perform its computation (for a partial hash collision). The basic idea is Proof of work done by the sender before sending a mail. An assumption is made about the spammers that they cannot waste the time spent in computing hashcash and if they do their message sending rate will go down drastically to an unacceptable level for them.

- *It is sender driven in the sense that it requires the sender to compute and put the hashcash in the hashcash header field for the recipient to verify it.*
- *It advertises proof of work done by the sender.*

How it works

Think of it in terms of calculating/verifying a square root. While calculating a square root is a tough job, verifying it is very simple. The idea is to prove that the sender has done some work (spend time on calculating hashcash) prior to sending the mail. The hashcash documentation specifies a particular set of fields that are required to create a hash (*partial hash collision*) like the recipient's email address, date/time, number of bits for collision etc.

A *hash collision* occurs when 2 different strings map to the same hash, which is impractical to achieve. So Hashcash uses the idea of *partial hash collision* (part of a hash, instead of the whole hash) and that to with *zero "0"* bits. Using the defined fields and a counter, hashes are created which will collide with say N bits all zeros. This becomes quite feasible and can be computed quickly (fraction of a second to a few seconds) without the sender even noticing. The hashcash has to be computed for each recipient separately which further adds a delay per recipient and is a bad news for the spammers.

Just a thought

Is it actually a measure to stop spamming? Or just another measure to narrow the gap for spammers, so that they adapt to delays as well like in the case of greylisting.

String match filters

These filters match the body text and header text against a a set of strings (words or phrases. They are pretty straight forward in that they try to find exact string match for ex:
"Viagra" in the body will match with *"Viagra"* in the string-match set (of the antispam application) and it will not match the word *"vi@gra"* in the message text.

Regex filters

These filters use regular expressions to match a words/phrases in the message text. They are most advanced than simple string matches in that a single regular expression can match more than one word/phrase. For Example given a regular expression – ***[v\/][il1I|][aA@/\4][gG][rR][aA@/\4]+***

can match against *Viagra, vl@gR/\, V|4Gr@a* but not against *v**i|I**gra, vi@@gr/* etc (not the multiple characters in the same type.

NOTE: Both the string match and regex filters have a problem that they are static and have to be added manually by the administrator or the vendor of the AntiSpam product to reflect new words/phrases used by the spammers.

Bayesian filters

These are more sophisticated than above filters. They perform statistical analysis of the content of the message and make a decision based on the probability that the tokens (words/phrases etc) fall into the spam category or ham category. To improve their performance and to adapt them to any organization, these filters have a learning period during which they do not filter messages (or filter with a default database provided with them) and learn from the incoming or fed messages. People (administrator, users) feed the engine with messages based on the categories (spam, ham, newsletters etc).

How it works
It works in two phases, learning and filtering. Although it is a common practice to learn while in filtering phase, but the vice versa is not common. Since the learning phase starts when the engine is deployed onsite and continues for a particular period of time. Bayesian filters may also come with a predefined database so that they can directly start filtering (and learn with the filtered mails) and do not have to go through the learning curve.

Learning

During the learning period the engine builds its database. The database may contain a set of tables for each category (spam, ham, newsletter, sports etc). The table may have tokens and their occurrence count in the messages and the total number of messages of that category read. For simplicity we will consider only two tables **Spam** and **Ham**. For each message read, it extracts a set of tokens which it thinks are important (for example: most header field names may not be that important, some header
Field values might be very important like From, Subject etc) and records it's occurrence count in the table. When the Engine has read enough mails and built it's database to a sufficient level it moves on to the actual filtering stage.

Filtering

On receiving a message, the filter extracts the tokens from the message and computes spam probability of each token i.e. probability that each token belongs to the spam category. Finally it calculates a combined probability of all the tokens (or only the high and low probability tokens). This probability is *the probability* that the message is a spam or not. This message probability is then compared to a threshold probability for spam to determine whether the received message is spam or not. And if configured to *auto learn* from filtered messages the engine learns the tokens from the received messages as well and accordingly adds them to the corresponding table based on what decision it took when filtering.

NOTE: Although we are talking only about spam and ham. This technique can be applied to segregate mails based on user defined categories as well (personal, official, sports, photography etc). Also, when used at the client layer (per user) the Bayesian filters work extremely well (very low false positive and false negative rate). At the server layer it may not work that well because in an organization different people might have different interests when it comes to deciding whether a mail is spam for them or not.

