Detection of Network Buffer Overflow Attacks: A Case Study

Maroš Barabas, Ivan Homoliak, Matej Kačic, Petr Hanáček
Brno University of Technology, Faculty of Information Technology
Božetěchova 2, CZ-612 66, Brno, Czech Republic
{ibarabas, ihomoliak, ikacic, hanacek}@fit.vutbr.cz

***Abstract*— This paper presents an automated detection method based on classification of network traffic using predefined set of network metrics. We proposed the set of metrics with focus on behavior of buffer overflow attacks and their sufficient description without the need of deep packet inspection. In this paper we describe two laboratory experiments of automated detection of buffer overflow attacks on vulnerable network services and their description by proposed set of network metrics. We present the principles of several chosen network metrics and their application on experimental attacks according to their nature in comparison to valid communication.**

***Keywords*— buffer overflow, detection, ids, network metrics**

I. Introduction

The aim of our work is to design and develop a novel real-time intrusion prevention system for detection of advanced network attacks using behavioral characteristics of network communication. In previous articles [1][2] we proposed an idea of framework architecture that would be used for detection of various network threats. The papers presented the novel Automated Intrusion Prevention System (AIPS) which uses honeypot systems for the detection of new attacks and the automatic generation of behavioral signatures based on network flow metrics. The detection method is based on extraction of partial communication and creation of behavioral signature from the network flow by previously defined set of network metrics. We have successfully experimented with the architecture of the AIPS system and we defined 167 metrics divided into five categories according to their nature. These metrics are used to describe properties of detected attack not upon the fingerprint of common signature, but based on its behavior. Metrics are formally specified and extraction of them can be generally realized for each data flow. The specification includes statistic, dynamic, localization and especially behavioral properties of network communication. For the learning phase of classification, we use simulated and captured set of buffer overflow attacks and new attacks extracted from shadow honeypots deployed in protected network. These shadow honeypots are capable of detection either previously unknown buffer overflows using dynamic taint analysis [4].

This paper discusses two use cases of buffer overflow attacks simulated in laboratory conditions and their behavioral characteristics created by previously published set of network metrics. Together with the attack communications we present valid communications for further comparison of detection approaches and to emphasize the particular metrics used for the attack detection. We compare various chosen network metrics to visualize methods of detection buffer overflow attack vectors and their differences from valid communication. We chose several metrics that approximate the communication and visualized them (e.g. polynomial approximation of output communication in output direction from the side of attacked machine). The second contribution of this paper is the possibility of detection new zero-day attacks by successful and sufficiently abstract description of these attacks to detect the similarities in behavioral characteristics of various attack vectors of the same type (e.g. buffer overflow).

II. Method of Detection

For classification of malicious attack vectors in network flow, the captured traffic must proceed through several steps of extraction as follows:

1) First, the captured traffic from mirrored network interface is divided into separate connections by meeting specific criteria (e.g. three-way handshake). Metrics extraction process considers communications data stored in libpcap format [3]. This process is closely defined in our previous article [1].

2) The second step is the extraction of metric data from from each TCP connection by predefined set of network metrics. The output of the extraction is an array of integers, each number representing one metric or its part (for more complex metrics as polynomial representation, etc.)

3) All these metrics (extracted data) create the behavior signature of the particular connection that can be matched with known attack patterns. The comparison with the database is done by using artificial intelligence to suppress threshold evasion methods used by attackers.

4) For each matched attack the strongest features used for the detection are strengthen inside the detection model to increase the detection ratio (e.g. by increasing its weight).

The initial set of attack signatures is gathered and later updated by shadow honeypot systems deployed within the monitored network. Honeypot systems are very good source of attack vectors due to their nature of detecting buffer overflow issues within the monitored services. We use honeypot system

Argos [5], which was earlier reprogrammed for our purposes. This honeypot system is able to detect buffer overflow type of attacks within the virtualized operating system and its applications. The system is deployed with the set of services commonly used in the organization. The nature of the system indicates that all network traffic and triggered events are always malicious, because regular users do not use these systems. We use the systems as source of expert knowledge, where each buffer overflow event indicates an attack and can be analyzed and integrated into the detection model as it was briefly described above in the step 4. in this section.

In following chapters we describe the laboratory experiments with honeypot systems and their integration to our detection system for creation of behavioral signature of each attack. The main goal of this research is to bring a method and the set of metrics that sufficiently describe each network attack abusing buffer overflow type vulnerabilities using the nature of its behavior to lower the false positive ratio of the classification.

