6th International Scientific Conference ITEMA 2022 – Proceedings
DOI: https://doi.org/10.31410/ITEMA.2022.105

An Approach to Improving Network Security Using Log Analysis

Marija Zajeganović[1]
Milan Pavlović[2]
Danica Mamula Tartalja[3]
Silva Kostić[4]

Received: November 21, 2022
Accepted: January 31, 2023
Published: June 12, 2023

Keywords:
Log messages;
Troubleshooting;
Malicious activities.

Abstract: *Troubleshooting is the process of detecting, identifying and resolving problems within a computer network by means of specific methods, tools and operations. Troubleshooting implies following a set of procedures or steps that conform to the security standards and policies of a company. Diagnosing the source of a problem can be done by tools for system monitoring, recording log messages, manual testing of device configuration, as well as by tools for device operation analysis. The procedure for using log messages to resolve both common problems and those caused by attacks is explained in this paper. Furthermore, this paper describes the way security threat management systems use the contents of log messages to analyze hardware problems and malicious activities.*

1. INTRODUCTION

It is almost impossible to imagine any segment of our lives without computer networks. There are different realizations of a properly designed and configured computer network. Generally, all networks can be defined by their hardware, software and network protocols which enable both proper communication in a computer network and the use of numerous applications. Since the appearance of the first computer networks, the way of communication and the functioning of the networks themselves have been changed and adapted several times to current trends, i.e. demands and standards of large corporations. In the beginning, there were several manufacturers of hardware equipment for computer networks and each of them developed its standards and protocols for communication. Over time, all components of computer networks, protocols and hardware have been significantly improved. Today, there is a large number of new network devices, routers and switches of newer generations, firewall devices, WiFi routers, etc. (Panek, 2020)

Once a computer network is set up, there are numerous challenges and issues that IT staff can encounter. With the realization of modern complex computer networks, the risk of network problems increases, either as a result of bad connectors and/or damaged cables at the physical level, wrong device configuration or as a result of threats and attacks on the computer network (Simpson et al., 2011).

[1] The Academy of Technical and Art Applied Studies Belgrade, Department School of Information and Communication Technologies, Zdravka Čelara 16, Belgrade, 11000, Republic of Serbia

[2] The Academy of Technical and Art Applied Studies Belgrade, Department School of Information and Communication Technologies, Zdravka Čelara 16, Belgrade, 11000, Republic of Serbia

[3] The Academy of Technical and Art Applied Studies Belgrade, Department School of Information and Communication Technologies, Zdravka Čelara 16, Belgrade, 11000, Republic of Serbia

[4] The Academy of Technical and Art Applied Studies Belgrade, Department School of Information and Communication Technologies, Zdravka Čelara 16, Belgrade, 11000, Republic of Serbia

Challenges and problems in a computer network can be divided into those that occur due to various situations that do not depend on people:

a. Power cut - unplanned works, power supply failure, a blown fuse;
b. Natural disasters - fires and earthquakes;
c. Device overload – too high level of received voltage;
d. Component failure – hardware components with a manufacturing defect.

On the other hand, computer network problems can be caused by human error:

a. Configuration errors – misconfiguration, wrong IP address, deleted configurations;
b. Targeted attacks and network failures - Zero-day, viruses and trojans, phishing, DDoS;
c. Poor planning and improper project implementation - poor assessment of requirements and inadequate analysis of the necessary resources for flawless network operation.

The consequences can be almost imperceptible, with minimal packet loss, but in some cases, the mere functioning of a computer network can be affected. In order to detect errors or in the network as well as potential attacks, it is necessary to continuously analyze network traffic by observing the content of log messages (Ranjbar, 2015). Quick and accurate detection of problems implies taking adequate countermeasures to eliminate them. The main steps in the problem solving procedure will be explained in this paper.

2. COMPUTER NETWORK TROUBLESHOOTING PROCEDURE

Troubleshooting is a process of solving problems. When IT staff troubleshoot problems, they use specific methods and operations to detect the cause of a problem and eliminate it. Troubleshooting consists of a series of steps and procedures in accordance with the standards and policies of a company or an organization (Ranjbar, 2015). It includes certain tools for system monitoring, log messages, testing of devices and system and analyzing. There are several steps in a troubleshooting process, as shown in Figure 1:

1. First, it is necessary to diagnose a problem after the monitoring system notifies a problem on specific devices or in the application;
2. Then, relevant information about the problem should be gathered. It is important to detect which device is affected, if there are one or more affected devices, when the problem occurred, if the problem occurred for the first time, etc.;
3. After that, the cause of the problem should be identified and a plan to solve the problem defined;
4. Next, solutions should be implemented and processes repeated and changed if necessary;
5. Finally, the entire troubleshooting process should be documented.

