Micro-signatures: The Effectiveness of Known Bad N-Grams for Network Anomaly Detection

Richard Harang[1(✉)] and Peter Mell[2]

[1] United States Army Research Laboratory, Adelphi, MD, USA
richard.e.harang.civ@mail.mill
[2] National Institute of Standards and Technology, Gaithersburg, MD, USA
peter.mell@nist.gov

Abstract. Network intrusion detection is broadly divided into signature and anomaly detection. The former identifies patterns associated with known attacks and the latter attempts to learn a 'normal' pattern of activity and alerts when behaviors outside of those norms is detected. The n-gram methodology has arguably been the most successful technique for network anomaly detection. In this work we discover that when training data is sanitized, n-gram anomaly detection is not primarily anomaly detection, as it receives the majority of its performance from an implicit non-anomaly subsystem, that neither uses typical signatures nor is anomaly based (though it is closely related to both). We find that for our data, these "micro-signatures" provide the vast majority of the detection capability. This finding changes how we understand and approach n-gram based 'anomaly' detection. By understanding the foundational principles upon which it operates, we can then better explore how to optimally improve it.

Keywords: Network intrusion detection · Anomaly detection · Microsignatures

1 Introduction

Anomaly based intrusion detection systems attempt to learn a 'normal' pattern of activity and then produce security alerts when behavior outside of those norms is detected. This has been an active area of research since at least the late 1980's [1–3]. In the late 1990's, the use of n-grams was discovered to be useful for host based anomaly detection [4]. N-grams are simply a collection of arrays of length *n* obtained by applying a sliding window of length *n* to whatever activity is being monitored (e.g., system calls) [5], and were first applied to analyze network payloads in the PAYL model [6] in 2004 but were limited to 1-grams, as the number of different n-grams that can be acquired can approach a^n where *a* is the number of characters available (UTF-8 encoding has 1,114,112 code points [7]). In 2006, the seminal Anagram approach introduced using an *n* of greater than 1 by storing the acquired n-grams in Bloom filters [8]. As a minor enhancement, Anagram introduced the idea of using known bad n-grams to augment its ability to identify malicious traffic.

F. Cuppens et al. (Eds.): FPS 2016, LNCS 10128, pp. 36–47, 2017.
DOI: 10.1007/978-3-319-51966-1_3

While not discussed in [8], use of the bad n-grams represented a significant shift away from pure anomaly detection. In this approach, a corpus of malicious packet payloads or intrusion signatures can be used to generate n-grams and these n-grams are filtered to exclude any that have been observed to occur in normal benign traffic. The remaining n-grams are then added to a known bad content filter comprised of what we refer to as automatically generated micro-signatures. In operation, any time an n-gram is analyzed that matches one in the bad content filter, the score for the overall packet increases pushing it towards being labelled anomalous/malicious. The good n-gram filter works in an opposite fashion in that the score for the overall packet increases if an n-gram is evaluated that is not in the good filter (thus truly looking for anomalies).

In our work, we re-implement the Anagram approach and attempt to reproduce the results of [8]. We experiment with the Hypertext Transfer Protocol (HTTP) to provide a direct comparison with one of the protocols examined in [8]. We also analyze Anagram performance relative to Domain Name System (DNS) requests, which was not previously done. This traffic is of interest because – while previous work [9] found that n-gram analysis of purely binary protocols was not feasible – DNS requests contain both binary components (e.g., record types and number of requests are encoded as byte fields) and textual components (e.g., the domain name itself is encoded in human-readable ASCII text with length fields acting as separators).

We focus our evaluation on the extent to which the automatically generated micro-signatures in the known bad n-gram filter contribute to the overall performance of Anagram. To do this, we constructed a standalone intrusion detection system (IDS) consisting only of the known bad filter. Unlike the original Anagram, which used bad n-grams derived from IDS signatures and virus samples, we derived our n-grams from known malicious packets to align it with our data sets. We refer to our known bad n-gram IDS as the micro-signature approach.

