Messaging, Malware and Mobile Anti-Abuse Working Group

M³AAWG Best Current Practices For Building and Operating a Spamtrap

Version 1.2.0
Updated August 2016

Table of Contents

Introduction

Computer security researchers have long made use of "honeypots," servers and/or networks designed as traps to detect, deflect, or in some way counter and research the abusive use of information systems. To an outsider, a honeypot generally looks like an ordinary service such as a Web server, mail server or network server, but it has additional instrumentation for close monitoring. In some cases, the honeypot can be an ordinary system doing productive work with the addition of this monitoring instrumentation. At other times, a honeypot is a special system that only does monitoring.

In the email abuse field such honeypots are usually called "spamtraps" and they are servers that receive spam and other types of email abuse. A spamtrap is designed with specific goals in mind and the researcher monitors the spamtrap to ensure the necessary data is collected. While some purists may assume that spamtraps are exclusively for the detection of unsolicited email, also known as spam, email abuse is now far broader than just spam and we will use the term spamtrap as a honeypot designed to capture any sort of email abuse.

The researcher uses the information captured by the server monitoring instrumentation to achieve research objectives. However, depending on the researcher's goals, implementation strategies, and heuristics, researchers often find numerous pitfalls with spamtraps. This document attempts to lay out the best

practices in operating a spamtrap such that researchers will achieve accuracy in their research, whatever their specific goals.

The audience for this document is currently active and potential spamtrap operators and those researchers. They generally make use of the data generated from spamtraps for purposes such as research, evidence collection, infected machine mitigation or mail list leakage and list quality control.

Spamtrap Goals/Purpose

While there are almost as many possible goals for a spamtrap as there are email abuse researchers, some common examples are:

- Refining local spam filters
- Creating reputation lists, including DNS-Based Black Lists (DNSBLs), based on a variety of heuristics
- Monitoring client bulk mail lists
- Capturing and analyzing virus and other malicious payloads
- Identifying and eradicating phishing
- Identifying and detecting malicious URLs and domains
- Detecting data leakage

The researcher's goals play a major factor in the spamtrap's design. For example, simplistic counting or checksumming of emails does not work well in a combined spamtrap-production environment where large numbers of real users could legitimately receive identical emails (e.g., outsourced corporate communications). Likewise, in the case of confirmed malware, the existence of a valid recipient email address does not matter.

Spamtrap Addresses

Spamtrap email addresses are collected in a number of different ways. Analysis heuristics that work with some collection methods will not work as well with others. This section attempts to describe some of the major pitfalls and to provide guidance in selecting and using a collection method.

Each address collection method has a different history and some are more useful for specific goals. The information each can provide is different in subtle and sometimes not-so-subtle ways. Some types of spamtraps require special management and have special caveats for their use.

Spamtrap address categories include:

- *Pure (or pristine) addresses:* Addresses that have never existed as a legitimate destination for email but are still receiving email. This occurs due to misspellings of addresses and parsing errors of automatic address harvesters; for example, truncating parts of email addresses, collecting Message-IDs as addresses, etc.
- *Created (seeded) addresses:* Addresses that have been deliberately "offered" by listings on websites (especially via tools like "wpoison"), by deliberately injecting machines with lots of preloaded email addresses, and by other methods.
- *Repurposed addresses:* Originally valid email addresses that are now being used as trap addresses.
- *Existing addresses:* Using instrumentation from production mail servers delivering email to real users.

Issues with Addresses

Even when pristine trap addresses are used in a spamtrap, the heuristics in the spamtrap must **not** assume that all email to these addresses is invalid. A trap used to detect bot-emissions via malware fingerprints does not need to care about the legitimacy of the recipient addresses, but a trap that identifies spam emitters by counting emails from given IP addresses should ensure that the email is checked for consistency with the goals/heuristics of the trap.

Any address that receives spam can also be forged as the sender of future spam. If this occurs, the addresses that have been forged will receive bounces of spam or responses from autoresponders. While purists may believe that a server bouncing spam is as bad as one originating it, operators have to know of these issues and decide, based on their goals, whether it matters or not and share this with users of their data. (For example, see the transparency requirements in RFC 6471.)

