# Characteristics of Buffer Overflow Attacks Tunneled in HTTP Traffic

Ivan Homoliak*, Daniel Ovsonka*, Karel Koranda* and Petr Hanacek*
*Faculty of Information Technology,
Brno University of Technology,
Bozetechova 1/2, 612 66 Brno, Czech Republic
Email: {ihomoliak, iovsonka, ikoranda, hanacek}@fit.vutbr.cz

***Abstract*—The purpose of this article is to describe characteristics of obfuscated network buffer overflow attacks in contrast with characteristics of directly simulated attacks. The obfuscation was performed by tunneling of malicious traffic in HTTP and HTTPS protocols. These protocols wrap a malicious communication between an attacker situated outside of an intranet and a callback located inside of an intranet. The detection analysis which we perform is based on features extraction from network packets dumps and it employs a behavioral and statistical analysis of communications' progress in time and packet index domain. There were performed experiments in four scenarios simulating traffic shaping, traffic policing and transmission on unreliable network channel to make properties of direct attacks and obfuscated attacks as various as possible. Next part of this article is comparison of obfuscated and direct attacks classification by our previously designed ASNM network features with state-of-the-art features set of A. Moore, both representing statistical and behavioral based experimental academic kernels for NBA. Presented results show better classification accuracy of ASNM features in all kinds of experiments.**

***Keywords*—protocol tunneling, network vulnerabilities, buffer overflow, obfuscation, NBA, AIPS, ASNM.***

## I. Introduction

The category of buffer overflow attacks is considered as the most dangerous one, because it provides an attacker the ability to inject and execute attack code. The injected attack code runs with the privileges of the vulnerable application and enables the attacker to execute any functionality necessary to control the compromised PC [1]. Buffer overflow attacks are easy to perform and there exists many tools and tutorials for buffer overflow attacks execution [2], [3], [4], [5].

The most of the previous and current research on the field of network attacks detection and traffic classification presents various methods pursued in their performance in contrast of lacking operational deployment of such systems [6]. These methods employ different machine learning algorithms and data mining techniques to examine and analyze some dataset without concerning how dataset was collected and traffic simulation performed. The reasons why these systems are not deployed in real traffic environment can be different. For example, it can be caused by over-learned classifiers or by polymorphism and metamorphism of the network malware or by the fact, that zero-day attacks are incessantly introducing different behavioral characteristics.

These thoughts and facts bring us to consideration of making malicious traffic simulation as various as possible. We suppose to use diverse obfuscation techniques of network attacks causing classifiers trained without knowledge of these modifications unable to provide acceptable response. Providing of supposed data modifications to a training phase of a classifiers can strengthen their detection capabilities. In the other words, we want to prepare classifiers for occurrence of various obfuscation techniques of an attacker. In this article we use a protocol tunneling technique of attacks' obfuscation.

In our previous article [7] we proposed a method called Automated Intrusion Prevention System (AIPS) which performs network buffer overflow attacks detection by statistical and behavioral analysis of network communications using network features called Advanced Security Network Metrics (ASNM) originally designed in [8]. Our next paper [9] examined the detection properties of this system on publicly available dataset CDX 2009 [10] which was collected during warfare competition, in which one of the goal was to generate labeled dataset. Another objective of this paper was to evaluate and compare detection properties of AIPS and state-of-the-art features of A. Moore called discriminators [11] on the CDX 2009 dataset. The result of these experiments show similar detection properties of both of the features sets (offering high attacks' detection capability). The dataset used in this paper was not generated using any obfuscation techniques, therefore arise new challenge for comparative and evaluation study using mentioned feature sets.

The paper is organized as follows. The section II discusses related work in the field of network traffic obfuscation with emphasis on protocol tunneling and detection of tunneled traffic. In the section III we briefly describe the principles of our custom obfuscation system which performs tunneling. Section IV mentions specific network vulnerabilities, which we exploited. In the section V we depict virtual network architecture established for a traffic simulation and we describe malicious and legitimate traffic simulation with various scenarios covering real network conditions. In the section VI we focus on characteristics of obfuscated malicious traffic and we present some examples of features. Next section VII summarize traffic classification experiments comparing properties of ASNM and discriminator features. The last section VIII contains conclusion of this paper.

