Distributed Content Validation

A decentralized method for reducing spam

Galen Hussey

ghussey@gatech.edu

Abstract

Current anti-spam techniques (mainly filtering) only cause spammer adaptation.[6] A few years ago, this wasn't extremely important, but now that we have seen predicted increases of 63% for this year over the 13.4 billion spams per day sent in 2006, it has become a source of tremendous revenue for various unscrupulous organizations.[5] This has led to attacks on any major group that manages to threaten their revenue, such as Israel-based Blue Security (which was aggressively targeted and eventually shut down), and the SpamHaus project. The problem with current anti-spam services is their reliance on central servers to handle filtration and coordination of their endeavors, as any central point of attack is vulnerable to the massive infected computer ("bot") networks controlled by spammers[4]. This paper proposes a distributed architecture that will allow fast response times to new spam campaigns, provide resilience against distributed denial-of-service (DDOS) attacks, and will place the anti-spam mechanisms that typically require central servers onto the very infrastructure that the spammers rely on for delivery of their messages.

1. Introduction to the Spam Problem

I'm sure that we've all experienced the annoyance of an inbox clogged with unsolicited emails, bulletin boards covered with pages of advertising posted by spammers, or perhaps even had to deal with an infestation of malware used to send spam. Of course, our filtering techniques are improving every day - our annoyance is becoming limited to picking out the occasional legitimate email from our spam folders, or perusing anti-spam logs on our forums. The problem is, out of sight and out of mind doesn't mean non-existent. The more we filter out, the more spammers send. While sending email costs them nothing, the cost of carrying the traffic and filtering it falls to the ISPs and therefore to us.

Until fairly recently, the people involved in the spam community were mostly unorganized. However, the latest trends show that spammers have banded together with groups involved in credit fraud, identity theft, malware, and extortion. This gives them far more power and resources than ever before. Since large organized groups are now in control of massive bot networks (such as the Storm botnet), spammers and other groups are able to rent time on the networks for any purpose, be it DDOSing a rival, sending spam, or breaking passwords.[10] There are many examples of this activity, such as the attacks on 419Eater, ScamWarners, CastleCops, scam.com, scamfraudalert.com, and Artists against 419. [9] Three thousand bots were even rented from the Reactor net's crime syndicate to send millions of pro-Ron-Paul political spam emails. [8]

Naturally, these large-scale attacks and operations often have extreme effects on the internet infrastructure. For example, when Blue Security was taken down, researchers claim that a third of the internet also went down for a short time during the DDOS attack. Since the spammers attacked major traffic routing points to cut Blue Security off from the rest of the internet, other major sites became inaccessible as well.[11] For perspective, a group running a large botnet attacked two of the root name servers - one of the most robust and important servers on the internet - merely to test the botnet's capabilities.[7] They have no ethical limitations, and their networks are difficult to filter, since they are built from our own compromised home and business computers.

2. An Effective Solution

It is a difficult situation when the very people you are trying to protect are inadvertently providing resources to their own attackers. It is also probable that the majority of users on any peer-based solution will be malicious. So, there are three main points to consider:

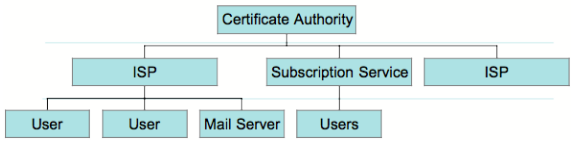
1. A minority must be able to trigger the system in the face of an overwhelming majority
2. The ability of the overwhelming majority to falsely trigger the system must be minimized

3. The system must not be rendered non-functional by an attack on the minority

There are two time periods over which these items need to be addressed. There must be an initial response to new spam campaigns in the short term - anywhere from five minutes to five days. After this period is over, however, most spam campaigns have run their course, and no longer need to be dealt with in the same manner. This limits the data set that the fast-response mechanism is required to consider. A different mechanism should be used to deal with the ever-increasing set of information required after the short-term period has elapsed.

3. High-level Architecture

The most robust way to handle these two distinct time periods is by using a tiered model. Long-term data should be handled by peers that are always on, have good internet connections, and are resistant to attack from malicious entities. Internet service providers (ISPs) are perfect for this middle tier, since they are already well-equipped for these tasks. These long-term providers, due to their small number and very static nature, should be fairly easy to authenticate using the top tier, which is a central certificate authority.