When a valid email address or domain expires, this does not mean that everyone who had a valid reason to send email to those addresses immediately knows that the address no longer exists. When the address was valid, the user may have signed up for mailing lists, purchased items online, sent email to friends and family, etc. Therefore, it must be assumed that upon expiration, some email that is sent to the address is legitimate, even after the address is repurposed as a spamtrap.

Most simple spamtraps, including most spamtraps used for DNSBLs, use basic counting/similarity techniques in their heuristics. Repurposed addresses must be conditioned before use in such a spamtrap. The intent behind conditioning is to give current users, who legitimately send mail to a now-repurposed address, fair warning that the address is no longer valid. The recommended practice is that email to these repurposed addresses should receive standard SMTP "no such user" 500-level rejection messages (see RFC 5321) or otherwise ensure that no mail is accepted for delivery for a minimum of twelve contiguous months. 500-level RCPT rejections do not necessarily mean that no useful information can be derived from such connections; however it usually precludes saving copies of the email.

Sometimes, twelve contiguous months is not long enough to catch infrequent mailings. Hence with repurposed email addresses, the probability of an email being legitimate never reaches zero (see Hints and Pitfalls).

The contents of valid email messages to valid email addresses used as spamtraps obviously represent potentially severe breaches of privacy. This is still true with repurposed spamtrap addresses with respect to the previous owner of the address. The full ramifications of this are out of the scope of this document due to varying legal jurisdictions, but will be touched upon briefly in subsequent sections.

Implementation Considerations

A spamtrap is a server that processes and logs, monitors or collects email for the purpose of spam analysis. It is a best practice that the domains of email addresses used for the spamtrap have their MX records point directly to the spamtrap server so that inbound messages for those addresses do not traverse intermediate servers. Forwarding of email from non-spamtrap servers, acting as a domain's MX to the spamtrap server, can be done; however, some useful information is invariably lost. If it is necessary to forward email, ensure that this action still supports the goals and techniques of the spamtrap.

The spamtrap can be co-resident with production service and email flows, as long as the design, goals and heuristics are consistent with the mixture of valid and invalid email addresses and the need to deliver valid email.

Larger volume spamtraps generally should be built with mail server software capable of being specifically customized for spamtrap use. While standard mail servers can be used, the operator may find that the facilities for extended logging, archiving and fingerprinting are limiting, awkward, or non-existent.

Note that customizing production email servers or software for spamtraps may present difficulties in terms of future upgrades, support and versioning. This can also present issues with production email performance, especially if the spamtrap addresses come under attack or during naturally-occurring spam spikes.

Spamtraps should be adequately provisioned in performance and storage capacity. They should be capable of handling very high volume spam spikes, which are frequent in the wild, as well as point failures. Trends cannot be monitored if there are large holes in the data. Thus, consideration should be given to redundant spamtrap servers with considerable extra capacity over regular flow rates.

Careful thought should be given to how the spamtrap reacts to incoming spamtrap email. The server could simply silently accept it, but, depending on the trap heuristic and goals, it may be preferable to reject it. Examples include gaining useful information during spamtrap conditioning or distinguishing senders that do not respect "no such user" returns.

Some spamtrap implementations consist of many spamtrap servers, often for different sets of trap addresses, that aggregate information to centralized analysis engines. Care should be taken to ensure that the trap address creation methods of individual traps do not conflict with overall trap heuristics and goals.

Analysis

The evidence generated by properly managed spamtraps is among the best for identifying which IPs and domains are involved in spam as well as other forms of email abuse. The archives and logs show:

- Which IP addresses are sending the abuse
- What types of abuse they are sending, including links and domains of malicious content
- Contents of the abusive messages
- The behavior of specific IPs or domains, which may indicate the type of software they are using (mailing software, botnet, operating system, geolocation, sender identification, etc.)
- Whether they are part of a recognized spam network or botnet

The spamtrap operator should ensure that the trap is logging at least as much detailed information as is required for analysis, based on their goals.

The methods used to analyze collected information vary widely, depending on the volumes that must be processed and the intended end-goal of the analysis. Obviously, there are many types of analysis that can be performed. Descriptions of specific algorithms are out of the scope of this document. This section will cover just a few specific points. The final section in this paper contains more details to be considered.