## II. State of the art

Payload tunnels are covert channels that tunnel one protocol in the payload of another protocol. One of the purposes of these channels is circumventing firewalls that limit outgoing traffic to few allowed application protocols. A variety of tools exist for tunneling over application protocols that are often not blocked such as ICMP or HTTP [12].

One of the first approaches for tunneling protocols over ICMP was Loki, which tunneled data in the payload of ICMP echo messages [13]. Zelenchuk implemented an indirect IP over ICMP tunnel [14]. The covert sender sends echo request packets to a bounce host with spoofed source address (set to the address of the covert receiver) and the covert data encoded in the payload. The bounce host then sends echo replies to the covert receiver with the same payload as in the requests.

Another popular method is to tunnel protocols over HTTP. Padgett developed a tool that tunnels SSH through HTTP proxies [15]. Dyatlov and LeBoutillier implemented tools for tunneling UDP and TCP over HTTP [16], [17]. Lundstrm implemented a tool that can establish a bi-directional tunnel over the exchange of emails [18].

The authors Crotti et. al. in their work [19] proposed application of a statistically-based traffic classification technique to detect tunneled protocols inside of HTTP by the analysis of inter-arrival time, size and order of the packets crossing a gateway. They describe the technique which effectively enhance application level gateways and firewalls, helping to better apply network security policies on such tunneled traffic.

The Web Tap's [20] focus is aimed on detecting attempts to send significant amounts of information via HTTP tunnels to rogue web servers from within firewalled internal network. A related goal of Web Tap is to help detect spyware programs, which often send out personal data to servers using HTTP transactions and may open up security holes in the network. Based on the analysis of HTTP traffic over a training period, the authors use filters to help in detection of anomalies in outbound HTTP traffic using metrics such as request regularity, bandwidth usage, inter-request delay time, and transaction size. The Web Tap can be evaded by the adversary by monitoring and analysis of users outbound traffic and then mimic the access patterns of a legitimate site.

Dusi et al. presented a mechanism called Tunnel Hunter, which can successfully identify protocols tunneled inside tunneling protocols such as HTTP, DNS and SSH [21]. It is performed by statistical analysis of simple IP level flow features. Their technique suffers from the problem of sensitivity to packet-size and timing value manipulation.

Sohn et al. [22] used the SVM-based approach to evaluate the accuracy of detecting covert channels embedded in ICMP echo packets and achieved classification accuracy of up to 99 percent when training a classifier on normal and abnormal packets from Windows, Solaris and Linux.

Pack et al. [23] proposed detecting HTTP tunnels by using behavior profiles of traffic flows. Behavior profiles are based on a number of metrics such as the average packet size, ratio of small and large packets, change of packet size patterns, total number of packets sent and received, and connection duration. If the behavior of a flow under observation deviates from the normal HTTP behavior profile it is likely to be an HTTP tunnel.

The authors Wagner D. et al. [24] introduced the notion of a mimicry attack which can cloak the attacks behavior to avoid detection of IDS by generating usual system calls into system calls sequence of an attack. Next, they develop a theoretical framework for evaluating the security of an IDS against mimicry attacks.

Fogla et al. [25] realized the obfuscation of network attacks by proposal of a new class of polymorphic attacks, called polymorphic blending attacks (PBA), that can effectively evade byte frequency-based network anomaly IDS. The attacks match the statistics of the mutated attack instances to the normal profiles. They demonstrated the efficiency of PBA attacks on PAYL (byte-frequency based IDS). In the next paper [26] they showed that in general, generating a PBA that optimally matches the normal traffic profile is a hard problem (NP–complete), but can be reduced to SAT or ILP problems. They presented a formal framework for PBA attacks and they also proposed a technique to improve the performance of an IDS against PBAs.

Some of the related work was revealed in the survey of covert channels and countermeasures in computer network protocols collected in the year 2007 [12].

## III. Tunneling obfuscation

Our custom obfuscation system was created for the purpose of tunneling examined data. The system consists of two modules: an attackers module and a callback module. The attackers module act as a fake web server and waits for connection from the callback module. When the connection is established, all selected protocols are tunneled by carrying protocols (HTTP or HTTPS). Tunneled traffic is bypassed and processed on low system level making it transparent to higher network layers. Therefore, communicating entities can not notice the data is tunneled and they do not need to process obfuscated data.