The entire communication between two devices is realized thanks to a large number of protocols that are defined within the layers of the OSI model, i.e. the TCP/IP stack of protocols (Ranjbar, 2015). Errors and problems that arise in the computer network are also observed through these reference models. Cabling problems often occur at the physical layer. It is not unusual to use inadequate types or damaged cables. The connectors on the cables themselves may be poorly connected. Hardware problems might occur as well.

Components can fail due to poor quality, maintenance, or long-term load. At the connection layer, the second layer of the OSI model, the most common problems are: mismatched speed and/or duplex settings on the switch port itself, incorrect Spanning Tree Protocol settings, problems related to ARP attacks and MAC Spoofing or DHCP Spoofing (Ranjbar, 2015). At the network layer, potential problems could be duplicate IP addresses and misrouting due to misconfiguration of routing protocols or implementation of static routes. At the transport layer where TCP/UDP ports are used, there are frequent problems related to disabling the transfer of certain data due to ACL lists or activated policies on the firewall. At the other layers (session, presentation and application layer), most problems are connected with the applications or specific protocols. Since there is a large number of protocols at OSI or TCP/IP layers, this can be the cause of many problems. Server configuration problems (Apache, Nginx...) might occur as well.

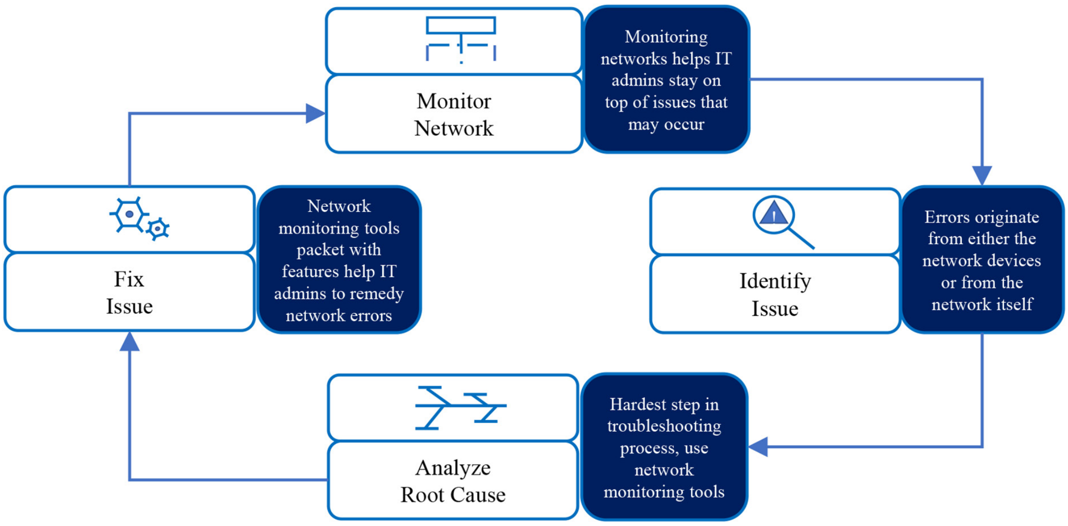

Monitor Network
Monitoring networks helps IT admins stay on top of issues that may occur
Identify Issue
Errors originate from either the network devices or from the network itself
Analyze Root Cause
Hardest step in troubleshooting process, use network monitoring tools
Fix Issue
Network monitoring tools packet with features help IT admins to remedy network errors

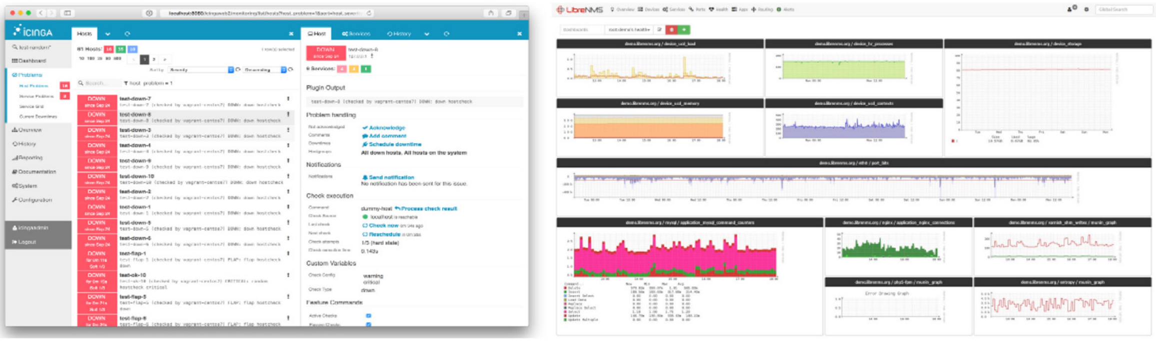
Figure 1. The main steps in network monitoring and troubleshooting

