Intrusion Detection Evasion Techniques and Case Studies

GIAC (GCIA) Gold Certification

Author: Pierce Gibbs, pierce.m.gibbs@gmail.com
Advisor: Richard Carbone

Accepted: January 17th 2017

Abstract

The number of security breaches is increasing significantly each year. Global Internet traffic is expected to be on the order of zettabytes for 2016 and then doubling by 2020. In addition to increased traffic, the percentage of attack traffic is also increasing.

The sophistication of attacks is also increasing. Attacks range in complexity from simple protocol, insertion, or desynchronization attacks that exploit the vagueness and incompleteness of the RFCs to polymorphic blending attacks that camouflage attack and exfiltration traffic to match normal traffic for that particular network.

Various evasion techniques have been described in articles within this field of study, but there has not been a collective discussion on the variety of evasion techniques. A comprehensive compilation of the most common evasion techniques is needed to aid Intrusion Detection System providers and to assist various decision makers as they determine how best to apply limited resources to protect assets.

This paper is a case study analysis designed to detail the most common intrusion evasion techniques that exist in the wild today.

1. Introduction

There has been a rapid expansion in Internet traffic and that growth is forecast to increase at an even faster pace going forward. The Cisco Visual Networking Index reports that the zettabyte (10^{21}) threshold will be surpassed in 2016 and then doubled by 2019. The percentage of attack traffic will also increase (Karimi, Niyaz, Weiqing Sun, Javaid, & Devabhaktuni, 2016, p. 0522). In addition to the growth in traffic, there will be a corresponding growth in devices connected to the Internet. Three million devices (i.e. appliances, automobiles, smart devices…) are expected to be connected to the Internet by 2022 (Gendreau & Moorman, 2016, p. 84).

Defense in depth is a widely-regarded security principle. Firewalls, anti-virus systems and honeynets are all different and important security components. However, firewalls require holes in them to allow traffic through which means they can fail to detect or prevent many types of attacks (Mallissery, Prabhu, & Ganiga, 2011, p. 224). Signature-based anti-virus systems are only effective against malicious code for which a signature has been written. Honeynets can lure attackers and can occupy an attacker's time but do not protect assets against attacks. Attackers often invade networks undetected. Intrusion Detection Systems (IDS) are a complementary component in a defense in depth strategy (Mavroeidakos, Michalas, & Vergados, 2016).

Intrusion detection is concerned with monitoring hosts or networks for indicators of violations or potential violations of computer or network security policy (Scarfone, K. A, 2007, p. ES-1). There are two general types of intrusion detection systems – host-based intrusion detection systems (HIDS) and network intrusion detection systems (NIDS). Host-based IDS are generally concerned with endpoint security, privilege escalation, changes to the host configuration, application behavior, and traffic into and out of the host. Anti-virus protection, file integrity monitors, host-based firewalls and kernel call monitoring are aspects of HIDS. NIDS monitor network traffic in search of malicious or abnormal traffic that may violate the organization's security policy (Jaiganesh & Mangayarkarasi, 2013, p. 1629). The focus of this paper is network intrusion detection and evasion.

2. Background Theory

2.1. Detection Theory

There are two general detection approaches employed in IDS systems – signature-based and anomaly. Signature-based detection is a rules based approach where predefined patterns of known traffic or behavior are searched for. This approach is simple and has a low rate of false positives. It is not well suited to detect zero day attacks or other attacks for which the system has no signature.

Anomaly-based detection is a behavioral approach where the IDS must learn what normal behavior looks like and then alerts when abnormal behavior is detected. Anomaly-based detection can be effective against zero-day attacks but generally has a high rate of false positives.

A primary role of the NIDS is to predict the internal state of the hosts it protects based on collected network traffic. If the NIDS determines that the end host will transition to an insecure or compromised state, then the NIDS will alert. However, it is difficult for a NIDS to accurately make such predictions. One reason state prediction is so difficult is the sheer size of the state vector for an end host. Hardware, Operating system (OS), applications and services are part of a host's state vector. Hardware details might include which BIOS and CPU a motherboard contains. OS details might include its running state, version and patch level, the status of all ports, the set of applications running, and the set of services exposed. TCP details can include connection-oriented details such as port number and IP address of both hosts in the connection, sequence numbers and window sizes.

