2010 International Conference on Broadband, Wireless Computing, Communication and Applications

Behavior Shaver: An Application Based Layer 3 VPN that Conceals Traffic Patterns Using SCTP

Mamoru Mimura
Graduate School of Information Security
Institute of Information Security
Yokohama, JAPAN
Email: dgs104101@iisec.ac.jp

Hidehiko Tanaka
Graduate School of Information Security
Institute of Information Security
Yokohama, JAPAN
Email: tanaka@iisec.ac.jp

***Abstract*—In recent years, distributed systems are connected by VPN (Virtual Private Network) through the Internet, and construct complicated information systems. These information systems bring benefit and security risks to many users. Representative security risks, vulnerabilities are closely related to application software installed in information systems. If a malicious adversary identifies the application software, he can seek the vulnerabilities easily. Thus, to ensure security of information systems, it is necessary to conceal application software installed in information systems. On the other hand, some attempts have been proposed to identify application software or protocol without scanning the payload. These proposed methods can analyze encrypted traffic, because the methods scan traffic patterns such as packet sizes and transmission intervals. While there are some legitimate uses for encrypted traffic analysis, these methods also raise problems about the confidentiality of encrypted traffic. Many researchers proposed countermeasures against traffic analysis to ensure anonymity in a public network. They indicated how to alter traffic patterns in the main. However, a few researcher indicated how to implement the method. Indeed, though previous VPN applications protect payloads against an eavesdropper, do not conceal side channel information including traffic patterns. Our work applies these proposed countermeasures and shows how to implement a secure VPN application that conceals traffic patterns. To alter traffic patterns, it is necessary to control packet sizes. Many popular application based VPN encapsulates packets by TCP or UDP. However, TCP cannot control packet sizes strictly. Though UDP can control packet sizes without difficulty, does not ensure reliable data transmission. A secure application based VPN requires a protocol that can control packet sizes strictly and can ensure reliable data transmission in untrusted networks. SCTP (Stream Control Transmission Protocol) is a suitable solution for these requirements. This paper proposes the behavior shaver, an application based layer 3 VPN that conceals traffic patterns using SCTP. The results of experiments show the performance.**

I. Introduction

In recent years, many information systems are constructed by organizations. Some distributed systems are connected by VPN (Virtual Private Network) through the Internet, and construct complicated information systems. VPN applications play an important role in distributed systems through untrusted networks like the Internet. In general, security of VPN depends on strength of the cryptographic communication. These information systems join the Internet to users and bring benefit to many users. At the same time, the benefit often bring security risks to many users. To cope with both benefit and security is a great problem to be solved. Many information systems have many security risks such as vulnerability. Today, vulnerabilities are reported from day to day and anyone can find security information easily from the Internet. A zero-day attack, a computer threat that tries to exploit computer application vulnerabilities that are unknown to others, undisclosed to the software vendor, or for which no security fix is available, has lost much of its novelty now. Vulnerabilities are closely related to OS (Operating System) or application software installed in information systems. If a malicious adversary identifies the OS or the application software of an information system, he can seek the vulnerabilities easily. Thus, to ensure security of information systems, it is necessary to conceal OS or application software installed in information systems.

On the other hand, some attempts have been proposed to identify application software or protocol without scanning the payload. These proposed methods analyze traffic patterns such as packet sizes and transmission intervals. If a malicious adversary can analyze encrypted traffic and reveal the application software by these techniques, they can become threats to ensure security. Though previous VPN applications e.g. OpenSSH[1], OpenVPN[2] protect payloads against an eavesdropper, do not conceal side channel information including traffic patterns. Threfore, previous VPN applications cannot ensure security of information systems.

This paper makes threats of traffic analysis in untrusted networks clear and consider constructing secure communication route to ensure security of distributed systems. Our goal is developping a secure VPN application that conceals traffic patterns.