We find that the Anagram and micro-signature approaches have very similar performance characteristics with the Receiver Operating Curves (ROC) of the full Anagram approach providing only small improvements over the micro-signature ROC curves. In the best case, we found that the Anagram approach provided an increase in area under the ROC of only .0028 compared to the micro-signature approach. This means that the vast majority of the detection capability of Anagram (at least when applied to our data sets) is derived from the known bad component.

The known bad component was portrayed in [8] as a minor enhancement and has received little attention in any other paper in our literature survey. Thus, our results suggests that future research and development of network anomaly detection systems should focus on n-grams of malicious behavior as a primary detection method to be supplemented by the more traditional n-grams of normal behavior. This is the reverse of what is currently discussed in the literature. The utility of automatically generated micro-signatures is an open area for future IDS research.

We should emphasize that this result does not imply that the known good content filters are useless or that the Anagram approach is flawed. On the contrary, the full Anagram approach slightly outperforms the micro-signature approach across a wide range of false positive rates (e.g., false positive rates of .0001 to .001). However, the vast majority of true positives are still attributed to the micro-signature approach, supporting the primary conclusion of our paper. In the worst case with one data set, at a false positive rate of 0 the micro-signature true positive rate exceeds Anagram's by .03. In the best case with the same data set, the Anagram true positive rate exceeds the micro-signature's by .12 at a false positive rate of .0001. For the vast majority of non-zero false positive rates, the Anagram true positive rate had some small advantage over the micro-signature true positive rate.

In summary, the primary findings of this paper are the following:

1. The Anagram approach of using high order n-grams remains effective for analysis of two ubiquitous network protocols (HTTP and DNS).
2. For our data sets, the known bad n-gram filter accounts for the vast majority of the detection capability, and the identification of anomalous n-grams provides only an incremental improvement.
3. Automatically generated micro-signatures provide an effective detection abstraction and can be used to create standalone n-gram based IDSs that can be viewed as an interesting hybrid of an anomaly detection technique (n-grams), standard signature detection, and automated analyses of malicious packets.

The rest of this paper is organized as follows. Section 2 discusses our data. Section 3 summarizes our experiments and Sect. 4 discusses the results. Section 5 provides related work and Sect. 6 concludes.

2 Data

We used three sets of data to compare the effectiveness of Anagram and the micro-signature approach. The first two are sets of HTTP requests and the last is a set of DNS requests.

For the first set, we collected HTTP requests to an operational web server over the course of 24 h. We obtained 769,838 distinct requests, of which 605 were known to be malicious due to signature-based analysis. The first 10,000 requests in this file were closely examined by hand by network security personnel to verify that they were benign.

For the second set, we generated a data set of 393,814 malicious requests from a combination of scanning, vulnerability assessment, fuzzing, and exploit tools from the Kali Linux distribution that were targeted at a virtual machine running an identical web stack to the operational web server. We restricted the data to consider only incoming TCP packets delivered via port 80; all packet headers were discarded, and no stream reassembly was performed.

For the third set, we obtained DNS requests gathered via monitoring traffic crossing the network boundary of an active network. Due to the high volume of requests, we examined only the first 3,000,000 requests that were obtained, and restricted the data to UDP packets with a destination port of 53 and containing a valid DNS header. As with the HTTP data, all transport layer headers were discarded. Based on a pre-existing analysis, 28 packets were known to be malformed and deliberately altered to contain non-DNS data from the 32$^{\text{nd}}$ byte onwards. An additional 72,914 packets were known to contain properly formatted requests, but requested domain names that encoded non-DNS information in some fashion (such as base16 or base32 encoding). Many of these appeared to be commercial security products using DNS as a communication channel, presumably to avoid being blocked by firewalls.

3 Experiment Design

For each of the two types of data sets (HTTP and DNS), we define training sets used to generate the n-gram based content filters (both 'good' and 'bad'). We then construct both Anagram and micro-signature IDSs from the content filters and, from each of the three data sets, we define a validation set used to test the IDSs' effectiveness.