Another factor that makes intrusion detection difficult is that the traffic that an IDS and the various hosts receive can be different, particularly if there are intervening devices between the IDS and end host such as routers and switches. Additionally, how an end host processes a packet is based on its OS (Ptacek & Newsham, 1998, p. 6).

The role of the anomaly-based NIDS is to detect when traffic varies sufficiently from normal traffic patterns to warrant an alert. One key to the performance of a NIDS is its ability to learn what normal traffic is for the portion of the network it protects. There are two general categories of anomaly-based intrusion detection approaches – detection using counters and detection using feature distributions.

Counters provide simple statistics such as the minimum, maximum and average counts or rates for bytes or packets. This could include discarded, forwarded or fragmented packet counts or bytes. Detection through the use of feature distributions often includes more complex statistical methods such as correlations. Statistical methods are applied to a variety of traffic metadata such as header information or flow information (Bereziński, Jasiul, & Szpyrka, 2015, p. 2369 - 2370).

2.2. Evasion Theory

The primary function of the IDS is to predict the state of the hosts it protects. If the IDS can accurately predict a host's state, it can accurately predict how a host will respond to an event such as the receipt of network traffic. IDS evasion techniques focus on attacking an IDS' ability to predict the change in state a host will make based on incoming traffic.

Ambiguity is a contributing factor that leads to an IDS' inability to predict future states of a machine. There are differences in how operating systems implement the various request for comments (RFC) documents. RFCs are vague and incomplete suggestions rather than strict, complete, unambiguous requirements. RFCs are even interpreted differently between different versions of the same OS (Shankar & Paxson, 2003).

Insertion is an evasion technique where the attacker provides network traffic that the IDS processes and the end host does not process. This can be the result of the end host never receiving the traffic or receiving the traffic and discarding it.

Evasion is a technique where an attacker provides network traffic that the IDS does not process but the end host does. By using fragmentation techniques and differences in reconstruction algorithms, an attacker can craft packets such that when the host and IDS reconstruct them, the packets are discarded by the IDS but accepted by the host. Another evasion technique is to overwhelm the IDS with traffic so that IDS cannot process it all.

Overlapping IP fragments and overlapping TCP sequences provide opportunities for IDS and end hosts to reconstruct messages differently. If the result of IDS reconstruction yields a benign message and the result of the end host reconstruction yields a malicious message, then the IDS has been evaded.

3. Techniques for Evading Intrusion Detection

3.1. Time To Live Manipulation

The Time-To-Live (TTL) field in the Internet Protocol (IP) header is designed to prevent IP packets from bouncing around the Internet indefinitely. The field represents a number of hops rather than a time of life. When a router receives an IP packet, the router will decrement the TTL value by one and if the resulting TTL value is greater than zero, the router forwards the packet, according to its routing table, to the next hop. That next hop could be another router or the final destination. If the receiving router decrements the packet's TTL to zero, then it is discarded.

The TTL attack is an attack against the synchronization of the IDS and the end host and is dependent upon a router being between the IDS sensor and the end host. Network reconnaissance can help an attacker determine hop counts to router and end host. A packet crafted with a TTL equal to the number of hops of the router will result in a packet being examined by the IDS but never reaching the end host, thus desynchronizing the end host and the IDS. This attack can be thwarted by a NIDS that examines the TTL field and understands the network topology. However, this requires additional processing resources. A NIDS usually protects more than one host and as the number of hosts increases, so do the demands on the processing resources of the NIDS.

3.2. IDS MAC Address Attack

Layer 2 switches on an Ethernet-based local area network (LAN) typically forward Ethernet frames based on the media access control (MAC) address found in the frame header. Modern switches typically forward packets only to the host specified by the MAC address in the frame header or another switch if it is using a cascading switch network architecture. Ethernet hosts, unless configured in promiscuous mode, only process frames addressed to them. Therefore, frames addressed to the IDS will only be seen by the IDS. For IP traffic, IP header and payload are encapsulated in the Ethernet frame body. Most IDS examine IP addresses while ignoring MAC addresses.

The MAC address attack is an insertion attack. The attacker crafts an Ethernet frame containing the MAC address of the IDS and the IP address of the end host the attacker wants to desynchronize from the IDS. The IDS will receive the frame but the end host will not resulting in desynchronization.

3.3. IP Fragmentation Attacks

The Ethernet protocol is the most pervasive data link protocol for LAN technology in use today. The maximum Ethernet frame size is 1518 bytes. For networks using IP over Ethernet, IP packets are encapsulated in the Ethernet frame. The IP header has a 16-bit total length field allowing for packet sizes to be as large as 65,535 bytes. To accommodate transmission of IP packets that are larger than the underlying frame size, the IP has a provision for packet fragmentation.