II. Scope

Traffic analysis methods can be classified into payload analysis, header analysis and behavior analysis. A countermeasure against payload analysis is cryptographic communication. The second method is mainly called OS fingerprinting[3], and a protocol scrubber[4] is one of many countermeasures against this method. However, there is a few countermeasure against behavior analysis that infers application software or protocol. Our work belongs to this category.

Our purpose is not anonymity in a public network but confidentiality in a private network. Many researchers proposed

978-0-7695-4236-2/10 $26.00 © 2010 IEEE
DOI 10.1109/BWCCA.2010.152

countermeasures against traffic analysis to ensure anonymity in a public network. They indicated how to alter traffic patterns in the main. However, a few researcher indicated how to implement the method. Our work applies these proposed methods to conceal traffic patterns and enhance confidentiality. This paper shows how to implement a secure application based layer 3 VPN that conceals traffic patterns, and does not study how to alter traffic patterns thoroughly.

There are some cases where implementations of cryptographic communication reveal vulnerabilities to an eavesdropper. One of the causes is ambiguity and the other is a particular traffic pattern generated by packet sizes and their inter-arrival times. We will use the term ”behavior” to refer to traffic patterns that packet sizes and their inter-arrival times generate.

III. Threat

This section reviews behavior analysis methods and makes threats of traffic analysis in untrusted networks clear.

Recently, several methods that analyze behavior such as packet sizes and transmission intervals were developed. These methods need only packet sizes and their inter-arrival times, and thus, there is no need to scan payloads. Thus, these methods can analyze encrypted traffic[5]. Moore and Zuev[6], [7] provided a robust traffic classification scheme based on Bayesian analysis techniques, which requires access only to packet header data. Shizuno[8] and Kitamura[9] proposed the application identification methods based on analyzing traffic flow behavior such as packet sizes and transmission intervals. Still more, some researchers use transition patterns of packet sizes to classify network applications[10], [11], [12]. Bernaille[13], [14] proposed a technique that uses only the size of the first few packets to identify the application. Recent techniques for the network traffic analysis include machine learning to classify traffic. Early[15], McGregor[16] and Moore[17] used machine learning to classify traffic by an application of the network traffic analysis. Wright[18], [19] developed their own traffic classification system with a hidden Markov model. Kohara[20], [21], [22] and Sena[23] attempted to classify traffic by using Support Vector. These methods analyze traffic patterns generated by the implementation of application software or protocol. The implementation of application software or protocol generates packets in its own sizes and their inter-arrival times. In particular, a pattern of packet sizes is large, and many early application identification methods use only packet sizes.

These methods are novel and worthwhile. However, a malicious eavesdropper can also analyze encrypted traffic by these methods to identify the application software. As previously stated, if a malicious adversary identifies the the application software on an information system, he can seek vulnerabilities easily. Especially, vulnerabilities of each application software are often reported. Such a few vulnerability of each application software has a great potential for vulnerabilities of the whole information system. Then, he can attempt an intrusion on the information system abusing the vulnerabilities. After intruding the information system, he may access confidential information or personal information. Therefore, to ensure security of information systems, it is necessary to conceal application software installed in information systems. Moreover, to ensure security of distributed systems connected by VPN, it is necessary to develop a secure VPN application that conceals traffic patterns.

IV. Related works

This section briefly reviews previous works about a secure VPN application that conceals traffic patterns and brings up problems to be solved.

A malicious eavesdropper can analyze encrypted traffic and identify the application software, because the implementation of application software generates a particular traffic pattern. To disturb identifying the application, it should conceal a particular traffic pattern. In particular, a pattern of packet sizes should be concealed.

There are many countermeasures for the purpose of anonymity. The concept of data anonimization through cryptography and forwarding was originally introduced by Chaum[24]. He proposed the use of a Mix, a computer proxy. The Mix collects a number of packets called batch, reorders packets and transmits fixed size packets. However, the reality is that a Mix cannot always get sufficient packets efficiently from users. Hence, it is suggested that users send dummy messages of random and meaningless data. Many anonymous systems use fixed length messages by padding in order to conceal message size. Some anonymous systems use dummy messages (cover traffic) to conceal the correlation between source and destination. A common tactic for mitigating such threats is to pad packets to fixed sizes or to send packets at fixed intervals. Though these methods were proposed for anonymity, some methods can be applied to confidentiality. They indicated how to alter traffic patterns in the main.

