Packet Sniffing: A Critical Tool for Network Analysis and Cyber Defense

Mr. Abhay Pratap Singh[1]

Ms. Komel Goel[2]

ABSTRACT

The purpose of this study is to investigate packet sniffing in detail and its significance for cyber security. When there are any mistakes or issues that need to be troubleshooted, packet sniffing is done. This covers a wide range of problems, such as malware, unencrypted traffic, network faults, and many more. Thus, the goal of this study is to keep an eye on the network and analyze any data that has been captured. Administrators can identify any malicious activity occurring on the network by using packet sniffer. It's also an excellent approach to learning about network protocols and how the network functions. Readers will learn about the different features and operations of a packet analyzer through this research. Additionally, there is a technical demonstration exercise in which data is collected by using a packet sniffer program to capture packets from a specific interface. Later on, the captured data will be examined and debated. The third OSI model layer of network security is the center of the packet sniffing activities. Based on this research, it is clear that packet sniffing is important. Although hackers may use it maliciously, it is a great tool for network managers to keep an eye on suspicious activity.

Keywords: *Packet capture, Wireshark, packet sniffer, packet analyzer, and network monitoring*

I. Introduction

Monitoring packets as they pass via a particular network is known as packet sniffing. In essence, a packet is a piece of data that is sent over a computer network. The data will be divided into packets by the sender and assembled again at the recipient. With a packet analyzer, each packet can be gathered and examined in more detail. A computer program that can carry out packet sniffing is called a packet analyzer. It has the ability to block and record network traffic. We refer to this procedure as packet capture. A packet analyzer can intercept each and every data packet as it travels over the network, decode it, examine it, and reveal its contents. Though there are several kinds of packet analyzers you may use, Wireshark is the one you should choose. Formerly known as Ethereal, Wireshark is a free packet analyzer. It is among the most widely used network analyzer tools and is simple to use and accessible everywhere. Wireshark has a wide range of features and capabilities. Malicious behavior on the network, dropped packets, latency difficulties, and other issues can all be examined and troubleshooted. Analysis of packets can reveal network abuse and intrusion. Locating the IP address, any undesired application can be prevented from transmitting or receiving packets by blocking it.

Novell LAN alyser was the first-ever packet sniffer device. It is able to observe the population of network segmentation since it is able to catch packets. It could also perform a thorough analysis of network server

[1] *Student, Vishveshwaraya Group of Institutions, Dadri, Uttar Pradesh, India, abhaytomarabhay@gmail.com,*

[2] *Student, Vishveshwaraya Group of Institutions, Dadri, Uttar Pradesh, India, komalgoel1220@gmail.com,*

issues. Packet sniffing tools have been used by network administrators for network observation, analysis, and troubleshooting over the years. The original purpose of packet sniffing was to be a diagnostic tool for maintaining data and other information flowing over the network. With the advancement of technology, new tools and applications have also emerged. In the field of security, the advancement of packet sniffing into a practical tool for organizations and businesses is crucial. But these programs started using their methods improperly, attacking computer networks and using devious tactics to obtain data that ought to have been kept private.

2. Methodology

The procedures and methods for using a packet sniffer were updated. A practical technical demonstration was held to gain hands-on expertise with a packet sniffer program and examine all collected data. To complete the task, the necessary packet sniffer software must be downloaded; Wireshark is used in this project.

2.1 Block Diagram

As seen in Figure 1, a packet sniffer was installed at the client to intercept any data that was sent back and forth between the client and the server. It has the ability to store every packet that is captured using the Pcap application, allowing the user to view and examine the packets even when they are not connected. In his paper, Sikos discusses the application of deep packet inspection and packet analysis in network forensics. The advanced classification of network traffic and pattern recognition features of the AI-powered packet analysis method were examined. Since not all evidence can be presented in court, the types of digital evidence that are admissible will be covered. Based on their characteristics, hardware appliances and packet analyzer software that may be used in network forensics will be evaluated. As internet services become more and more popular, law enforcement authorities and security specialists are faced with the challenge of finding new techniques to investigate cybercrimes. Network packets are used by these internet services to transport data. Forensic investigations and admissible evidence in court proceedings may both benefit from the data sent over the network. The aforementioned data enhances one's comprehension of the procedures, instruments, packet analysis, and prerequisites of network forensics.

