Leveraging Protocol Structures for Enhanced Network Traffic Analysis

Apurba Ranjan Biswal,Biswadarsi Biswal,Fernadis Mishra
Dept. of Computer Science and Engineering, GIFT Autonomous, Bhubaneshwar, 752054, India
Email:biswadarsi@gift.edu.in

Abstract

There are situations where network administrators don't have their analysis toolset following their organization's goals. There are existing applications for network traffic capture and analysis. However, the alerting system on these applications is not added. A user not experienced with networking concepts will not be able to understand the generated output in these existing traffic capture systems. This project will develop an application to monitor the traffic in a user laptop connected to an Ethernet or wireless Internet. The application will generate a report with the details of internet traffic; Ethernet, IP, ICMP/or UDP/ or TCP, and Application layer services. It will also rank the used application layer protocols from the one that utilized more bandwidth to the one that utilized the least bandwidth. We will create a loop that keeps on looping to listen for any data that comes across the network connection. Then, this captured data, an Ethernet frame that has IP packet inside which has TCP information, will be passed to various unpacking functions.

Keywords: Analyzer, Packet Sniffer, Network Analyzer, Packet Analyzer

Introduction

In Modern Society, Computers Are No Longer Treated As Stand-Alone Machines. Instead, They Are Communicating To Share Resources And Data Through Computer Networks. Network Packets Are Units Of Data Travelling In These Computer Networks, Carrying All The Essential Information From Its Source To Its Final Destination. Besides The Packet Payload (The Actual Data), Which Contains Lots Of Helpful Information, The Packet Headers Themselves Also Have A Wealth Of Information About The Network Infrastructure And Network Topologies And May Also Indicate Some General Behaviour Of The Network Traffic. For Example, The Header Information Was Used To Discover The Congestion Sources In The Network Traffic In [5,6,7], And To Analyze The Quality Of Routing In The Internet In [9]. Another Use Of The Packet Header Information Is In Genesis, A Distributed Network Simulation System [14,15,16], Including Wireless Networks [11].A Packet Analyzer, Also Known As A Packet Sniffer, Protocol Analyzer, Or Network Analyzer,[1][2][3][4][5][6][7] Is A Computer Program Or Computer Hardware Such As A Packet Capture Appliance, That Can Intercept And Log

Traffic That Passes Over A Computer Network Or Part Of A Network.[8] Packet Capture Is The Process Of Intercepting And Logging Traffic. As Data Streams Flow Across The Web, The Analyzer Captures Each Packet And, If Needed, Decodes The Packet's Raw Data, Showing The Values Of Various Fields In The Packet, And Analyzes Its Content According To The Appropriate RFC Or Other Specifications.

Advantage: Hardware Agnostic

SNMP and Net Flow both require support at the network hardware level. While both technologies enjoy wide support, they are not universally available. There may also be differences in how each vendor implements them. On the other hand, packet capture does not require specialized hardware support and can take place from any device that has access to the network.

Disadvantage: Large File Sizes

Full packet capture can take up large amounts of disk space – sometimes up to 20 times as much space as other options. Even when filtering is applied, a single capture file may take up many gigabytes of storage. This can make packet captures unsuitable for long-term storage. These large file sizes can also result in lengthy wait times when opening a .pcap in a network analysis tool.

Disadvantage: Too Much Information

While packet captures to provide a very complete look at network traffic, they're often too comprehensive. Relevant information can often get lost in vast sums of data. Analysis tools have features order, sort, and filter capture files, but many use cases might be better served by other options. It's often possible to troubleshoot a network or spot signs of an attack with just the summarized versions of network traffic available in other monitoring solutions. One common approach is to use a technology like Net Flow to monitor all traffic and turn to a full packet capture as needed.