However, a few researcher indicated how to implement the method. Csaba[25] adopted Traffic Flow Confidentiality mechanisms, and implemented the functions (padding, fragmentation, dummy packet generation and artificial alteration of the packet delay) in IPsec[26]. The IPsec supports fragmentation by reusing the IPv6 fragmentation header. There is some possibility that a malicious eavesdropper reassembles the fragmented packets. IPsec is a network based kernel space VPN, and does not ensure reliable data transmission by itself. Hense, a VPN application that ensures reliable data transmission is also necessary. However, suchlike popular application based VPN does not even conceal a pattern of packet sizes.

Accordingly, a tunnel that conceals traffic patterns was proposed[27]. Their tactic that padding packets to fixed sizes and sending packets at fixed intervals is not novel. However, they brought up some problems to implement an application based user space VPN that conceals traffic patterns. Their tunnel divides arriving packets with a fixed size and sends the fragmented packets at fixed transmission intervals. The fragmented packets are encapsulated by TCP or UDP in the same popular application based VPN. However, in the case of stream-oriented protocols such as TCP, it is difficult to control the packet sizes strictly. Datagram-oriented protocols such as UDP can control the packet sizes without difficulty. UDP does not ensure reliable data transmission in untrusted networks. Thus, the tunnel cannot ensure enough security for distributed systems connected by VPN. A secure application based VPN requires a protocol that can control packet sizes strictly and can ensure reliable data transmission in untrusted networks.

V. Behavior Shaver

In this section, we propose the behavior shaver that conceals traffic patterns using SCTP (Stream Control Transmission Protocol)[28].

A. Proposal

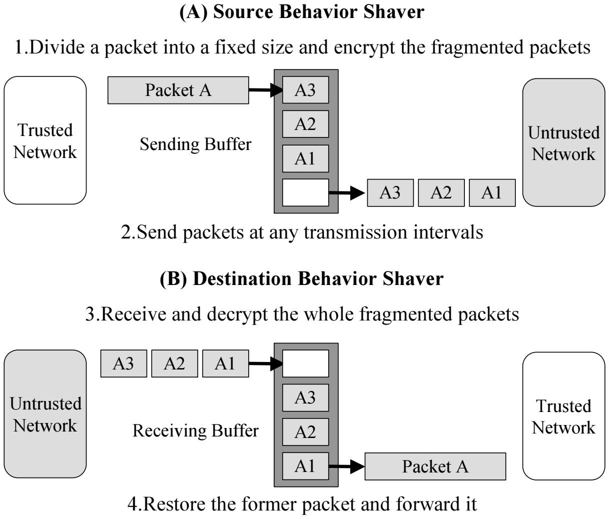
The proposed tunnel[27] does not encrypt packets and encapsulates packets by TCP or UDP in the same popular application based VPN. For that reason, their tunnel cannot control packet sizes strictly or cannot ensure reliable data transmission in untrusted networks. To solve these problems, we propose an application based layer 3 VPN that encapsulates packets by SCTP. The SCTP is a transport layer protocol, serving in a similar role as the popular protocols TCP or UDP. It provides some of the same service features of TCP, ensuring reliable, in-sequence transport of messages with congestion control. Moreover, the SCTP may be characterized as record-oriented, meaning it transports data in terms of messages, in a similar fashion to the UDP. Therefore, the SCTP can control packet sizes strictly and can ensure reliable data transmission.