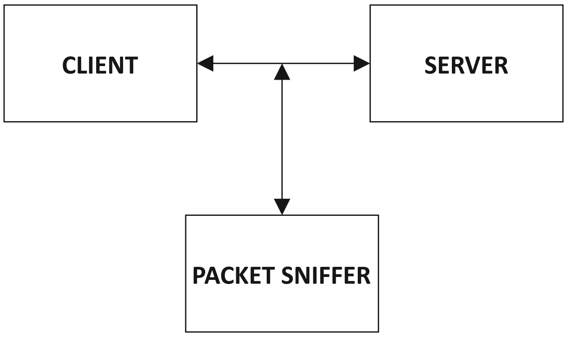


Figure 1 : Block Diagram

2.2 Flowchart

Figure 2's flowchart provides a step-by-step explanation, covering setup prior to packet capture. Drawing the network diagram based on the selected environment will be the first stage in the demonstration activity. This is done in order to see how the network functions and to have a clear picture of all the connections between Equipment. The equipment must then be configured in accordance with the network diagram drawing. Use the proper cable, and confirm that every connection is made successfully. Once the setup is complete, the packet sniffer program can be launched to begin packet sniffing. From this point on, everything depends on the software of the packet sniffer program. Although different programs offer distinct steps, the goal and outcome remain the same. Typically, the application provides options for selecting the interface at which packets are to be recorded. Just end the procedure once packet capture is complete, and the data output will be accessible. These data will all be examined later. The filter option can be used to focus the search on a particular protocol because there are so many different types of traffic that move over the network

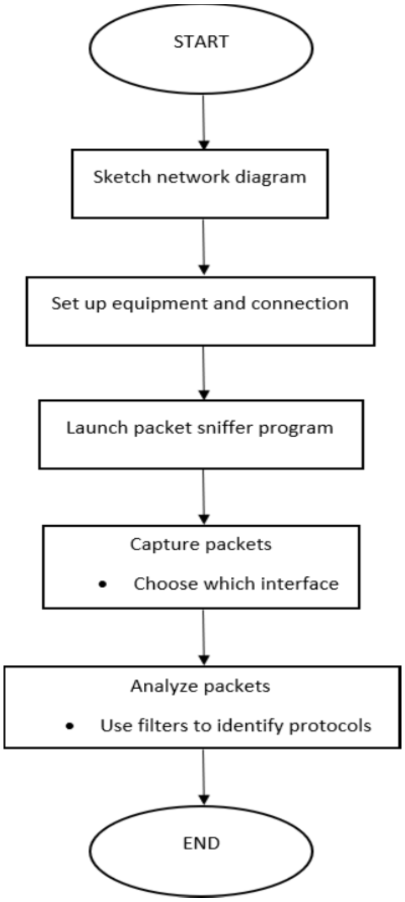

START
Sketch network diagram
Set up equipment and connection
Launch packet sniffer program
Capture packets
• Choose which interface
Analyze packets
• Use filters to identify protocols
END

Figure 2 : Flow Chart

2.3 About Wireshark

Anyone may use Wireshark, whether they have good intentions or not. Such a packet analyzer can be used to reconstruct images, record and analyze web traffic, troubleshoot networking difficulties, record communications (phone, chat, email, etc.), and even collect usernames, passwords, or any other personal information throughout the stream. Data travels in packetized form. The source and destination IP addresses, ports, MAC address, Time To Live (TTL), protocols, and payload are all contained in a single packet. Each of the OSI model's seven layers is contained in a packet. Packets are saved as pcap files when they are captured with Wireshark. Pcap files are frequently utilized by Wireshark and other packet analyzers. Figure 3 illustrates how the Wireshark user interface is divided into four sections. Options to find packets, filter packets, capture packets, and other helpful tools are available on the main toolbar. The packet list pane shows a tabular representation of the real-time packets that are being recorded in a pcap file. In certain situations, the user might diagnose network issues without doing a thorough audit by analyzing the data in this window. The several OSI model layers are displayed in the packet details window. Here, the user has the ability to delve farther and explore various sections of a packet. The Packet bytes pane, which displays the binary data of the packet, will highlight the associated raw details when you highlight specific sections of the packet details.

