Features for Behavioral Anomaly Detection of Connectionless Network Buffer Overflow Attacks

Ivan Homoliak, Ladislav Sulak, and Petr Hanacek

Faculty of Information Technology, BUT,
Bozetechova 1/2, 612 66 Brno, Czech Republic
{ihomoliak,xsula04,hanacek}@fit.vutbr.cz
http://www.fit.vutbr.cz/.en

Abstract. Buffer overflow (BO) attacks are one of the most dangerous threats in the area of network security. Methods for detection of BO attacks basically use two approaches: signature matching against packets' payload versus analysis of packets' headers with the behavioral analysis of the connection's flow. The second approach is intended for detection of BO attacks regardless of packets' content which can be ciphered. In this paper, we propose a technique based on Network Behavioral Anomaly Detection (NBAD) aimed at connectionless network traffic. A similar approach has already been used in related works, but focused on connection-oriented traffic. All principles of connection-oriented NBAD cannot be applied in connectionless anomaly detection. There is designed a set of features describing the behavior of connectionless BO attacks and the tool implemented for their offline extraction from network traffic dumps. Next, we describe experiments performed in the virtual network environment utilizing SIP and TFTP network services exploitation and further data mining experiments employing supervised machine learning (ML) and Naive Bayes classifier. The exploitation of services is performed using network traffic modifications with intention to simulate real network conditions. The experimental results show the proposed approach is capable of distinguishing BO attacks from regular network traffic with high precision and class recall.

Key words: Buffer overflow • Connectionless traffic • SIP • TFTP UDP vulnerabilities • NBAD • Naive Bayes

1 Introduction

Buffer overflow attacks belong to the category of the most dangerous network attacks. They are typically used for the execution of an attack code remotely by overflowing a piece of memory. It occurs when a program attempts to write beyond the end of an array with the purpose of changing an instruction flow [5]. In the case of vulnerable application running with root privileges, the attacker will also get root account permissions. As a result, the whole machine may be compromised, which can have very harmful consequences.

There exist various methods for detection of such attacks. One of the approaches is to analyze receiving data from traffic and to search for the known payload signatures of attacks. However, successful detection of unknown threats (zero-day), obfuscated attacks or encrypted data is restrained or not even possible. Also, the whole process can be very time-consuming in this approach. Another approach for detection of mentioned attacks is Network Behavioral Anomaly Detection (NBAD), which will be focused in this paper. NBAD is a technique used for analyzing network traffic without knowledge of transferring data, therefore problems mentioned above may be reduced noticeably. On the other hand, the possibility of evasion (false negatives) and denial of legitimate traffic (false positives) can be increased in NBAD [4].

A connection-oriented communication [7] includes the session initiation phase at the very beginning of each communication and the session destruction phase at the end of a session. TCP is representative of connection-oriented communication where it is possible to track the beginning and the end of a connection thanks to the presence of handshake and endshake. After the successful establishment of a TCP connection, the useful data are delivered and reconstructed to be in the same order as they are sent. Moreover, some protocols are able to determine the current state of connection-oriented communication, like in FTP, Telnet or Samba. It should be noted that some protocols are designed to be primarily connection-oriented, but they can be switched to connectionless mode, too.

This work is primarily focused on connectionless communication, because this problem has been omitted in the past [6, 10] and there is little research primarily aimed at this problem. The typical representative of connectionless communication is UDP protocol which does not contain any session establishing or destruction, and thus poses a problem in the assignment of packets to particular connection records. Regarding NBAD, we propose a collection of features which describe the behavior of connectionless buffer overflow attacks in the way which enables us to distinguish between legitimate communications and attack ones. A tool for extraction of the designed features collection is proposed. Gathered data containing features of connections' records serve as the input for the ML process, which is a fundamental step in the data mining phase.

The paper is organized as follows. We describe related work with a focus on NBAD intrusion detection and ML based internet traffic classification in Section 2. Next, Section 3 characterizes proposed features with a description and categorization of them. Section 4 contains detailed information about our experiments performed in a virtual network environment as well as about vulnerabilities and exploits. In Section 5, we describe data processing and the analysis of our collected dataset. Section 6 presents a summary of achieved results and the last part – Section 7 – concludes the paper.

