Deep Packet Inspection: A Key Issue for Network Security

14

Hannah Bartus

Abstract

As the number of cyber-attacks continue to increase, the need for data protection increases as well. Deep Packet Inspection is a highly effective way to reveal suspicious content in the headers or the payloads in any packet processing layer, except when the payload is encrypted. DPI is an essential inspector of packet payloads as it is applied to many different layers of the OSI model. The DPI tasks include intrusion detection, exfiltration detection and parental filtering. This can be a great advantage as layer-independent attacks are becoming more prevalent. It allows for inspection of all layers for attacks. However, there are challenges that come with Deep Packet Inspection. Some include the decrease of throughput of the system, attacks through the Secured Socket Layer and intrusion fingerprint matching. These challenges do not constitute as grounds to eliminate DPI as a method, but instead obstacles to be aware of in case difficulties with implementation prevails.

Keywords

Deep Packet Inspection · Network intrusion detection system · Secure socket layer · Network security · Network traffic · Pattern matching

14.1 Introduction

In approaching the investigation of network attacks, the use of Deep Packet Inspection (DPI) must be analyzed. This technique is used to closely examine packets and its fields that flow within the network. This will allow for anomaly detection within the network flow. The anomalies can be used to determine important information for dealing with and responding to security incidents. This could be in the form of IP addresses involved, the type of attack, the time of the incident as well as the incident duration. With all of the data from this inspection as well as what is gathered during the incident, security professionals are then able to mitigate risks and consequently, incidents [1].

At this point in time, the use of DPI is limited to end-host. This is because the edge and core routers do not have the processing power needed to inspect the entire content of a packet at wire speed. The edge router typically only examines the head of that packet, whereas the core routers examine the packet's destination address for forwarding. The problem arises when these routers perform at such high line speeds that this leaves very few nanoseconds to analyze the entire packet content. As the line rates increase between an edge router and a core router, the task becomes more difficult as the time to process each byte of the packet decreases [2]. Usually, an intrusion detection system for Deep Packet Inspection consists of two different parts. The first is a header rule that includes a 5-tuple packet classification being performed on the packet's header. The second focuses on content at given points within the packet's payload. However, only 60–80% of instructions are executed in the fraction of time of 40–70% with network intrusion detection [3]. Any type of design for improving network security must have efficiency in mind. A high throughput is very prevalent and cannot be the solution of throwing more processing power at string matching due to other restrictions in the network. This would cause an increase in cost as the need for cooling would increase with maintenance expenses.

Deep Packet Inspection must be implemented in such a way in order to avoid the main problems that come with it. DPI is quite complex and is extremely difficult to customize. Without a team directly working on implementation of a custom network intrusion detection system, users ought to depend on commercial products to perform the actions

H. Bartus (✉)
Robert Morris University, Moon, PA, USA
e-mail: hcbst109@mail.rmu.edu

S. Latifi (ed.), *Information Technology – New Generations*, Advances in Intelligent Systems and Computing 738,
https://doi.org/10.1007/978-3-319-77028-4_14

needed within DPI. However, one of the biggest concerns is DPI of the Secure Socket Lay (SSL). Due to the encryption of the HTTPS packet, SSL creates a blind spot for the firewall. This occurs because the firewall inspects the data broken into packets and a traditional firewall cannot inspect encrypted traffic on its own—and anything behind that encryption will enter the network untouched [4]. As the need for encrypted services like HTTPS increases, unfortunately the security risks also increase due to the level of insecurity behind the secure socket layer.

14.2 Literature Review

The goal of this research is to explore solutions to the obstacles with Deep Packet Inspection. DPI has been implemented in many workplace settings to aid in the security of the network. However, there is not one solution that results in 100% accuracy regarding network monitoring through DPI.

14.2.1 Pattern Matching Algorithms

The standard function of a Network Intrusion Detection System (nIDS) is based on a set of signatures, each describing one known intrusion threat. The nIDS examines the network's traffic for any matches to known intrusion attempts. NIDS's rule set is a two-dimensional data-structure chain that tests chain headers against packet header rules. When the packet header rule is matched, a pattern matching algorithm begins. However, this is the most financially detrimental operation of the nIDS [3]. No single algorithm performs best in all conditions, however a possible hybrid of multiple algorithms may be the best solution for such an obstacle.

