Anagram: A Content Anomaly Detector Resistant to Mimicry Attack[1]

Ke Wang Janak J. Parekh Salvatore J. Stolfo

Computer Science Department, Columbia University
500 West 120th Street, New York, NY, 10027
{kewang, janak, sal}@cs.columbia.edu

Abstract. In this paper, we present *Anagram*, a content anomaly detector that models *a mixture of high-order n-grams* ($n > 1$) designed to detect anomalous and "suspicious" network packet payloads. By using higher-order n-grams, Anagram can detect significant anomalous byte sequences and generate robust signatures of validated malicious packet content. The Anagram content models are implemented using highly efficient Bloom filters, reducing space requirements and enabling privacy-preserving cross-site correlation. The sensor models the distinct content flow of a network or host using a semi-supervised training regimen. Previously known exploits, extracted from the signatures of an IDS, are likewise modeled in a Bloom filter and are used during training as well as detection time. We demonstrate that Anagram can identify anomalous traffic with high accuracy and low false positive rates. Anagram's high-order n-gram analysis technique is also resilient against simple mimicry attacks that blend exploits with "normal" appearing byte padding, such as the blended polymorphic attack recently demonstrated in [1]. We discuss *randomized n-gram models*, which further raises the bar and makes it more difficult for attackers to build precise packet structures to evade Anagram even if they know the distribution of the local site content flow. Finally, Anagram's speed and high detection rate makes it valuable not only as a standalone sensor, but also as a network anomaly flow classifier in an instrumented fault-tolerant host-based environment; this enables significant cost amortization and the possibility of a "symbiotic" feedback loop that can improve accuracy and reduce false positive rates over time.

1 Introduction

The current generation of Network Intrusion Detection Systems (NIDS) are typically ill-suited for stealthy worms and targeted attacks. Misuse and anomaly detectors that analyze packet headers and traffic flow statistics may be too slow to react to reliably detect worms that are designed to evade detection by shaping their behavior to look like legitimate traffic patterns [2]. Furthermore, signature scanners are vulnerable to zero-day exploits [3] and polymorphic worms/stealthy attacks with obfuscated exploit code [4]. Consequently, there has been an increasing focus on payload analysis to detect the early onset of a worm or targeted attack. Ideally, one would hope to detect the very first packets of an attack, rather than accumulating sufficient statistics about connection flows to detect a zero-day attack.

A number of researchers (e.g., [5-8]) have focused on payload-based anomaly detection. Approaches that have been studied include specification-based anomaly detection [7] as well as techniques that aim to detect "code-like" byte sequences in network payloads [6, 9]. In our work, we have focused on automated statistical learning approaches to efficiently train content models on a site's "normal" traffic flow without requiring

[1] This work has been partially supported by a grant with the Army Research Office, No. DA W911NF-04-1-0442.

significant semantic analysis. Ideally, we seek to design a sensor that automatically learns the characteristics of "normal" attack-free data for any application, service, network or host. Consequently, a model learned for "normal" attack-free data may be used to identify "abnormal" or suspicious traffic that would be subjected to further analysis to validate whether the data embodies a new attack.

In our previous work we proposed PAYL (short for "PAYLoad anomaly detection") that modeled the "normal" attack-free traffic of a network site as 1-gram, byte-value frequency distributions [10], and demonstrated an ability to effectively detect worm behavior via ingress/egress and cross-site correlation [11]. The sensor was designed to be language-independent, requiring no syntactic analysis of the byte stream. Furthermore, PAYL was designed to be efficient and scalable for high-speed networks and applicable to any network service. Various experiments demonstrated that PAYL achieved a high detection rate and with low false positives for "typical" worms and exploits available at the time.

However, most researchers[2] correctly suspected that PAYL's simplicity would be easily blinded by *mimicry attacks*. Kolesnikov, Dagon and Lee [1] demonstrated a new blended, polymorphic worm designed to evade detection by PAYL and other frequency distribution-based anomaly detectors. This demonstration represents a new class of "smart worms" that launch their attack by first sniffing traffic and shaping the datagram to the statistics specific to a given site to appear normal. The same principles may be applied to the propagation strategy as well as in, for example, parasitic worms. Since PAYL only models 1-gram distributions, it can be easily evaded with proper padding to avoid detection of anomalous *byte sequences*. As a countermeasure, we conjecture that higher-order n-gram modeling may likely detect these anomalous byte sequences. Unfortunately, computing a full frequency distribution for higher order n-grams is computationally and memory-wise infeasible, and would require a prohibitively long training period even for modest gram sizes.