Disadvantage: Fixed Fields

The most recent iterations of Net Flow allow for customizable records, meaning network adman's can choose what information to capture. Since packet capture is based on the existing structure of an IP packet, there is no room for customization. This may not be an issue, but again depending on the use case, there may not be a need to capture all fields of an IP packet. Packet capture is invaluable from a troubleshooting and security perspective but should never be the sole tool that a network admin or security engineer relies on. The increased use of encryption for both legitimate and illegitimate purposes limits the effectiveness of tools like Wire shark. Packet captures also do not give incident responders much of an idea of what actions have taken place on a host. Files could have been modified, processes hidden, and new user accounts created without generating a single packet.

Literature Review

Network packets hold more than just communication data and metadata; files that traversed through a network can be reconstructed from network packet streams (network carving) (Beverly et al., 2011) using purpose-designed network carvers or packet analyzers that support file export from packet capture. This, together with other options to find traces of network data transfer, makes packet analysis a primary trace back technique in network forensics. It can assist in finding traces of nefarious online behavior and breaches affecting an organization, determining the source of network security attacks, and acquiring host-based evidence of malicious actions (Johansen, 2017), although making sense of encrypted network traffic is far more challenging than the analysis of unencrypted traffic (van de Wiel et al., 2018). For example, network traffic classification based on packet analysis and port numbers alone is infeasible for encrypted VoIP applications, such as Skype (Alshammari and Zincir-Heywood, 2015), although even encrypted network traffic can be classified using machine learning (Dong and Jain, 2019).Packet sniffing is a method of tapping packet flows, i.e., packets as they flow across a communication network (Ansari et al., 2003), and even re-transmitted packets, such as with different TCP properties. This can be utilized for reconstructing data transferred over the network and might even be used as an anti-forensic measure.

Because packet capture files often contain sensitive data, such as network users' personal data, information about an enterprise network's internal structure, etc., privacy restrictions, policies, and laws make it impossible to share packet capture files. There are approaches and solutions to automatically scramble network packet capture data while preserving binary integrity, such as SafePcap,14 which complies with the Europe Union's General Data Protection Regulation (GDPR)15 and NIST's NISTIR 8053 "De-Identification of Personal Information." 16 Safe Pcap performs data modifications in a break-proof manner by recalculating the lengths, checksums, offsets and all other services for all affected packets and protocol layer fields on the fly.

A full packet capture is imperative when investigating what has happened in a network at a particular point in time and who was actually involved in an online activity because the IP address of a suspect's computer alone cannot serve as the basis of forensic investigations due to the dynamic nature of IP addresses, and because they often cannot be linked directly to an individual (Clarke et al., 2017) and often not even to an exact geographical location (Afanasyev et al., 2011). Nevertheless, following the TCP stream of the simultaneous use of SMTP and a particular IP address can identify the address associated with the From tag of the email header. Furthermore, email headers contain the name of the sender, which may reveal the suspect's real name. Emails sent by the user can be reconstructed, including any attachments. The manufacturer of a suspect's computer can be identified with high certainty based on the Organizational Unique Identifier (OUI) part of the device's MAC address,17 although this cannot be used in many cases, particularly in corporate networks. Based on the packet data, it can be determined when the suspect logged in to the network. If the password of the suspect was encoded in Base64, it can be converted to UTF-8 to reveal the actual password that was used to log in. Ultimately, such information can help build a profile of the suspect's identity.

Network packet analyzers

Generally, each packet analyzer performs four steps to process packets (Yang et al., 2018): Open a packet capture socket: select a network device and open it for live capture, retrieve the network address and subnet mask, convert the packet filter expression into a packet filter binary, and assign the packet filter to the socket

Packet capture loop: determine the datalink type and start the packet capture

Parse and display packets: set a character pointer to the beginning of the packet buffer and move it to a particular protocol header by the size of the header preceding it in the packet, and map the header to the appropriate header structure (IP, TCP, UDP, ICMP, etc.) by casting the character pointer to a protocol-specific structure pointer

Terminate the capturing process: send interrupt signals and close the packet capture socket