3. SYSTEM MONITORING AND LOG MESSAGES

Collecting information about the overall operation of the computer network in the form of log messages is one of the most important steps in the entire process of solving problems and locating errors. All available information about the cause of the problem and its consequences is collected. The obtained information is then analyzed. Based on the analyzed data, further preparation for the following steps is carried out in order to eliminate the located problems. The accuracy of information gathered from log messages is of crucial importance (Jing et al., 2015).

The network monitoring step can include many sub-steps, tools, systems or software, and the ways and methods of gathering information depend on the environment. There are several ways to gather relevant information about a network's behavior. It might be through a monitoring system – which consists of one or more programs that perform different operations like monitoring, information gathering, notifying administrators, etc.

The information can refer to the use of the server memory and load of its processor, device availability or problems with TLS certificates. Such information is most often "extracted" from the devices themselves, mostly using the SNMP protocol. With the SNMP protocol, some other basic and more advanced information can also be obtained, such as the device name, IP address, VRF, QoS, port information on the switch, etc. Examples of such monitoring systems are IcingaWeb2, LibreNMS and Grafana. Then, relevant information about a network behavior might be gathered via data collection agents – i.e. software/applications installed on end devices that collect and forward log messages to the central log message management system.



A) B)

Figure 2. Display of the interface of different monitoring tools: A) Web interface of the IcingaWeb2 monitoring tool (source: https://icinga.com); B) Web interface of the LibreNMS monitoring tool (source: https://alternativeto.net/software/librenms)

Agents are configured to check specific log files (which are often in plain text document format) and forward them to the central system. It is possible to indicate exactly which type of log messages needs to be collected. It is not necessary to collect all log messages since they can occupy additional memory space. Information can also be gathered by log message management systems – the central system where log messages are collected, normalized and analyzed.

Log messages are special texts recorded and stored in log files. Log analysis is a process of analyzing and extracting information from log messages. There are several types of log messages depending on the type of event being recorded: system, network, technical or security ones (Vaarandi et al., 2018). Optimization of log messages is considered to be the essential procedure since it is necessary to know which type of log messages will be collected, monitored and analyzed. A piece of the log information is forwarded to centralized log systems directly or through various log message transmission mechanisms, while other information is obtained from the operating system as part of the overall log message generation process. Log messages are written in log files and these files can be generated by the system or applications, or generated by the administrator (Vaarandi et al., 2018). By default, log messages use UDP port 514 for communication, while the default port for TCP protocol is 6514. It is possible to use other ports for the transmission of log messages. Log analysis takes place during the process of searching log messages and looking for specific information, but it also takes place when examining a specific incident that occurred due to a network failure.

Log messages can be divided into the following main groups: System log messages (Activities in the system, End devices, Applications), Network log messages (Email, Firewall, VPN), Technical log messages (Proxy, DNS, HTTP, DHCP, Web and SQL), Security log messages (Antivirus, Network intrusion detection system – NIDS, Host intrusion detection system - HIDS, Data loss protection - DLP). Additionally, log messages may be divided according to the severity level indicated by the content of a log message. A log message can be informative as well as displayed as a warning or an error in an application or a system. The severity level of a log message ranges from 0 to 7 with the lower the level, the greater the severity (Suh-Lee et al., 2016).

