



Applying Collaborative Anti-Spam Techniques to the Anti-Virus Problem

Adam J. O'Donnell, Director of Emerging Technologies, Cloudmark, Inc.
Vipul Ved Prakash, Founder and Chief Scientist, Cloudmark, Inc.

ABSTRACT

One of the most effective techniques available for combating spam is the widespread application of collaborative filtering, where members of a community identify and vote on spam messages. A collaborative system is built on the, now proven, premise that individual users can, with high accuracy, determine the difference between spam and legitimate email. However, it is not obvious that this also holds true for email-borne malware threats, whose sole indicator is often only a malicious attachment to an otherwise seemingly legitimate email. We present data and analysis of our success in applying a collaborative filter, originally designed for anti-spam, to the anti-virus problem. We also present our results from specific case studies, including an analysis of the CME-24 outbreak of early 2006. We show that not only is a collaborative filter effective for filtering viruses, but also that the community begins filtering the virus within minutes of its initial detection—and with an extremely low false positive rate.

INTRODUCTION

It is widely accepted that computer users operate in a dangerous world, where expressions like “rapidly evolving threats” are no longer considered sensationalism. Computer security professionals have difficulty keeping up with the latest threats, with confusion frequently arising over something as simple as the names of new threats. The problem has become severe enough that the US Federal Government, through the contractor MITRE, has interceded and normalized the naming schema through the formation of the Common Malware Enumeration initiative. As domain experts struggle to keep pace, ordinary users are left in a world of confusion.

The security issues that home users faced over a decade ago were quite different than those faced today. Typically, viruses and other malicious code propagated through infected floppy disks. The transmission of viruses was slow and limited by the rate at which physical media moved. From a containment perspective, the slow rate of viral transmission meant the anti-virus (AV) community could contain and control viruses through its standard channels of delivering software updates.

Both empirical evidence and improved analytical models of viral propagation show that viral epidemics cannot be avoided solely due to low transmission rates. As software became more complex, it became easier to compromise. The adoption of this extremely complex software by the general public, coupled with the introduction of pervasive networks has dramatically increased virus propagation rates and expanded the number of possible vulnerable software targets. An improved understanding of social network behavior shows that any computer virus that propagates across a human social network, such as email viruses, has the potential to turn into an epidemic [1].

The computer security industry has evolved considerably in response to these challenges. There are thousands of security products on the market, available in hundreds of form-factors, to defend against all manner of threats in all manner of environments. Network firewalls, authentication systems, encryption systems, anti-virus tools, and anti-spyware tools are almost universally deployed. Even with all of these available technologies, virus and malware problems persist because Internet security attacks spread faster than the small security teams working in operation centers can react.

Consider spam. In conventional security terms, spam constitutes unauthorized access to an unauthenticated service—namely, our inbox. In practical terms, however, we all recognize that our mailboxes are clogged everyday with unwanted messages. Users see daily evidence—the volume of spam and virus-laden email—that conventional security solutions are outstripped by the rate and innovation of spam and viruses. Entire sub-industries, including the messaging security industry of which the authors are members, have developed to combat these two security threats.

CONVENTIONAL ANTI-VIRUS AND ANTI-SPAM METHODOLOGY

Conventional anti-virus software works on a basic principle. The software examines every file that comes into the machine and generates a unique signature for each file. This signature is then checked against a database of signatures of known viruses. Engineers isolate and analyze samples of computer viruses to create these signatures. Ideally, signatures uniquely identify a virus strain without colliding with legitimate software. If the database is updated frequently and the signature is sufficiently selective, then all viruses are filtered out of the system before they can do any harm; otherwise, critical system software may be flagged as malicious or, conversely, virus variants may evade the filter. An in-depth treatment of this topic can be found in Szor's *The Art of Computer Virus Research and Defense* [2].

The anti-spam industry has settled on three major methodologies:

- Network-Layer Analysis
- Heuristics and Machine Learning
- Fingerprinting

Network-Layer Analysis

Network-layer analysis encompasses IP blacklisting, mail delivery rate limiting, and several other techniques that depend upon traffic analysis at the network layer alone. IP blacklisting works by blocking all incoming traffic from mail servers that are known to send spam or that have the potential to send spam due to misconfiguration. Servers that want to reject mail from known spam mail servers can fetch a list of these mail servers by IP address (commonly referred to as RBLs, or Real-time Blackhole Lists) and block all inbound connections from these systems. While this is a powerful tool for stopping the most abusive mail servers on the internet, blocking every incoming message from a specific mail server, even one predominately used for spam, may increase the false positive and critical false positive rate of the spam filter. Traffic analysis techniques also encompass email delivery rate limiting, where MTA connections are throttled if a single connection attempts to deliver a large volume of mail to a large number of users. While it does slow down the rate at which spam is delivered, this technique is difficult to apply for large volume customers, such as enterprise users.