B. Design

Figure 1 shows the design of the behavior shaver. The source behavior shaver divides an arriving packet with a fixed size and adds to a sending buffer. If the arriving packet sizes is less than the fixed size, a gap is padded with random numbers. The fragmented packets are encrypted and encapsulated by SCTP. It then sends the encapsulated packets at any transmission intervals. The destination behavior shaver receives the encapsulated packets and adds to a receiving buffer. The encapsulated packets are decapsulated and decrypted. After reception of the whole fragmented packets, the destination behavior shaver restores the former packet and forwards to the destination.

C. Quantization of Transmission Intervals

Converting packet sizes to a fixed size, variation of traffic patterns diminish dramatically. Because an eavesdropper can get only packets in the same size. However, transmission intervals may generate a particular pattern. The proposed tunnel[27] sends packets at fixed intervals. This method is inefficient and cannot ensure enough throughput. Then, we adopt variable transmission intervals, and quantize transmission intervals to 3 kinds of ranges. Table I shows the empirically derived ranges. Transmission interval ranges are inversely proportional to the corresponding buffer space rate as in Table I. We set the transmission interval by a random generator within the range. Therefore, original transmission intervals are quantized smoothly.



(A) Source Behavior Shaver
1.Divide a packet into a fixed size and encrypt the fragmented packets
Trusted Network
Packet A
Sending Buffer
A3
A2
A1
A3
A2
A1
Untrusted Network
2.Send packets at any transmission intervals
(B) Destination Behavior Shaver
3.Receive and decrypt the whole fragmented packets
Untrusted Network
A3
A2
A1
Receiving Buffer
A3
A2
A1
Packet A
Trusted Network
4.Restore the former packet and forward it

Fig. 1. A design of the behavior shaver

TABLE I
Transmission intervals

Buffer Space Rate	Transmission Interval Range
Less than 50%	0
Less than 90%	0-1ms
else	1-10ms

D. Implementation

We implemented the behavior shaver with C programming language on a Fedora 10[29] system. Table II shows the development environment and specifications. We adopted AES[30] algorithm to encrypt payloads, and the Mersenne Twister[31] with a hush function, a random number generator. Before starting, we can select parameters, packet sizes and tunneling protocols. In the case of SCTP, the number of streams is always one. We adopted the lksctp-1.0.9[38], an implementation of the Stream Control Transmission Protocol in the Linux kernel. The behavior shaver has 2 network interfaces in a trusted network and an untrusted network. One interface in a trusted network receives or sends plain packets. Another interface in an untrusted network receives or sends encrypted packets. The behavior shaver has 2 buffers, namely a sending buffer and a receiving buffer. The sending buffer stores encrypted packets to send into an untrusted network. The receiving buffer stores decrypted packets to send into a trusted network.

VI. Experiment

In this section, we discuss the performance of the behavior shaver. We have conducted experiments and the result shows performance of the behavior shaver.

TABLE II
DEVELOPMENT ENVIRONMENT AND SPECIFICATIONS

OS	Linux-2.6 (Fedora 10)
Programming Language	C (gcc-4.3.2)
Encryption Algorithm	AES
Random Number Generator	Mersenne Twister
Packet Sizes	128/256/512/1024 bytes
Transmission Intervals	0-10 ms
Tunneling Layer	L3 (Network Layer)
Tunneling Protocol	UDP/SCTP