To provide the fastest response to new threats, our mechanism needs to have access to as many data collection points as possible, with as little delay as possible. For this reason, the fast-response mechanism will reside on every user's computer. That way, the load is distributed, and the larger the spam campaign (and, therefore, the threat), the larger the number of peers that can report the spam to the network. Since the fast-response mechanism only stores short-term data, it will be sufficiently light-weight to run on computers typically found in homes and workstations without using an undue amount of resources. This architecture solves all three issues mentioned above:

A minority must be able to trigger the system The protocol only contains messages that mark content as invalid, so there is no way to falsely allow malicious content to be validated by other peers.

The system must minimize false triggers The middle tier of the command hierarchy will maintain whitelists, to prevent legitimate traffic from being invalidated. Messages that attempt to invalidate content must provide a context for the invalidation, and the sender's and receiver's contexts must match.

The system must weather attacks The upper tiers are run on the very infrastructure the spammers rely on for the delivery of their messages, so there will be no benefit in attacking them. That way, the whitelists and long-term blacklists will stay intact. The lower tiers are the people they are attempting to deliver the messages to, so attacking them also has little benefit at high peer populations. At lower peer populations, the checks and balances on invalidation messages should help.

4. Protocols

This architecture was designed with Java's JXTA library in mind, which takes care of the peer-to-peer framework.[2] However, since this approach is modular, any peer-to-peer technology that supports peer discovery, broadcasting, and messaging should be sufficient.

4.1 Certificate authority

The certificate authority maintains a list of ISPs and middle-tier entities. It is responsible for issuing and signing certificates for these entities, and for providing peers with updated certificates when the need arises. To provide peers with updated certificate information, the certificate authority uses the same messaging structure as the middle-tier nodes.

4.2 Whitelist and blacklist authorities

The middle-tier protocol differs from the general peer protocol in several ways, to provide additional functionality. Any of these messages sent from a middle-tier peer must be signed with a certificate.

Validate certificate :

When this message is received, this node must check to see if the certificate is valid or not. If it is known to be valid, or if it is known to have been revoked, a validate message is constructed (with flags altered to show whether the certificate is valid or not) with the queried certificate as payload. This message is then padded and signed, and broadcast or sent to the requesting peer.

Authoritatively validate content :

This message is the middle-tier counterpart to the bottom-tier "Validate content" message. If we receive a validation query from a peer, and this node's blacklist or whitelist contains an entry matching the message's payload, then we must respond. A message is constructed with the proper message type, flags set depending on whether the content is valid or invalid, and the content as the payload. The message is then padded, signed, and broadcast. Note that this message is broadcast, in order to pre-emptively store this in the short-term cache on all peers.

4.3 Bottom-tier peers

The bottom-tier peers should comprise the majority of non-malicious entities in the peer group. They are designed to continue functioning, albeit in only a fast-response capacity, even if the upper tiers are suppressed. There is only one message type required for these peers to work effectively.

Validate content :

The query for this message contains a payload that consists of the content to be validated. If this message is received, this node's database/cache is queried to see if we have decided this content is invalid. If we believe the content is invalid, we send a message back with the same type, different flags, and the context/fingerprint (identifying why we consider the content to be invalid) as the payload.

If we receive a response to one of these messages, the payload is checked against the fingerprint/context of the content we're attempting to validate. If the two contexts match, the content is considered to be invalid, and appropriate action is taken to both store this information in our database and prevent the content from being delivered.

5. Usability

5.1 Wide-scale deployment

Obviously, the primary objective of a system like this is to eliminate coordinated attacks against internet infrastructure. In order to do so, the cooperation of a fairly large number of peers is recommended, but this makes initial adoption problematic. If adoption begins with the major internet entities and ISPs, there is a period where the number of peers will be low enough to allow effective DDOS attacks. However, support from the middle tier would provide a necessary base for adoption by individual users. If consumers are the initial peers, it is likely that they would not be considered a sufficient threat in time to counter their growing numbers. It is possible, though, that consumer growth would be slow without the long-term stabilization a well-populated middle tier would give.