2 Related Work

Indeed, there exists hardly any research thoroughly dealing with up-to-date behavioral anomaly intrusion detection in connectionless network traffic. The re-

ason can be the fact that prevalent amount of network services are primarily using TCP and, therefore, are more attractive for attackers. Another reason may be related with unavailable information about the beginning and end of the connectionless flow as discussed above.

2.1 NBAD in Connectionless Network Traffic

The first known work which performs behavioral-based network anomaly intrusion detection primarily aimed at connection-oriented traffic, but including connectionless too, is contained in paper [8]. The approach of this work simplifies the connectionless traffic by treating each UDP packet as one connection record and, therefore, misses all behavioral and statistical characteristics of related packets in the connectionless flow. Also, the data set of this work is not up-to-date. Latter related work [3] dealing with network intrusion detection including connectionless traffic is based on collecting of flows' statistics with accumulation of a concern index value which is assigned to host machines. When a concern index of a host is exceeded, an alert is raised. No results presenting performance of a designed system are available and there is not described any approach of determining a session establishment or destruction.

2.2 NBAD in Connection-oriented Network Traffic

There exist several papers which discuss and propose various non-payload based behavioral anomaly intrusion detection models for connection-oriented traffic. Most of them evaluate their performance on DARPA dataset. DARPA dataset was considered useful for evaluation of intrusion detectors in 2008 [14], but it is arguably these days. Examples of works using DARPA dataset include: PHAD [9], ADAM [2], APAN [13] etc.

PHAD [9] is anomaly detection algorithm that learns the normal ranges of values for each packet header field at data link, network, and transport layers. It detects 72 of 201 instances (29 of 59 types) of attacks with only approx. 10 false positives per day. It detects some attacks (6 types) based on anomalous IP addresses. ADAM [2] is based on Naive Bayes classifier as supervised machine learning model. It monitors port numbers, IP addresses, subnets, and TCP state. The paper presents its successful improvement using pseudo-Bayes estimators with aim on reduction of false positives and detection of new attacks. Evaluation is performed on DARPA 1998 training data and tested on DARPA 1999. Shin et al. [13] introduced a novel probabilistic approach which uses Markov chain for probabilistic modeling of abnormal events in network traffic. Performance of the proposed approach was evaluated by a set of experiments using the DARPA 2000 dataset. It achieved a high detection performance while representing level of attacks in stages. Bayesian Neural Networks classifier designed in [1] is discussed to be a part of a complete Intrusion Detection System. However, performed experiments were primarily aimed at network traffic classification. The work uses Moore's 2005 datasets discussed in [10]. The paper [6] describes a formal definition of the ASNM features describing various properties of

the connection-oriented communication intended for network anomaly intrusion detection. Later, performance is evaluated using the Naive Bayes classifier and CDX 2009 dataset [12]. In contrast to discussed group of non-payload based NBAD there exists group of NBAD whose representatives analyze application payload of packets too, but this group is not considered as related to our approach.

2.3 Network Traffic Classification

One of the related research branches is dealing with network traffic classification. The representative of connectionless traffic classification based on statistical properties of the flows is designed in [16]. The authors do not define handling of beginning and end of the UDP flows, but on the other hand, they mention the UDP flow has to contain at least two packets. Another work which includes connectionless & connection-oriented traffic classification using statistical approach is work [15] dealing with online game traffic classification methods. The authors of the paper define end of the UDP flow as 60s timeout in the sending of packets. Next example of work in connection-oriented & connectionless traffic classification group is work [11] which uses Linear and Quadratic Discriminate Analysis. A.Moore et al. proposed behavioral-based features for network traffic classification called discriminators [10]. Discriminators designed in this work perform analysis of connection-oriented network traffic only [1]. The authors discuss connectionless classification as future work, but they never performed it because of not easily identified beginning and end of the UDP flow.