In this paper we present a new sensor, Anagram, which introduces the use of Bloom filters and a binary-based detection model. Anagram does not compute frequency distributions of normal content flows; instead, it trains its model by storing all of the distinct n-grams observed during training in a Bloom filter without counting the occurrences of these n-grams. Anagram also stores n-grams extracted from known malicious packets in a second *bad content Bloom filter*, acquired by extracting n-grams from openly available worm detection rules, such as the latest Snort rulesets [12]. At detection time, packets are scored by the sensor on the basis of the number of unobserved n-grams the packet contains. The score is weighted by the number of malicious n-grams it contains as well. In this paper, we demonstrate that this semi-supervised strategy attains remarkably high detection and low false positive rates, in some cases 100% detection with less than 0.006% false positive rate (per packet).

The use of Bloom filters makes Anagram memory and computationally efficient and allows for the modeling of a *mixture of different sizes of n-grams* extracted from packet payloads, i.e. an Anagram model need not contain samples of a fixed size gram. This strategy is demonstrated to exceed PAYL in both detection and false positives rates. Furthermore, Anagram's modeling technique is easier to train, and allows for the estimation of when the sensor has been trained enough for deployment. The Bloom filter model representation also provides the added benefit of preserving the privacy of shared content models and alerts for cross-site correlation.

[2] Including ourselves; a proposal to study counter-evasion techniques led to the work reported herein.

Of particular interest here is that Anagram is shown to be robust against existing mimicry attack approaches. We demonstrate Anagram's ability to counter the simple mimicry attacks levied against PAYL. Furthermore, Anagram is designed to defeat training and mimicry attacks by using *randomized n-gram modeling*. The approach presented raises the bar against the enemy, making it far more difficult to design an n-gram attack against Anagram. By randomizing the portion of packets that Anagram extracts and models, mimicry attackers cannot easily guess how and where to pad malicious content to avoid detection. We also describe the use of a feedback loop between the Anagram sensor and host-based detectors, thereby updating Anagram models over time and improving its detection performance. Thus, the combination of model randomization and a feedback loop makes it more difficult to evade detection by training and mimicry attacks. The contributions of this paper include:

- A new statistical, language-independent, efficient content-based anomaly detector based upon semi-supervised training of higher-order n-gram analysis that is shown to be resistant against existing mimicry attacks. The sensor does not rely upon a specification or semantic analysis of the target applications;
- Robustness against future mimicry attacks by the use of a novel, low-overhead randomized testing strategy, making it difficult for the attacker to guess *where* or *how* to pad content;
- Development of a run-time measure of the "stability" of a network's content flow, providing a reasonable estimate of when the sensor has been well enough trained and is ready for deployment.
- A robust means of representing content-based alerts for cross-site alert sharing and robust signature generation using a Bloom Filter representation of anomalous byte sequences[3];
- A new defensive strategy showing how a symbiotic relationship between host-based sensors and a content-based sensor can adapt over time to improve accuracy of modeling a site's content flow.

The rest of the paper is organized as follows. Section 2 details the Anagram sensor and its relevant algorithms. Performance, detection rate, and false positive characteristics are presented testing Anagram against real network traffic traces infected with a collection of worms and viruses. Section 3 describes Anagram's robustness against the cleverly crafted blended polymorphic worm reported in [1], previews the possibility of new customized mimicry attacks being crafted against Anagram, and describes randomization techniques for defeating such attacks. In section 4 we present the concept of coupling a "shadow server" with Anagram and discuss how the combination can effectively support the techniques presented in section 3, as well as support robust signature generation and patch generation. Section 5 discusses related work, while section 6 concludes the paper with a call for collaboration among researchers at distributed sites.

2 Anagram – Modeling a Mixture of N-grams

Anagram's approach to network payload anomaly detection uses a mixture of higher order n-grams (n>1) to model and test network traffic content. N-gram analysis is a well-known technique has been used in a variety of tasks, such as system call monitoring [15-17]. In Anagram, the n-grams are generated by sliding windows of arbitrary lengths over a stream of bytes, which can be per network packet, per request session, or other type of data unit.

[3] The representation also permits patch generation systems to share anomalous data for local patch generation across an "application community" [13, 14].

In our previous work on network payload anomaly detection, PAYL [10, 11], we modeled the length-conditioned 1-gram frequency distribution of packet payloads, and tested new packets by computing the Mahalanobis distance of the test packet against the model. This approach is effective at capturing attacks that display abnormal byte distributions, but it is likely to miss well-crafted attacks that focus on simple CPU instructions and that are crafted to resemble normal byte distributions. For instance, although a standard CodeRed II's buffer overflow exploit uses a large number of "N" or "X" characters and so appears as a peak in the frequency distribution, [18] shows that the buffer can instead be padded with nearly any random byte sequence without affecting the attack vector. Another example is the following simple phpBB forum attack:

```
GET /modules/Forums/admin/admin_styles.php?phpbb_root_path=http
://81.174.26.111/cmd.gif?&cmd=cd%20/tmp;wget%20216.15.209.4/cri
man;chmod%20744%20criman;./criman;echo%20YYY;echo|..HTTP/1.1.Ho
st:.128.59.16.26.User-Agent:.Mozilla/4.0.(compatible;.MSIE.6.0;
.Windows.NT.5.1;)..
```