III. Experiments

For this article we have chosen two experiment attacks (in laboratory conditions) on two different network services, both vulnerable to buffer overflow type of attacks. The first attack was taken against CaesarFTP 0.99g with publicly documented vulnerability CVE-2006-2961[1] together with a public exploit. In the Figure 1 is shown the attack on the service together with the valid communication by the size of transferred packets. The inbound communication is above the x-axis (positive size of packets) and outbound below the axis (negative size). The peak (12th packet) is the packet containing the buffer overflow exploit data. Transferred data represents the active FTP data connection, which is intentionally separated from ongoing control connection.

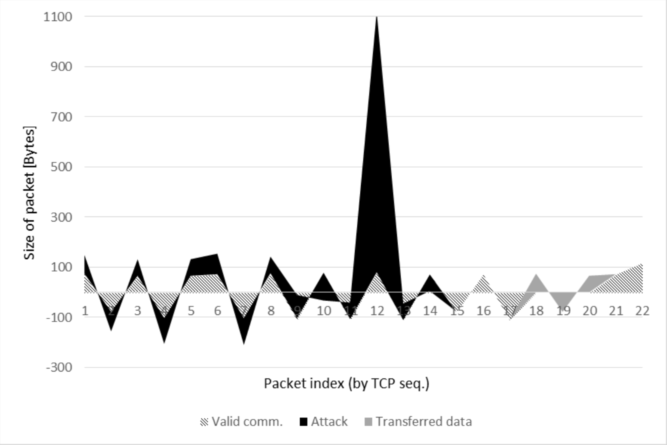

Figure 1. The attack on FTP service in comparison with the valid communication

The second presented experiment is SSH service FreeSSHd 1.0.9. This service is vulnerable to buffer overflow attack according to CVE-2006-2407[2] with public exploit. The attack is shown on the Figure 2, where the inbound communication is above the x-axis and outbound below the axis.

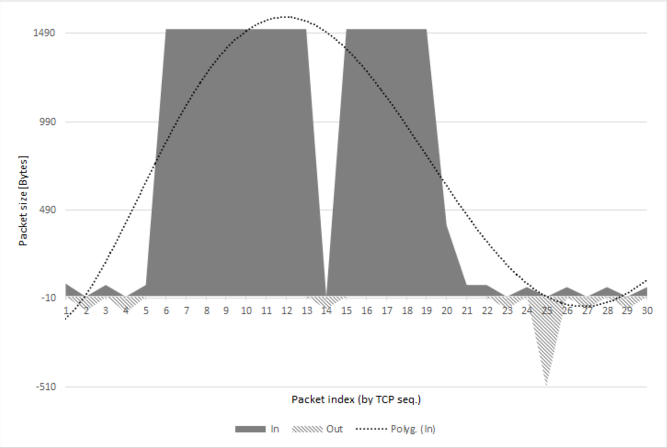

Figure 2. The attack on SSH service (inbound = positive size; outbound = negative size) and the 5th order polynomial approximation.

IV. Network Metrics

The signatures created by AIPS are specific and unique for their behavioral nature. Each signature is a vector composed of dozens of numbers, each corresponding to a value of a specific metric. Each metric is a characterization of the network flow and could be specified as an extension of the NetFlow protocol describing not only statistical properties of the network flow, but it also includes dynamic, localization and behavioral specifics. These metrics are briefly described in the following text. For a better orientation we divided the metric set into five categories based on the nature of particular metrics.

A. *Statistical*

The first category includes various statistical properties of TCP connection. Most metrics are characteristics of TCP parameters representing by mathematical statistic functions as count, mean, mode, median and standard deviation of packets within a session or during the predefined time interval (e.g. bytes per second). As the example of statistical metrics used for a better attack classification we can mention metrics based on the size of fragmented packets, which counts all bytes (before applying mentioned mathematical functions) within fragmented packets exceeding the defined threshold (MTU). These metrics are specifically successful in detection of buffer overflow exploit packets which are usually much bigger than average packets in similar traffic.

Experiments: Using our examples the average number of bytes of malicious traffic packets exploiting buffer overflow vulnerability is larger by more than 150% then average number of bytes in a valid communication (In our particular example it’s 2,47 more bytes in case of attack, as it can be seen in Figure 3). In our experiments this metric had 99.75% overall accuracy in detection of buffer overflow attacks. This seems to

[1] URL: http://cvedetails.com/cve/2006-2961/

[2] URL: http://cvedetails.com/cve/2006-2407/

be a consequence to the packet size of buffer overflow exploit, however, this metric is usable only without variability of bytes in the service protocol. In the picture we can see the size of packets of the attack (solid fill) and the valid communication (pattern) by their index (sort by TCP sequence number) and the difference in the 12 packet which contains the exploit.