Packet analyzers are designed for various purposes and differ in terms of capabilities and features, hardware resource utilization, processing speed (Goyal and Goyal, 2017), supported protocols, user- friendliness, supported operating systems, supported network types, interface, license, and implementation type. Many packet analyzers support both live capture and offline analysis. The deep inspection of packets and the analysis of various types of network traffic are available only in those analyzers that support hundreds of protocols. Those packet analyzers that intercept traffic on wireless networks are called wireless analyzers (WiFianalyzers), e.g., Aircrack-ng,18 and Kismet.19 For Bluetooth, there is a purpose-built packet sniffer called FTS4BT.20

Some tools support data carving, capture file quality assessment, anomaly detection, protocol encapsulation, and flexible packet aggregation. The list of supported file formats varies between packet analyzers, and some tools even provide on-the-fly gzip decompression.21The analyzers that come with a GUI feature typically have a packet browser to visualize the packet content, and various display filters to show only the information relevant for a particular task, rather than everything captured. Some packet analyzers can differentiate between frame types, and visualize them using color schemes.

In terms of licensing, packet analyzers are either open source, freeware, or commercial. Common license types associated with packet analyzers include the GNU General Public License22 and proprietary licenses.There are both hardware appliances and software implementations for packet analysis, although software tools are far more common than hardware implementations.

Methodology

Packet Capture Formats

While packet capture tools like Wireshark can be used to inspect traffic in real-time, it's more common to save captures to a file for later analysis. These files can be saved in a variety of formats.

.pcap files are the most common and are generally compatible with a wide range of network analyzers and other tools. .pcapng builds on the simple .pcap format with new fields and capabilities and is now the default format when saving files in Wireshark. Some commercial tools may also use proprietary formats.

Libraries

Libraries like libpcap, winpcap, and npcap are the real stars of the packet capture show, hooking into an operating system's networking stack and providing the capability to peer into packets moving between interfaces. Many of these libraries are open-source projects, so you may find them in a wide variety of both commercial and free packet capture tools. In some cases, you may need to install the library separately from the tool.

Filtering

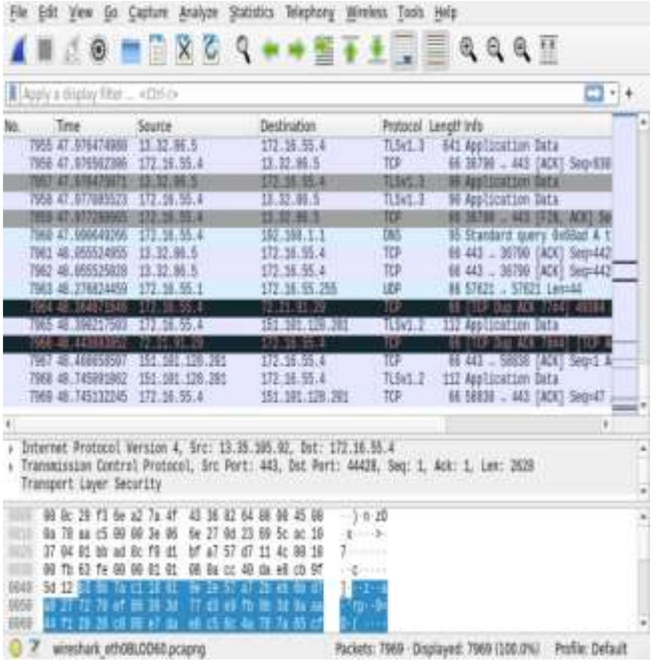
Full packet capture can take quite a bit of space and demand more resources from the capturing device. It's also overkill in most cases – the most interesting information is typically only a small portion of the total traffic being observed. Packet captures are often filtered to weed out the relevant information. This can be based on everything from the payload to IP address to a combination of factors.

Packet Capture Tools

A large number of different tools are available to capture and analyze the packets traversing your network. These are sometimes known as packet sniffers. Here are some of the most popular:

Wireshark

The quintessential packet tool, Wireshark is the go- to packet capture tool for many network administrators, security analysts, and amateur geeks. With a straightforward GUI and tons of features for sorting, analyzing, and making sense of traffic, Wireshark combines ease of use and powerful capabilities. The Wireshark package also includes a command-line utility called tshark.