3 Design of Features for Connectionless Communications

A collection of features designed for describing the characteristics of connectionless network buffer overflow attacks' behavior is detailed in this section. The input serving for computation of the features include information from packets' headers, packets' lengths and counts, time-arrival measurements and various information about simultaneous traffic, too. Features themselves are computed on UDP connections, because it is the most common connectionless protocol running at the transport layer of the TCP/IP protocol stack. Some of the other basic protocols, such as TCP and ICMP, are processed as well, but only for providing some auxiliary information about the behavior of a given UDP connection. We considered the nature and principles of two UDP based network services (SIP and TFTP) when designing and especially improving the feature set.

3.1 Categorization of the Features

There are 117 features defined and categorized into six groups according to their principles and input data used for their computation:

(a) **Data from packets' headers.** These features are computed from information of network and transport layers headers of UDP packets. Some fields of packets' headers, like MAC and IP addresses, are not taken into account because our NBAD approach is based on analysis of statistical and behavioral properties of UDP flows without considering any host-localization information. Also, port numbers are processed, as they are useful in another group of features.

(b) **Data from packets' counts and sizes.** This category contains various features which are based on statistical information considering the number of packets and payload lengths taken from data packets only. A data packet refers to any packet with a non-zero length application layer's payload.

(c) **Analysis of fragmented packets.** This category is a bit problematic in connectionless traffic because in some cases there is not a 100% assurance that a given fragmented packet belongs to the connection assigned to it. Therefore, only a few features are designed in this category.

(d) **Data from time slope and arrival time of packets.** Analysis of such data makes sense in the field of buffer overflow attacks, because malicious communication does not have to be regular and stable. On the other hand, it is arguable, because the time slope of any particular communication can have various characteristics which depend on the actual network traffic situation or possible obfuscation. Therefore, we will perform network traffic modifications of executed BO attacks in order to simulate real conditions, and thus this group of features will not be favored in data mining phase.

(e) **Analysis of quartile time segments.** This category is basically just a slightly sophisticated version of the previous one. It represents extraction of various features over 4 time segments (quartiles) for a period of each connection's duration.

(f) **Data from related connections in specified time context.** These features examine connections started before the beginning, during the progress and after the end of an analyzed UDP connection. It includes, for example, the presence of TCP connections, ICMP packets, and the detection of destination port change in the time context of analyzed connection. The time context is specified to include packets of connections occurred 30s before beginning and 30s after end of analyzed UDP connection. Connections related to the analyzed one are restricted by the time context. This category was designed in order to be useful when backdoor communication occurs or there exists a hidden relation between analyzed communication and another one.

The enumeration of categorized features is depicted in Table 1 and 2. Almost all the features listed in the tables consider packets from both directions of a connection together during their computation. Exceptional features which are computed for each direction separately are marked in the tables by an asterisk. The quartile-time flow analysis group contains features regarding one quartile only, but actually, the features are computed on all 4 quartiles. One of the most important issues taken into account during designing and improving features set is a connection searching mechanism which has to consider the fact that there is

Table 1: Summary of designed features (part 1/2)

Source	Group	Features
Type of Service	(a)	• mode,* median,* mean*
Time to Live	(a)	• minimum,* maximum,* sum,* median,* mode,* • mean,* standard deviation*
Port numbers	(a)	• source and destination port numbers
Packet counting	(b)	• the number of all incoming packets,* • the number of all outgoing packets,* • the ratio of incoming and outgoing packets, • the number of data packets, • the number of non-data packets, • the ratio of data packets and non-data packets counts
Payload lengths	(b)	• sum, mode, median, mean, standard deviation
Fragmentation	(c)	• the number of fragmented packets, • the number of non-fragmented packets, • the ratio of fragmented and non-fragmented packets counts, • sum of fragmented packet payload lengths, • the ratio of non-fragmented and fragmented packets' payload lengths
Packets' IAT	(d)	• minimum,* maximum,* median,* • mean,* standard deviation*

*Computed separately for each direction.

no exact sign of connections' boundaries in connectionless traffic. Therefore, we created a mechanism based on a time-out value for determining the end of each connection. The beginning of each connection is simply determined by occurrence of a packet with flow parameters (ports, IPs) not contained in the current list of running connections. It turned out that the number of found connections is very sensitive about the time-out value. For example, using our further dataset with the value of 30 seconds resulted in 900 connections being found, while using the value 180 seconds resulted in 500 connections being found. The feature extraction process also provided different values in both cases. Therefore, according to some of the Linux-based operating systems[1], Netgear VPN Firewall[2] and MikroTik[3] devices, we decided to utilize an approach respecting the following statements:

- if the transfer is in progress in one direction only, then the time-out is set to 30 seconds, as per default,
- if at least 1 packet occurred in the opposite direction, then the value is set to 180 seconds and the connection is now considered to be established.