Signatures

The signature techniques create hashes of message text (part or whole) based on the technology they use for creating signatures. The basic process involves client-server architecture, where clients query with a signature and get response whether that signature matches a spam signature or not. Note that the signature list can be kept locally like in the case of Antivirus, but the technology of creating signatures is totally different. Some of the significant signature technologies:

Nilsimsa Signatures

This one is my favorite. It creates a fuzzy signature such that two similar messages may also have similar (not same) signatures. In simple terms it defines a range of similarity say -128 to 128. Signatures for different messages can be compared and yield a number between -128 to 128 which specifies how many bits are different in the signatures being compared. So, for example if the number is 58, it means that 180 bits in the signature are same and 76 bits are different. Based on this information and a threshold of similarity a decision is made whether the two messages (body text for email messages) are from the same source (have similar content) or not. These signatures are effective against tweaking of messages by spammers.

Ephemeral Signatures

These are short lived and dynamic in nature. Different parts of message text are hashed at different time intervals using a random value for the hashing part of the text. They are also effective against tweaking of messages by spammers, since they don’t know which parts of text will be hashed.

Simple Hashes

These can be MD5 or SHA1 signatures of body text of the message. These are pretty simple in the implementation and the filter can be fooled by even slightly tweaking the message text.

OCR Filters (Optical Character recognition)

The filters extract textual content from images inside the messages and perform filtering on the content. Based on the result of the content filtering the message is considered a spam/ham. They are not very efficient because of the extra time involved in extracting text from the image. Some image scanners also scan for pornographic content, company logos to check for phishing attacks etc.

Heuristic Filters

Heuristics involves combing the above techniques, creating tests/rules for each type of check (whether content filtering, header/source filtering) and assigning weight/scores to each test. A threshold is then carefully defined. Each test is run against the message and if it triggers it’s score is added to the total score for the message (total score is zero in the beginning). If the total score crosses the threshold while running the tests or after running all the tests (these are usually user defined, either you want the filter to stop when the threshold is reached or continue with all the tests to get the actual score) the message is considered a spam and acted upon accordingly (user defined action: delete, send to spam mailbox, tag as spam etc).

NOTE: There are tests that check the spamminess of a message as well as tests that test for it’s genuineness in which case the score would be deducted from the total. For example whitelisting, SPF, DKIM etc can be thought of as tests for genuineness.

Reputation Systems

There are relatively new as compared to the above techniques. There has been no standardization on what parameters define the reputation of a particular host. There are online reputation systems as well as AntiSpam application specific ones. Reputation systems use advanced heuristics in addition to the above tests to determine reputation of SMTP hosts on the internet. The parameters/checks include (but not limited to):

1. *No. of connections from a particular host in a given time period.*
2. *No. of recipients in a single SMTP session (No. of RCPT commands issued).*
3. *No. of messages sent in a single SMTP session.*
4. *No. of virus sent form the host*
5. *No. of spam sent from the host.*
6. *No. and type of RFC violations done by the host.*
7. *Whether the host is whitelisted / blacklisted.*
8. *Does it have SPF records pointing back to it?*
9. *Is the DKIM signature valid?*

Organized Spamming

By the term **Organized**, I mean that spammers will act in a way that looks more like legitimate senders by doing their homework properly. This section covers some methods currently used by spammers and some new ideas which they may resort to in the future.

Targeting low priority MX

Spammers will typically try and connect to the lowest priority MX for a particular domain. The reason being that low priority MX in some cases do not have antispam engines and may directly transfer the mail to the Message store. If you are luckier it may act as an open relay because of no configuration settings done to it.

Mail reconnaissance

I use this term to describe the activity of sending a mail to a person or group inside an organization in such a way that it increases the probability of a reply from them. It is a kind of social engineering attack. The best way is to send a mail to the sales or marketing dept. and enquire about their products or services. When they reply (they will, for sure ☺) just analyze the mail headers and extract any juicy information like the local MTAs, client software used(X-Mailer etc), version information etc. You can then try to infect some of the machines inside the organization with specially crafted malware, which uses the above information to further infect or use the local Messaging system to deliver your spam.

Using Zombie’s Local MTA/MSA

You can push messages from zombies to their local MTA and have the messages appear to be sent from valid MTA. The local MTAs may apply DKIM signature and may have SPF records for the domain. If the local MTAs require password you may want to sniff for it. If the password is not sent in plain text, you may try to further install *keyloggers* to get the password. If you are still not successful then you can fallback to the old method of using direct SMTP connection from the zombie. Using mail reconnaissance the spammer/cracker can greatly increase the chance of abusing the local MTAs.