14.2.1.1 The Boyer-Moore Algorithm

The Boyer-Moore Algorithm is the most well-known pattern matching algorithm for examining an input against a single pattern. This algorithm starts at the rightmost character of the search pattern and analyzes leftward. When a mismatch occurs, both heuristics are triggered. The bad character heuristic is the first one triggered. This heuristic operates by shifting the search pattern to the rightmost position of where it appears if the mismatching character appears in the search pattern. However, if the mismatching character does not appear in the search pattern, then it is shifted to one position past that character. The good suffixes heuristic, when triggered due to a mismatch in the middle of the search pattern, shifts the search pattern to the next occurrence of the suffix in the pattern [3]. The Boyer-Moore algorithm has been adjusted many times, whether it was reduced to its bad character heuristic solely, or a modified version of the algorithm then integrated and tested with an enterprise internet connection.

14.2.1.2 The Wu-Manber Algorithm

The Wu-Manber algorithm is used in some variant in the nIDS known as Snort. This algorithm is based on the bad character heuristic of Boyer-Moore but uses up to two-byte bad shift tables. These were created to process the entirety of the patterns instead of just one at a time. This creates a table similar to that of a rainbow list that can then be used to detect intrusion threats. The Wu-Manber algorithm "performs a hash on the two-character prefix of the current input to index into a group of patterns, which are then checked starting from the last character, as in Boyer-Moore" [3]. Although this algorithm performs well on large sets, it struggles with short patterns in rules.

14.2.2 Hardware

The A10 network middlebox is not only praised for the throughput that it can handle, but also the additional tools and operations that can be utilized. A10 allows for visibility and security in the form of hardware, software or in the cloud. This network analyzer has both the ability to scan for intrusions and DoS attacks, as well as scan through the SSL. This hardware can decrypt packets to further analyze their contents and therefore better secure the network they are passing through [5]. Many federal contractors use this hardware because of its protection against their highest threat of intrusion through the SSL.

The FortiGate is a similar solution to the A10 network middlebox. However, FortiGate's hardware is known as a Next Generation Firewall (NGFW). This hardware also has the ability to decrypt the SSL for further security. The NGFW can be combined with pattern-matching algorithms to perform at its best and create a secure and intrusion-free environment [6]. Both of these hardware solutions are ones that can be implemented in any size organization or network due to their flexibility.

14.2.3 Software

Although hardware is a great permanent DPI solution, software can also help an organization test out what they may need in a DPI tool. For example, SolarWinds is a network managing tool that can analyze network performance as well as track network traffic. SolarWinds offers performance monitors that would assist in finding anomalies within the network [7].

A second software option is Snort, an open source nIDS. This free software can help sort out where possible vulnerabilities are. This software can also perform real-time traffic analysis and packet logging on IP networks as well as protocol analysis to detect anomalies within the network and catch intrusions [8]. Although each of these software systems are not permanent solutions, they can assist in quick scans of the network to then perform a more intense evaluation of vulnerabilities.

14.3 Proposed Solution

To begin, the highest vulnerabilities must be assessed. Across the board, one of the greatest vulnerabilities is the SSL. Although this allows for information to be encrypted and protected, it also causes a blind spot in network detection. HTTPS and other encryption protocols have grown rapidly over the past years and therefore protected the private data in those encrypted packets from eavesdroppers. However, this category also includes middleboxes, like the A10 network, which are also, by definition, eavesdropping on the network traffic. Therefore, malicious encrypted packets are not inspected and are able to accomplish their malicious tasks. However, many middleboxes "mount a man-in-the middle attack on SSL and decrypt the traffic at the middlebox" which is an incredibly insecure way of attempting to support HTTPS [9]. However, HTTPS must be supported through some available method. Below is a diagram of how middleboxes interact with the SSL and their rules detailed in the rule generator.

Different DPI tools perform different tasks. This being established, one must know and understand the company's needs prior to being able to choose the tool that best suits them. A company must have an organization-wide security assessment to accurately install what is needed. For example, if throughput is not a major concern due to the fact that it is a different network than what customer-facing employees use, then throughput does not need to be analyzed as heavily. The organization must make sure they are not using too many resources with DPI and SSL inspection. Therefore, it is important to know the traffic primarily.

14.3.1 Know the Traffic

In understanding the traffic, it is worth noting how much traffic is expected and how much of that traffic is encrypted. From here, the allowance of encrypted traffic can be edited and customized [4]. The first step in DPI implementation is testing out open source software to see if the data produced on the user interface is what is imperative for the organization. The Fortigate user interface details the IP address of the source as well as the destinations in which they are headed. Besides the source, this interface also shows the amount of data sent and received. This is the most significant step in remaining aware of the traffic in the network.