Heuristics and Machine Learning

Heuristic techniques are human-written rules that look for certain behavioral differences between legitimate mail and spam. For example, the increasingly large volume of image-based spam has driven many anti-spam vendors to create ad-hoc rules for determining the disposition of attachments. Heuristics that are automatically updated and refined by machines based upon training sets are classified as Machine Learning (ML) techniques. One popular ML technique, namely Naïve Bayesian classifiers, tokenizes mail content into words and phrases (or other linguistic units) and registers the probability of the appearance of various words and phrases in spam and legitimate messages. The learned set of linguistic units and their corresponding probabilities constitute the "hypothesis" used to classify incoming mail. While machine learning techniques must be trained incrementally by the recipient, the training events are rare, compared to the frequency of incoming mail. Most implementations come with a built-in hypothesis that serves as a starting point to offset the training requirements of the tool. Once trained, machine learning-based systems are quite accurate at identifying legitimate communications and reasonably good at identifying spam. They are known to perform best in single-user environments where the training corpus accurately reflects specific user preferences. Most real-world deployments of statistical text classification are augmented with orthogonal classifiers, such as blacklisting, to derive acceptable spam detection performance.

Fingerprinting

Fingerprinting methodologies are similar to anti-virus signatures in that the fingerprint uniquely identifies strains of spam by extracted portions of the message content. Unlike classic virus signatures,

fingerprints are automatically generated from the e-mail content. Due to their rapid generation and dissemination, fingerprints do not need to be as tolerant to evasion by mutation as virus signatures. Two methods exist for automatic spam fingerprint generation, bulk detection and collaborative filtering. Bulk detection works by automatically issuing a fingerprint for any content that matches certain traffic characteristics, such as rapid transmission. Collaborative filtration, the method employed by the author's company, relies on a community of users to submit fingerprints identifying spam messages and to issue alerts, or contest a fingerprint, when legitimate e-mail has been identified as spam.

MOTIVATIONS FOR ANTI-SPAM AND ANTI-VIRUS TECHNOLOGIES

While viruses and spam are both unsolicited and unwanted content that arrives at our computers, they exist in somewhat dissimilar ecologies. We find that the underlying social and economic factors that drive the creation of spam and viruses are fundamentally different. As a result, the emergent spam and virus threats are distinct, and have been addressed distinctly by technologists. Consider the following differences driving the proliferation of spam versus viruses:

Virus writers typically target computers, while spam writers typically target minds. The goal of a classic computer virus is to alter the execution of software, such as causing the operating system to delete all the files on the hard drive on a given day. The goal of spam, on the other hand, is to convince someone to take a specific action, such as buying a product, or, in the case of phishing, replying with bank account information.

Many more people can write spam than can write viruses. Virus writers are extremely computer proficient as compared to the general populace. Virus writers discover new or detect existing flaws in an operating system to propagate an attack. Even simply altering or mutating an existing virus requires an understanding of computer code. By contrast, spammers are marketers. The skills required to create a spam message are the same as those needed to write an e-mail or create a graphic. These skills are possessed by a far larger population.

The combination of these two factors leads to a far greater number of spam mutations than virus mutations. Thankfully, many more people can recognize a spam message than can recognize a virus. The opinion of the general population as to what messages are spam and what messages are not spam forms an accurate depiction of the content's disposition, and can be used to generate an accurate anti-spam filter, as detailed below.

CLOUDMARK AUTHORITY ANTI-VIRUS: COLLABORATIVE FILTERING

Conventional anti-virus software relies on teams of highly trained experts to extract computer virus signatures from binary sources. Anti-spam solutions that use heuristic engines rely on a similar expert team to create regular expressions. Alternatively, we can use the large pool of e-mail readers to differentiate between spam and legitimate mail, and then generate precise, orthogonal fingerprints that accurately identify a message and its mutations, as spam has always been relatively easy to recognize. This allows for the creation of a collaborative filtering-based, anti-spam solution—where the masses of e-mail users can decide, as a group, on the nature of individual messages by nominating messages as either spam or “not spam”. Rather than waiting on an expert to detect and extract a virus feature, as is the case with conventional anti-virus, collaborative filtering anti-spam techniques can leverage the proverbial “Wisdom of the Crowd” and quickly and efficiently flag what is spam and what is “not spam” and then automatically filters the messages associated with these fingerprints.