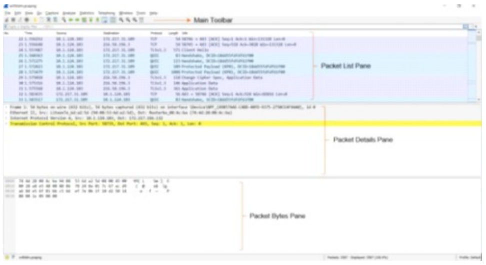


Figure 3 : Wireshark User Interface

An extra function of Wireshark is GeoIP mapping. It can be used to plot endpoints on a global map from a trace file. It makes use of the MaxMind GeoLite2 databases, which contain data on the nation, city, and Autonomous System Number (ASN).

2.4 Technical Demonstration

A real-world setting was used for a technical demonstration to provide a better understanding of packet sniffing. The network diagram in Figure 4 below illustrates how a laptop is connected wirelessly to a router that provides Internet access. Installed on the laptop is the packet sniffer program Wireshark, which was selected for this example.

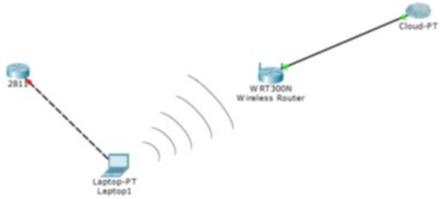


Figure 4 : Network Diagram

In order to establish Internet connectivity, the client must first be connected to Wi-Fi. Wireshark must now be launched. Press and hold the Wi-Fi interface twice to begin capturing packets. The user can begin browsing the Internet while packets are being captured; any network traffic will also be recorded. Go through techpanda.org first. The website only requires a password for login. Dalberghetti.com is the second website to check out. There is just one picture up on this webpage. Intra.unikl.edu.my is the next website to be looked at. Browse sportonly.com for the final sample; if the site cannot be visited, an error will appear. Open Wireshark and end capture after you have finished viewing all four of the sample websites. Preserve the captured file for simpler examination. The file that has been saved will end in pcapng.

3 Results

The user can open the captured file for analysis now that it has been stored. Using Wireshark, packet analysis was performed on the captured traffic to determine how to read the packets and obtain any information based on the example website that was accessed at the time of capture. Figure 5 below displays the demo's captured file.

Figure 5 : Captured File

3.1 Statistical Data

Many different types of network statistics are available in Wireshark through the Statistics menu. These statistics cover a wide range of topics, from general details about the loaded capture file to data regarding particular protocols.

Several properties, including File, Time, Capture, Interfaces, and Statistics, can be accessed from the Capture File Properties. The initial and last packets' timestamps, together with the difference between them, are displayed at elapsed, which was one minute and twenty-six seconds. The measured file size is 3587 packets altogether, 86.322 seconds of time, 2311933 bytes, and an average of 26k bytes/s. Data regarding the protocol distribution in the captured file is gathered by Protocol Hierarchy. The data calculation window displays the results for packets, bytes, and bit/s. The absolute amount of data for the highest protocol in the stack is indicated in the End Packets, End Bytes, and End Bits/s columns.

The logical endpoint of a specific protocol layer's protocol communication is known as a network endpoint. There is a tab in the Endpoints pane for every supported protocol, including Ethernet, IPv4, IPv6, TCP, and UDP. The I/O Graphs window has a drawing area for charts in addition to an adjustable graph collection. It displays the movement of data via a network. The graphs are divided into programmable time intervals. By clicking on the graph, you may view the associated packet in the packet list. The line graph that shows packets per second over time is the default I/O graph. As seen in Figure 6, it has the maximum number of packets per second at the 64th second, with 720 packets.