Three header fields are key for implementing the fragmentation scheme – flags, offset, and identification. The flags field indicates whether fragmentation is allowed and whether there are more fragments. The IP Identification field identifies to which packet a fragment belongs. All fragments of a particular packet will have the same IP Identification. The offset field indicates where in the reconstructed packet a particular fragment belongs.

3.3.1. Reassembly Attack

The Internet Protocol RFCs do not specify how to reassemble overlapping fragments. Hence, different operating and IDS systems handle reassembly differently

(Shankar & Paxson, 2003). Consider two fragments with the same IP Identification number. Assume Fragment 1 has 20 bytes of data and an offset of 20. Fragment 1's data should occupy bytes 20 – 39 in the reassembled packet. Assume Fragment 2 has 10 bytes of data and an offset of 25. Fragment 2's data should occupy bytes 25 – 34. Two different fragments have specified bytes 25 – 34. Some OSs will not allow previous defined data to be overwritten by subsequent fragments while others will. If the IDS does not employ the same reassembly strategy as the end host, desynchronization will occur.

3.3.2. Do Not Fragment Attack

The maximum transmission unit (MTU) specifies the largest amount of data that can traverse a link. When a router receives a packet to forward, it examines the MTU of the link the packet should be forwarded to. If the packet's size is larger than the MTU and the do not fragment flag is set, the packet is discarded.

When the IDS is positioned between two routers on networks with different MTUs, desynchronization is possible as the IDS cannot ascertain whether traffic is discarded or reaches the end host.

3.4. Encryption

The use of encryption by malware is increasing. Two hosts can communicate using IPSec or other forms of encrypted tunnels and the NIDS would have no means to inspect the traffic unless provided with decryption keys (Studer, McLain, & Lippmann, 2007, p. 1). Some malware use a key and bitwise XOR function to encrypt. Given adequate resources, in some cases a NIDS could decrypt the packets. Such resources generally are committed to decryption in NIDS. Advantages of XOR encryption include that is that it is computationally fast and easy and most common hardware architectures provide the function. Disadvantages are that malware encrypted with XOR can be detectable and breakable using byte frequency and other statistical analysis. IPSec-encrypted traffic can be detected using entropy measurements, although compressed data can have similar levels of entropy and lead to false positives (Thurner, Grun, Schmitt, & Baier, 2015, p. 27). Breaking the encryption is out of scope for the NIDS.

3.5. Polymorphic Blending Attack

Signature-based IDS look for attack traffic by searching for predefined patterns in the traffic. Polymorphic attacks are attacks in which every instance of the malware is different, but performs the same malicious function in an attempt to evade signature-based IDS that alert based on a fixed or calculable signature. Encryption, byte substitution and code obfuscation are all code transforming techniques employed to ensure the attack traffic not match the signature (Fogla, Sharif, Perdisci, Kolesnikov, & Lee, 2006, p. 1).

Behavioral-based intrusion detection systems develop statistical and heuristic profiles that define normal behavior during a learning period. Byte frequency distribution, byte sequence occurrence, packet lengths and character distribution are examples of the statistical characteristics included in the profile. Once the network is profiled, the intrusion system detection will alert when traffic patterns deviate beyond some configurable threshold (Bose, Bharathimurugan, & Kannan, 2007).

Polymorphic malware variants are often detectable by behavioral intrusion detection systems. Byte frequency distributions will often be the same for the various instances of an attack. The byte values may be different due to substitution or encryption but there will often be a common distribution curve across the instances (Yu, Zhou, Liu, Yang, & Luo, 2011, p. 640).

Polymorphic blending attacks are polymorphic attacks that are created to match the normal profile for that network. The malware that generates the attack malware monitors the network to develop a normal profile. Once the generating attack malware has developed a representative normal profile for the network, the generating attack malware creates polymorphic instances of attack software that match the representative normal profile. Blending techniques include padding of generated malware to match byte count and frequency statistics. The CLET polymorphic engine is a freely available tool that employs byte padding and ciphering with different length keys between the variants in an effort to blend and increase the difficulty of detection (Song, Locasto, Stavrou, Keromytis, & Stolfo, 2009, p. 184).