[1] `http://www.iptables.info/en/connection-state.html`
[2] `http://documentation.netgear.com/fvx538/enu/202-10062-08/pdfs/FullManual.pdf`
[3] `https://www.mikrotik.com/testdocs/ros/2.8/ip/conserv.php`

Table 2: Summary of designed features (part 2/2)

Source	Group	Features
Transfer time	(d)	• duration of connection, • bytes per second, • the sum of time without data packets transfer, • the sum of time without data packets transfer – included into the sum only if higher than 2000ms, • the ratio of two above features
Quartile-time segments	(e)	• minimum, maximum, median, mode, mean and standard deviation of Ethernet data lengths, • ratio of means of Ethernet packets' data length in one quartile and the whole connection, • the number of all incoming packets,* • the number of all outgoing packets,* • the number of incoming data packets,* • the number of outgoing data packets,* • the ratio of incoming and outgoing packets counts, • the ratio of incoming and outgoing data packets counts
Related connections	(f)	• occurrence of at least 1 following TCP packet, • occurrence of at least 1 following ICMP packet, • port change detected in the related connection, • the number of previous related connections, • the number of following related connections
Consequent connections	(f)	• the number of packets, • the mean of packets' payload lengths, • the sum of packets' payload lengths, • the sum of packets' sizes, • the mean of packets' sizes, • standard deviation of packet sizes, • bytes per second

*Computed separately for each direction.

4 Network Traffic Simulation Experiments

Experiments were performed in a virtual network environment and were aimed at collecting a dataset containing UDP network buffer overflow attacks. The *VirtualBox*[4] tool was used for safe and legal network vulnerability exploitation. The simple scheme of network infrastructure that we used is illustrated in Figure 1. The only vulnerable virtual machine which was running under operating system Windows XP with Service Pack 3 was utilized. *Wireshark*[5] tool was used for capturing network traffic between the legitimate machine and attacker's machine – the one with operating system Ubuntu 14.10. The Windows XP machine

[4] `https://www.virtualbox.org/`
[5] `https://www.wireshark.org/download.html`

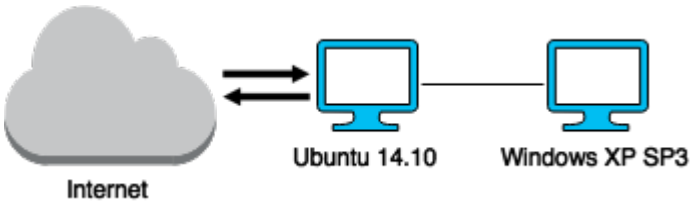

Fig. 1: Network scheme of exploitation environment

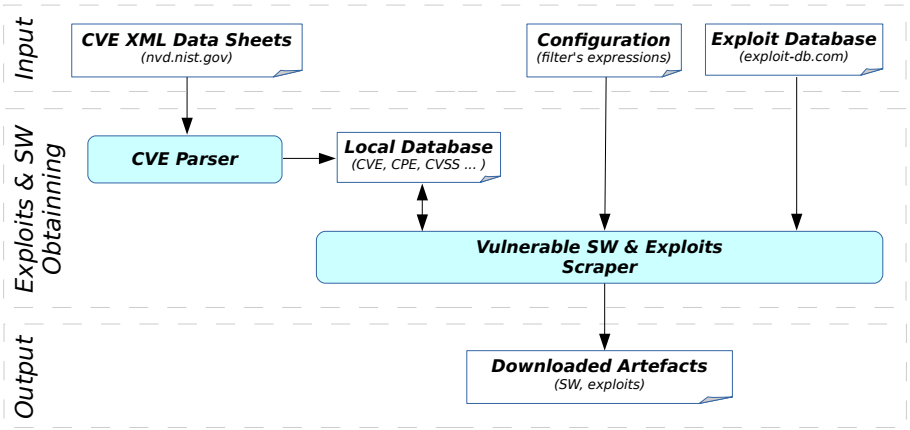

Fig. 2: Collection of exploits and vulnerable SW

was equipped with vulnerable applications and the firewall was disabled on it. A snapshot of the machine was taken after its initial setup, which served for a quick recovery after successful exploitation of the machine.