The core of the implementation is based on `libnetfilter_queue` library which allows us to move packet processing from kernel space to user space. All packets are filtered with `iptables`, thus the only obfuscated packets are sent for further processing of our system. A connection is initiated from the callback module when a request is sent. Request also contains encapsulated meta-data to identify callback module at web-server and it can contain data gathered from internal network e.g. response from another host in private network. An attacker encapsulates data ready for obfuscation into carrying protocol and sends them to the callback module as a standard web-server response. The callback module restore the encapsulated data and distributes them into private network. The callback module also catches the responses from internal network, encapsulates them and sends them back to the fake web-server. This technique makes it really hard to detect tunneled malicious traffic with classic signature based approaches. Experiments using SNORT IDS [27] were performed and all obfuscated attacks on internal networks' vulnerabilities were undetected.

### A. Callback module installation

The callback module have to be installed on a machine situated inside of an internal network. The successful deployment of the callback module mostly depends on privilege escalation of a host machine. It can be performed by various approaches e.g. an attacker can use an infected website which exploits

| Service | Connections | | |
|---|---|---|---|
| | **Legitimate** | **Attack** | **Obfuscated attack** |
| **Apache** | 263 | 101 | 72 |
| **BadBlue** | 166 | 4 | 10 |
| **DCOM RPC** | 222 | 4 | 8 |
| **Samba** | 18 | 20 | 8 |

TABLE I. TRAFFIC CLASSES DISTRIBUTION (TCP CONNECTIONS).

hosts browser and consequently installs a backdoor to the host. Another way of the backdoor deployment can be executed by a user who installs it as a part of useful application from an untrusted source (trojan horse). A host machine can be also exploited directly in the case it have an exposed open port to public network which is bounded to a vulnerable application. When the attacker gain access to a host machine, the callback module can be deployed there.

Successfully installed callback module starts to send periodic requests to the fake web-server representing an attackers machine. The name callback is derived from the character of communication with command and control server of an attacker. All communications have to be initiated from the callback module, therefore we can evade firewall which usually drops all incoming communication from untrusted networks. In our case we use wide-spread HTTP/HTTPS protocols making the probability of dropping the callbacks traffic very low.

## IV. VULNERABILITIES

There were selected four vulnerable applications for our experiments. The main criteria of vulnerable application selection was existence of public exploit which is easy detectable by classic signature based IDS. The set of vulnerable application we used consists of Apache web-server (version 1.3.20) with `mod_ssl` plug-in (version 2.8.4). This version of httpd server is vulnerable to buffer overflow in certificate verification. Attacker can execute arbitrary code via special crafted certificate. Second vulnerability is included in BadBlue simple web-server (version 2.72b) which is vulnerable to stack-based buffer overflow and it is possible to execute arbitrary code via special crafted long query string. These web-servers were selected because of we want to prove the obfuscation technique can encapsulate even HTTP into HTTP protocol without detection by IDS. Third tested vulnerability is part of standard Windows 2000 (Service Pack 4) installation. DCOM RPC interface is vulnerable to buffer overflow which allows attacker to execute arbitrary code on a host machine. The last application is Samba server (version 2.2.0) which is also vulnerable to buffer overflow attack. All tested vulnerabilities allow to fully compromise target machines and even remote attacker can gain root privileges.

## V. TRAFFIC SIMULATION

Designed virtual network consists of two separate networks interconnected through the gateway node. The first network represents private network and the second one represents public untrusted network. We used Metasploit and one publicly known exploit to perform attacks on vulnerable applications. All gathered data were dumped by Wireshark to PCAP files from gateway node because of all important packets were routed through this node. The traffic class distribution of all collected dump files is listed in the Table I.

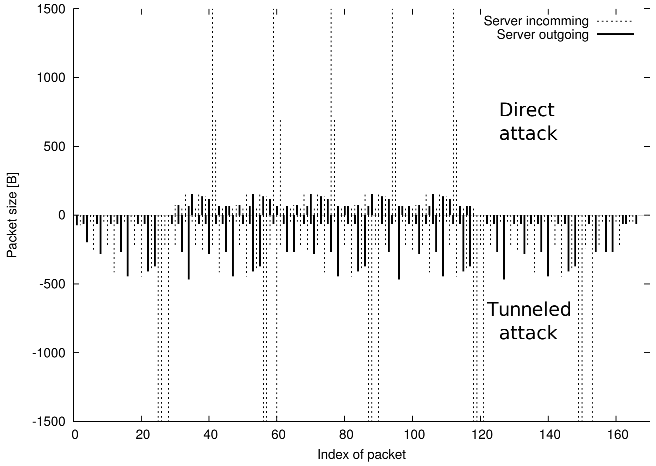

Fig. 1. Traffic flows of direct and tunneled attack on Samba service.