3.1 Training Sets

For HTTP, we constructed a 'normal traffic' training set consisting of the 10,000 hand verified requests along with 100,000 randomly selected requests from the remaining 759,838 requests (excluding the 605 known bad requests). We constructed a malicious training set consisting of 10,000 randomly chosen requests from the set of 393,814 generated malicious requests.

For DNS, we constructed a 'normal traffic' training set consisting of 10,000 randomly selected requests from the 2,927,058 not known to be malicious. We constructed a malicious training set consisting of 10,000 randomly chosen requests from the set of 28 malformed and deliberately altered requests and the 72,914 requests encoded with non-DNS information.

3.2 Intrusion Detection System Construction

We followed the procedure as described in [8] to build and train the known good and known bad content filters and to construct the Anagram IDS. We used just the known bad filter to construct the micro-signature IDS. Note that different good and bad content filters were generated specific to each protocol (HTTP and DNS). The Bloom filters used for the content filters were constructed using a 2^{24} bit index with 3 hash functions per item and using SHA-1 as the hash function, as in [8]. We used an n-gram size of 5 as [8] cited this as being good for 'general' purpose experiments.

For HTTP, we first generated n-grams from the 10,000 hand-verified known good requests to populate what we call a 'gold' content filter (there was no name for this in [8]). We then constructed the bad content filter by generating n-grams from our malicious request training set and including them in the filter if they didn't also occur in the gold filter. Note that this is the only use of the gold filter and it is not used again. Finally, we generated n-grams from the normal traffic training set. For each HTTP request, if the proportion of n-grams that appear in the bad content filter is less than 5%, then we add those n-grams to the good content filter that did not occur in the bad content filter. If this ratio exceeded 5% then the entire HTTP request was ignored.

For DNS, we followed the same methodology. For this though, we had no gold content filter and we populated the known bad filter directly from the malicious request training set. The good content filter was constructed using the 'normal traffic' training set using the same methodology as with the HTTP traffic (using the 5% rule and the bad content filter to eliminate candidate n-grams).

To score a particular request (HTTP or DNS) using our content filters, we recorded the number of n-grams in that request that were not found in either content filter (i.e., new n-grams), the number found in the known bad filter, and the total number of n-grams in the request. Three scoring rules were applied: the Anagram rule of $\frac{5\times\#\{\text{bad ngrams}\}+\#\{\text{new ngrams}\}}{\#\{\text{ngrams in packet}\}}$, our micro-signature rule of $\frac{\#\{\text{bad ngrams}\}}{\#\{\text{ngrams in packet}\}}$, and a "non-normalized" micro-signature rule which involves simply counting the number of bad n-grams without normalization against the size of the packet. However, this final non-normalized version produced worse results, which we do not display.

3.3 Validation Data Sets

To test the effectiveness of the Anagram and micro-signature approaches, we generated three validation data sets. For each set, receiver operator characteristic (ROC) curves were generated for the Anagram and micro-signature approaches.

The first HTTP validation set was focused on testing IDS effectiveness in detecting 'real' attacks. It consisted of 659,838 requests. This was the total set of 769,838 distinct requests minus the 110,000 used in the training set for the good content filter. This set includes the 605 known malicious requests. There was thus no overlap between the training and testing data.

The second HTTP validation set was focused on testing IDS effectiveness in detecting our generated attacks (from the use of security and exploit tools). It consisted of 1,043,652 requests. We started with the total set of 769,838 distinct HTTP requests and removed the 605 known malicious requests. Then we removed the 110,000 requests used in the training set for the good content filter. Lastly, we added the 383,814 generated malicious requests that were not used in the training set for the bad content filter.

The DNS validation set consisted of 2,980,000 requests. We started with the 3,000,000 total requests and subtracted the 10,000 used to train the good content filter and then subtracted the 10,000 used to train the bad content filter. This then included 62,942 known malicious requests that were not used in any of the training sets.