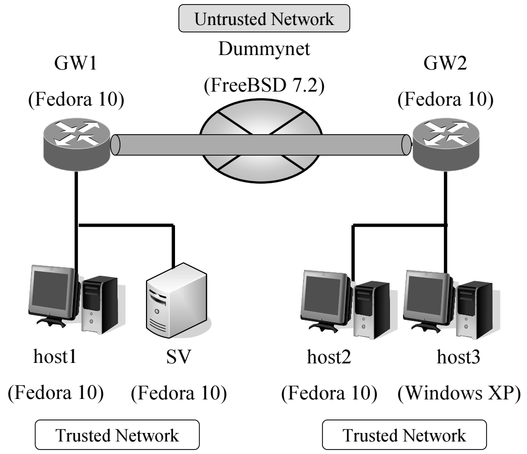


Fig. 2. An experimental network

A. Environment

We first describe the experimental environment. Figure 2 shows the network topology. Any hosts are connected with 100BASE/T Ethernet. The OS is displayed in the figure. GW1 and GW2 are routers where the behavior shaver is installed in. A network between GW1 and GW2 is a untrusted network that is eavesdropped by a malicious adversary. The behavior shavers construct a secure VPN tunnel between GW1 and GW2. The untrusted network is simulated by dummynet[33]. Dummynet simulates queue and bandwidth limitations, delays, packet losses and so on. The main platform for dummynet is FreeBSD. Other networks are trusted networks that do not need encryption. SV provides E-mail, DNS, World Wide Web or NTP services. Other hosts are clients that use these services.

B. Experiment

1) Function: We confirm that any hosts can use E-mail, DNS, World Wide Web or NTP services provided by SV through the secure VPN tunnel. Then, we also analyze traffic patterns and confirm that all packets are uniform size.

2) RTT: We estimate RTT (Round Trip Time) between host2 and SV for every packet sizes. In order to measure RTT, host2 sends ICMP echo requests 100 times and calculate the mean. The delays and packet losses in the untrusted network are altered by dummynet.

TABLE III
RESULTS OF A FUNCTIONAL TEST

Protocol	Result
ICMP	OK
SMTP, POP, IMAP	OK
DNS	OK
HTTP, HTTPS	OK
NTP	OK

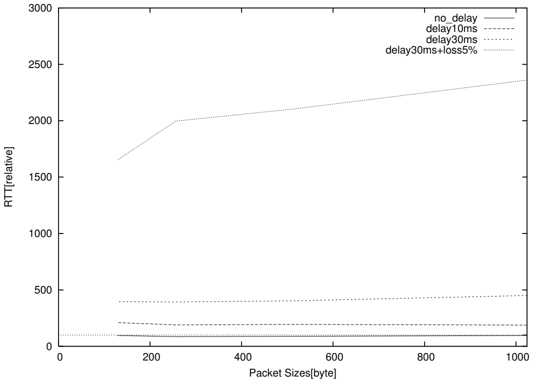


Fig. 3. RTT of the behavior shaver

3) Throughput: We estimate throughput between host2 and SV for every packet sizes. In order to measure throughput, we use Iperf-2.0.4[34]. Iperf was developed by NLANR/DAST (The Distributed Applications Support Team) as a modern alternative for measuring maximum TCP and UDP bandwidth performance. The delays and packet losses in the untrusted network are altered by dummynet.

C. Result

1) Function: Table III shows the results of a functional test. All of protocols or applications in any OS worked normally. We analyzed traffic patterns and confirmed that all packets were uniform size.

2) RTT: Figure 3 shows the RTT for every packet sizes. Let the performance of the previous tunnel[27] equal 100. The previous tunnel divides packets and encapsulates the fragmented packets by UDP. The parameter of packet sizes is set 1024 bytes with no delay. The results are based on the performance. The x-coordinate corresponds to the packet size, and the y-coordinate corresponds to the relative value of RTT. In the case of no loss, the RTTs are almost regular values correlating without packet sizes. There is not much to choose between the RTT at no delay and the base value. When packets are lost, the RTTs are regularly long regardless of packet sizes. This is because packet loss caused retransmission. Therefore, the implementation using SCTP does not cause excessive overhead.