In my opinion, the best option for real-world deployment would start with the all-in-one ISPs. These companies (such as AOL) offer their users a complete package for internet connectivity, including anti-virus and anti-spyware tools. If these companies adopted the system and packaged bottom-tier peer software with their toolkits, there would be both long-term support and attack resistance simultaneously. Since their audiences are a more common target for spammers, they would also stand to see greater benefit than other services.

5.2 Other applications

This architecture is not limited to email spam filtration. A broad bottom tier would be the perfect platform from which to detect fast-flux based hosting[3], DNS poisoning[1], or even differentiate between commercial and user-created postings on website forms. All that is required is the creation of a fingerprinting method for the context of the content in question.

5.3 Vulnerabilities

While the system is resistant to outright attack, care must be taken when deciding on a fingerprinting method. A method that is too strict could allow almost-matching messages to slip through the cracks, resulting in invalid content being delivered. Spammers will try to "game the system", as they have done with search engine optimization.

With the recent debates over internet neutrality, the motivations and capabilities of ISPs are not as clear as they once were. If a middle-tier entity were to go rogue and maintain blacklists and whitelists that were at odds with its peers, for financial gain or damage to rivals, all peers could suffer.

During the initial period, bottom-tier peers will have to deal with spammers' increasingly complex attempts to undermine the effectiveness of the network. The main vulnerability in this instance is a function of the short-term cache - if the cache is too large or if access times are too high, a flood of false invalidation messages could bog down the consumer's computer.

5.4 Traffic constraints

Since this system does use broadcasts, extremely large peer groups will require very large amounts of traffic, depending on the size of local caches, response speeds of the middle tier, etc. Pre-emptive caching, such as that done by the middle tiers on authoritative validation messages, could be used by the lower-tier peers to reduce the number of broadcasts surrounding an email campaign. However, it could open up an opportunity for malicious peers to flood that cache, rendering it completely ineffective. The obvious solution is to issue certificates and require registration for each peer on the network, and cull any peers that consistently respond maliciously. Dealing with the complexities of such a mechanism would be at odds with the simplicity of architecture this system is attemping, and could open up other avenues of attack, to say nothing of consuming significantly more resources. Still, if it is done in a commercial environment it may be worth it.

6. Future work

This architecture provides a robust framework for attack-resistant content validation. The method of implementation for the various modules required by any specific solution will greatly influence the effectiveness of a real-world deployment. The majority of work will be creating fingerprinting modules for each type of content to be validated. A significant amount of research has already been done for email spam, but fast-flux detection is a fairly new problem, as is detecting man-in-the-middle DNS attacks. Secure signing

methods need to be researched and applied to this problem, especially for middle-tier peers. Traffic patterns need to be evaluated with different populations for each tier, as well as during different attack types. Research into the best database technologies and sizes for the short-term cache would also be beneficial.

References

[1] 68,000 open recursive dns servers behaving maliciously. http://www.circleid.com/posts/malicious_open_recursive_dns_servers/.

[2] Jxta(tm) community projects. https://jxta.dev.java.net/.

[3] Know your enemy, fast flux service networks. http://www.honeynet.org/papers/ff/fast-flux.html.

[4] Okopipi anti-spam project. `http://www.okopipi.org/`.

[5] The can-spam act: Requirements for commercial emailers. `http://www.ftc.gov/bcp/conline/pubs/buspubs/canspam.shtm`, April 2004.

[6] Adblock.org. `http://www.adblock.org/`, April 2007.

[7] Dns attack: Only a warning shot? http://www.darkreading.com/document.asp?doc_id=116685&WT.svl=news1_1, February 2007.

[8] Researchers track ron paul spam back to reactor botnet. http://arstechnica.com/news.ars/post/20071206-researchers-track-ron-paul-spam-back-to-reactor-botnet.html, December 2007.

[9] Spammers launch denial of service attacks against anti-spam sites. http://arstechnica.com/news.ars/post/20070912-spammers-launch-denial-of-service-attacks-against-antispam-sites.html, September 2007.

[10] goombah99. Slashdot — sec halts trading on spam driven stocks. `http://it.slashdot.org/article.pl?sid=07/03/09/0235222`, March 2007.

[11] Robert Lemos. Blue security folds under spammer's wrath. `http://www.securityfocus.com/news/11392`, May 2006.

Special thanks to Chris Lee and Frank Rietta