Exploits and some vulnerable applications associated with them were obtained by a tool which had been designed for this purpose. A scheme of the tool is illustrated in Figure 2. *CVE Parser* component builds a relational database from data sheets available at `nvd.nist.gov`. Consequently, *Vulnerable SW & Exploits Scraper* component utilizes the database by queries specifying UDP buffer overflow vulnerabilities with a high impact score. Then, fetched information is used for web scraping in a public database of exploits *exploit-db*[6] resulting in the download of exploits and vulnerable applications if they are available. We observed that the number of UDP network vulnerabilities present at nvd.nist.gov and exploit-db is not so high as in the case of TCP. Except for exploits downloaded from exploit-db, we used *Metasploit* framework[7] for some attack executions, too. As a result, we successfully executed 223 buffer overflow attacks, performed on various SIP and TFTP network services, which resulted into 433 UDP connection records in total. The attacks utilized various network traffic modifications: spre-

[6] `https://www.exploit-db.com/`
[7] `http://www.rapid7.com/products/metasploit/`

Table 3: Testing dataset distribution

Network Service	Count of UDP Records		
	Legitimate	**Attacks**	**Summary**
TFTP	52	356	408
SIP	65	77	142
Other UDP Traffic	5012	n/a	5012
Summary	5129	433	5562

ading out packets in time; segmentation & fragmentation; changing of packets' order; simulation of unreliable network channel; packets' loss; packets' duplication as well as their combinations. The summary of vulnerable applications together with CVEs are listed in Table 6 of Appendix.

Also, we collected legitimate network traffic from several public sources like *pcapr.net*[8] and *wireshark.org*[9] as well as from campus network environment. The final dataset is summarized in Table 3.

5 Data Processing and Analysis

Collected dataset was used as an input for the data mining process. A new tool was designed to extract proposed features' values for each UDP communication record. The tool is implemented in Python 2.7 using library *dpkt*[10] for parsing of packet headers. The scheme in Figure 3 illustrates the whole process which consists of the following three parts:

- The upper segment represents the input of the whole process. It contains files with captured network traffic, which was collected during a simulation of attacks and legitimate communications. Another part of this segment serves for obtaining ground truth (expert knowledge) from metadata of each connection. The ground truth provides information about the type of communication (attack or legitimate) and identification of vulnerability (CVE-ID). Directory structure has been designed specifically for this purpose as well as metadata included in the input with additional files.
- The middle segment performs data processing. The *Communication Extractor* module executes parsing of tcpdump files. It also aggregates UDP packets from input network traffic into connection records. Then, with the use of *Expert Knowledge Processor*, each connection record is labeled by all necessary data. Such objects are passed to *Feature Extraction Preprocessor*, which prepares some of the embedded data for easier and faster processing. The last module in this segment is *Features Extractor* which performs the extraction process itself on each UDP connection record and outputs the CSV file.

[8] `http://pcapr.net/home`
[9] `https://wiki.wireshark.org/SampleCaptures`
[10] `https://code.google.com/p/dpkt/`

