© 2016 IJSRSET | Volume 2 | Issue 6 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Improving NIDS Rules for Protocols with Detection of Abnormal Traffic in Real Time Traffic Using Snort

Ankita Choubey, Navi Singh Thakur

Shri Ram Institute of Science & Technology, Jabalpur, Madhya Pradesh, India

ABSTRACT

Network intrusion detection system (NIDS) has attracted much attention in recent years due to ever-increasing amount of network traffic and ever-complicated attacks. Numerous studies have been focusing on accelerating pattern matching for a high-speed design because some early studies observed that pattern matching is a performance bottleneck. However, the effectiveness of such acceleration has been challenged recently. This work therefore re-examines the performance bottleneck by profiling popular NIDSs, Snort, with various types of network traffic in detail. In the profiling, we find pattern matching can be dominant in the Snort execution if the entire packet payloads in the connections are scanned, while executing the snort rules is an obvious bottleneck in the snort execution. This work suggests three promising directions towards a high-speed NIDS design for future research: a method to precisely specify the possible locations of the signatures in long connections, a compiler to transform the policy scripts to efficient binary codes for execution, and an efficient design of connection tracking and packet reassembly.

Keywords: NIDS, Snort, Network Traffic, Profile, Snort Rules.

I. INTRODUCTION

As one part of protecting web servers, a system administrator needs to know when her system is under attack and has been (or is in danger of being) compromised this is intrusion detection.

IDS protecting web servers must detect novel attacks without human intervention, and that anomaly detection systems are a solution that addresses this requirement. All anomaly detection systems share a common trait of learning a model of normal behavior. Over the years, researchers have tried many different algorithms for learning this model. Some of these algorithms have promise for HTTP; others have limitations that prevent them from ever working in this domain.

Whenever the set of training data representing normal behavior for the anomaly detection system is incomplete or the actual set is infinite, the anomaly detection system must perform generalization. Although researchers have realized the usefulness of generalization, the extent to which they have investigated it is limited, the large variety of anomaly detection algorithms is a symptom of the fact that we lack a comprehensive theory of intrusion detection and anomaly detection to provide guidance

II. METHODS AND MATERIAL

A. Intrusion detection

Computer intrusion detection started in 1972 with a paper by Anderson [7] identifying the need for what would evolve into today's intrusion detection systems. Early IDS researchers focused on statistics of system and user behavior on a given machine (a host-based IDS) to address insider threats. In practice, these early systems had high false-positive rates.

The inability to protect web servers led to explorations of other approaches. Some researchers restricted HTTP to a presumably safe subset of the protocol. Other

approaches monitored the HTTP data stream at the ISO network application layer, in spite of the perceived difficulty of using this data stream. Some of these approaches treated web servers as a generic network service. Incoming and outgoing traffic were modelled as a stream of bytes or as discrete packets. Some approaches looked for patterns in the first few packets of connections. Others compared character distributions between the payloads of similar-sized packets.

In contrast with these protocol-independent approaches, some researchers focused on the HTTP requests in the network application layer, for example combining statisti- cal characteristics of common gateway interface (CGI) program request parameters. However, these anomaly detectors must be trained on data without attacks. This re- quirement is problematic because the normal background of today's Internet contains large numbers of old attacks, most of which are ineffective against properly patched servers. Signature-scanning intrusion-detection systems (IDS) such as snort can be used to filter out known harmless attacks; however, the high accuracy required for training requires frequent updates to the attack signature database and careful site-specific tuning to remove rules that generate false alarms. This manual intervention reduces the main advantage of using anomaly detection.

B. Related Work

Earlier anomaly detections systems assume that the data they use is stationary. Researchers working with no stationary data use generalization in order to tolerate the novel instances that are a hallmark of no stationary data. Some researchers working with no stationary data include Mahoney and Chan [11], who used an exponential decay of learned probabilities of features in network data. Lane (sometimes with Brodley) showed that user event data is no stationary, and they identified methods for measuring the magnitude and direction of the drift [15]. Much of the work on no stationary data focuses on eliminating the old portions of the model (forgetting), e.g., work by Salganicoff [13]. Littman and Ackley [14] looked at cases in which the problem can be divided into two parts, variable and invariant, although this approach would not apply to environments (such as HTTP) where little is invariant. Denning [68] described a system that modelled the data recorded by an audit system. The data were generalized by several statistical measures of patterns in the audit records. She recognized the problem presented by no stationary data (e.g., adding new users to a protected system), and she proposed approaches that might improve the situation. However, she did not report results showing the effectiveness of these approaches. She, like most of the following researchers, did not characterize how the generalization chosen affected accuracy of result.