- Ensure that the spamtrap email addresses used are consistent with the analysis engine heuristics. For example, production systems receive legitimate email in bulk (such as outsourced corporate emailings to employees); hence, simple statistical counts of email flow from a given netblock or of an identical subject would inaccurately identify senders as abusive. In other words, ensure that the characteristics of the target emails are well understood and that they are consistent with the set of tools available with which to identify them.

- It is unwise to attempt to open hyperlinks contained in emails, especially in automated systems. In spamtraps that rely on statistical analysis for decision-making, opening tagged links to confirm the existence of a particular recipient address may subvert spamtrap security, confuse matters as to the opt-in status of a particular recipient, and have other undesirable side effects leading to invalidating the goal of the trap. Link retrieval is perhaps safest when the analysis system is aware that it is a malware download link. It must not be done with ordinary spam, because link retrieval that would normally be done by human action could be construed as consent to receive further mailings.
- Size does matter. Even very large spamtraps do not always generate truly statistically relevant sample sizes for given issues, e.g., originating IP addresses or on specific threats. Furthermore, even large spamtraps can differ greatly in what email is sent to them. Beware of the problems of overgeneralizing from too-small sample sizes. A basic understanding of statistics and sampling is helpful.

See RFC 6471 on listing policy transparency.

Security

Spamtrap systems and addresses, especially those used for reputation generation, need to maintain operational security so that they are not subverted by abusive third parties. If security is breached, at best the spamtrap will lose its effectiveness for the intended purpose; at worst, innocent third parties can be severely affected with maliciously injected false information.

- Do not allow the identities (domains, addresses or MX server[s]) of the spamtraps to become publicly known. If identities are compromised, they may need to be retired from spamtrap use.
- While best practices regarding DNSBLs, such as RFC 6471, encourage transparency with analysis heuristics, the trap operator should keep details of these heuristics confidential.
- Data or heuristics detail should be shared with extreme caution. (See next section.)
- Access to the spamtrap servers, logs, archives and heuristic analysis should be carefully limited to trusted entities only. Non-disclosure Agreements (NDAs) should be established.

Modern security techniques, such as system hardening, minimal attack cross-section, etc., must be implemented for both operational integrity and privacy requirements.

Information Sharing

There are several entities or groups with whom a spamtrap operator might wish to share data, such as:

- Other spam/abuse researchers, consistent with long term goals
- Law enforcement agencies
- Mitigation groups, such as ISP security groups, anti-phishing organizations, etc.
- Commercial/bulk email senders, including listees in the case of traps generating reputation information
- Service providers, such as ISPs, email providers, hosters, etc.

Important: It is important to keep in mind that there are entities on **all sides** of the email ecosystem who may wish to subvert your spamtrap data for their own use and gain. Take great care in deciding who you trust to receive your sensitive data, should you decide to share it at all.

If data is to be shared, identities of the spamtrap addresses should generally be withheld from all entities with whom data is shared. This implies that you should remove all indications of what the recipient address was (domain, server) before sharing. However, in some cases, it may be desirable to "reveal all" for the purpose of legal evidence collection or to other organizations that have a need to know and are trusted by the operator. Only trusted entities with NDAs or law enforcement agencies should receive data that has not been cleansed.

Issues surrounding spamtrap identification involve more than only the TO: addresses. Identifier tags are frequently included inside emails, including special headers, links with hidden identifiers, VERP (Variable Envelope Return Path) and other areas. Additionally, given the width of IPv6 addresses, and the size of provisioned customer nodes, it is possible to encode recipient data into the source address of the IPv6 packet. Therefore, the best practice is not to include copies of headers or bodies unless the emails have been carefully examined and any possible identifiers of the recipient address, domain or server are removed.

This issue is particularly important in the case of repurposed spamtrap addresses because the email contents may be subject to privacy, safety and security issues. No automated process can guarantee 100 percent reliability of this type of cleansing. Share message headers and bodies only after manual cleansing and with extreme caution. It is often possible to identify a spamtrap email just by revealing an exact time of receipt.