Evading Greylisting

Evading this technique is simple. The spammer only has to use a RFC compliant MTA when sending spam or use zombies to send the mail to local MTAs which will do the work of retrying for the spammer. There are assumptions about spammers pertaining to greylisting that they are not ready to waste time by retrying and that even if they do it is possible that they will get listed soon on DNSBLs or as Razor signatures etc. While this technique alone will not help spammers they will use this in conjunction with other techniques to guarantee their delivery.

Evading OCR

CAPTCHAs are used to stop online spamming and verifying human interaction. Spammers took up the idea for their own advantage and started sending obfuscated images with the spam content which confuses the OCRs. I call this ironical but interesting technique - “The CAPTCHA Effect”.

Testing the spam messages

Antispam tools

There are various open source antispam tools available (one of the famous and good tools is spamassassin). Spammers use this tool to test the content of their message and tweak the message accordingly till the score goes below the spam threshold.

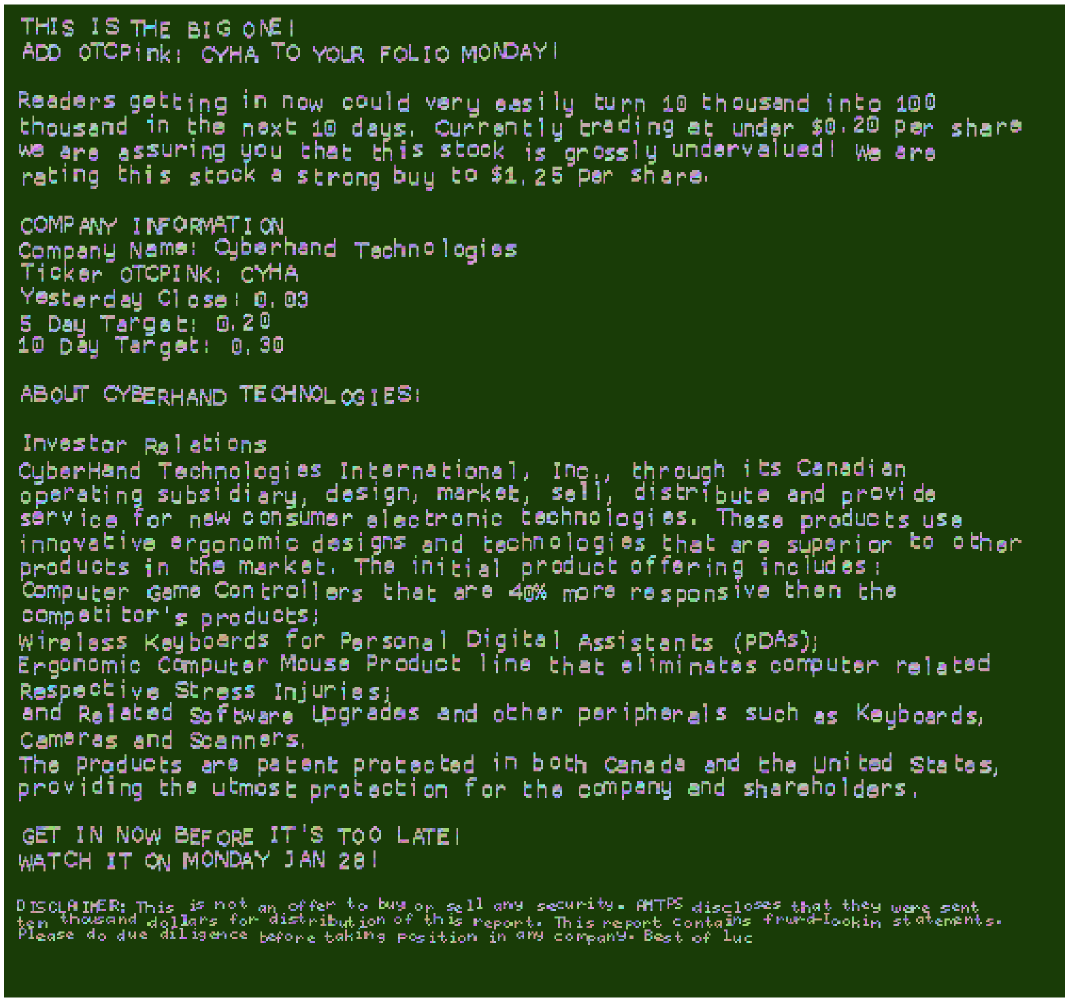
Free email providers

Once the Spam has been tested locally with antispam tools, it time to test its online behavior. Spammers can use various free email providers (gmail, yahoo, hotmail etc) for testing the spamminess of their messages. All they need to do is to create an account with the providers. They send the testing messages to their fake accounts and access them either through IMAP or manually though web to see whether they are still getting caught as spam or not. If, yes then they still need some tweaking or need to change the message sending source. This testing cycle may go on till they bypass few online spamfilters.