False positives are an indication of under generalization or insufficient training. Axels- son [14, 15, and 16] noted that an intrusion detection system must be very accurate to avoid producing many more false alarms than true positives. In addition to controlling false positives through targeted generalization, some anomaly detection systems use techniques to control false positives. Two approaches have been proposed to address this problem.

The second approach is correlating alarms. The idea is to group alarms into classes representing behavior, and assume that all of the alarms in one class represent the same behavior. The idea is that a system administrator can view a single exemplar from the class and determine if it represents normal or abnormal behavior Robertson et al. [2] went one-step further, using heuristics to identify the attack type associated with a class of HTTP requests. Julisch [3] looked at the problem of false alarms overwhelming human operators. He clustered alarms to identify the root cause and was successful, reducing the future alarm load by 82%. The theoretical aspect of his paper showed that general alarm clustering is N P -complete.

III. PROPOSED WORK AND RESULTS

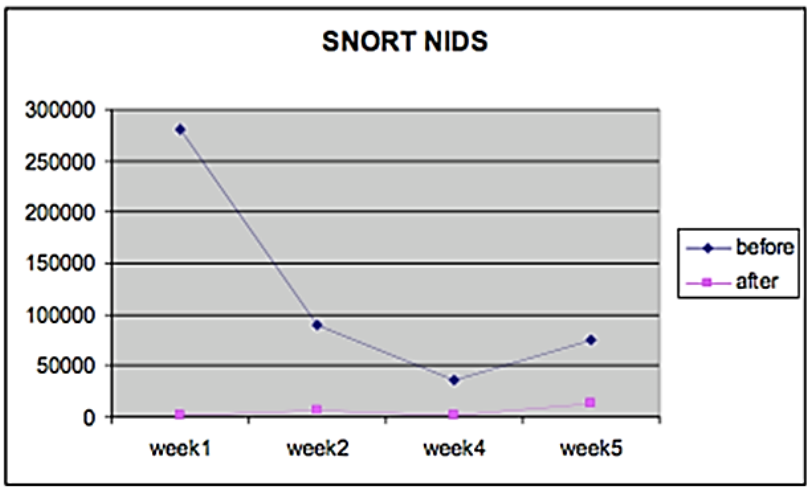
Anomaly Based Detection: An anomaly detection [1] system first creates a baseline profile of the normal system, network, or program activity. Thereafter, any activity that deviates from the baseline is treated as a possible intrusion. Anomaly detection systems offer several benefits. First, they have the capability to detect insider attacks. For instance, if a user or someone using a stolen account starts performing actions that are outside the normal user-profile, an anomaly detection system generates an alarm. Second, because the system is based on customized profiles, it is very difficult for an attacker to know with certainty what activity it can carry out without setting off an alarm. Third, an anomaly detection system has the ability to detect previously unknown attacks. This is because a profile of intrusive

activity is not based on specific signatures representing known intrusive activity. An intrusive activity generates an alarm because it deviates from normal activity, not because someone configured the system to look for a specific attack signature. Anomaly detection systems, however, also suffer from several drawbacks. The first obvious drawback is that the system must go through a training period in which appropriate user profiles are created by defining "normal" traffic profiles. Moreover, creating a normal traffic profile is a challenging task. The creation of an inappropriate normal traffic profile can lead to poor performance. Maintenance of the profiles can also be time-consuming.

Algorithm of proposed IDS

Step 1: Input data are taken from network packet
Step 2: Implementing the data into Snort
Step 3: Snort performs pre-processing and analyses whether the data is attacked or a normal one
Step 4: Applying snort rules to detect the attack in the application layer
Step 5: Applying further improvement in rule structure
Step 6: Finally, we perform the detection process and drop the attack packet and the new rule is generated through intrusive packets.

(a) This graph represent snort performance with time and payload data processed after applying ICMP rules on real time traffic (x axises represent time and y axises represent payload data in bytes)



(b) This table represent comparison of HTTP and FTP protocol based on rules in SNORT.

Comparison of Snort execution with the default and new configurations (execution time in CPU cycles).