4 Results

The results of our experiments indicate that, for our data sets, the Anagram and micro-signature approaches provide very similar performance. This is surprising as the micro-signature approach is portrayed in the literature as simply a minor augmentation to the overall Anagram approach. We first provide a look at the overall area under the ROC curves for all data sets and then look specifically at the HTTP and DNS results in more detail.

The area under the ROC curves for all three validation sets are shown in Tables 1 and 2. In the very best case for Anagram, it has an area under the curve of just 0.0028 more than the micro-signature approach. In the worst case (DNS), it slightly underperforms micro-signatures, although by such a thin margin that the difference is likely

Table 1. HTTP results

	Area under ROC curve per data set	
	Real attacks	Generated attacks
Anagram	0.9648	0.9998
Micro-signatures	0.9620	0.9997

Table 2. DNS results

	Area under ROC curve
Anagram	0.999963
Micro-signatures	0.999996

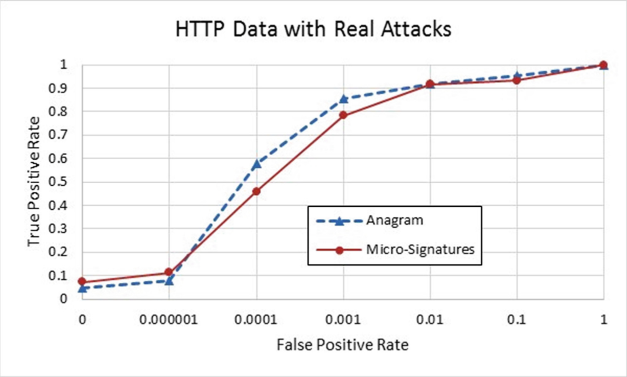

Fig. 1. Full ROC curve for the HTTP validation set with real attacks

not significant. It is especially interesting that the micro-signature approach was able to achieve this performance having been trained on only 10,000 malicious packets and tested against data sets with up to 3 million requests.

In Fig. 1 we show the full ROC curve where Anagram has the greatest advantage over micro-signatures with respect to the area under the curve (the HTTP validation set with the real attacks). Notice the similarities between both algorithms. While some differences do exist, the visual impact of this is highly exaggerated by the logarithmic scaling of the x-axis. While such scaling is not common for ROC curves, we will do this to highlight the importance of the true positive rate (TPR) values at very small false positive rates (FPRs). This is because for network anomaly detection, the number of packets is typically very high and an IDS will thus only be useful if it can operate at extremely low false positive rates [10].

We now review each of the validation sets in detail and analyze the portions of the ROC curves that best show the distinctions between the two approaches under analysis.

4.1 HTTP Results

As shown in Fig. 1 for the real HTTP attacks, Anagram has a better TPR than the micro-signature approach at FPRs at .0001 and above. At extremely low FPRs (.000001 and below), the micro-signature approach has a slight advantage. The largest Anagram advantage is at a FPR of .0001 where the Anagram TPR exceeds that of the micro-signatures by .12. The largest micro-signature advantage is at a FPR of 0 where the micro-signature TPR exceeds that of Anagram by .03. Both methods converge in performance at an FPR of .01.

In Fig. 2 we see how, for generated attacks, the Anagram TPR at most exceeds that of the micro-signature TPR by .0015 at a FPR of .0001. At the lowest observed FPRs, micro-signatures once again have an extremely small advantage over Anagram. Both methods converge in performance at FPRs of .001 and higher.