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

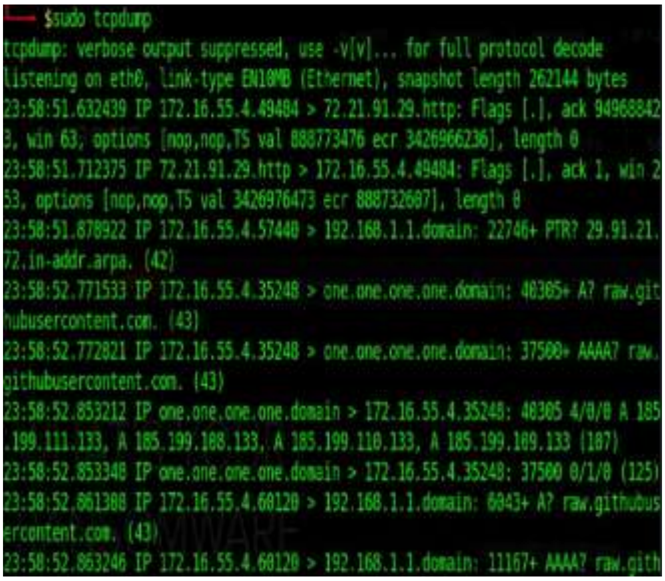
```
No.   Time         Source           Destination      Protocol Lengtl Info
 7955 47.976474980 13.32.86.5       172.16.55.4      TLSv1.3  641 Application Data
 7956 47.976502306 172.16.55.4      13.32.86.5       TCP       66 36790 → 443 [ACK] Seq=638
 7957 47.976479071 13.32.86.5       172.16.55.4      TLSv1.3   90 Application Data
 7958 47.977085523 172.16.55.4      13.32.86.5       TLSv1.3   90 Application Data
 7959 47.977280985 172.16.55.4      13.32.86.5       TCP       66 36790 → 443 [FIN, ACK] Se
 7960 47.990649266 172.16.55.4      192.168.1.1      DNS       95 Standard query 0x6bad A t
 7961 48.055524855 13.32.86.5       172.16.55.4      TCP       66 443 → 36790 [ACK] Seq=442
 7962 48.055525828 13.32.86.5       172.16.55.4      TCP       66 443 → 36790 [ACK] Seq=442
 7963 48.276824459 172.16.55.1      172.16.55.255    UDP       86 57621 → 57621 Len=44
 7964 48.364871946 172.16.55.4      72.21.91.29      TCP       66 [TCP Dup ACK 7764] 49484
 7965 48.398217589 172.16.55.4      151.101.128.201  TLSv1.2  112 Application Data
 7966 48.443893852 72.21.91.29      172.16.55.4      TCP       66 [TCP Dup ACK 7844] [TCP K
 7967 48.468658507 151.101.128.201  172.16.55.4      TCP       66 443 → 58838 [ACK] Seq=1 A
 7968 48.745891862 151.101.128.201  172.16.55.4      TLSv1.2  112 Application Data
 7969 48.745132245 172.16.55.4      151.101.128.201  TCP       66 58838 → 443 [ACK] Seq=47

▸ Internet Protocol Version 4, Src: 13.35.105.92, Dst: 172.16.55.4
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 44428, Seq: 1, Ack: 1, Len: 2628
  Transport Layer Security

0000  00 0c 29 f1 6e a2 7a 4f  43 36 82 64 08 00 45 00   ··)·n·z0
0010  0a 78 aa c5 00 00 3e 06  6e 27 0d 23 69 5c ac 10   ·x······>·
0020  37 04 01 bb ad 8c f8 d1  bf a7 57 d7 11 4c 80 18   7·······
0030  00 fb 63 fe 00 00 01 01  08 0a cc 40 da e8 cb 9f   ··c·····
0040  5d 12
0050
0060

wireshark_eth0BLOD60.pcapng          Packets: 7969 · Displayed: 7969 (100.0%)    Profile: Default
```