Note that the above best practice for removing data from reports is directly counter to RFC 5965, which defines the Abuse Report Format (ARF) and encourages full disclosure. A spamtrap operator may choose to use ARF only with the providers they trust not to abuse spamtrap information and/or use RFC 6590 (abuse reporting redaction methods) for data removal. Automated reporting without prior consent of the report recipient is abusive, and even with consent must have the ability to throttle the volume of reports to not overwhelm the recipient. See RFC 6650 for further guidance.

Data shared with other abuse researchers can contain just about any information. For example, spamtraps designed to identify botnets may send copies of URLs appearing in botnet emails to researchers that identify and mitigate malicious websites. It is up to the spamtrap operator to determine that the information released is appropriate for the other researcher's goals and that the researcher knows what they are allowed to do with the data; for example, by stipulating prohibitions against opening links and other issues. An NDA specifying acceptable use is recommended.

In a previous section, we identified that the email to repurposed spamtrap addresses might include personal information of the original user of the address. Sharing such information must be avoided wherever possible.

Sharing agreements or NDAs should, in general, indicate:

- That leakage of personal information is a possibility and the shared information must be kept secure on that basis by all parties who have access to the data
- The allowable uses of the information; for example: further reporting of malware payloads, subject lines, links, anti-phishing takedowns, etc.
- Limits to data retention

With ESPs (Email Service Providers), ISPs, and other large-scale email senders, it is frequently desirable to give them information identifying who sent the email and sufficient clues or hints to identify how the address was acquired. With such information, the customer can be identified and appropriate remediation steps can be taken. The best practice is to recommend that the sender identify how the addresses were collected, repair the problem, and/or educate the customer as appropriate so that the problem does not repeat. If the

ISP just removes a few offending recipient email addresses, it does not address the root cause of the issue and it does not prevent it from being repeated.

A spamtrap operator should consider providing some of the following information in response to requests. Experience tends to indicate that these items are usually more than sufficient for an ESP/ISP to identify the sender and list acquisition problems:

- Approximate timestamps
- FROM address, removing recipient encoding, if present
- Subject lines or portions thereof, removing recipient encoding, if present
- Representative content – e.g., links or content fragments – or known malware types, removing recipient encoding, if present
- Some information that the spamtrap operator knows about the recipient address but that does not identify the address, such as:
    - If address is expired and how long ago it expired, which may indicate a purchased list or bad bounce management
    - Possible typographical error of a valid email address, i.e., non-existent/non-functional closed-loop confirmation
    - If address was seeded or harvested – e.g., email addresses only appearing on Usenet, use of Message-IDs as addresses, harvest bait tools like "wpoison," and harvesting parsing errors – that may indicate purchased lists or deliberate spamming
- Approximate message counts for a given incident
- Routing information of the email trail within the sender's network, such as Received lines, X-Originating-IP, etc. This information is not always available and, in some cases, it can be considered forged and unreliable, even if it is present.

Spamtrap operators should consider which information would be most useful to provide in specific cases. In some cases, a spamtrap operator may choose to send reports to internet providers who have announced their willingness to receive them.

Many spamtrap operators consider a mail sender's request or insistence on information beyond the above recommendations as an attempt to subvert the spamtrap's integrity and purpose. It is recommend that spamtrap operators publish their information-release policies alongside their other trap and contact information and refer out-of-policy requests back to the policy.

Hints and Pitfalls

Spamtrap operators should consider these issues when building their spamtraps and developing heuristics to meet their goals:

- Originally-valid email addresses can be retained by legitimate senders for a surprisingly long time. A twelve contiguous month conditioning period is a compromise between practicality and accuracy; valid email may still arrive at a spamtrap years after an email address has been repurposed. Examples include:
    - The time between bursts of election-related emails can be four years or longer.

- In the case of product recalls or class action lawsuits, court-mandated emails to affected individuals may be sent a decade or more after the address has expired and been repurposed.
- Alert emails to Emergency Management System (EMS) providers, such as firefighters, EMTs, or disaster teams, may be years between uses.