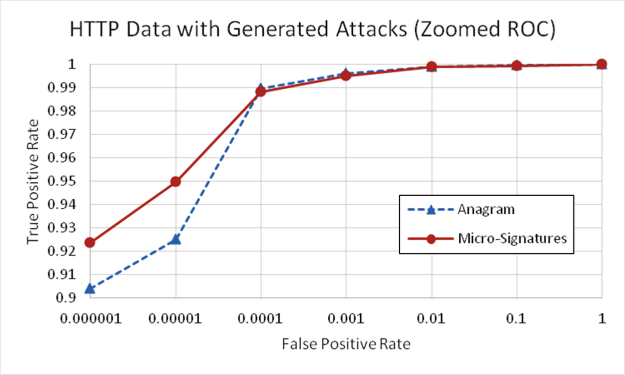

Fig. 2. Zoomed ROC curve for HTTP validation set with generated attacks

4.2 DNS Results

Analogous to our previous results, Fig. 3 shows how Anagram exceeds the micro-signature approach with our DNS dataset, but only by a very small margin. Note how at best Anagram has a TPR .013 higher than the micro-signature approach at the same FPR. The DNS data is of particular interest as it contains a mixture of binary encoded header information with (mostly) textual domain information at the end of the packet. Previous work [9] has suggested that binary protocols are difficult to analyze with n-gram methods; however, it appears that in this particular case the distributions over malicious and benign traffic in both the textual and binary encoded portions of the payloads are sufficiently dissimilar to permit accurate classification. In contrast to the HTTP data, Anagram outperforms the micro-signatures (albeit by extremely fine margins) at all FPR values. Note the extremely small range of y-axis values in Fig. 3, indicating the close similarity of the two approaches at all FPRs.

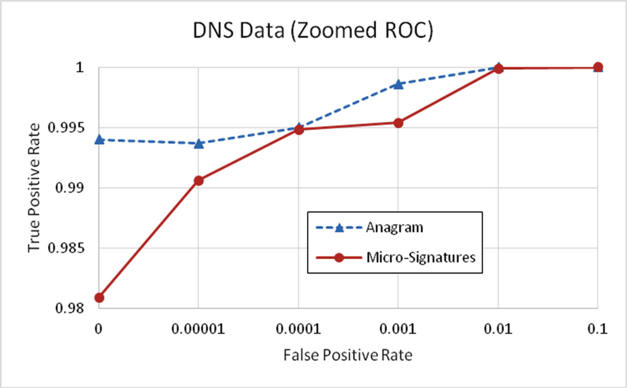

Fig. 3. Zoomed ROC curve for DNS validation set

5 Discussion

The results clearly show the effectiveness of the overall Anagram approach for the HTTP and DNS request datasets. Quite surprisingly, the micro-signatures performed almost as well as the full Anagram approach when considering the area under the ROC curves. At the very best with the HTTP generated attack validation set, Anagram had an ROC curve with an area .0028 greater than the corresponding micro-signature ROC curve. This overall similarity hides the fact that Anagram does outperform micro-signatures at most (but not all) operating points (although our data sets are substantially smaller than those described in [8], which may also account for the slightly poorer performance of the anomaly detection portion). In the best case at a .0001 FPR with the HTTP real attack validation set, Anagram achieves a TPR .12 higher than the micro-signatures. However, the micro-signatures still account for the vast majority of detections at all operating points and validation sets tested. This means that, relative to our datasets, the seminal Anagram anomaly detection system that proved the usefulness of n-grams for network packet inspection achieves the majority of its effectiveness from a subsystem that is effectively signature based.

However, this signature based subsystem is very different from typical signature based systems. The signatures are automatically generated from known malicious packets and are very small in size. It is the presence of groups of signatures that are indicative of an attack, not just single signatures as is the case with standard signature based IDSs. This means that, while clearly signature based, micro-signatures can also potentially generalize to new attacks. This micro-signature paradigm is then a hybrid anomaly-signature system that, in our literature survey, has not been explicitly investigated before. Micro-signatures are not a new discovery (having been included within Anagram in 2006), but they were not highlighted as a major contributor and were not separately evaluated. In this work, we have empirically shown the importance of this component and suggest that micro-signatures can provide a new avenue of IDS research.