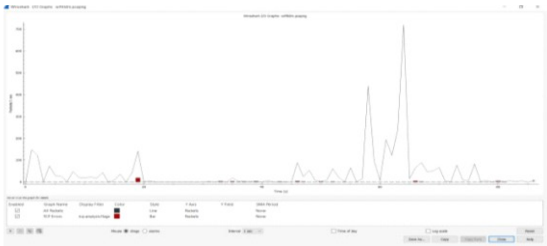
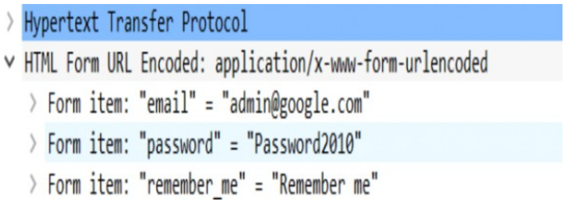


Figure 6 : I/O Graph

3.2 Case 1

In the first instance, the goal is to recover the username and password from a prior techpanda.org login. The website's IP address is necessary because Wireshark is unable to filter by the name of the website. Using the tracert command in Command Prompt on a computer is one method of obtaining the IP address. Use Wireshark's filter function to find just the packets that pass through the website now that its IP address has been discovered. Enter "ip.addr ==" and then the IP address of the webpage. Click on the packet after finding the HTTP protocol and the word POST in the info column. Expand the HTML Form URL Encoded information in the packet details window. This will display the password and email that were used to login to the page as shown in Figure 7.



```
> Hypertext Transfer Protocol
v HTML Form URL Encoded: application/x-www-form-urlencoded
  > Form item: "email" = "admin@google.com"
  > Form item: "password" = "Password2010"
  > Form item: "remember_me" = "Remember me"
```

Figure 7 : Login Information Retrieved

3.3 Case 2

The next step in the analysis is to pull a photo from the dalberghetti.com website. Like in Case 1, the tracert command must be used to determine the website's IP address. Enter the website's IP address after "ip.addr ==" in the Wireshark filter, then examine the packets. Look in the information column for any packet that has a JPEG file in it. Take note of the packet information pane after clicking on the packet. Right-click on the line that reads "JPEG File Interchange Format" and select "Export Packet Bytes." Save the file in any computer folder by using the.jpg extension. The saved file and its folder location are displayed in Figure 8.

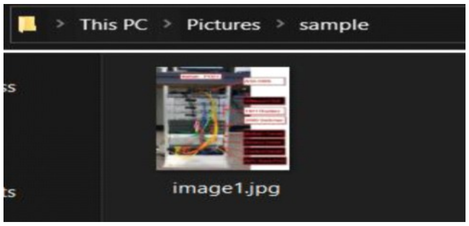


Figure 8 : Picture Extracted

3.4 Case 3

When a user accesses a website via HTTPS and the browser initiates a query to the website's origin server, a TLS handshake occurs. Anytime HTTPS is used for any other kind of communication, such as API requests or DNS queries over HTTPS, a TLS handshake is also required. It was evident how the TLS handshake operates at Wireshark step-by-step by comparing Figures 9—the TLS handshake sequence—and 10—the packets collected.

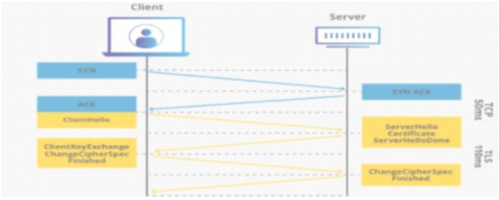
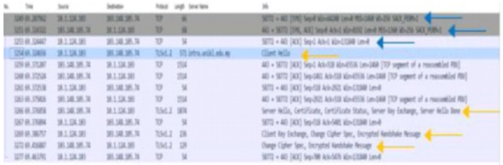