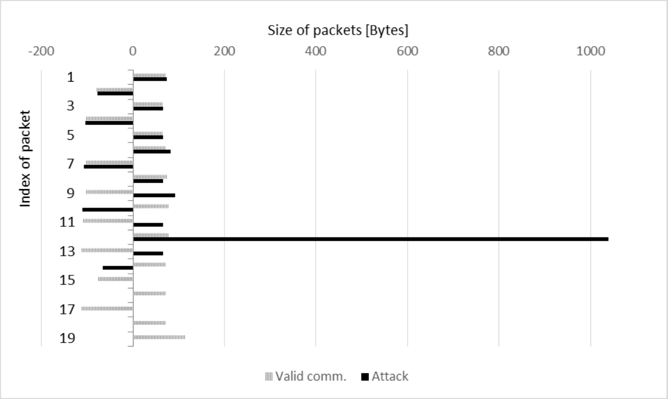

Figure 3. The difference between attack (solid fill) and valid (pattern fill) communication

B. Dynamic

Dynamic metrics transform properties of TCP connection by using time representation describing the dynamic behavior of the connection. In comparison to statistical, dynamic metrics often look on all connections within the analyzed time frame (i.e. the connection context). This is particularly helpful in analysing simultaneous connections typical for data transmitting type of services (e.g. FTP) where related simultaneous connections are common.

Experiments: We have chosen the metrics corClosed, which if true means the session was correctly closed (by ACK,FIN,ACK handshake). The one of nature characteristics of buffer overflow attacks is (but not always) the termination of remote service in advance of creating remote shell for the attacker . This behavior lies behind the creation of metric that is 0 if the connection is not properly closed (by FIN/ACK sequence), which means the service (thread or sub-process) was terminated. Both of example attacks has the corClosed false and this metric has the 83% accuracy (based on existence of attacks without termination of services).

C. Localization

The group of localization metrics represents properties of communicating machines which are related to their locality as well as localization of the service (e.g. IP address, port number). These metrics also take in consideration whether the machines are located in local network and the direction of the communication. Our experiments were taken in laboratory conditions therefore these metrics were not applicable, however, we can see their potential in classification according to IP reputation and other localization features.

D. Distributed

These metrics distribute connection preferences (e.g. packets or their size) into the fixed count of intervals in logarithmic scale (1, 4, 8, 16, 32 and 64 seconds). This group of metrics also contains vector metrics. Together we defined 34 of metrics in this category which are result of parameterization of 2 functions, which accepts parameters as unit time, threshold, direction and context of analyzed TCP connection.

Experiments: For our examples we chose four distributed metrics for the first experiment:

- InPkt1s10i and OutPkt1s10i - distribution of inbound and outbound packets into ten segments by 1s.
- InPktLen1s10i and OutPktLen1s10i - distribution of inbound and outbound packet size (length) into 10 segments by 1s.

In the Figure 4 we depicted the distribution of packets over time intervals of 1s (into 10 intervals). The distribution of packets can carry important information of the connection, particularly in buffer overflow type of attacks the packet size is clustered into few intervals. We can see that in the first interval there came the same number of packets as in third interval, but the size of packets in third interval is much higher.

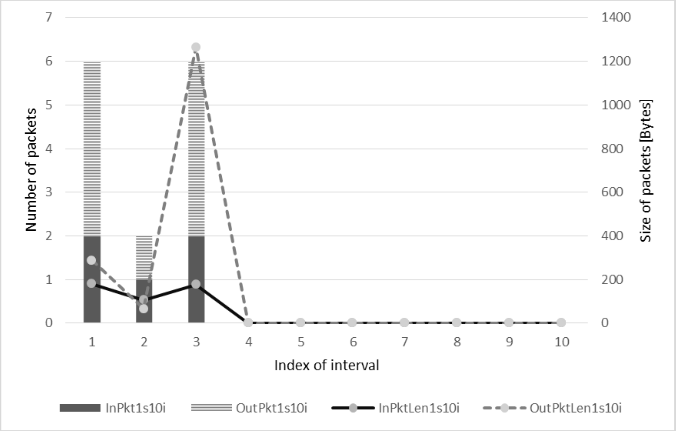

Figure 4. The distribution of inbound and outbound packet and their size over interval segments

E. Behavioral

Behavioral metrics is a group of metrics based on the description of the properties directly associated with the TCP connection behavior. Examples include legal or illegal connection closing, number of flows at defined time intervals, polynomial approximation of the length of packets in time domain or in index of occurrence domain. For polynomial representation we have chosen 3rd, 5th, 8th and 13th order polynoms. Other examples of behavioral metrics are coefficients of Fourier series in trigonometric representation with distinguished direction of analyzed TCP connection, standard deviation of time intervals between TCP connections going on the same ports and IP addresses and normalized products of the analyzed communication with 1 - n Gaussian curves with regard of the direction.