tcpdump



```
└─ $sudo tcpdump
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
23:58:51.632439 IP 172.16.55.4.49484 > 72.21.91.29.http: Flags [.], ack 94968842
3, win 63, options [nop,nop,TS val 888773476 ecr 3426966236], length 0
23:58:51.712375 IP 72.21.91.29.http > 172.16.55.4.49484: Flags [.], ack 1, win 2
53, options [nop,nop,TS val 3426976473 ecr 888732687], length 0
23:58:51.878922 IP 172.16.55.4.57440 > 192.168.1.1.domain: 22746+ PTR? 29.91.21.
72.in-addr.arpa. (42)
23:58:52.771533 IP 172.16.55.4.35248 > one.one.one.one.domain: 40305+ A? raw.git
hubusercontent.com. (43)
23:58:52.772821 IP 172.16.55.4.35248 > one.one.one.one.domain: 37500+ AAAA? raw.
githubusercontent.com. (43)
23:58:52.853212 IP one.one.one.one.domain > 172.16.55.4.35248: 40305 4/0/0 A 185
.199.111.133, A 185.199.108.133, A 185.199.110.133, A 185.199.109.133 (107)
23:58:52.853348 IP one.one.one.one.domain > 172.16.55.4.35248: 37500 0/1/0 (125)
23:58:52.861388 IP 172.16.55.4.60120 > 192.168.1.1.domain: 6043+ A? raw.githubus
ercontent.com. (43)
23:58:52.863246 IP 172.16.55.4.60120 > 192.168.1.1.domain: 11167+ AAAA? raw.gith
```

Lightweight, versatile, and pre-installed on many UNIX-like operating systems, tcpdump is a CLI junkie's dream come true when it comes to packet captures. This open source tool can quickly capture packets for later analysis in tools like Wireshark but has plenty of its own commands and switches to make sense of vast sums of network data.

SolarWinds Network Performance Monitor

This commercial tool has long been a favorite for its ease of use, visualizations, and ability to classify traffic by application. Though the tool only installs on Windows platforms, it can sniff and analyze traffic from any type of device.

ColaSoftCapsa

ColaSoft makes a commercial packet sniffer aimed at enterprise customers, but also offers a pared- down edition aimed at students and those just getting into networking. The tool boasts a variety of monitoring features to aid in real-time troubleshooting and analysis.

Kismet

Kismet is a utility devoted to capturing wireless traffic and detecting wireless networks and devices. Available for Linux, Mac, and Windows platforms, this tool supports a wide range of capture sources including Bluetooth and Zigbee radios. With the right setup, you can capture packets from all of the devices on the network.

Packet Capture and Packet Sniffer Use Cases

While the term Packet Sniffer may conjure up images of hackers covertly tapping into sensitive communications, there are plenty of legitimate uses for a packet sniffer. The following are some typical use cases for packet sniffers:

Asset Discovery/Passive Reconnaissance

Packets by their very nature include source and destination addresses, so a packet capture can be used to discover active endpoints on a given network. With enough data, it's even possible to fingerprint the endpoints. When done for legitimate business purposes, this is called discovery or inventory. However, the passive nature of a packet capture makes it an excellent way for malicious attackers to gather information for further stages of an attack. Of course, the same technique can be used by red teamers testing an organization's security

Troubleshooting

When troubleshooting network issues, inspecting the actual network traffic can be the most effective means of narrowing down the root cause of a problem. Packet sniffers allow network administrators and engineers to view the contents of packets traversing the network. This is an foundational network protocols such as DHCP, ARP, and DNS. Packet captures do not, however, reveal the contents of encrypted network traffic. Sniffing packets can help verify that traffic is taking the correct path across the network, and is being treated with the correct precedence. A congested or broken network link is often easy to spot in a packet capture because only one side of a typically two- sided conversation will be present. Connections with a large number of retries or dropped packets are often indicative of an overused link or failing network hardware.