Blending normal and attack traffic increases the amount of false alarms (Varshovi, Rostamipour, & Sadeghiyan, 2014, p. 50). False positives waste resources, as the event has to be investigated, needlessly. False negatives are problematic in that attack traffic crosses the network unnoticed.

4. Case Studies

Three case studies were selected for review. The first case study is a survey of security controls, particularly intrusion detection systems, put in place by the Kuwaiti government on selected servers. The second case study reports on the adaptation of data mining techniques to the intrusion detection problem and an analysis of the application of those techniques to data taken from an Indonesian government website. The third case study reports on applying Shannon entropy theory to the intrusion detection problem and the detection results when this theory is applied to known and unknown data sets.

4.1. Case Study I – Kuwait Government IDS Survey

Many countries target Kuwait government websites in search of critical information about the country, largely because of Kuwait's prominence as an oil producer. A survey was designed to assess the use of IDS in the protection of government computer systems in Kuwait. 90 employees across 16 different governmental agencies were surveyed. The agencies surveyed covered a broad range from the Public Authority for Youth and Sports to the ministries of Justice, Electricity, and Justice to the Public Authority of Industry. The majority of the employees were system software personnel with 5 to 9 years of experience. About half of those surveyed had bachelor degrees in Information Technology. 73% of the respondents reported that there had been no attacks in their organizations, but 27% reported that there had been attacks. 86% reported using firewalls to detect and prevent attacks while 14% reported not using firewalls for that purpose. 69% reported using IDS for detection while 31% reported not using IDS. 55% reported using NIDS while 45% reported not using NIDS. 73% reported having connections to other organizations while 27% reported not having connections to other systems. 81% reported that 5 or more employees use a remote access service while 19%

reported that they did not use the remote access feature. Half of the respondents reported that their systems could not detect some attacks. 16% of the organizations do not do updates. 13% do not protect the IP range of the network. 31% of the organizations do not change default passwords for network users. 82% reported using wireless (Al-Enezi, Al-Shaikhli, Al-Kandari, & Al-Tayyar, 2014).

The case study concluded that there are many weaknesses in the security of the systems of the organizations surveyed. Increasing employee awareness, improving the effectiveness of how the organizations employs IDS, and performing routine security maintenance such as applying updates, expiring and enforcing strong passwords, and developing a stronger security policy were the recommendations of the case study authors.

4.2. Case Study II – Detection Via Rule-Based Sequential Pattern Analysis

Pramono and Suhardi developed a log file data mining approach to anomaly detection (Trio Pramono & Suhardi, 2014). While this is basically a host-based intrusion detection approach, the process can easily be applied to NIDS as well. The approach involved mining user access logs for statistically related sequential patterns that could be captured to represent user interaction patterns. The algorithm is called a Sequential Pattern Mining (SPM) algorithm. The sequences are symbolic representations of rules. A rule growth data-mining algorithm is applied to the data to generate a comprehensive set of normal and abnormal rules.

A foundational assumption is that the majority of traffic in a data set is normal traffic. Abnormal sequences will be statistical outliers from the normal traffic. Pattern mining and rule growth constitute the learning phase of the anomaly detection system. Subsequent traffic is compared to the mined and generated rules to determine whether traffic is normal or abnormal.

The intrusion detection approach has two phases – pre-processing and core processing. The pre-processing phase has two components – data cleaning and data transformation. Data cleaning is concerned with removal of unnecessary data from the log file. Reduction of the data improves time and space processing efficiency. Data transformation converts user behavior into sequences of symbols that represent user interactions with the website. The standard, symbolic data representation also improves time and space processing efficiency. The symbolic data is then passed to core processing.

In the first phase of pre-processing a vulnerability selection and log transformation was performed. The log file was parsed for entries that perform HTTP requests that cause a web application to invoke vulnerable components access to administrator pages. Vulnerable components were determined by consulting exploit-db.com for applicable exploits and vulnerabilities. The selection reduced a log of 10910 entries to 4237. Vulnerable components of interest allowed user account creation and authorizes administrator login. The parsing activity also reduced the number of IP addresses of interest to 5.

In the second phase of pre-processing, user behavior was symbolized. The raw log file is parsed into sequential data or user transactions represented by a regular language. Symbols are used to represent detailed information and reduce algorithmic and processing complexity. Data items represented by symbols include IP addresses, access dates, URL access pages, HTTP request types, HTTP error codes, referrer URLs, user agents.