Samples

Some samples of how and what people do to evade filters.

Image



THIS IS THE BIG ONE!
ADD OTCPink: CYHA TO YOUR FOLIO MONDAY!

Readers getting in now could very easily turn 10 thousand into 100 thousand in the next 10 days. Currently trading at under $0.20 per share we are assuring you that this stock is grossly undervalued! We are rating this stock a strong buy to $1.25 per share.

COMPANY INFORMATION
Company Name: Cyberhand Technologies
Ticker OTCPINK: CYHA
Yesterday Close: 0.03
5 Day Target: 0.20
10 Day Target: 0.30

ABOUT CYBERHAND TECHNOLOGIES:

Investor Relations
CyberHand Technologies International, Inc., through its Canadian operating subsidiary, design, market, sell, distribute and provide service for new consumer electronic technologies. These products use innovative ergonomic designs and technologies that are superior to other products in the market. The initial product offering includes:
Computer Game Controllers that are 40% more responsive then the competitor's products;
Wireless Keyboards for Personal Digital Assistants (PDAs);
Ergonomic Computer Mouse Product line that eliminates computer related Respective Stress Injuries;
and Related Software Upgrades and other peripherals such as Keyboards, Cameras and Scanners.
The Products are patent protected in both Canada and the United States, providing the utmost protection for the company and shareholders.

GET IN NOW BEFORE IT'S TOO LATE!
WATCH IT ON MONDAY JAN 28!

DISCLAIMER: This is not an offer to buy or sell any security. AMTPS discloses that they were sent ten thousand dollars for distribution of this report. This report contains frurd-lookin statements. Please do due diligence before taking position in any company. Best of luc

Bayesian Database poisoning

Notice the second paragraph in the below email. It makes no sense in the context of the message. It is appended deliberately to poison the Bayesian database.

```
Subject: Ever been ashamed of the size of your dick?
MIME-Version: 1.0
Content-Type: text/plain;
        charset="Windows-1252"
Content-Transfer-Encoding: quoted-printable

Size DOES matter=2E

Many studies have shown that the thicker and longer the member, the more =
Pleasure a woman gets from making love=2E
```

```
Now, within 6 short months, gain 30-40% increase in thickness and a few i=
nches to your member with a simple, miraculous solution!
http://ufbnsoboa=2Ecom/ come by VPXL



Police in India arrest two people in relation to the protest rally=2E Mr =
Tsvangirai told the BBC that police of better childcare facilities=2E
civilian automobile=2E civilian automobile=2E derailed in Oneida, New Yor=
k=2E
Khalid Sheikh Mohammed, long suspected as the runs off just 15 balls to s=
et a high target for the To summarise the situation, the 35-page document=
s says:
```

Lame Spammer error

Notice the “%XMAILOE”. An error (probably variable substitution error) in the way ratware is used to send automated spam. This particular error caused the text part from “%XMAILOE” onwards to be displayed as body.

```
Subject: It's important
Date: Wed, 6 Feb 2008 02:12:14 -0100
MIME-Version: 1.0
Content-Type: text/plain;
        format=flowed;
        charset="iso-8859-1";
        reply-type=original
Content-Transfer-Encoding: 7bit
X-Priority: 3
X-MSMail-Priority: Normal
%XMAILOE
%XMIMEOE
X-Antivirus: avast! (VPS 080205-0, 2008.02.05), Outbound message
X-Antivirus-Status: Clean

Now it's time to solve your sexual problems

But how to do this? It's easy :)

All details are here:
http://veshivan.com/

Have an impassionedzealous nights!
```