14.3.2 Be Selective

Secure Socket Layer inspection should only be placed where it is needed. This will not only assist the throughput in the system, but also the policy limitations that are caused by such an inspection [4]. This selectivity could also be related to the amount of customization added to any type of hardware used.

14.3.3 Use Hardware Acceleration

For most SSL security with DPI, a hardware accelerator is the best step in inspection. However, with this accelerator, it is important for customization that matches the business needs of the company. This is where custom or known algorithms can be added for the greatest amount of security [4]. Depending on the company, the algorithm needs as well as the hardware needs may vary. Once these needs are determined, implementation can occur.

14.3.4 Test Real-World SSL Inspection Performance

Using the hardware accelerator, the best way to enforce this policy would be to gradually deploy SSL inspection to test SSL inspection performance. The SSL Inspection performance test would need to be managed like any other security protocol. This inspection may cause a decrease of the allowed throughput during the actual process, but it would become alike to any other scan made on a default schedule.

Security Information exchange programs can help explore algorithms just as many do with patches when attacks happen on the firewall. However, creating continued algorithms customized to the organization can be the greatest factor for SSL inspection [10].

14.4 Conclusion

When all of the above steps are performed, DPI can be properly initiated and SSL will no longer be as serious of a security issue now that the proper controls are in place. Although many DPI tools were tested throughout our research, we were not able to test every DPI tool. We also met with a representative from SolarWinds, but decided it was not the right solution due to the lack of possible uses with DPI and

SSL inspection. With the increase of ecommerce throughout the workplace, SSL inspections are more necessary than they have ever been. Luckily with the progressive technology of DPI, encrypted data is able to be decrypted, pattern matched and then re-encrypted and sent wherever needed.

References

1. G.A.P. Rodrigues, R. de Oliveira Albuquerque, F.E.G. de Deus, R.T. de Sousa Jr., G.A. de Oliveira Júnior, L.J.G. Villalba, T.-H. Kim, Cybersecurity and network forensics: Analysis of malicious traffic towards a Honeynet with Deep Packet Inspection. Appl. Sci. **7**(10), 1082 (2017)
2. A. Kennedy, X. Wang Z. Liu, B. Liu, Ultra-high throughput string matching for Deep Packet Inspection, in *2010 Design, Automation & Test in Europe Conference & Exhibition (2010)*, Dresden, 2010, pp. 399–404
3. S. Antonatos, K.G. Anagnostakis, E.P. Markatos, Generating realistic workloads for network intrusion detection systems. SIGSOFT Softw. Eng. Notes **29**(1), 207–215 (2004)
4. V. Martin, Why you should use SSL inspection—Fortinet Cookbook. [Online] Fortinet Cookbook (2017). http://cookbook.fortinet.com/why-you-should-use-ssl-inspection
5. A10 Networks, Thunder SSLi|A10 Networks (2017) [Online]. https://www. a10networks.com/products/thunder-series/ssl-decryption-encryption-and-inspection-ssl-insight
6. Fortinet, Next-Generation Firewalls (2017). [Online]. https://www.fortinet.com/products/next-generation-firewall.html
7. Solarwinds, IT Management Software & Monitoring Tools|SolarWinds (2017). [Online]. https://www.solarwinds.com/?&CMP=KNC-TAD-GGL-SW_NA_US_PP_CPC_LD_EN_PBRDE_DWA-X-X_X_X_X-X-775928844_40237439985_g_c_Solarwinds-e~185579782075~&kwid=iDVonkDn&gclid=CjwKCAiAxarQBRAmEiwA6YcGKOxRegq5tt6yVv_1LfkFMzd51MDaZX-JCVc0l2077adKbim-1GosPhoCL58QAvD_BwE
8. Snort.org, What is Snort? (2017) [Online]. https://www.snort.org/faq/what-is-snort
9. J. Sherry, C. Lan, R.A. Popa, S. Ratnasamy, Blindbox: Deep packet inspection over encrypted traffic. Comput. Commun. Rev. **45**(5), 213 (2015). https://doi.org/10.1145/2829988.2787502
10. M. Pyatkovskiy, Fast SSL testing using precalculated cryptographyc data (2017). Patents, [Online] p. 9. https://www.google.com/patents/US8649275