CLOUDMARK AUTHORITY ANTI-VIRUS SYSTEM

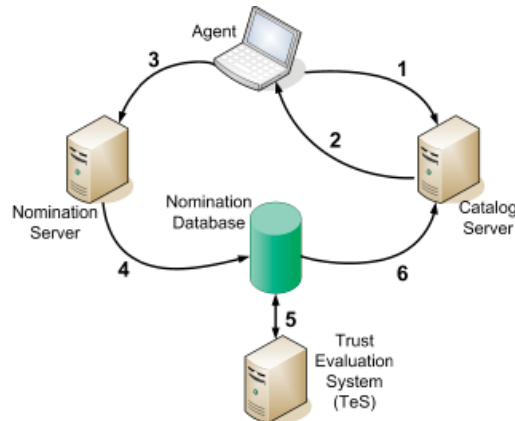


Figure 1: The process flow of the Cloudmark system.

An agent residing on a user's desktop computes a fingerprint of a new e-mail and submits this fingerprint (1) to the Spam Fingerprint Catalog server. If the Catalog server has the fingerprint in its database, the server tells the user (2) that the message has been flagged by the community as spam. If the fingerprint is not in the Catalog server, and the recipient feels that the message is spam, the recipient sends the fingerprint (3) to the Nomination server, which in turn, inserts the fingerprint into the Nomination database (4). The Cloudmark Trust Evaluation System®, or TES, continually watches (5) the Nomination database to see if there are any new fingerprints that have been submitted by multiple trusted e-mail recipients. If enough trusted recipients submit the same fingerprint, the fingerprint is promoted to the Catalog server, and the process continues. The system does not depend solely on human reporters. Honeypots are frequently added to the community and treated exactly the same as a high-volume human reporter. The rapid response and wide scope of Cloudmark's honeypots increases accuracy while reducing the number of responses required from humans.

The concepts behind Cloudmark's collaborative filter, known as the Cloudmark Network Classifier™ [3], are relatively simple. Users first submit a set of fingerprints derived from incoming e-mail. If the fingerprint is already in the catalog of malicious content, then the user is told that the e-mail is spam and it is moved to the user's spam folder. If the fingerprint is not in the database, and the user feels the content is spam, then the user nominates it as such, and its associated fingerprint is added to a database for temporary storage of new fingerprints. If a sufficient number of community members agree that the content is spam, then fingerprints are moved to the spam catalog and the process continues.

Community members who correctly identify spam in a timely manner are rewarded by becoming trusted members of the filtering community. Their feedback is weighted more than community members who have a lower trust rating. Conversely, if they report incorrectly, their trust level decreases, and their opinions count less in the future. The process is surprisingly fast, and allows for new spam to be identified and filtered in a matter of minutes. False positive reports (legitimate e-mail incorrectly identified as spam) are quickly remedied and unblocked by the larger community of highly-trusted

users, while the trust rating of the original reporter is significantly lowered.

APPLYING COLLABORATIVE ANTI-SPAM TO ANTI-VIRUS

The previous assumptions regarding the skills that have differentiated virus and spam writers have diminished over the past five years. The number of easily exploitable software packages has decreased, so now virus creators transmit more viruses via e-mail and attempt to convince the recipient to open attachments containing malware [4]. Additionally, the use of high level programming languages and the subsequent open-source distribution of virus code has enabled neophytes to easily modify preexisting virus code—to the point that it evades previous signatures. This is commonly seen in the multiple MyDoom and MyTob variants that seem to endlessly propagate across our inboxes.

E-mail viruses are often delivered with a spam message, where the goal of the spam is to convince the recipient to open the virus. The collaborative community doesn't actually need to determine if the attachment itself is a virus; they only need to recognize that the message that contains the attachment looks like spam. The back-end system that collects all of the spam nominated by the community can extract the attachment, use an algorithm to determine if it is a piece of computer code, and then add it to a table of computer virus signatures. By examining the bits of spam that come before an e-mail virus attachment, the community becomes a large, distributed, anti-virus research lab which is capable of quickly identifying new viruses in the wild. Using collaborative filtering for combating viruses is not so much a revolutionary idea in theory, as it is a revolutionary change in practice. Academic researchers have discussed collaborative defense techniques as a critical element in any virus and worm mitigation strategy [5]. The community's ability to recognize spam implicitly allows the community to recognize viruses as well. As long as individuals in the community recognize the content as being something that should be filtered, it is possible to generate a signature scheme that can fingerprint the content. For example, it is possible to generate a fingerprinting scheme that is specific to executables arriving as an e-mail attachment. If recipients recognize the body of an e-mail as spam, then they can submit fingerprints for both the e-mail and the executable attachment to the Cloudmark back-end.