Intrusion Detection

Suspicious network traffic can be saved as packet capture and fed into an IDS, IPS, or SIEM solution for further analysis. Attackers go to great lengths to blend in with normal network traffic, but a careful inspection can uncover covert traffic. Known malicious IP addresses, telltale payloads, and other minute details can all be indicative of an attack. Even something as innocuous as a DNS request, if repeated at a regular interval, could be a sign of a command and control beacon.

Incident Response and Forensics

Packet captures provide a unique opportunity for incident responders. Attackers can take steps to cover their tracks on endpoints, but they can't unsend packets that have already traversed a network. Whether it's malware, data exfiltration, or some other type of incident, packet captures can often spot signs of an attack that other security tools miss. As a packet header will always contain both a source and destination address, trace the path of an attacker through the network, or spot signs of data being exfiltrated out of the network.

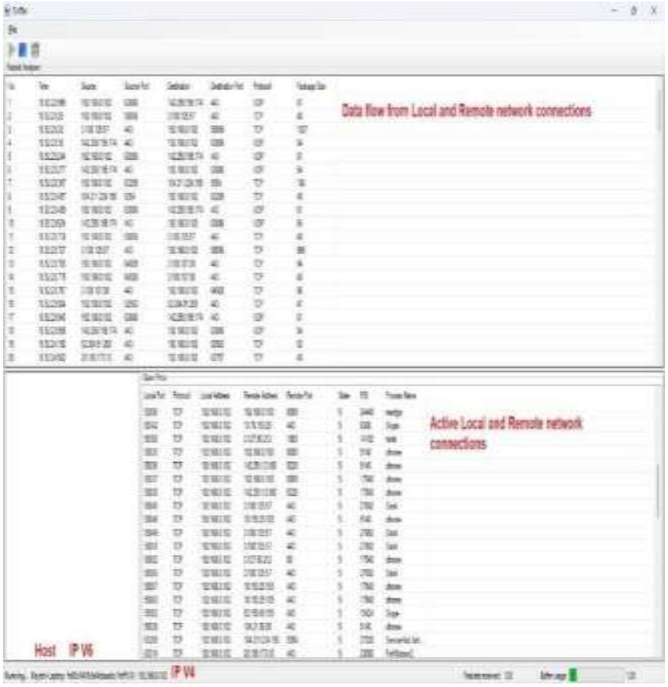
Packet analyzer software

Among the packet analyzer software tools, there are purpose-designed packet analyzers and network tools that provide features for packet capture and analysis. Such network tools include intrusion detection software, proxies, vulnerability assessment tools, network scanners, and network monitoring tools, which are used in network forensics (Joshi and Pilli, 2016).

In 1997, the Federal Bureau of Investigation (FBI) implemented its customizable packet sniffer as part of the system called Carnivore (which was later renamed to DCS1000). It monitored users' Internet traffic, including emails. It was phased out by 2005. In 1998, Gerald Combs developed Ethereal, a free and open-source packet analyzer, which was renamed to Wireshark in 2006 (Orebaugh et al.,

2006). Over the years, Wireshark has become one of the most widely used graphical packet capture and protocol analysis tools (Shimonski, 2013), featuring a highly intuitive GUI for packet analysis (Sanders,

2017). This GUI has a customizable packet browser that displays a maximum of three panes simultaneously, including a packet list and the packet details and packet bytes of the currently selected packet.



Data flow from Local and Remote network connections

Active Local and Remote network connections

Host IP V6

IP V4

Conclusion

Packet capture is invaluable from a troubleshooting and security perspective, but should never be the sole tool that a network admin or security engineer relies on. The increased use of encryption for both legitimate and illegitimate purposes limits the effectiveness of tools like Wireshark. Packet captures also do not give incident responders much of an idea of what actions have taken place on a host. Files could have been modified, processes hidden, and new user accounts created without generating a single packet.