In the core-processing phase, Rule-Growth Sequential Pattern Mining Analysis (SPMA) is used to analyze parsed data from the pre-processing phase. A fundamental assumption is that most data is normal and that outliers in a data set are malicious. Known malicious traffic can be input by itself to categorize malicious traffic. Two inputs to the SPMA algorithm are “support level” and “confidence level”. Support level is an indicator of the frequency a symbol appears in the data. Confidence level is an indicator of how often a sequence is determined to be true.

The SPMA and rule growth algorithms were applied to three publicly available, real life data sets. In each case, the algorithms generated rule patterns based on the selected data set.

Next, the algorithms were applied to data obtained from the Badan Pusat Statistik (BPS) Kabupaten Tangerang website. The BPS Tangerang website is an Indonesian governmental website. The log data was collected from February 28, 2014 through March 6, 2014. During that period, the website was defaced by an attacker who reportedly gained access using vulnerable web application components. The rule patterns were mined and grown using the open source data mining algorithms developed by Philippe Fournier-Viger. The software can be found at www.philippe-fournier-viger.com/spmf. The algorithms generated all possible rule patterns for the IP addresses identified as having access to the vulnerable components.

During analysis of the user behaviors, statistical methods applied to the results from the data mining algorithms showed that the behavior of one IP address fell into the anomaly category. The data mining algorithms successfully revealed all possible valid rules from a given data set.

This case study was a proof of concept study. There is a gray area between normal and abnormal that could lead to false results. Performance is also an issue. The case study greatly reduced the data set by eliminating traffic that did not cause the invocation of several vulnerable components. Preprocessing and core processing times greatly benefitted from the reduction in data. Lastly, this data mining approach requires tuning through the setting of two parameters – confidence and support. This case study did not address how to determine the proper setting of those parameters. Despite the aforementioned shortcomings, the results appear very promising.

4.3. Case Study III - Entropy-Based Intrusion Detection

Basicevic et al. developed an entropy-based intrusion detector as part of the open source Snort IDS. The entropy-based detector is an anomaly detector that performs statistical correlation on entropy values of basic communication parameters or features such as source and destination IPv4 addresses and TCP port numbers (Basicevic, Kostovic, Popovic, & Ocovaj, 2013, p. 125).

Entropy is a measure of randomness, uncertainty, disorder, or chaos. Entropy calculation can be used to succinctly represent feature distributions in a single value (Bereziński, Jasiul, & Szpyrka, 2015, p. 2371).

Entropy typically ranges from roughly 0 to ($\log_2$ n)-1, where n is the maximum value of the parameter being measure. For instance, the IPv4 address space has a size of 2^{32} so the entropy range for number of addresses is roughly 0 to 31. Similarly, the TCP ports are specified by 16-bit values and the port range size is 2^{16} and the maximum entropy is approximately 15. Entropy value of 0 indicates there is no randomness and all measured values are the same. Entropy increases as the impurity of the class of traffic increases. In general, the class of traffic that consists of normal, benign traffic will have lower entropy than the class of traffic that includes both normal and attack traffic. (S.SyedNavaz, Sangeetha, & Prabhadevi, 2013, p. 45)

Two types of entropy measurements were computed in this case study – the entropy based on the number of used ports and entropy based on the number of packets. The number of packets measurement is composed of two calculations: the first calculation relative to number of packets reflects the change in packets entropy over two 1-second intervals, while the second entropy calculation relative to number of packets is a number of packets entropy over ten 0.1-second intervals.

Three modules were developed to extend the Snort intrusion detection system – trace files processing, entropy calculation and live analysis. The trace files processing module reads and parses packet capture (Pcap) traffic files into input suitable for entropy calculation. The live analysis module parses output from the Snort packet capture engine into input suitable for entropy calculation. The entropy calculation module calculates the entropy values based on the parsed data it receives as input.

Two publicly available sets of captured data were provided for the case study. The first data set was made by MIT Lincoln Laboratory during a DARPA Intrusion Evaluation Event and contains three known attack events. One of the events is a SYN flood denial of service (DoS) attack. The second data set was the result of monitoring 20 university campus computers at the University of Brescia, Italy and there was knowledge of attack events beforehand.

Analysis of the first data set shows an abrupt increase in the entropy of source ports during the time of the SYN flood DoS attack which is consistent with that type of attack. Multiple source ports are used to begin the three-way handshake without completing the handshake or establish a TCP connection. The destination host awaits the completion of the three way handshake which consumes resources. The goal of the SYN flood DoS attack is to exhaust the destination host's critical resources, primarily CPU and memory.