We aimed to simulate real network traffic, therefore we created custom set of experiments which covers different situations reflecting real networks aspects. Testing scenarios differ in the modification degree of the traffic on gateway node. All packets remained unchanged in the first scenario. The second scenario simulates network overload by the means of traffic shaping, thus the gateway node was under heavy load causing change of time differences between packets. The third scenario represents transmission on unreliable network channel, thus we used random damage for 25% of transmitted packets. The last scenario simulates network policing so we have to drop selected packets on gateway node. Every malicious and legitimate communication was tested in each testing scenario.

## VI. CHARACTERISTICS' ANALYSIS

There were used ASNM [9] features of AIPS and discriminators features of A. Moore [11] for the purpose of traffics characteristics analysis in order to evaluate and plot the behavioral and statistical properties of simulated traffic. Both features sets represent academic experimental NBA based on statistical and behavioral analysis of the traffics flow. Representative example depicting the traffic flow differences between tunneled and direct malicious traffic is Samba service whose traffic is illustrated in the Fig. 1. The possitive sign of the y axis indicates direct malicious traffic and negative sign indicates tunneled malicious traffic.

We examined the value density distribution of each feature with emphasis on distinctiveness of obfuscated and direct malicious traffic. For the purpose of value density examination, we used kernel density estimation using Gaussian kernels. We selected some features presenting the interesting characteristics of each traffic. In some features we performed raw value density examination by frequency analysis because of clearer interpretation. At first, we present discriminating features enabling good malicious and legitimate traffic separation. Further, we present features representing specific characteristics of obfuscated attacks in comparison with direct attacks making the detection of such obfuscated attacks more complicated from the view of NBA.

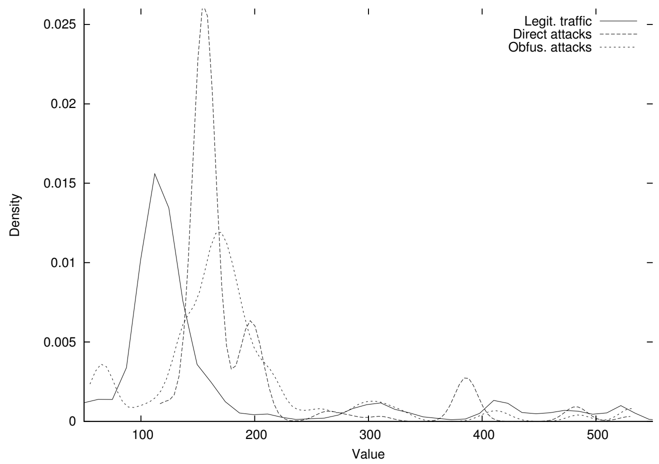

Fig. 2. Mean of total bytes in IP packets of all traffic.

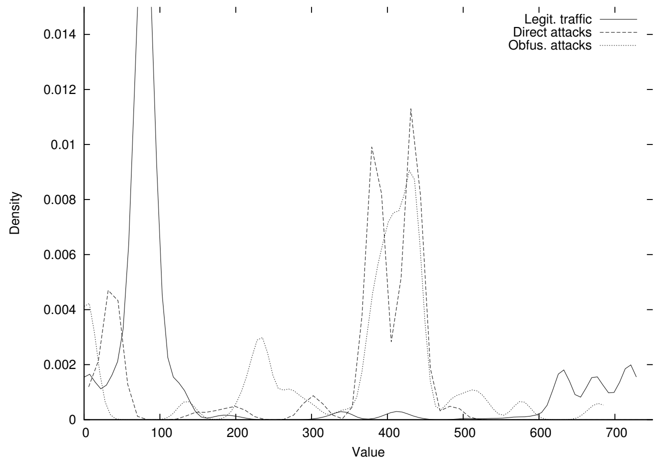

Fig. 3. Standard deviation of packets' lengths from server to client.