3) Throughput: Figure 4 shows the throughput for every packet sizes. Let the performance of the previous tunnel[27] equal 100. The results based on the performance. The x-coordinate corresponds to the packet size, and the y-coordinate

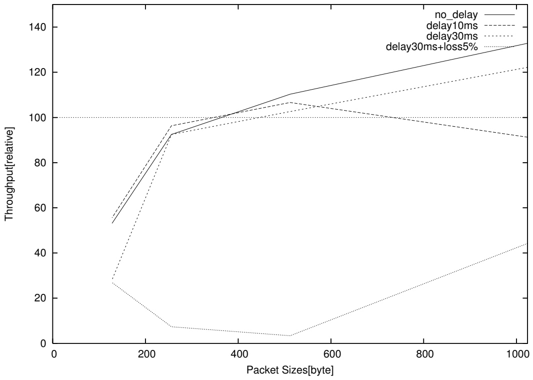


Fig. 4. Throughput of the behavior shaver

corresponds to the relative value of throughput. In the case of no loss, the throughputs are almost regular values correlating without packet sizes over 256 bytes. There is a rapid drop at 128 bytes. The cause can be our implementation decreases the goodput. When packets are lost, the throughput are regularly low regardless of packet sizes. This is because packet loss caused retransmission. Thus, the cause might not be our implementation decreases the goodput. Therefore, the implementation using SCTP does not cause excessive overhead under the condition that the parameter of packet sizes is set over 256 bytes.

D. Discussion

Previous VPN applications ensure confidentiality by encryption. However, these VPN applications forward packets without altering behavior. Hence, these VPN applications may reveal the application software that generates a particular traffic pattern on the VPN tunnel. Thus, these VPN applications are vulnerable to traffic analysis. Moreover, some implementations of cryptographic communication are vulnerable to timing attacks[35].

The behavior shaver transmits only packets in the same size. Thus, the behavior shaver entirely defeats simple behavior analysis methods[13], [14] using only packet sizes. Because their method uses only the size of the first few packets to identify the application. The methods using transition patterns of packet sizes[10], [11], [12] are also defeated entirely. Moreover, the transmission intervals were altered by a random number generator in order to defeat timing attacks. Thus, the behavior shaver is more confidential than previous VPN applications.

IPsec with Traffic Flow Confidentiality mechanisms[25] is also more confidential than previous VPN applications. The IPsec is a network based kernel space VPN. The behavior shaver is an application based user space VPN, and plays another role. If you want to ensure reliable data transmission of unreliable protocols or applications, the behavior shaver will enable that.

However, there is some doubt to defeat more sophisticated behavior analysis methods using machine learning or statistical analysis for a long term. Particularly, to defeat statistical analysis for a long term entirely, low-latency VPN applications probably has to pay compensation. Because most traffic over a general VPN tunnel is interactive and does not permit much delay.

VII. Conclusion

In this paper, we made the threat that behavior analysis revealed vulnerabilities of information systems clear, and indicated that previous VPN applications could not ensure security of information systems. We focused on implementation of application based layer 3 VPN and proposed the behavior shaver that concealed traffic patterns using SCTP. Our works proposed not how to alter traffic patterns but how to implement an application based user space VPN that controls packet sizes strictly. Moreover, we implemented the behavior shaver and the results of experiments showed the performance.

The confidentiality of the cryptographic communication mostly depends on crypto, and the behavior shaver enhanced the confidentiality. If an implementation of the cryptographic communication has fatal vulnerabilities, the behavior shaver may conceal the vulnerabilities and ensure confidentiality. The behavior shaver prevents behavior analysis attacks and disables identifying the application or the protocol. Thus, it is difficult for a malicious adversary to seek the vulnerabilities of information systems. The behavior shaver forces a malicious adversary to cost a long time. In the meantime, the administrator of the information system can detect the sign or take countermeasures. Security Requirements for NGN (Next Generation Network)[36] provide three kinds of security zones, namely Trusted, Trusted but Vulnerable and Untrusted. Our work prevents information leaks from trusted zones to untrusted zones or trusted but vulnerable zones, and ensures security of trusted zones.

However, the evaluation of security is not enough. We still cannot declare that the behavior shaver entirely defeat traffic analysis. To follow up this matter further would involve us in a discussion of implementation and would take us beyond the scope of this work. However, behavior analysis methods that identify the application or the protocol without using packet sizes are not practical. We also need to conduct experiments in other various environments. Our implementation is based on a tactic that padding packets to fixed sizes and sending packets at random intervals. However, this approach is less efficient than Wright's method[37]. How to implement the new methods is a future work. How to realize efficiently is considered as a future research which requires considerable effort.