- Addresses previously used as spamtrap emails may again become valid end-user addresses. This may arise after the spamtrap domain is allowed to expire and is re-registered by an unsuspecting third party. This means that the third party unexpectedly discovers that the domain comes "preloaded" with a substantial (and possibly service-crippling) volume of email abuse. High volume spamtrap domains should be disposed in a responsible fashion – e.g., transferring it to another spamtrap operator or warning the new owner of the problem.
- Not all email bounces have the MAIL FROM set to null ("<>"), despite being a violation of RFC 5321. Often, the spamtrap may need to resort to looking for "mailer-daemon," "postmaster" and other terms in the FROM: lines.
- Most senders and servers will eventually identify SMTP permanent "550 no such user" rejections as indicating that the user's email address no longer exists. It is less likely that they will identify domains with non-existent or non-functional mail servers as representing no-longer-valid email addresses. Twelve contiguous months of "550 no such user" is more effective than twelve months of "no mail server for domain" for repurposed spamtrap address conditioning.
- Email addresses are frequently mistyped. Thus, emails to never-existing addresses can sometimes be typing mistakes or indicative of non-confirmed opt-in lists, but are not necessarily the type of abuse the spamtrap is intended to find.
- Spamtrap data for repurposed addresses may be personal information protected under law. Privacy issues must be identified and retention/protection policies developed, including NDAs for data that is shared with others.
- Virtually any email address that receives spam can also be forged as the sender of spam. This means that the spamtrap will probably be receiving bounces of spam sent to others, auto-responses, and mailing list opt-in confirmation requests. The spamtrap operator might choose to ignore bounces and autoresponders, based on the spamtrap goals. A spamtrap operator should ignore opt-in confirmation requests. Most bounces can be readily identified (see RFC 5321). However, many mail servers generate bounces incorrectly. Autoresponders and opt-in confirmation requests are much more difficult to identify reliably. Fortunately, they are much less frequent than bounces.
- Some spam has null return addresses, making a spam message look like a bounce. Careful analysis is required to identify such spam.
- Email abusers – botnets in particular – attempt to hide the originators' identities and locations. The headers of abusive emails often contain forged headers to hide or obfuscate such information. Spamtrap operators must understand what parts of the message they can trust, paying particular attention to data being extracted by automation.

  For example, spamtraps should not depend on headers to automatically determine the source IP address of an email. The only reliable indicator is the IP address of the machine that sent it to the MX address of the spamtrap, and that should be acquired from the spamtrap server's operating system, which is the "peer address" or the Received header the spamtrap server inserts (the last

routing header). Previous Received headers should not be trusted, unless the system can determine whether they are trustable, for example, by knowing something about the chain of servers involved. This is far more difficult than it appears, so it is always safer to ignore received lines prior to the one the spamtrap inserts.

- Simple counting methods for reputation generation often suffer from sample sizes being too small to be statistically relevant. Do not assume too much from small spamtraps, those receiving less than 10,000-50,000 emails per day. The spamtrap operator should always be aware of this fact and should continuously monitor their spamtrap and analysis results for accuracy and reliability.

Conclusion

Spamtraps are a valuable tool in abusive email research. There are a number of technical and operational considerations in building, operating and using spamtraps that are not immediately evident to first time or even experienced researchers. This document attempts to distill many of these considerations into one resource.

References

RFC 5321 Klensin, J., "Simple Mail Transfer Protocol." http://tools.ietf.org/html/rfc5321, October 2008.

RFC 5965: Shafranovich, Y., Levine, J., and M. Kucherawy, "An Extensible Format for Email Feedback Reports," http://tools.ietf.org/html/rfc5965, August 2010.

RFC 6449: Falk, J., Ed., "Complaint Feedback Loop Operational Recommendations," http://tools.ietf.org/html/rfc6449, November 2011.

RFC 6471: Lewis, C., Sergeant M., "Overview of Best Email DNS-Based List (DNSBL) Operational Practices," http://tools.ietf.org/html/rfc6471, January 2012.

RFC 6590: Falk, J., Kucherawy, M., "Redaction of Potentially Sensitive Data from Mail Abuse Reports," http://tools.ietf.org/html/rfc6590, April 2012.

RFC 6650: Falk, J., Kucherawy, M., "Creation and Use of Email Feedback Reports: An Applicability Statement for the Abuse Reporting Format (ARF)," http://tools.ietf.org/html/rfc6650, June 2012.

VERP: D. J. Bernstein, "Variable Envelope Return Paths" https://cr.yp.to/proto/verp.txt, February 1999.

M3AAWG075-A