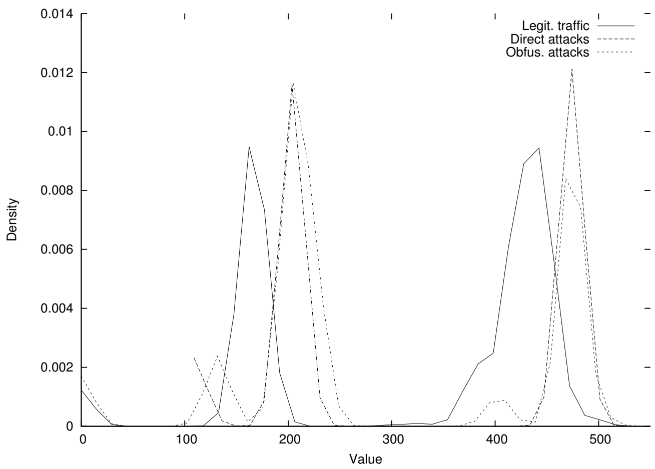

Fig. 4. Maximum segment size of TCP connection from client to server.

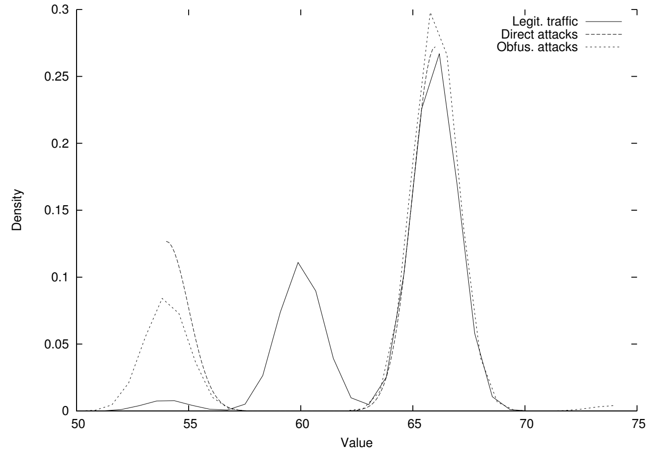

Fig. 5. Minimum number of bytes in Ethernet packet from server to client.

### A. Discriminating features

The first discriminating feature we present is mean of total bytes in IP packets. The Gaussian kernel density estimation of this feature is shown in the Fig. 2. It is expected, that tunneled attacks dispose by higher values of bytes in IP packets. The obvious result of this feature represents both of the attacks' classes having slightly different values than legitimate traffic has. The similar characteristics has another example of discriminating feature – standard deviation of packets' lengths from server to client illustrated in the Fig. 3.

The maximum segment size observed during the lifetime of the connection is the next example of these features. It is shown in the Fig. 4. Values of this feature are very similar for obfuscated and direct attacks classes. There exist value differences between both attacks classes and legitimate class.

Next feature representing this group is the maximum of bytes in Ethernet packet. This feature indicates very similar value distribution like maximum segment size of TCP connection, but it differs in higher values caused by addition of Ethernet headers to overall byte sum. The Gaussian kernel density estimation is not shown because it is almost identical to previous one.

The next interesting feature from actual group represents minimum number of bytes in Ethernet packet transmitted from server to client. The Gaussian kernel density estimate of this feature is illustrated in the Fig. 5. The third Gaussian curve does not dispose any differentiate capabilities, but the other two do. The first one shows the similar properties of subset of direct and obfuscated attacks. The second one models subset of legitimate traffic.

### B. Obfuscated features

The second group of features is aimed on quality of obfuscation. The first example of this group is the fast Fourier transformation (FFT) of inter arrival time of all traffic. The histogram of this feature is shown in the Fig. 6. This feature represents just the magnitude of the third frequency of the FFT. The most important result shows similar values of tunneled attack instances and direct attacks instances, which can be considered as successful obfuscation of malicious traffic imitating the behavior of legitimate traffic. Another observation shows differences between majority of tunneled attacks instances and direct attack instances resulting into good features discrimination of these two classes.