Experiments: For example we have chosen 5th order polynomial approximation, which is shown in Figure 5 for approximation of the first experiment (FTP). The approximation trend-line of the attack is compared to valid communication. In the Figure 2 is depicted the 5th order polynomial approximation (in packet's index domain) of the attack communication (on SSH service).

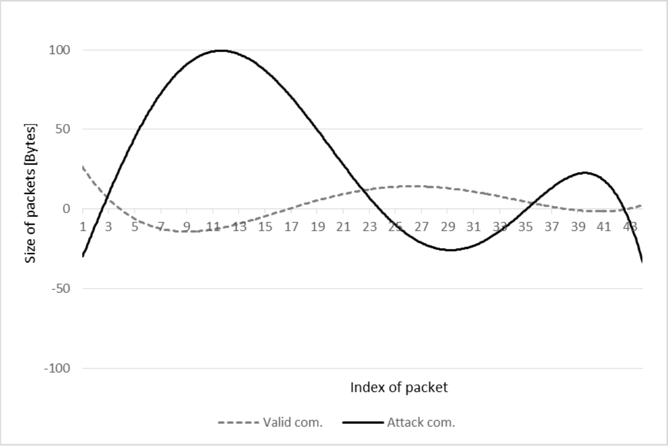

Figure 5. The difference between attack and valid 5th ord. polynomial approximation

V. Conclusions

In the article, we outlined the possibility of automation of buffer overflow network attacks detection. The key idea is to apply a set of network metrics on each network connection keeping its context related to other connections and by continuously updating the expert knowledge of underlying AI detection engine by Honeypot systems, gradually decreases the false positive rate of detection of ongoing similar attacks.

After the laboratory experiments, it seems, that some of the proposed metrics could have a high success rate in the detection of buffer overflow attacks, however, most of them could be easily evaded if used alone. We proposed 167 network metrics without using techniques of deep packet inspection and with using data mining we can adjust the best fitting subsets of metrics each used for a particular type of analyzed attack keeping the process fully automated without the need of human intervention.

The reason of high classification capability of few metrics is, that the classification of buffer overflow attacks is highly predicable by the size of data in fragmented packets, necessary to cause the overflow and the nature of valid communication with small number of fragmented packets.

Our future work will focus on extending the metrics set to achieve more sufficient results in detection of buffer overflow attacks and we plan to perform more experiments with actual metric sets extended with real attacks captured in campus network. The efficiency of current detection method was tested only on a small number of attacks. In the near future, we plan to create a public detection set that would create a challenge in the development of detection algorithms to detect unknown attacks.

Acknowledgment

This article was created within the research plan of the Security-Oriented Research in Information (MSM0021630528). This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and by the project Advanced Secured, Reliable and Adaptive IT (FIT-S-11-1) and with a financial support from the Czech Republic state budget through the Ministry of Industry and Trade by the research plan FR-TI1/037.

References

[1] Barabas, M., Homoliak, I., Drozd, M., Hanáček, P.: Automated Malware Detection Based on Novel Network Behavioral Signatures, In: International Journal of Engineering and Technology, Vol. 5, No. 2, 2013, Singapore, SG,p. 249-253, ISSN 1793-8236

[2] Barabas, M., Drozd, M., Hanáček, P.: Behavioral signature generation using shadow honeypot, In: World Academy of Science, Engineering and Technology, Vol. 2012, No. 65, US, p. 829-833, ISSN 2010-376X

[3] Luis Martin Garcia. Programming with libpcap - sniffing the network from our own application. Hakin9 - Computer Security Magazine, 2-2008(2-2008):9, 2008.

[4] X. Zhang, L. Zhi, a D. Chen, "A Practical Taint-Based Malware Detection," in Apperceiving Computing and Intelligence Analysis, 2008. ICACIA 2008. International Conference on, s. 73 -77, 2008

[5] G. Portokalidis, A. Slowinska, a H. Bos, "Argos: an Emulator for Fingerprinting Zero-Day Attacks," in Proc. ACM SIGOPS EUROSYS'2006, 2006