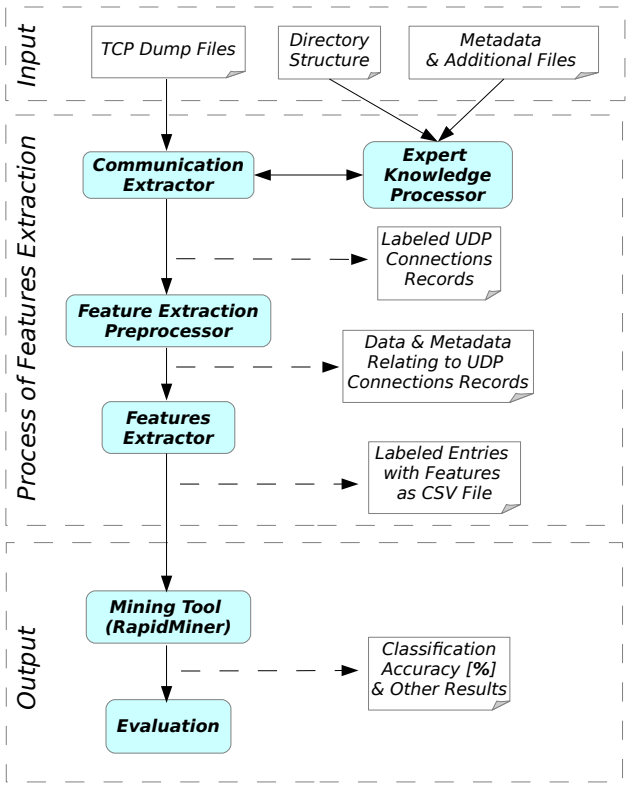

Fig. 3: Scheme of data processing and analysis

– The bottom segment illustrates the mining and assessment process. It produces results based on data stored in the CSV file. In this work we use *RapidMiner*[11] tool for machine learning process and for evaluation purposes.

6 Summary of the Results

The purpose of this stage is to perform supervised classification in order to evaluate discrimination properties of proposed features by analyzing feedback of a classifier. All experiments were performed by the Naive Bayes classifier employing kernel weighting function for feature-independent estimation of value density distribution, which is considered as non-parametric estimation model. Considering our testing dataset distribution, 5-fold cross validation method was selected for all classification experiments. We performed Forward Feature Selection (FFS) method for selection of the best features. FFS started to run with an empty set of features and in each iteration added a new feature contributing

[11] `https://rapidminer.com/products/studio/`

by the best improvement of average recall of all classes. The average recall of all classes was computed using the underlying 5-fold cross validation evaluating the Naive Bayes classifier. In FFS, we allowed the acceptance of one iteration without improvement, as we wanted to alleviate the possibility of the selection process becoming stuck in local extremes. Our experiments consisted of two executions of the FFS. The first execution took as input the whole testing dataset. The second one took as input testing dataset excluding other UDP traffic. We performed only little optimization of Naive Bayes model, and in the result, set fixed bandwidth of kernel function to 0.1.

The performance result of the first FFS experiment is depicted in Table 4. The confusion matrix obtained in this experiment shows high precision and

Table 4: FFS with 5-fold cross validation of the whole dataset

Classification Accuracy: 99.75% ± 0.13%		**True Class**		**Precision**
		Legitimate	**Attacks**	
Predicted Class	**Legitimate**	5122	7	99.86%
	Attacks	7	426	98.38%
	Recall	99.86%	98.38%	$F_1 = 98.38\%$

recall of both classes. Note that F_1 notation represents *F-measure₁* which equally balances precision and recall measures for positive class (Attacks).

We excluded other traffic from the second execution of FFS, and thus validated discrimination properties of our features only for TFTP and SIP traffic. The associated confusion matrix is depicted in Table 5. Comparing the result of

Table 5: FFS with 5-fold cross validation of all SIP and TFTP traffic

Classification Accuracy: 99.82% ± 0.36%		**True Class**		**Precision**
		Legitimate	**Attacks**	
Predicted Class	**Legitimate**	117	1	99.15%
	Attacks	0	432	100.00%
	Recall	100.00%	99.77%	$F_1 = 99.88\%$

this experiments to the previous one, we can see that better discrimination property between malicious and legitimate UDP traffic was achieved. Thus, we can observe that explicit training data of specific service comprised of both malicious and legitimate instances provides a better performance results than training data containing only representatives of one class.

7 Conclusion

The main objective of this work is to design a non-payload based collection of features describing statistical, behavioral and contextual characteristics of connectionless network traffic aimed at anomaly detection of buffer overflow attacks. The work also deals with the connection searching problem taking into consideration the fact that there is no exact sign of the connections' boundaries in connectionless traffic. This problem is resolved using a time-out approach inspired by existing devices and Unix based operating systems.

The performance of the designed feature set is evaluated by Naive Bayes classifier and 5-fold cross validation method. The summary of the presented results shows that our feature set is successfully utilized in classification of UDP network traffic into malicious and legitimate classes.