What happens if a virus is not preceded by a piece of spam? The honeypot pool, augmented with abandoned accounts from large enterprise customers, provides an effective sensor network for the detection of new viruses. Additionally, individual community members knowledgeable in anti-virus will, as they have in the past, submit samples of new viruses to the back-end.

FINGERPRINTING A VIRUS

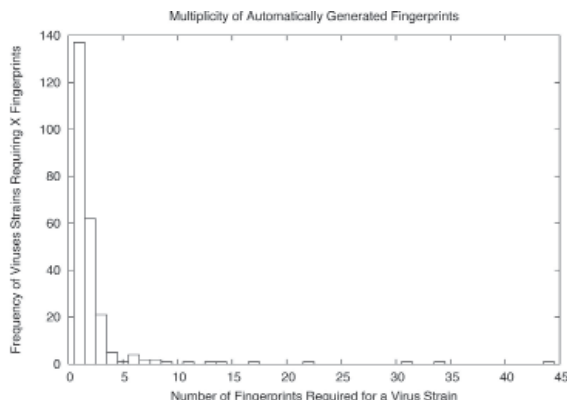


Figure 2: The frequency of virus strains requiring a given number of fingerprints to cover the outbreak.

The vast majority of virus outbreaks only require a single fingerprint to cover the virus and its variants, while a handful of viruses require dozens of fingerprints to cover the mutations.

The goal of fingerprint-based, anti-spam schemes is to locate, using either probabilistic or deterministic algorithms, the invariant aspects of spam messages and extract these for the fingerprint. Presently, Cloudmark uses seven different fingerprinting algorithms that use orthogonal methodologies to identify and encode invariant information. The majority of our anti-spam fingerprinting schemes were designed to work at the byte level, making them content, encoding, and format agnostic. We have discovered that these alone provide excellent coverage for x86 executables—as well as script and interpreted language viruses.

While non-executable specific fingerprinting schemes have shown to be extremely effective at tracking viruses, Cloudmark developed a specific fingerprinting scheme for x86 binaries that extracts instructions from the code section of the binary while skipping non-critical instructions. This fingerprinting scheme disassembles an executable and extracts potentially invariant sections of the code. The algorithms can't produce fingerprints with the same multiplicity as human-generated fingerprints; nor can it map to all possible mutations of the same virus. However, the low rate of fingerprint collision between malicious and innocuous content, referred to as cross-class collision [3], is low enough to allow for inclusion in a largely autonomous system. As shown in Figure 2, the majority of viruses are covered with a single signature, while a handful of viruses may require more than 10 signatures to cover all of its variants.

PERFORMANCE OF CLOUDMARK AUTHORITY ANTI-VIRUS

Testing the performance of a collaborative anti-virus solution is not a trivial task, as it depends on a steady stream of new viruses for determining its primary performance metric—specifically its observed time of coverage of new viruses as compared to other available solutions. We can measure the coverage of our approach by examining the amount of time it takes for publicly available products to “label” a binary that the community has identified as being a virus. It is also possible to infer the timeliness of our virus fingerprints by examining the rate at which the fingerprint is found in mail and looking for the standard profile of an epidemic outbreak; if we do not observe an increase in global infection attempts, followed by a steady decrease as systems are scrubbed of the contagion, then our virus fingerprint was issued far too late to be effective.

Signature Timeliness

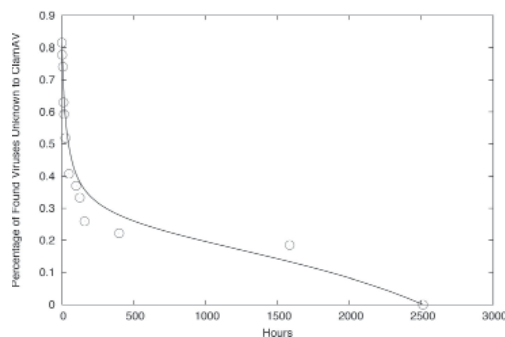


Figure 3: The fraction of viruses identified by Cloudmark Authority Anti-virus system and not by ClamAV, as a function of time.