Analysis of the second set of data showed some abrupt entropy changes but it was inconclusive as to whether the entropy changes were due to attack traffic or normal changes in network activity.

The case study concludes that analysis of simple packet distributions can be effective in intrusion detection but by itself is not sufficient. Changes in entropy can be caused by normal network traffic. Additional techniques are required to increase detection accuracy and reduce false positives. While some attacks can be sufficiently detected solely using entropy-based methods, it is imperative to take into account the statistical characteristics of the network traffic. (Basicevic, Kostovic, Popovic, & Ocovaj, 2013, p. 128)

5. Conclusion

Global internet traffic is increasing significantly. As the volume increases, so does the importance of the data and the machines that are on the internet. Beyond the obvious e-commerce and social networking, we see growing internet reliance by the infrastructures of even third world nations. Communication systems, power grids, hospitals and patient monitoring are other examples.

Almost a decade ago, we first saw cyber-attacks precede physical invasions such as the 2008 Georgian-Russian war. The US intelligence community claims foreign entities tried to influence the 2016 presidential election. Cyber has joined land, sea, air and space as a military domain of US military doctrine.

The sophistication of the attacks are growing as well. Malware now morphs into signature defeating variants and communicates with other nodes via encrypted or side channels. Malware also blends into normal network traffic to go unnoticed.

The internet was not built with security in mind. Ethernet, IP, TCP, ICMP, ARP and HTTP are all insecure protocols. The RFCs developed for the internet community are suggestions and best practices rather than unambiguous and verifiable requirements. Vendors are free to come to their own interpretations of the RFCs. In fact, some operating systems even implement RFCs differently between versions.

Intrusion detection is just one layer of a good defense-in-depth security solution and will never be a sufficient protection on its own. Signature based IDS are relatively easy to defeat and anomaly detection devices can have high rates of false positives that consume analysts time while trying to determine if an alert is valid or not. Despite the limitations, intrusion detection is a necessary and important aspect of security.

The first case study illustrated that government entities still struggle with the technical and administrative aspects of cyber security. Many governments across the globe could be surveyed with similar results. Based on responses from those surveyed, the network and assets were poorly protected and there was significant disagreement on which protection measures were in place. The survey illustrates the need for user education and a comprehensive security program.

The second case study presented a novel and promising approach to intrusion detection using data mining techniques. There exists major concern about the performance of the algorithms and whether they can be used in real-time for intrusion prevention.

The third case study presented an innovative approach to intrusion detection using entropy calculations. The results were presented for a DoS attack from multiple, external machines and an increase in entropy in IP addresses is to be expected. While three preliminary results looked good for DoS attacks, the study did not address whether the entropy calculation would be an effective indicator of intrusion for other types of attacks.

Innovative research into traffic measurement analysis is yielding good results. Current research is improving false positive, false negative and detection rates. Intrusion detection is not a panacea. However, intrusion detection should be a layer in the defense-in-depth approach to securing systems.

References

Al-Enezi, K. A., Al-Shaikhli, I. F., Al-Kandari, A. R., & Al-Tayyar, L. Z. (2014). A Survey of Intrusion Detection System Using Case Study Kuwait Governments Entities. *2014 3rd International Conference on Advanced Computer Science Applications and Technologies*. doi:10.1109/acsat.2014.14

Basicevic, I., Kostovic, Z., Popovic, M., & Ocovaj, S. (2013). Effect of nonstationarity of network traffic in entropy-based intrusion detection (case study). *2013 21st Telecommunications Forum Telfor (TELFOR)*. doi:10.1109/telfor.2013.6716188

Bereziński, P., Jasiul, B., & Szpyrka, M. (2015). An Entropy-Based Network Anomaly Detection Method. *Entropy*, *17*(4), 2367-2408. doi:10.3390/e17042367

Bose, S., Bharathimurugan, S., & Kannan, A. (2007). Multi-Layer Integrated Anomaly Intrusion Detection System for Mobile Adhoc Networks. *2007 International Conference on Signal Processing, Communications and Networking*. doi:10.1109/icscn.2007.350763

Bukac, V. (2010). *IDS System Evasion Techniques* (Master's thesis, Masarykova Univerzita). Retrieved from http://is.muni.cz/th/172999/fi_m/MT_Bukac.pdf

Casenove, M. (2015). Exfiltrations using polymorphic blending techniques: Analysis and countermeasures. *2015 7th International Conference on Cyber Conflict: Architectures in Cyberspace*. doi:10.1109/cycon.2015.7158479

Fogla, P., Sharif, M., Perdisci, R., Kolesnikov, O., & Lee, W. (2006). Polymorphic Blending Attacks. *in 15th USENIX Security Symposium*.