The Gaussian kernel density estimation of feature representing the discrete fast Fourier transformation (DFFT) of packets' length is illustrated in the Fig. 7. The feature denotes

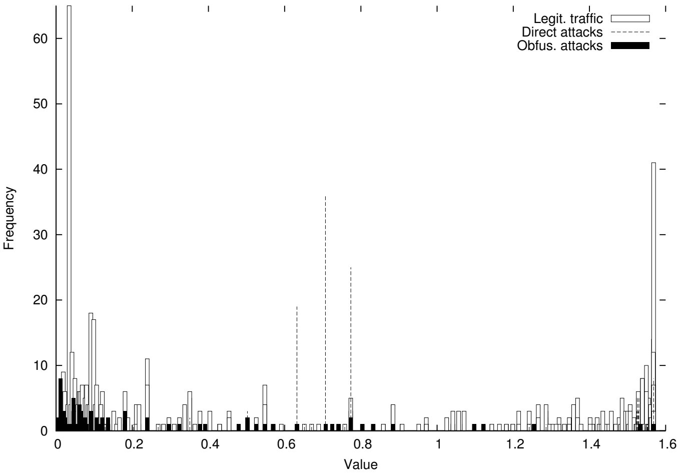

Fig. 6. FFT of inter arrival time for all traffic.

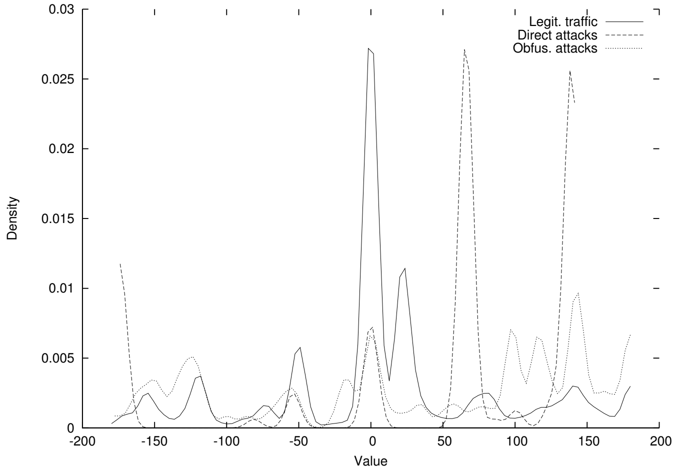

Fig. 7. DFFT of packets' lengths in direction from client to server.

the angle of goniometric representation of the fourth frequency of DFFT. The important matter observable from this figure is the specific distribution of direct attacks instances and wide spread value distribution of the tunneled attack instances reflecting very similar properties than in the legitimate traffic class. This characteristic stands for superior obfuscation of tunneled attacks and therefore makes harder to detect these attacks using actual feature. The same statements can be claimed about feature representing the Gaussian curves convolution with all traffic, which is illustrated in the Fig. 8.

## VII. Traffic classification

There were performed classification experiments comparing the detection capabilities of ASNM features with discriminator features of A. Moore [11]. The 5-cross fold validation was applied for our experiments using the same conditions and dataset. There were simultaneously performed metrics extraction and discriminators extraction processes resulting into two datasets of TCP connections' features. To ensure having the same dataset for both experiment categories, we performed intersection of both previously obtained datasets resulting into two dataset files with the same number of entries representing the same connections' entries. The first file contained metrics features of connections and the second one contained discriminators features of these connections. The datasets were analyzed by RapidMiner Studio [28] tool using forward feature selection of Naive Bayes classifier limited to fifteen generations and constrained by maximally two generations without improvement. We used attributes discretization into specified numbers of bins.

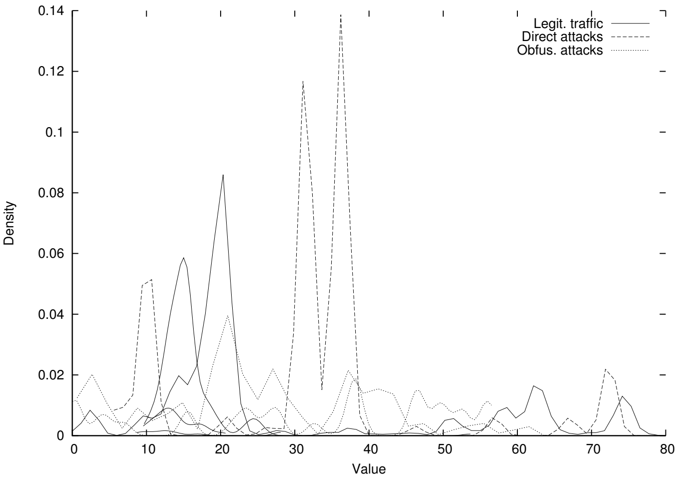

Fig. 8. Gaussian curves convolution with all traffic.