References

[1] " OpenSSH, " http://www.openssh.org/

[2] " OpenVPN, " http://openvpn.sourceforge.net/

[3] Fyodor Dostoevsky,
" Remote OS Detection via TCP/IP Fingerprinting(2nd Generation), " http://insecure.org/nmap/osdetect/

[4] David Watson, Matthew Smart, G. Robert Malan and Farnam Jahanian, "Protocol Scrubbing: Network Security Through Transparent Flow Modification, " IEEE/ACM Transactions on Networking, Vol.12, No.2, pp.261–273 (2004).

[5] Matthew Gebski, Alex Penev and Raymond K. Wong, ” Protocol Identification of Encrypted Network Traffic, ” Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, pp.957–960 (2006).

[6] Andrew W. Moore and Denis Zuev, ” Internet Traffic Classification Using Bayesian Analysis Techniques, ” Proceedings of the ACM SIGMETRICS 2005, (2005).

[7] Denis Zuev and Andrew W. Moore, ” Traffic Classification Using a Statistical Approach, ” The 6th Anuual Passive and Active Measurement Workshop (PAM 2005), (2005).

[8] Takayuki Shizuno, Tsutomu Kitamura and Toshiya Okabe, ” An Application Identification Method based on Flow Behavior Analysis for an Aggregation of Flows, ” Technical Report of the Institute of Electronics, Information and Communication Engineers, NS, Vol.105, No.627, pp.9–12 (2005).

[9] Tsutomu Kitamura, Takayuki Shizuno and Toshiya Okabe, ” Application Classification Method based on Flow Behavior Analysis, ” Technical Report of the Institute of Electronics, Information and Communication Engineers, NS, Vol.105, No.470, pp.13–16 (2005).

[10] Tsutomu Kitamura, Takayuki Shizuno and Toshiya Okabe, ” Traffic Identification Method based on Packet-type Transitions, ” Technical Report of the Institute of Electronics, Information and Communication Engineers, NS, Vol.106, No.41, pp.25–28 (2006).

[11] Shinnosuke Yagi, Yuji Waizumi, Hiroshi Tsunoda and Yoshiak Nemoto, ” An Evaluation of Transition Pattern of Payload Legnth for Network Application Identification, ” Technical Report of the Institute of Electronics, Information and Communication Engineers, TM, Vol.107, No.313, pp.1–6 (2007).

[12] Shinnosuke Yagi, Yuji Waizumi, Hiroshi Tsunoda and Yoshiak Nemoto, ” Classifying Network Application using Transition Pattern of Payload-length, ” The Special Interest Group Technical Reports of Information Processing Society of Japan, DSM, Vol.38, pp.83–88 (2007).

[13] Laurent Bernaille, Renata Teixeira, Ismael Akodkenou, Augustin Soule and Kave Salamatian, ” Traffic Classification on the Fly, ” ACM SIGCOMM Computer Communication Review, Vol.36, Issue 2, pp.23–26 (2006).

[14] Laurent Bernaille, Renata Teixeira and Kave Salamatian, ” Early Application Identification, ” Proceedings of the 2006 ACM CoNEXT Conference, No.6 (2006).

[15] James P. Early, Carla E. Brodley and Catherine Rosenberg, ” Behavioral Authentication of Server Flows, ” Proceedings of the 19th Annual Computer Security Applications Conference, pp.46–55 (2003).

[16] Anthony McGregor, Mark Hall, Perry Lorier and James Brunskill, ” Flow Clustering Using Machine Learning Techniques, ” In the 5th Anuual Passive and Active Measurement Workshop (PAM 2004), (2004).

[17] Andrew W. Moore and Konstantina Papagiannaki, ” Toward the Accurate Identification of Network Applications, ” Proceeding of the Passive and Active Measurement Workshop, pp.41–54 (2005).