As shown in Figure 3, 80% of the new fingerprints are novel to the open-source ClamAV product at the time of their identification by the Cloudmark community. Of the 80% which are not initially identified by ClamAV, over half are not identified after 2 days. Similar delays were observed in the commercial products we have tested as well. As the AV industry shifts toward requiring sub-hour response for outbreak prevention, a multiple-day gap will no longer be acceptable.

TRACING THE EVOLUTION OF AN OUTBREAK

Recording the number of times each fingerprint is seen by the Cloudmark network is an important metric for deciding when a fingerprint should be “retired”, or aged out of the system. A benefit of this design is that it is possible to estimate the prevalence of a virus over time by tracking the number of times fingerprints associated with infection attempts are checked by client-side software. While this does not provide a direct measure of the number of infected systems globally, it is sufficient for observing the typical virus lifecycle of infection and remediation.

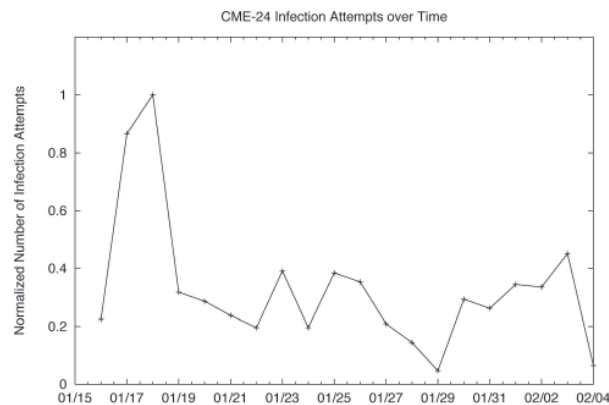


Figure 4: The normalized rate of infection transmission by CME-24 hosts to Cloudmark users. The period of large-scale infection attempts on the 17th was captured by the system.

While reports of new variants are received almost daily, the vast majority of virus outbreaks are not significant enough to warrant the attention of the media and the security industry as a whole, thus limiting our ability to validate our observations. The CME-24 outbreak provided us with our first opportunity to track the lifecycle of an openly discussed e-mail worm event.

Figure 4 provides a plot of the normalized infection propagation attempts observed by our users. The graph begins when our customers first noticed and submitted a sufficient number of reports to block the virus, specifically at 13:08 GMT on January 16th, 2006. The outbreak reached a local maximum two days later, as media reports began to surface that discussed a new and potentially dangerous worm that had been designed to destroy user’s personal files on the 3rd of February. Even at this peak, a relatively small number of infection attempts were even recorded; less than .25% of Cloudmark Authority Anti-virus users saw virus propagation attempts. Later media reports confirmed that the outbreak had largely been a bust.

CONCLUSION

Both conventional anti-spam and conventional anti-virus systems are dependent upon the knowledge of a select and expert few, who constantly tune their systems and add new signatures to combat evolving threats from mass mailers and virus writers. Our work over the past five years has shown that the collaborative filtering paradigm works exceedingly well for both spam and viruses. More importantly, the changing economics of the spam environment will necessitate solutions that are able to rapidly adapt to new threats; currently the collaborative filtering architecture is the only one suited to this new landscape.

The collaborative filtering architecture is not limited to combating viruses and spam. Phishing, spyware, and a whole host of other security problems that are recognizable by individuals in the community can be solved using the same overarching concept of leveraging community consensus against these threats. As the time to remediation for security threats continues to decrease, collaborative security frameworks, such as the Cloudmark Network Classifier, will prove to be one of the only means of delivering the security response required on the time scales demanded by customers.

ACKNOWLEDGEMENTS

The authors would like to thank Jason Harbert for his work in collecting data on the CME-24 infection profile, and Sophy Ting O'Donnell for her assistance editing this paper.

REFERENCES

1. Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos, "Epidemic spreading in real networks: An eigenvalue viewpoint," in 22nd Symposium on Reliable Distributed Systems, IEEE Computer Society, October 2003.
2. P. Szor, *The Art of Computer Virus Research and Defense*. Addison-Wesley Professional, 2005.
3. V. V. Prakash and A. O'Donnell, "Fighting spam with reputation systems," *Queue*, vol. 3, no. 9, pp. 36–41, 2005.
4. L. Birdwell, "10th annual computer prevalence survey," tech. rep., ICSA Labs, 2004.
5. D. Moore, C. Shannon, G. Voelker, and S. Savage, "Internet quarantine: Requirements for containing self-propagating code," in Twenty-Second Annual Joint Conference of the IEEE Computer and Communication Societies (INFOCOM), pp. 1901–1910, March – April 2003.

For more information
visit us at www.cloudmark.com