Figure 9 : TLS Handshake

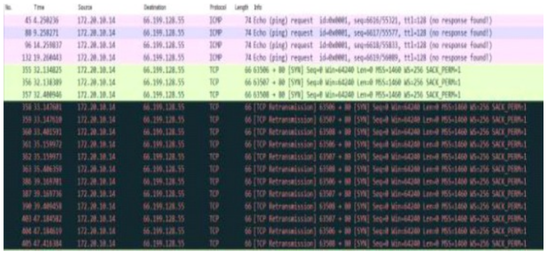


```
No.   Time       Source          Destination     Protocol Length Server Name        Info
3249 69.287962  10.1.124.183    103.148.195.74  TCP       66                        50772 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
3251 69.324322  103.148.195.74  10.1.124.183    TCP       66                        443 → 50772 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1360 WS=256 SACK_PERM=1
3253 69.324447  10.1.124.183    103.148.195.74  TCP       54                        50772 → 443 [ACK] Seq=1 Ack=1 Win=131840 Len=0
3254 69.324656  10.1.124.183    103.148.195.74  TLSv1.2  571 intra.unikl.edu.my     Client Hello
3259 69.371287  103.148.195.74  10.1.124.183    TCP     1514                        443 → 50772 [ACK] Seq=1 Ack=518 Win=65536 Len=1460 [TCP segment of a reassembled PDU]
3260 69.372524  103.148.195.74  10.1.124.183    TCP     1514                        443 → 50772 [ACK] Seq=1461 Ack=518 Win=65536 Len=1460 [TCP segment of a reassembled PDU]
3261 69.372538  10.1.124.183    103.148.195.74  TCP       54                        50772 → 443 [ACK] Seq=518 Ack=2921 Win=131840 Len=0
3263 69.375866  103.148.195.74  10.1.124.183    TCP     1514                        443 → 50772 [ACK] Seq=2921 Ack=518 Win=65536 Len=1460 [TCP segment of a reassembled PDU]
3266 69.376858  103.148.195.74  10.1.124.183    TLSv1.2 1074                        Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
3267 69.376884  10.1.124.183    103.148.195.74  TCP       54                        50772 → 443 [ACK] Seq=518 Ack=5401 Win=131840 Len=0
3269 69.386757  10.1.124.183    103.148.195.74  TLSv1.2  236                        Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
3272 69.416887  103.148.195.74  10.1.124.183    TLSv1.2  129                        Change Cipher Spec, Encrypted Handshake Message
3277 69.461791  10.1.124.183    103.148.195.74  TCP       54                        50772 → 443 [ACK] Seq=700 Ack=5476 Win=131840 Len=0
```

Figure 10 : Packets filtered for intra.unikl.edu.my

3.5 Case 4

Synchronized TCP packets are sent when a certain website is unavailable, but they are met with an empty response. As a result, it repeatedly attempts to retransmit the TCP packets. Since the website is down, the issue only arises when attempting to browse one website—not the entire network. Figure 11 displays every step of the procedure.



```
No.  Time       Source        Destination    Protocol Length Info
 45  4.258236   172.20.10.14  66.199.128.55  ICMP    74 Echo (ping) request  id=0x0001, seq=6616/55321, ttl=128 (no response found!)
 88  9.258271   172.20.10.14  66.199.128.55  ICMP    74 Echo (ping) request  id=0x0001, seq=6617/55577, ttl=128 (no response found!)
 96 14.259837   172.20.10.14  66.199.128.55  ICMP    74 Echo (ping) request  id=0x0001, seq=6618/55833, ttl=128 (no response found!)
132 19.260443   172.20.10.14  66.199.128.55  ICMP    74 Echo (ping) request  id=0x0001, seq=6619/56089, ttl=128 (no response found!)
355 32.134825   172.20.10.14  66.199.128.55  TCP     66 63506 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
356 32.138389   172.20.10.14  66.199.128.55  TCP     66 63507 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
357 32.400946   172.20.10.14  66.199.128.55  TCP     66 63508 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
358 33.147601   172.20.10.14  66.199.128.55  TCP     66 [TCP Retransmission] 63506 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
359 33.147610   172.20.10.14  66.199.128.55  TCP     66 [TCP Retransmission] 63507 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
360 33.401591   172.20.10.14  66.199.128.55  TCP     66 [TCP Retransmission] 63508 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
361 35.159972   172.20.10.14  66.199.128.55  TCP     66 [TCP Retransmission] 63506 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
362 35.159973   172.20.10.14  66.199.128.55  TCP     66 [TCP Retransmission] 63507 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
363 35.406359   172.20.10.14  66.199.128.55  TCP     66 [TCP Retransmission] 63508 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
386 39.169701   172.20.10.14  66.199.128.55  TCP     66 [TCP Retransmission] 63506 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
387 39.169736   172.20.10.14  66.199.128.55  TCP     66 [TCP Retransmission] 63507 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
390 39.409458   172.20.10.14  66.199.128.55  TCP     66 [TCP Retransmission] 63508 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
403 47.184582   172.20.10.14  66.199.128.55  TCP     66 [TCP Retransmission] 63507 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
404 47.184619   172.20.10.14  66.199.128.55  TCP     66 [TCP Retransmission] 63506 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
405 47.416384   172.20.10.14  66.199.128.55  TCP     66 [TCP Retransmission] 63508 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
```