4. CENTRALIZED SYSTEMS FOR MONITORING LOG MESSAGES

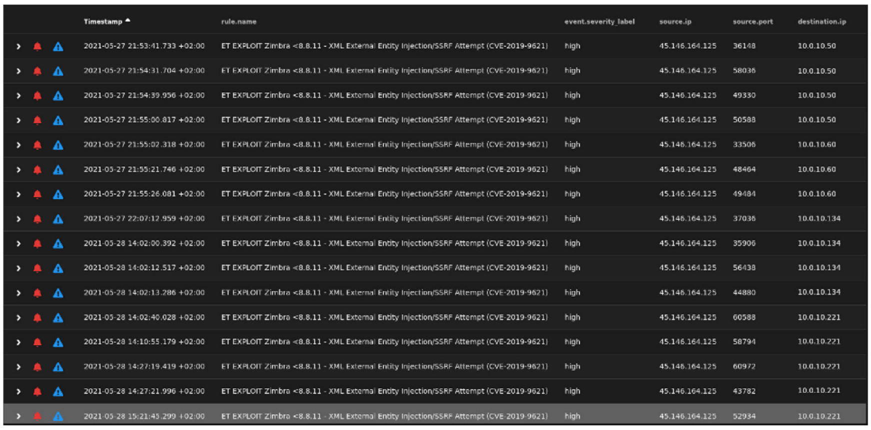
The system for monitoring and analyzing events in the network consists of agents for collecting log messages and the system for managing log messages, as explained above.

Security Information and Event Management (SIEM) is a system for managing security information and events. SIEM represents a centralized system for collecting regular and security log messages. It provides different types of notifications and warnings, uses special algorithms (AI and ML) for analyzing data and has possibilities for isolating malicious packets and analyzing packet capture files. This system processes data in real time and data can include almost any type of log messages, whether it is data from routers, switches, mobile phones, or applications (Bhatt, 2014).

The features of a SIEM system are identical to a log message management system, except that they mainly focus on security and (mostly) malicious activities. Activities can range from simple ping packets coming from the Internet to compromised services such as FTP, web servers, third-party applications, etc. In the SIEM architecture, there are agents for collecting log messages which send captured messages to the main center for processing and analysis. Further analysis is performed, data are compared, their correlation is performed, data groups are formed and activity alerts are created based on the extracted data. Different SIEM solutions depend primarily on the manufacturer, but all of those solutions include the following functions: data aggregation, threat intelligence, correlation and security monitoring, alert analytics, forensics, threat detection, incident response and Security Operations Center (SOC) automation (Bhatt, 2014). Every security engineer in a Security Analysis Center (SOC) should have insight into network activity, the way of communication between devices and its duration and the type of exchanged data, and know the standard procedure if any problems occur.

5. CASE STUDY

In this chapter, an example of potential threats and malicious activities in the network will be presented using the methods of analyzing and collecting data through the Security Onion platform using SIEM. Specifically, malicious activity in Security Onion was analyzed when an alarm „`ET EXPLOIT Zimbra <8.8.11 - XML External Entity Injection/SSRF Attempt (CVE-2019-9621`” with the severity level *HIGH* was detected.