In the first experiment we compared poly-nominal classification of both features set using the discretization into 5 bins. Notice by poly-nominal there is meant classification into 13 classes where communications on each vulnerable service may contain three classes (legitimate traffic, direct attacks and tunneled attacks) resulting into 12 classes. One additional class represents the other traffic which was captured during our traffic simulations. The achieved results show better classification accuracy for ASNM – 98.85% with precision ±0.61% in contrast with discriminators – 93.74% with precision ±1.36%.

The next experiment was dedicated to traffic classification into 3 classes representing direct and obfuscated malicious traffic and legitimate traffic. We aimed to optimize detection properties of both features set by using the most convenient discretization. As in previous experiment, we achieved better classification accuracy in ASNM case – 99.69% with precision ±0.26% in contrast with discriminators – 98.12% with precision ±0.78%. Note the best accuracy of ASNM was achieved with discretization into 5 bins and the best accuracy of discriminators was achieved with discretization into 20 bins. The confusion matrices for this experiments are depicted in the Table II and III.

| **Accuracy:**<br>99.69% ± 0.26% | **True**<br>**obfus. attack** | **True**<br>**direct attack** | **True**<br>**legitimate** | **Class**<br>**precision** |
|---|---|---|---|---|
| **Pred. obfus. attack** | 90 | 4 | 0 | 95.74% |
| **Pred. direct attack** | 4 | 121 | 2 | 95.28% |
| **Pred. legitimate** | 4 | 4 | 730 | 98.92% |
| **Class recall** | 91.84% | 93.80% | 99.73% | |

TABLE II. Confusion matrix of discriminators features.

| Accuracy:<br>98.12% ± 0.78% | True<br>obfus. attack | True<br>direct attack | True<br>legitimate | Class<br>precision |
|---|---|---|---|---|
| **Pred. obfus. attack** | 96 | 0 | 0 | 100.00% |
| **Pred. direct attack** | 0 | 129 | 1 | 99.23% |
| **Pred. legitimate** | 2 | 0 | 731 | 99.73% |
| **Class recall** | 97.96% | 100.00% | 99.86% | |

TABLE III. CONFUSION MATRIX OF ASNM FEATURES.

## VIII. CONCLUSION

This paper presents significant part of our research aimed on malicious network traffic obfuscation by tunneling in HTTP and HTTPS protocols. There were examined interesting characteristics of tunneled and direct buffer overflow network attacks. We divided observed characteristics into two categories. First one called discriminating features, represents properties which are useful in detection phase of obfuscated attacks. These features are directly able to differentiate between legitimate traffic class and malicious traffic class, therefore they specify direct characteristics of malicious traffic (obfuscated and direct). The later category called obfuscated features, represents features which were significantly influenced by our tunneling obfuscation technique and therefore can not be directly used to classify malicious and legitimate traffic. Examples of this features group serve as prove of obfuscations success. Thus, the tunneling obfuscation provides special feature space distribution and therefore it is important to include it into training phase of classifiers.

Consequently, we performed traffic classification experiments comparing performance of Naive Bayes classifier using ASNM metrics and discriminators of A. Moore subsequently. The obtained result of poly-nominal classification equals to 98.85% accuracy of ASNM metrics and 93.74% accuracy of discriminators. In the next experiment we employed three class classification and there was achieved 99.69% accuracy for ASNM metrics and 98.12% accuracy for discriminator features. Both experiments demonstrate better classification performance of ASNM features.

## ACKNOWLEDGMENT

The work was supported by the EU/Czech IT4Innovations Centre of Excellence project CZ.1.05/1.1.00/02.0070 and the internal BUT projects FIT-S-12-1 and FIT-S-14-2486.

## REFERENCES

[1] C. Cowan, P. Wagle, C. Pu, S. Beattie, and J. Walpole, "Buffer overflows: Attacks and defenses for the vulnerability of the decade," in *DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings*, vol. 2. IEEE, 2000, pp. 119–129.

[2] A. One, "Smashing the stack for fun and profit," *Phrack magazine*, vol. 7, no. 49, pp. 14–16, 1996.

[3] "Writing buffer overflow exploits – a tutorial for beginners," ECE/CIS labs, University of Delaware. [Online]. Available: http://www.eecis.udel.edu/ bmiller/cis459/2007s/readings/buff-overflow.html

[4] "How to write Buffer Overflows," Insecure.Org. [Online]. Available: http://insecure.org/stf/mudge_buffer_overflow_tutorial.html

[5] "Metasploit project," Rapid7 Community. [Online]. Available: http://www.metasploit.org

[6] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Security and Privacy (SP), 2010 IEEE Symposium on*. IEEE, 2010, pp. 305–316.