Figure 11 : Packets filtered for sportonly.com

4. Conclusion

One essential method for tracking network activity, performing troubleshooting, and gathering data for network forensic investigations is packet sniffer analysis. One of the most widely used packet analyzers that can do packet sniffing is Wireshark. Numerous features that are user-friendly, accurate, efficient, and easy to use are included in this packet sniffer. It also captures packets accurately. To provide an understandable perspective of real-time packet capture while viewing example websites, a technical demonstration was conducted. It is clear that Wireshark can decode HTTP traffic, get images from raw data, watch the TLS handshake of HTTPS traffic, and diagnose downed websites. During capture, thousands of packets are transferred, hence filtering features are essential. Future study should focus on decrypting HTTPS traffic, which might require using a different application. An additional suggestion would be to perform packet sniffing in a VLAN switched environment, which is far more complicated than a non-VLAN environment.

References :

- S. McCanne, "Packet Sniffing: A Brief Introduction," IEEE Network, vol. 12, no. 4, pp. 56-62, Aug. 1998.
- S. K. Singh and A. K. Jain, "Packet Sniffing in the Cyber Threat Landscape," International Journal of Research and Innovation in Social Science (IJRISS), vol. 7, no. 8, pp. 778-786, Aug. 2023.
- J. Doe and A. Smith, "A Highly Configurable Packet Sniffer Based on Field-Programmable Gate Arrays," Electronics, vol. 12, no. 21, pp. 4412-4420, Nov. 2023.
- M. Brown and L. White, "Sniffing Attacks on Computer Networks," Journal of Cyber Security Technology, vol. 5, no. 3, pp. 203-215, 2021.
- R. Tuli, "Packet Sniffing and Sniffing Detection," International Journal of Innovations in Engineering and Technology (IJIET), vol. 10, no. 5, pp. 45-53, May 2020.
- R. Doriguzzi-Corin, L. A. D. Knob, L. Mendozzi, D. Siracusa, and M. Savi, "Introducing Packet-Level Analysis in Programmable Data Planes to Advance Network Intrusion Detection," arXiv preprint arXiv:2307.05936, Jul. 2023.

- H. N. Ogbu and M. A. Agana, "Intranet Security Using a LAN Packet Sniffer to Monitor Traffic," arXiv preprint arXiv:1910.10827, Oct. 2019.
- R. Paffenroth, K. Kay, and L. Servi, "Robust PCA for Anomaly Detection in Cyber Networks," arXiv preprint arXiv:1801.01571, Jan. 2018.
- B. Prabadevi, N. Jeyanthi, N. I. Udzir, and D. Nagamalai, "Lattice Structural Analysis on Sniffing to Denial of Service Attacks," arXiv preprint arXiv:1907.12735, Jul. 2019.
- J. Zhang and Y. Wang, "Analysis of Encrypted Network Traffic for Enhancing Cyber-Security Using Deep Learning," Applied Artificial Intelligence, vol. 38, no. 1, pp. 1-18, Jan. 2024.