Gendreau, A. A., & Moorman, M. (2016). Survey of Intrusion Detection Systems towards an End to End Secure Internet of Things. *2016 IEEE 4th International*

Conference on Future Internet of Things and Cloud (FiCloud). doi:10.1109/ficloud.2016.20

Jaiganesh, V., & Mangayarkarasi, S. (2013). Intrusion Detection Systems: A Survey and Analysis of Classification Techniques. *International Journal of Advanced Research in Computer and Communication Engineering*, *2*(4). Retrieved from www.ijarcce.com

Karimi, A. M., Niyaz, Q., Weiqing Sun, Javaid, A. Y., & Devabhaktuni, V. K. (2016). Distributed network traffic feature extraction for a real-time IDS. *2016 IEEE International Conference on Electro Information Technology (EIT)*. doi:10.1109/eit.2016.7535295

Mallissery, S., Prabhu, J., & Ganiga, R. (2011). Survey on intrusion detection methods. *3rd International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom 2011)*. doi:10.1049/ic.2011.0085

Mavroeidakos, T., Michalas, A., & Vergados, D. D. (2016). Security architecture based on defense in depth for Cloud Computing environment. *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. doi:10.1109/infcomw.2016.7562097

Milenkoski, A., Jayaram, K. R., Antunes, N., Vieira, M., & Kounev, S. (2016). Quantifying the Attack Detection Accuracy of Intrusion Detection Systems in Virtualized Environments. *2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)*. doi:10.1109/issre.2016.39

Ptacek, T. H., & Newsham, T. N. (1998). *Insertion, evasion, and denial of service: Eluding network intrusion detection*. SECURE NETWORKS INC CALGARY ALBERTA.

Scarfone, K. A. (2007). *Guide to Intrusion Detection and Prevention Systems (IDPS)* (NIST Special Publication 800-94).

Shankar, U., & Paxson, V. (n.d.). Active mapping: resisting NIDS evasion without altering traffic. *Proceedings 19th International Conference on Data Engineering (Cat. No.03CH37405)*. doi:10.1109/secpri.2003.1199327

Song, Y., Locasto, M. E., Stavrou, A., Keromytis, A. D., & Stolfo, S. J. (2009). On the infeasibility of modeling polymorphic shellcode. *Machine Learning*, *81*(2), 179-205. doi:10.1007/s10994-009-5143-5

Studer, A., McLain, C., & Lippmann, R. (2007). Tuning Intrusion Detection to Work with a Two Encryption Key Version of IPsec. *MILCOM 2007 - IEEE Military Communications Conference*. doi:10.1109/milcom.2007.4455095

S.SyedNavaz, A., Sangeetha, V., & Prabhadevi, C. (2013). Entropy based Anomaly Detection System to Prevent DDoS Attacks in Cloud. *International Journal of Computer Applications*, *62*(15), 42-47. doi:10.5120/10160-5084

Thurner, S., Grun, M., Schmitt, S., & Baier, H. (2015). Improving the Detection of Encrypted Data on Storage Devices. *2015 Ninth International Conference on IT Security Incident Management & IT Forensics*. doi:10.1109/imf.2015.12

Trio Pramono, Y. W., & Suhardi. (2014). Anomaly-based intrusion detection and prevention system on website usage using rule-growth sequential pattern analysis: Case study: Statistics of Indonesia (BPS) website. *2014 International*

Conference of Advanced Informatics: Concept, Theory and Application (ICAICTA). doi:10.1109/icaicta.2014.7005941

Varshovi, A., Rostamipour, M., & Sadeghiyan, B. (2014). A fuzzy Intrusion Detection System based on categorization of attacks. *2014 6th Conference on Information and Knowledge Technology (IKT)*. doi:10.1109/ikt.2014.7030332

Yu, S., Zhou, S., Liu, L., Yang, R., & Luo, J. (2011). Detecting Malware Variants by Byte Frequency. *Journal of Networks*, *6*(4). doi:10.4304/jnw.6.4.638-645