			Timestamp ▲	rule.name	event.severity_label	source.ip	source.port	destination.ip
›	🔔	⚠	2021-05-27 21:53:41.733 +02:00	ET EXPLOIT Zimbra <8.8.11 - XML External Entity Injection/SSRF Attempt (CVE-2019-9621)	high	45.146.164.125	36148	10.0.10.50
›	🔔	⚠	2021-05-27 21:54:31.704 +02:00	ET EXPLOIT Zimbra <8.8.11 - XML External Entity Injection/SSRF Attempt (CVE-2019-9621)	high	45.146.164.125	58036	10.0.10.50
›	🔔	⚠	2021-05-27 21:54:39.956 +02:00	ET EXPLOIT Zimbra <8.8.11 - XML External Entity Injection/SSRF Attempt (CVE-2019-9621)	high	45.146.164.125	49330	10.0.10.50
›	🔔	⚠	2021-05-27 21:55:00.817 +02:00	ET EXPLOIT Zimbra <8.8.11 - XML External Entity Injection/SSRF Attempt (CVE-2019-9621)	high	45.146.164.125	50588	10.0.10.50
›	🔔	⚠	2021-05-27 21:55:02.318 +02:00	ET EXPLOIT Zimbra <8.8.11 - XML External Entity Injection/SSRF Attempt (CVE-2019-9621)	high	45.146.164.125	33506	10.0.10.60
›	🔔	⚠	2021-05-27 21:55:21.746 +02:00	ET EXPLOIT Zimbra <8.8.11 - XML External Entity Injection/SSRF Attempt (CVE-2019-9621)	high	45.146.164.125	48464	10.0.10.60
›	🔔	⚠	2021-05-27 21:55:26.081 +02:00	ET EXPLOIT Zimbra <8.8.11 - XML External Entity Injection/SSRF Attempt (CVE-2019-9621)	high	45.146.164.125	49484	10.0.10.60
›	🔔	⚠	2021-05-27 22:07:12.959 +02:00	ET EXPLOIT Zimbra <8.8.11 - XML External Entity Injection/SSRF Attempt (CVE-2019-9621)	high	45.146.164.125	37036	10.0.10.134
›	🔔	⚠	2021-05-28 14:02:00.392 +02:00	ET EXPLOIT Zimbra <8.8.11 - XML External Entity Injection/SSRF Attempt (CVE-2019-9621)	high	45.146.164.125	35906	10.0.10.134
›	🔔	⚠	2021-05-28 14:02:12.517 +02:00	ET EXPLOIT Zimbra <8.8.11 - XML External Entity Injection/SSRF Attempt (CVE-2019-9621)	high	45.146.164.125	56438	10.0.10.134
›	🔔	⚠	2021-05-28 14:02:13.286 +02:00	ET EXPLOIT Zimbra <8.8.11 - XML External Entity Injection/SSRF Attempt (CVE-2019-9621)	high	45.146.164.125	44880	10.0.10.134
›	🔔	⚠	2021-05-28 14:02:40.028 +02:00	ET EXPLOIT Zimbra <8.8.11 - XML External Entity Injection/SSRF Attempt (CVE-2019-9621)	high	45.146.164.125	60588	10.0.10.221
›	🔔	⚠	2021-05-28 14:10:55.179 +02:00	ET EXPLOIT Zimbra <8.8.11 - XML External Entity Injection/SSRF Attempt (CVE-2019-9621)	high	45.146.164.125	58794	10.0.10.221
›	🔔	⚠	2021-05-28 14:27:19.419 +02:00	ET EXPLOIT Zimbra <8.8.11 - XML External Entity Injection/SSRF Attempt (CVE-2019-9621)	high	45.146.164.125	60972	10.0.10.221
›	🔔	⚠	2021-05-28 14:27:21.996 +02:00	ET EXPLOIT Zimbra <8.8.11 - XML External Entity Injection/SSRF Attempt (CVE-2019-9621)	high	45.146.164.125	43782	10.0.10.221
›	🔔	⚠	2021-05-28 15:21:45.299 +02:00	ET EXPLOIT Zimbra <8.8.11 - XML External Entity Injection/SSRF Attempt (CVE-2019-9621)	high	45.146.164.125	52934	10.0.10.221

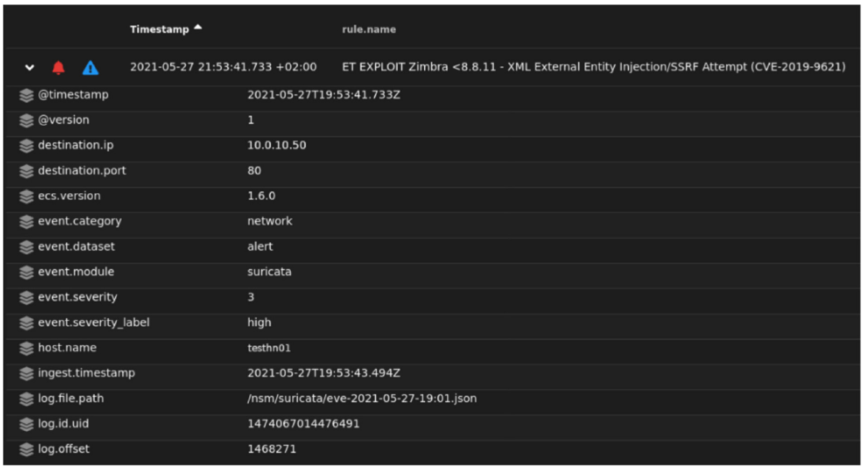
Figure 3. Security Onion Threat Alerts

The first step is to isolate the alarm and display only the alarm related to this malicious activity. As can be seen in Figure 3, all potential attacks came from the same IP address, while multiple IP addresses were targeted. In-depth analysis of the alarm message reveals detailed information about the target destination port, the type of event, the NIDS (Network intrusion detection system) which detected the attack, etc. First, a review of the local network should be done.

After research and other alarms, it is evident that only HTTP port 80 was targeted during the attack, as shown in Figure 4, which means that the attacker was targeting potential web servers.

Another important piece of information is the decoded packet which indicates that the POST header with the URL /Autodiscover/Autodiscover.xml link was sent to 10.0.10.50, and that the XML file tried to capture the /etc/passwd file, which is a file on Linux operating system that contains usernames, groups they belong to and similar essential information.