An interesting aspect of micro-signatures (that is newly discovered in this research) is that their accuracy can be extremely high even with a small training set. We used 385,751 bad n-grams for the HTTP data set and achieved similar HTTP detection results to the original Anagram paper [8], which used 30 million bad n-grams. For the DNS work, while we can't compare the results directly to [8], we achieve a high detection capability with only 78,532 bad n-grams. This suggests that the micro-signatures generalize to attacks not present in training data, although further research is necessary to quantify this. With a training set of just 10,000 malicious DNS requests, the micro-signatures were able to detect a set of 62,942 malicious requests with a TPR of .995 at a FPR of .0001. This may be potentially explained by considering micro-signatures as a form of supervised learning, while the anomaly detection component of Anagram is more closely related to a one-class unsupervised learning problem. Supervised learning approaches for intrusion detection using n-grams have been shown to be successful elsewhere [11], although they are typically significantly more complex than the simple set membership tests we consider here.

One consideration in the use of micro-signatures, is their resilience to evasion attacks. In particular, the normalization to packet length in our micro-signature approach could lead to an evasion attack where a malicious packet is stuffed with a lot of normal data; this "content mimicry" attack is considered within the original Anagram paper, where it is addressed via subsampling of the packet payload [8]. While the mimicry resistant approach suggested in the original Anagram paper will likely not be as effective for micro-signatures, another potential avenue for handling content mimicry might be through not normalizing the micro-signature counts to packet length. Not shown in this paper are results which find that this idea is effective, but has worse performance than normalized micro-signatures.

6 Related Work

The difficulty of applying machine learning in general to intrusion detection is discussed by Sommer and Paxson [12], which points out several features of intrusion detection problems that make it difficult to successfully apply machine learning; this includes the rareness of attacks, the high cost of diagnosing detected attacks, and the complexity of the input data. A more probabilistic argument is made in [10] in terms of the base rate fallacy. Nevertheless, multiple examples of anomaly-based and unsupervised network intrusion detection methods can be found in the literature.

One of the earliest n-gram approaches is that of the PAY-L system [6], which clusters network traffic based on the distribution of 1-grams. The Anagram system [8], which forms the basis of our analysis, extends the length of the n-grams to between 5 and 9, while also addressing the issue of "content mimicry". In perhaps the most general case, the issue of anomaly detection via n-grams in non-textual, binary protocols is considered by Hadžiosmanović et al. [9], building on the work of [6, 8]; this work examines classifiers that make no use of any protocol-specific domain knowledge and concludes that n-gram based methods generally perform poorly for binary protocols, with an unavoidable tradeoff between high detection rate and low false positive rate. This is formalized in the work of [14], which in addition to evaluating the settings in which n-gram based tools may be expected to perform well, also empirically examines a number n-gram based intrusion detection systems, including the Anagram system. While they do examine the “benign” filter alone and in conjunction with the malicious content filter, they do not examine the contribution of the malicious content filter alone. Finally, similarly to the clustering described in [6], the work of [13] examines the use of a self-organizing map for on-line clustering of packets.

Domain-specific knowledge, in the form of partial parses of protocols, can be used to extract more specific sets of features that help in the identification of anomalous content. In Robertson et al. [15], for instance, web requests are processed by specializing to particular web components, and then learning simple specialized models conditional on each field and component – in effect learning a mixture of site-specific ‘sub-protocols’ within HTTP. Guangmin [16] performs similar tokenization for use in an artificial immune system model. Ingham et al. [17] attempt to learn deterministic finite automata (DFAs) for normal HTTP traffic while detecting, parsing, and transforming known features (such as email addresses) in order to control complexity. The high degree of structure in the underlying grammar (HTTP) combined with the generally limited character set all contribute to the ability of such systems to be effective. However, these systems are also highly specialized to their particular domain of application and so cannot extend to more general intrusion detection scenarios.