References

[1] Afanasyev et al., 2011M. Afanasyev, T. Kohno, J. Ma, N. Murphy, S. Savage, A.C. Snoeren, G.M. VoelkerPrivacy-preserving network forensicsCommun. ACM, 54 (5) (2011), pp. 78-87, 10.1145/1941487.1941508

[2] Agrawal and Tapaswi, 2017N. Agrawal, S. TapaswiThe performance analysis of honeypot based intrusion detection system for wireless networkInt. J. Wirel. Inf. Netw., 24 (1) (2017), pp. 14-26, 10.1007/s10776-016-0330-3

[3] Al-Duwairi and Govindarasu, 2006B. Al-Duwairi, M. GovindarasuNovel hybrid schemes employing packet marking and logging for IP tracebackIEEE T. Parall. Distr., 17 (5) (2006), pp. 403-418, 10.1109/TPDS.2006.63

[4] Alhawi et al., 2018O.M.K. Alhawi, J. Baldwin, A. DehghantanhaLeveraging machine learning techniques for Windows ransomware network traffic detection

[5] Alshammari and Zincir-Heywood, 2015R. Alshammari, A.N. Zincir- HeywoodIdentification of VoIP encrypted traffic using a machine learning approach

[6] J. King Saud Univ. Comput. Inf. Sci., 27 (1) (2015), pp. 77-92, 10.1016/j.jksuci.2014.03.013

[7] Alsmadi et al., 2018I. Alsmadi, R. Burdwell, A. Aleroud, A. Wahbeh, M. Al-Qudah, A. Al-OmariNetwork forensics: lesson plans

[8] Practical Information Security: A Competency- Based Education Course, Springer, Cham (2018), pp. 245-282, 10.1007/978-3-319-72119-4_11

[9] J.R. Vacca (Ed.), Computer and Information Security Handbook (third ed), Morgan Kaufmann, Cambridge, MA, USA (2017), 10.1016/B978-0-12-803843-7.00062-4

[10] Salim et al., 2019M.M. Salim, S. Rathore, J.H. ParkDistributed denial of service attacks and its defenses in IoT: a surveyJ. Supercomput. (2019), 10.1007/s11227-019-02945-z

[11] Sanders, C., 2017. Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems. No Starch Press, San Francisco.

[12] Savage, S., Wetherall, D., Karlin, A., Anderson, T., 2001. Network support for IP traceback. IEEE ACM Trans. Netw. 9 (3), 226-237.

[13] H. Ralph, J. Sprague (Eds.), Proceedings of the 40th Annual Hawaii International Conference on System Sciences, IEEE Computer Society, Los Alamitos, CA, USA (2007), 10.1109/HICSS.2007.617

[14] G. Peterson, S. Shenoi (Eds.), Advances in Digital Forensics XIV, Springer, Cham (2018), pp. 183-197, 10.1007/978-3-319-99277-8_11

[15] P. Biljanovic, Z. Butkovic, K. Skala, B. Mikac, M.

Cicin-Sain, V. Sruk, S. Ribaric, S. Gros, B. Vrdoljak, M. Mauher, A. Sokolic (Eds.), 38th International Convention on Information and Communication Technology, Electronics and Microelectronics, IEEE (2015), pp. 1338-1343, 10.1109/MIPRO.2015.7160482 Jamalipour, D.-J. Deng (Eds.),

[16] Xiang et al., 2008Y. Xiang, W. Zhou, M. GuoFlexible deterministic packet marking: an IP traceback system to find the real source of attacksIEEE T. Parall. Distr., 20 (4) (2008), pp. 567-580, 10.1109/TPDS.2008.132

[17] Yang et al., 2018J. Yang, Y. Zhang, R. King, T. TolbertSniffing and chaffing network traffic in stepping-stone intrusion detection

[18] L. Barolli, M. Takizawa, T. Enokido, M.R. Ogiela, L. Ogiela, N. Javaid (Eds.), 32nd International Conference on Advanced Information Networking and Applications Workshops, IEEE Computer Society, Los Alamitos, CA, USA (2018), pp. 515-520, 10.1109/WAINA.2018.00137

[19] Yin et al., 2018C. Yin, H. Wang, J. WangNetwork data stream classification by deep packet inspection and machine learning