The threat was identified by the public CVE number 2021-2109, which can easily be searched and researched to learn more about this type of attack and methods of protection.



Timestamp	rule.name
2021-05-27 21:53:41.733 +02:00	ET EXPLOIT Zimbra <8.8.11 - XML External Entity Injection/SSRF Attempt (CVE-2019-9621)
@timestamp	2021-05-27T19:53:41.733Z
@version	1
destination.ip	10.0.10.50
destination.port	80
ecs.version	1.6.0
event.category	network
event.dataset	alert
event.module	suricata
event.severity	3
event.severity_label	high
host.name	testhn01
ingest.timestamp	2021-05-27T19:53:43.494Z
log.file.path	/nsm/suricata/eve-2021-05-27-19:01.json
log.id.uid	1474067014476491
log.offset	1468271

Figure 4. Detailed display of malicious activity alerts

After gathering some information about the attacker, the next step is to check if additional methods of intrusion were attempted from the same address. Thus, the sequence of movements and activities can be viewed chronologically. It can be determined where the attack started from, as well as when it happened. In this particular case, the first alarms were more informational and attempt to access non-existent web server pages were detected. This type of attack is mostly done with automated tools and scripts, so many alarms often appear, because it is a brute forcing technique in a way. After that, an attempt to establish sessions and communication from the target machine to the attacker was observed. And finally, there are attempts to exploit web server vulnerabilities. Further steps include the analysis of the web server itself, logs, data and scan for vulnerabilities and malicious files in order to determine whether there really was a breach, that is, whether the communication between the attacker and the target device was successful.

6. CONCLUSION

Collecting information in the form of recording and reading log messages is definitely the most important step in the entire problem solving process, and the accuracy of the information collected from log messages is of crucial importance.

Analyzing log messages directly from applications or text files is almost never done. For this purpose, log message management systems were created, as central points to which log messages are sent using log message collection agents. On these central systems, data is collected, the normalization process is carried out, and they are parsed and displayed in a readable format for further analysis. In accordance with the company's policies and legal regulations (e.g. GDPR), the data is stored for a certain period and after the expiration of the period, the data must be deleted.

Then, standard systems for managing log messages are not adequate and SIEM solutions (such as Security Onion) are used since they include more advanced tools for detection (and rarely prevention) of malicious activities. Such solutions have numerous integrated tools and specific agents for collecting not only log messages but also network traffic (packets), as well as metadata and system registry files. HIDS and NIDS solutions perform additional analysis and scanning of files on devices and send data to the central server for further analysis. SIEM are mandatory components of a computer network for monitoring events as well as recording and analyzing log messages. Detailed plans and strategies are needed for data storage and backup. Continuous monitoring of the system is crucial. Finally, it is essential to stay up-to-date with the latest trends and threats to network security in order to improve SIEM solutions.

References

Bhatt, S., Manadhata, P. K. & Zomlot, L. (2014). The Operational Role of Security Information and Event Management Systems. In *IEEE Security & Privacy*, vol. 12, no. 5, pp. 35-41, Sept.-Oct. 2014, https://doi.org/10.1109/MSP.2014.103

Jing, Y., Tingwen, L., Haoliang, Z., Jinqiao, S. & Guo, L. (2015). An automatic approach to extract the formats of network and security log messages. *MILCOM 2015 – 2015 IEEE Military Communications Conference*, pp. 1542-1547, https://doi.org/10.1109/MILCOM.2015.7357664

Panek, C. (2020). *Networking Fundamentals*. by John Wiley & Sons, Inc. ISBN: 978-1-119-65074-4

Ranjbar, A. (2015). *Troubleshooting and Maintaining Cisco IP Networks (TSHOOT) Foundation Learning Guide*, Pearson Education, Inc., ISBN-13: 978-1-58720-455-5

Simpson, M. T., Backman, K., & Corley, J. E. (2011). *Hands-On Ethical Hacking And Network Defense*. Course Technology, Cengage Learning. ISBN: 978-1-4354-8609-6.

Suh-Lee, C, Ju-Yeon, J., & Yoohwan, K. (2016). Text mining for security threat detection discovering hidden information in unstructured log messages. *2016 IEEE Conference on Communications and Network Security (CNS)*, pp. 252-260, https://doi.org/10.1109/CNS.2016.7860492

Vaarandi, R., Blumbergs, B., & Kont, M. (2018). An unsupervised framework for detecting anomalous messages from syslog log files. *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pp. 1-6, https://doi.org/10.1109/NOMS.2018.8406283