[18] Charles V. Wright, Fabian Monrose and Gerald M. Masson, ” HMM Profiles for Network Traffic Classification, ” Proceedings of the CCS Workshop on Visualization and Data Mining for Computer Security, pp.9–15 (2004).

[19] Charles V.Wright, Fabian Monrose and Gerald M.Masson, ” On Inferring Application Protocol Behaviors in Encrypted Network Traffic, ” The Journal of Machine Learning Research, Vol.7, pp.2745–2769 (2006).

[20] Masayoshi Kohara, Yoshiaki Hori and Kouichi Sakurai, ” Discussion about Flow Traffic Classification using Support Vector, ” Symposium on Cryptography and Information Security (SCIS 2006), 2E2-5, (2006).

[21] Masayoshi Kohara, Yoshiaki Hori and Kouichi Sakurai, ” Proposal of Traffic Classification Method using Packet Length, ” Symposium on Cryptography and Information Security (SCIS 2007), 1F2-2, (2007).

[22] Masayoshi Kohara, Yoshiaki Hori, Kouichi Sakurai, Heejo Lee and Jae-Cheol Ryou ” Flow Traffic Classification with Support Vector Machine by using Payload Length, ” In the 1st International Workshop on Network Traffic Control, Analysis and Applications (NTCAA 2009), (2009).

[23] Gabriel Gómez Sena and Pablo Belzarena, ” Early Traffic Classification using Support Vector Machines, ” Proceedings of the 5th International Latin American Networking Conference, pp.60–66, (2009).

[24] David L. Chaum, “ Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms, ” Communications of the ACM, Vol.24, No.2, pp.84–88 (1981).

[25] Csaba Kiraly, Simone Teofili, Giuseppe Bianchi, Renato Lo Cigno, Matteo Nardelli and Emanuele Delzeri, ” Traffic Flow Confidentiality in IPsec: Protocol and Implementation, ” The Future of Identity in the Information Society, Vol.262, pp.311–324 (2008).

[26] Stephen Kent and Karen Seo, ” Security Architecture for IP, ” RFC 4301, (2005).

[27] Mamoru Mimura and Yasuhiro Nakamura, ” A Tunnel that Conceals Packet ’ s Behavior Against Traffic Analysis Attack ” Memoirs of the National Defense Academy, Vol.49, No.2, pp.1-19 (2010).

[28] Randall Stewart, ” Stream Control Transmission Protocol, ” RFC 4960, (2007).

[29] ” Fedora 10, ” http://fedoraproject.org/

[30] Niels Ferguson, Richard Schroeppel and Doug Whiting, ” A Simple Algebraic Representation of Rijndael, ” Revised Papers from the 8th Annual International Workshop on Selected Areas in Cryptography, pp.103–111 (2001).

[31] Makoto Matsumoto and Takuji Nishimura, ” Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator, ” ACM Transaction on Modeling and Computer Simulation, Vol.8, No.1, pp.3–30 (1998).

[32] ” FreeBSD 7.2, ” http://www.freebsd.org/

[33] ” dummynet, ” http://info.iet.unipi.it/l̃uigi/ip_dummynet/

[34] ” Iperf, ” http://dast.nlanr.net/Projects/Iperf/

[35] David Brumley and Dan Boneh, ” Remote Timing Attacks are Practical, ” Computer Networks: The International Journal of Computer and Telecommunications Networking, Vol.48, Issue 5, pp.701–716 (2005).

[36] International Telecommunication Union Telecomminication Standardization Sector, “ Security Requirements for NGN release 1, ” ITU-T Recommendation Y.2701 (2007).

[37] Charles V.Wright, Scott Coull and Fabian Monrose, ” Traffic Morphing: An Efficient Defense Against Statistical Traffic Analysis, ” Proceedings of the Network and Distributed System Security Symposium (NDSS 2009), pp.237–250 (2009).

[38] ” The Linux Kernel Stream Control Transmission Protocol, ” http://lksctp.sourceforge.net/