Finally, as machine learning techniques have developed, anomaly-based IDS work has kept pace. More advanced approaches to the problem include that of Gornitz et al. [18]. Here, active learning is used to request that specific packets be labeled by an outside mechanism (e.g. a human analyst) thus maximizing the discriminative power of the learning algorithm within a limited budget of time and effort. While such systems do require more resources to train initially, they typically result in significantly improved performance over purely unsupervised systems. The use of the bad content model in the Anagram system [8] may be viewed as a non-active, simplified version of this semi-supervised approach.

7 Conclusion

The n-grams methodology has arguably been the most successful technique for anomaly detection within packet payloads, with Anagram [8] being the seminal work. We tested the Anagram anomaly detection system on two ubiquitous network protocols, confirming its effectiveness on HTTP requests and newly demonstrating its effectiveness on DNS requests. We analyzed the two primary components of Anagram and showed that, for our data, the known bad n-gram filter accounted for the vast majority of the detection capability and that the identification of anomalous n-grams provided only a marginal improvement. Furthermore, we showed that the automatically generated micro-signatures (comprising the known bad n-gram filter) provide an effective detection abstraction and can be used to create standalone n-gram based IDSs, which have performance comparable to Anagram under a wide range of operating conditions.

This study strongly suggests that the effectiveness of Anagram is not primarily due to its core anomaly detection filter but instead to a novel signature detection methodology (i.e., micro-signatures) that was never highlighted in the literature. Thus, this result may indicate a new avenue for IDS research that is not pure anomaly detection but that also deviates greatly from standard signature detection. Unlike anomaly detection, it uses signatures and requires some reference set of malicious traffic. Unlike standard signature detection, it neither looks for arbitrary and variable length substrings or patterns within packet data nor does it require humans to write complete descriptions of indicators of malicious traffic. Instead, it can automatically construct n-gram based signatures automatically from malicious traffic, once that traffic is identified.

In future work, we plan to evaluate how to most effectively use micro-signatures. We plan to create micro-signatures from existing IDS signatures and compare the micro-signature IDS performance against the standard signature based IDS performance. We also need to evaluate the extent to which a group of micro-signatures can hinder an attacker from creating variations of attacks that evade current signature sets. Building on this, we need to evaluate how much micro-signatures generalize within classes of attacks or even between different classes. The various parameters that can be set for the micro-signatures, including the length of the n-gram used, the parameterization of the Bloom filter (or other data structure), and methods for selecting the threshold parameter in the absence of extensive validation data, all require further study. Methods for providing additional situational awareness around positive results from micro-signatures should also be considered; we need to either identify the portions of the packet in which micro-signatures were found or (if the micro-signatures were created from existing signatures) find a way to link the micro-signature to source data for easier interpretation. Finally, the effectiveness of micro-signatures across multiple protocols must be examined, including the potential of combining micro-signatures for multiple protocols into a single, larger Bloom filter.

References

1. Smaha, S.E.: Haystack: an intrusion detection system. In: Aerospace Computer Security Applications Conference (1988)
2. Denning, D.E.: An intrusion-detection model. IEEE Trans. Softw. Eng. **2**, 222–232 (1987)
3. Vaccaro, H.S., Liepins, G.E.: Detection of anomalous computer session activity. In: IEEE Symposium on Security and Privacy (1989)
4. Forrest, S., Hofmeyr, S., Somayaji, A.: Computer immunology. Commun. ACM **40**(10), 88–96 (1997)
5. Damashek, D.: Gauging similarity with n-grams: language independent categorization of text. Science **267**(5199), 843–848 (1995)
6. Wang, K., Stolfo, S.J.: Anomalous payload-based network intrusion detection. In: Jonsson, E., Valdes, A., Almgren, M. (eds.) RAID 2004. LNCS, vol. 3224, pp. 203–222. Springer, Heidelberg (2004). doi:10.1007/978-3-540-30143-1_11
7. The Unicode Standard Version 6.0- Core Specification, February 2011. http://www.unicode.org/versions/Unicode6.0.0/ch01.pdf