[7] M. Barabas, M. Drozd, and P. Hanáček, "Behavioral signature generation using shadow honeypot," in *World Academy of Science, Engineering and Technology*, ser. Issue 65, May 2012, Tokyo, Japan, no. 65. World Academy Science Engineering Technology, 2012, pp. 829–833.

[8] I. Homoliak, "Metrics for intrusion detection in network traffic," Master's thesis, University of Technology Brno, Faculty of Information Technology, Department of Intelligent Systems, 2011, In Slovak Language.

[9] I. Homoliak, M. Barabas, P. Chmelar, M. Drozd, and P. Hanacek, "ASNM: Advanced security network metrics for attack vector description," in *Proceedings of the 2013 International Conference on Security & Management*. Computer Science Research, Education, and Applications Press, 2013, pp. 350–358.

[10] B. Sangster, T. OConnor, T. Cook, R. Fanelli, E. Dean, W. J. Adams, C. Morrell, and G. Conti, "Toward instrumenting network warfare competitions to generate labeled datasets," in *Proc. of the 2nd Workshop on Cyber Security Experimentation and Test (CSET09)*, 2009.

[11] A. W. Moore, D. Zuev, and M. Crogan, "Discriminators for use in flow-based classification," Technical report, Intel Research, Cambridge, Tech. Rep., 2005.

[12] S. Zander, G. J. Armitage, and P. Branch, "A survey of covert channels and countermeasures in computer network protocols." *IEEE Communications Surveys and Tutorials*, vol. 9, no. 1-4, pp. 44–57, 2007.

[13] Magazine, Phrack, "7 (51) September 01, 1997, article 06 of 17 [LOKI2 (the implementation)]."

[14] I. Zelenchuk, "Skeeveicmp bounce tunnel," 2004.

[15] P. Padgett, "Corkscrew," 2001. [Online]. Available: http://www.agroman.net/corkscrew/

[16] A. Dyatlov, "Firepass - is a tunneling tool," 2003. [Online]. Available: http://gray-world.net/pr_firepass.shtml

[17] P. LeBoutillier, "Httunnel," 2005. [Online]. Available: http://sourceforge.net/projects/httunnel/

[18] M. Lundstrm, "Mailtunnel," 2010. [Online]. Available: http://gray-world.net/tools/mailtunnel-0.2.tar.gz

[19] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli, "Detecting http tunnels with statistical mechanisms," in *Communications, 2007. ICC'07. IEEE International Conference on*. IEEE, 2007, pp. 6162–6168.

[20] K. Borders and A. Prakash, "Web tap: detecting covert web traffic," in *Proceedings of the 11th ACM conference on Computer and communications security*. ACM, 2004, pp. 110–120.

[21] M. Dusi, M. Crotti, F. Gringoli, and L. Salgarelli, "Tunnel Hunter: Detecting application-layer tunnels with statistical fingerprinting," *Computer Networks*, vol. 53, no. 1, pp. 81–97, 2009.

[22] T. Sohn, J. Moon, S. Lee, D. H. Lee, and J. Lim, "Covert channel detection in the icmp payload using support vector machine," in *Computer and Information Sciences-ISCIS 2003*. Springer, 2003, pp. 828–835.

[23] D. J. Pack, W. Streilein, S. Webster, and R. Cunningham, "Detecting http tunneling activities," DTIC Document, Tech. Rep., 2002.

[24] D. Wagner and P. Soto, "Mimicry attacks on host-based intrusion detection systems," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*. ACM, 2002, pp. 255–264.

[25] P. Fogla, M. Sharif, R. Perdisci, O. Kolesnikov, and W. Lee, "Polymorphic blending attacks," in *Proceedings of the 15th USENIX Security Symposium*, 2006, pp. 241–256.

[26] P. Fogla and W. Lee, "Evading network anomaly detection systems: formal reasoning and practical techniques," in *Proceedings of the 13th ACM conference on Computer and communications security*. ACM, 2006, pp. 59–68.

[27] "SNORT intrusion detection system." [Online]. Available: https://www.snort.org/

[28] "RapidMiner Studio." [Online]. Available: http://rapidminer.com/