In such situations, the abnormal byte distribution model is insufficient by itself to identify these attack vectors as abnormal data. However, invariants remain in the packet payloads: the exploit code, the sequence of commands, or the special URL that should not appear in the normal content flow to the target application. By modeling higher order n-grams, Anagram captures the order dependence of byte sequences in the network payload, enabling it to capture more subtle attacks. The core hypothesis is that any new, zero-day exploit will contain a portion of data that has never before been delivered to the application. These subsequences of new, distinct byte values will manifest as anomalous" n-grams that Anagram is designed to efficiently and rapidly detect.[4]

In the following sections we will give a detailed description of Anagram, which outperforms PAYL in the following respects:

- Accuracy in detecting anomalous payloads, even carefully crafted 'mimicry attacks' with a demonstrably lower false positive rate;
- Computational efficiency in detection by the use of fast (and incremental, linear-time) hashing in its Bloom filter implementation;
- Model space efficiency since PAYL's multiple-centroid modeling is no longer necessary, and Bloom filters are compact;
- Fast correlation of multiple alerts while preserving privacy as collaborating sites exchange Bloom filter representations of common anomalous payloads;
- The generation of robust signatures via cross-site correlation for early warning and detection of new zero day attacks.

In the following sections, we will describe the mechanisms in detail and present experimental results of testing Anagram against network traces sniffed from our local LAN.

2.1 High Order N-gram Payload Model

While higher order n-grams contain more information about payloads, the feature space grows exponentially as *n* increases. Comparing an n-gram frequency distribution against a model is infeasible since the training data is simply too sparse; the length of a packet is too small compared to the total feature space size of a higher-order n-gram.

[4] Note that some attacks may not include byte sequences that are "code-like", and hence testing content for such code-like data subsequences is not guaranteed to cover all attack cases. The language independence of anomalous n-grams may be broadly applicable to these and other attacks.

One TCP packet may contain only a thousand or so n-grams, while the feature space size is 256^n. Clearly, with increasing n, generating sufficient frequency statistics to estimate the true empirical distribution accurately is simply not possible in a reasonable amount of time.

In Anagram, we therefore do not model the frequency distribution of each n-gram. Rather, we observe each distinct n-gram seen in the training data and record each in a space efficient Bloom filter. Once the training phase terminates, each packet is scored by measuring the number of n-grams it did not observe in the training phase. Hence, a packet is scored by the following formula, where N_{new} is the number of new n-grams not seen before and T is the total number of n-grams in the packet:

$$Score = \frac{N_{new}}{T} \in [0,1]$$

At first glance, the frequency-based n-gram distribution may contain more information about packet content; one might suspect it would model data more accurately and perform better at detecting anomalous data, but since the training data is sparse, this alternative “binary-based model” performs significantly better than the frequency-based approach given the same amount of training data.

We analyzed the network traffic for the Columbia Computer Science website and, as expected, a small portion of the n-grams appear frequently while there is a long “tail” of n-grams that appear very infrequently. This can be seen in Table 1, which displays the percentage of the n-grams by their frequency counts for 90 hours of CS web traffic. Since a significant number of n-grams have a small frequency count, and the number of n-grams in a packet is very small relative to the whole feature space, the frequency-distribution model incurs relatively high false positives. Thus, the binary-based model (simply recording the distinct n-grams seen during training) provides a reasonable estimate of how “normal” a packet may be. This is a rather surprising observation; as we will demonstrate, it works very well in practice. The conjecture is that true attacks will be delivered in packets that contain many more n-grams not observed in training than “normal” packets used to train the model. After all, a true zero-day attack must deliver data to a server application that has never been processed by that application before. Hence, the data exercising the vulnerability is very likely to be an n-gram of some size never before observed. By modeling a mixture of n-grams, we increase the likelihood of observing these anomalous grams.

To validate this conjecture, we compare the ROC curves of the frequency-based approach and the binary-based approach for the same datasets (representing equivalent training times) as displayed in figure 1. We collected the web traffic of two CS departmental web servers, *www* and *www1*; the former serves the department webpage, while the latter serves personal web pages. Traffic was collected for two different time periods: a period of sniffed traffic from the year 2004 and another dataset sniffed in 2006. The 2004 datasets[5] (*www-04* and *www1-04*) contain 160 hours of traffic; the 2006 datasets (*www-06* and *www1-06*) contain about 560 hours. We tested for the detection of several real worms and viruses: CodeRed, CodeRed II, WebDAV, Mirela, a php forum attack, and a worm that exploits the IIS Windows media service, the nsiislog.dll buffer overflow vulnerability (MS03-022). These worm samples were collected from real traffic as they appeared in the wild, from both our own dataset and from a third-party.

[5] The 2004 dataset was used to train and test PAYL as reported in [11], and is used here for comparative evaluation.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.