8. Wang, K., Parekh, Janak, J., Stolfo, Salvatore, J.: Anagram: a content anomaly detector resistant to mimicry attack. In: Zamboni, D., Kruegel, C. (eds.) RAID 2006. LNCS, vol. 4219, pp. 226–248. Springer, Heidelberg (2006). doi:10.1007/11856214_12
9. Hadžiosmanović, D., Simionato, L., Bolzoni, D., Zambon, E., Etalle, S.: N-Gram against the machine: on the feasibility of the N-Gram network analysis for binary protocols. In: Balzarotti, D., Stolfo, Salvatore, J., Cova, M. (eds.) RAID 2012. LNCS, vol. 7462, pp. 354–373. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33338-5_18
10. Axelsson, S.: The base-rate fallacy and the difficulty of intrusion detection. ACM Trans. Inf. Syst. Secur. **3**(3), 186–205 (2000)
11. Chang, R., Harang, R.E., Payer, G.S.: Extremely lightweight intrusion detection (ELIDe), Army Research Laboratory (2013)
12. Sommer, R., Paxson, V.: Outside the closed world: on using machine learning for network intrusion detection. In: Security and Privacy (2010)
13. Bolzoni, D., Zambon, E., Etalle, S., Hartel, P.: Poseidon: a 2-tier anomaly-based intrusion detection system, arXiv.preprint.cs/0511043 (2005)
14. Wressnegger, C., Schwenk, G., Arp, D., Rieck, K.: A close look on n-grams in intrusion detection: anomaly detection vs. classification. In: 2013 ACM workshop on Artificial intelligence and security (2013)
15. Robertson, W., Vigna, G., Kruegel, C., Kemmerer, R.A.: Using generalization and characterization techniques in the anomaly-based detection of web attacks. In: NDSS (2006)
16. Guangmin, L.: Modeling unknown web attacks in network anomaly detection. In: Third International Conference on Convergence and Hybrid Information Technology (2008)
17. Ingham, K.L., Somayaji, A., Burge, J., Forrest, S.: Learning DFA representations of HTTP for protecting web applications. Comput. Netw. **51**(5), 1239–1255 (2007)
18. Görnitz, N., Kloft, M., Rieck, K., Brefeld, U.: Active learning for network intrusion detection. In: Proceedings of the 2nd ACM Workshop on Security and Artificial Intelligence (2009)
19. Axelsson, S.: Intrusion detection systems: a survey and taxonomy (2000)
20. Paxson, V.: Bro: a system for detecting network intruders in real-time. Comput. Netw. **31**, 2435–2463 (1999)
21. Roesch, M.: Snort: lightweight intrusion detection for networks. In: LISA (1999)
22. Rieck, K., Laskov, P.: Detecting unknown network attacks using language models. In: Büschkes, R., Laskov, P. (eds.) Detection of Intrusions and Malware & Vulnerability Assessment. LNCS, pp. 74–90. Springer, Heidelberg (2006)
23. Rieck, K., Laskov, P., Müller, K.-R.: Efficient algorithms for similarity measures over sequential data: a look beyond kernels. In: Franke, K., Müller, K.-R., Nickolay, B., Schäfer, R. (eds.) DAGM 2006. LNCS, vol. 4174, pp. 374–383. Springer, Heidelberg (2006). doi:10.1007/11861898_38
24. Cretu-Ciocarlie, G.F., Stavrou, A., Locasto, M.E., Stolfo, S.J.: Adaptive anomaly detection via self-calibration and dynamic updating. In: Kirda, E., Jha, S., Balzarotti, D. (eds.) RAID 2009. LNCS, vol. 5758, pp. 41–60. Springer, Heidelberg (2009). doi:10.1007/978-3-642-04342-0_3
25. Perdisci, R., Ariu, D., Fogla, P., Giacinto, G., Lee, W.: McPAD: a multiple classifier system for accurate payload-based anomaly detection. Comput. Netw. **53**(6), 864–881 (2009)