Acknowledgments

This article was created within the project Reliability and Security in IT (FIT-S-14-2486) and supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II); project IT4Innovations excellence in science - LQ1602.

References

1. Auld, T., Moore, A.W., Gull, S.F.: Bayesian neural networks for internet traffic classification. Neural Networks, IEEE Transactions on 18(1), pp. 223–239 (2007)
2. Barbara, D., Wu, N., Jajodia, S.: Detecting novel network intrusions using bayes estimators. In: SDM. pp. 1–17. SIAM (2001)
3. Copeland III, J.A.: Flow-based detection of network intrusions (2007), uS Patent 7,185,368
4. Corona, I., Giacinto, G., Roli, F.: Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues . Information Sciences 239, pp. 201–225 (2013)
5. Cowan, C., Wagle, P., Pu, C., Beattie, S., Walpole, J.: Buffer overflows: Attacks and defenses for the vulnerability of the decade. In: DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings. vol. 2, pp. 119–129. IEEE (2000)
6. Homoliak, I., Barabas, M., Chmelar, P., Drozd, M., Hanacek, P.: ASNM: Advanced security network metrics for attack vector description. In: Proceedings of the 2013 International Conference on Security & Management. pp. 350–358. Computer Science Research, Education, and Applications Press (2013)
7. ISO Standard IS 8072: Information Processing Systems - Open Systems Interconnection - Transport Service Definition. Tech. rep., International Organization for Standardization (1986)
8. Lee, W., Stolfo, S.J.: Data mining approaches for intrusion detection. In: Usenix Security (1998)
9. Mahoney, M.V., Chan, P.K.: Phad: Packet header anomaly detection for identifying hostile network traffic (2001)

10. Moore, A.W., Zuev, D., Crogan, M.: Discriminators for use in flow-based classification. Tech. rep., Technical report, Intel Research, Cambridge (2005)
11. Roughan, M., Sen, S., Spatscheck, O., Duffield, N.: Class-of-service mapping for qos: a statistical signature-based approach to ip traffic classification. In: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement. pp. 135–148. ACM (2004)
12. Sangster, B., OConnor, T., Cook, T., Fanelli, R., Dean, E., Adams, W.J., Morrell, C., Conti, G.: Toward instrumenting network warfare competitions to generate labeled datasets. In: Proc. of the 2nd Workshop on Cyber Security Experimentation and Test (CSET09) (2009)
13. Shin, S., Lee, S., Kim, H., Kim, S.: Advanced probabilistic approach for network intrusion forecasting and detection. Expert Systems with Applications (2012)
14. Thomas, C., Sharma, V., Balakrishnan, N.: Usefulness of darpa dataset for intrusion detection system evaluation. In: SPIE Defense and Security Symposium. pp. 69730G–69730G. International Society for Optics and Photonics (2008)
15. Williams, N., Zander, S., Armitage, G.: Evaluating machine learning methods for online game traffic identification. Centre for Advanced Internet Architectures, http://caia. swin. edu. au/reports C 60410 (2006)
16. Zander, S., Nguyen, T., Armitage, G.: Automated traffic classification and application identification using machine learning. In: Local Computer Networks, 2005. 30th Anniversary. The IEEE Conference on. pp. 250–257. IEEE (2005)

Appendix

Table 6: Vulnerable UDP network services

Application	Version	CVE*	Attacks
Tftpd32	$\leq$ 2.21	2002-2226	3
TFTP Server TFTPDWIN	0.4.2	2006-4948	38
3Com TFTP Service	2.0.1	2006-6183	19
Allied Telesyn TFTP	2.0	2006-6184	37
Quick TFTP Pro	2.1 & 2.2	2008-1610	36
TFTPUtil GUI	1.4.5	2010-2028	19
Serva 32 TFTP	2.1.0	2013-0145	18
Distinct TFTP	3.10	2012-6664	17
SIPfoundry's sipXezPhone	0.35a	2006-3524	18
SIPfoundry's sipXphone	2.6.0.27	2006-3524	18
Overall number of attacks			223

*`http://cve.mitre.org/cve/identifiers/index.html`