Traffic type	Decoding	Preprocessing	Mpse	Rule tree	Pcre	Others	Total
HTTP default (total)	2.58G	8.67G	5.65G	4.75G	0.92G	6.98G	29.55G
HTTP default (proportion) (%)	8.47	29.15	18.98	12.88	3.05	23.38	100
HTTP new (total)	3.31G	15.28G	202.50G	54.97G	5.17G	34.19G	315.42G
HTTP new (proportion) (%)	1.05	4.84	64.20	17.43	1.64	10.84	100
FTP default (total)	18.30M	38.01M	0.19M	0.34M	27.10K	9.16M	66.03M
FTP default (proportion) (%)	27.72	57.57	0.29	0.51	0.04	13.87	100
FTP new (total)	17.03M	56.62M	935.53M	10.77M	0	38.39M	1,058.34M
FTP new (proportion) (%)	1.61	5.35	88.40	1.02	0	3.62	100

IV. CONCLUSION

In network, traffic is a potential threat to a network or not, there is a need for IDS to have a method in differentiating whether it is malicious or not. Therefore, this research has introduced a new methodology to identify a fast attack intrusion using time-based detection. In this paper, we have the method used to identify anomalies based on the number of connection made in 1 second. The approach is then tested on real network traffic data and the result is then evaluated by using the Classification Table based on the logistic regression model. From the test and analysis, it is shown

that the model is suitable for predicting the normal and abnormal behavior in UDP and ICMP protocol.

V. REFERENCES

[1] Haitao Sun, Shengli Liu, JiayongChen and Changhe Zhang "HTTP tunnel Trojan detection based on network behavior", Elsevier, Proceedings to the Energy Procedia ESEP 2011: 9-10 December 2011, Singapore, pp. 1272 – 1281, 2011.

[2] Borders K and Prakash A. Web tap:detecting covert web traffic. Proc. ACM conference on Computer and Communications Security (CCS 04)2004;110-120.

[3] Kruegel C, Vigna G. Anomaly Detection of web-based attacks. Proc. ACM conference on Computer and Communications Security (CCS 03)2003;251-261.

[4] Wenke Lee. (1999). A Data Mining Framework for Constructing Feature and Model for Intrusion Detection System. PhD thesis University of Columbia.

[5] Cuppen, F. & Miege, A. (2002). Alert Correlation in a Cooperative Intrusion Detection Framewok. In Proceeding of the 2002 IEEE Symposium on Security and Privacy. IEEE, 2002.

[6] Cabrera, J.B.D., Ravichandran, B & Mehra R.K. (2000). Statistical Traffic Modelling for Network Intrusion Detection. In Proceeding of the IEEE Conference.

[7] Yeophantong, T, Pakdeepinit, P., Moemeng, P & Daengdej, J. (2005). Network Traffic Classification Using Dynamic State Classifier. In Proceeding of IEEE Conference.

[8] Farah J., Mantaceur Z. & Mohamed BA. (2007). A Framework for an Adaptive Intrusion Detection System using Bayesion Network. Proceeding of the Intelligence and Security Informatics, IEEE, 2007.

[9] Wang Y., Huang GX. & Peng DG. (2006). Model of Network Intrusion Detection System Based on BP Algorithm. Proceeding of IEEE Conference on Industrial Electronics and Applications, IEEE, 2006.

[10] Sekar, R., Gupta, A., Frullo, J., Shanbhag, T., Tiwari, A., Yang, H. & Zhou, S. (2002). Spesification-based Anomaly Detection: A New Approach for Detecting Network Intrusions. In Proceeding of CCS ACM Conference.

[11] Karl Levitt. (2002). Intrusion Detection: Current Capabilities and Future Direction. Proceeding of IEEE Conference of the 18th Annual Computer Security Application, IEEE, 2002.

[12] Garuba, M., Liu, C. & Fraites, D. (2008). Intrusion Techniques: Comparative Study of Network Intrusion Detection Systems. In Proceeding of Fifth International Conference on Information Technology: New Generation, IEEE, 2008.

[13] Robertson S., Siegel EV., Miller M. & Stolfo SJ. (2003). Surveillance

[14] Detection in High Bandwidth Environment. In Proceeding of IEEE Conference on the DARPA information Survivability and Exposition, IEEE, 2003.

[15] AXELSSON , S. On a difficulty of intrusion detection. In Recent Advances in Intru- sion Detection (1999).