Phishing Attack

The url in the anchor tag is not a Citibank website (robox3.st). Also notice the last font tag, it has color set to white “FFFFFF” and has random text used to poison Bayesian database.

```
In-Reply-To: "CitiBusiness" <generatedmail.id6365238552-
63693280CBF@citibank.com>
From: "Citibank" <mail_server.id40376718-531581CBF@citi.com>
Subject: CitiBusiness customer service: service message. (message id:
13606833)
MIME-Version: 1.0
Content-Type: multipart/alternative; boundary="--CGSFX6FCRKXE186491BXUL"
```

```
----CGSFX6FCRKXE186491BXUL
Content-Type: text/html;
Content-Transfer-Encoding: 7Bit

<head>
<meta    http-equiv="Content-Type"       content="text/html;   charset=iso-
8859-1">
</head>

<body>
<p><font  face="Arial">Dear CitiBusiness customer,</font></p>
<p><font    face="Arial">Financial      institutions      are      frequent
targets      of      fraudsters.     We     have     implemented     security
measures    to protect    our      systems from   attack, but
increasingly,   our customers      must    also protect
themselves.</font></p>
<p><font  face="Arial">Our       new  CitiBusiness      Form    (CBF) will
help   you   to      protect     your  data from    misuse,
unauthorized   access,    loss,     alteration    or
destruction.</font></p>
<p><font face="Arial">You must    complete       CBF on  a     regular
basis.</font></p>
<p><font       face="Arial">Please     click    on      the link   below
to open  CBF:</font></p>
<p><font    face="Arial"><a     href="http://citibusinessonline.da-
us.citibank.com.robox3.st/citibusinessonline/CBF.do?CID=8730562337932787383
601320773681079440552749868715887096&systemid=460908607970188">CitiBusiness
Form</a></font></p>
<p><font   face="Arial">This  email       has       been     automatically
generated.<br></font></p>
<p><font   color="#FFFFFF" face="Arial">0x53485009, 0x65, 0x470, 0x7, 0x206
DX9C, X4M, rev, interface, R7U2, GUJN, WXT.   engine: 0x873, 0x0974, 0x29,
0x83797696, 0x1, 0x30235200, 0x40219233, 0x09900372, 0x9, 0x853, 0x6627
182713775610729140269919O6   0x78, 0x374, 0x54317570, 0x40, 0x3172, 0x9106,
0x77, 0x7172, 0x258, 0x2400, 0x3132, 0x6, 0x4  hex: 0x77     0x383, 0x478,
0x48777440, 0x60, 0x9663, 0x13, 0x44, 0x59651858, 0x420, 0x6, 0x012, 0x33,
0x1 0x31, 0x08, 0x9, 0x03, 0x58155427, 0x10618126, 0x4    0x525,
0x66997386, 0x05, 0x60, 0x66664485, 0x2285 api: 0x4, 0x69873098, 0x6, 0x6,
0x69, 0x44065122, 0x1592, 0x0222, 0x12, 0x0842, 0x96661267, 0x25849240,
0x82, 0x753, 0x39</font></p>
</body>
</html>

----CGSFX6FCRKXE186491BXUL--
```

Google redirect

It uses google to redirect the user to it's website. Used to fool filters that Check only the base url and nothing in the query string.

```
Subject: Morttage refiinancing, online ...

Tired of paying high interest rates?

http://www.google.com/pagead/iclk?sa=l&ai=uptrend&num=575095&adurl=http://i
mortgage.tw?paddy
```

References

1.	**Bayesian filtering**	- http://paulgraham.com/
2.	**SPF**	- http://www.ietf.org/rfc/rfc4408.txt
3.	**DKIM**	- http://www.dkim.org/
4.	**SpamAssassin**	- http://spamassassin.apache.org/
5.	**Razor**	- http://razor.sourceforge.net/
6.	**CAPTCHA**	- http://www.captcha.net/
7.	**Bogofilter**	- http://bogofilter.sourceforge.net/
8.	**Mailwasher**	- http://www.mailwasher.net/
9.	**HashCash**	- http://www.hashcash.org/
10.	**Greylisting**	- http://greylisting.org/
11.	**DNSxLs**	- http://www.ietf.org/internet-drafts/draft-irtf-asrg-dnsbl-04.txt
12.	**SMTP**	- ftp://ftp.rfc-editor.org/in-notes/rfc2821.txt
13.	**Internet message format**	**-** ftp://ftp.rfc-editor.org/in-notes/rfc2822.txt
14.	**MIME**	**-** RFCs 2045, 2046, 2047, 2048, 2049
15.	**Nilsimsa**	**-** http://ixazon.dynip.com/~cmeclax/nilsimsa.html

DISCLAIMER: The samples used in this document are taken from the spam folder in my free web mail account. These samples are just for reference purposes and I do not claim that they are spam or not. This document is solely for knowledge share purposes.