An Innovative Approach to Anomaly Detection in Communication Networks Using Multifractal Analysis

Pawel Dymora * and Miroslaw Mazurek

Faculty of Electrical and Computer Engineering, Rzeszów University of Technology, al. Powstańców Warszawy 12, 35-959 Rzeszów, Poland; mirekmaz@prz.edu.pl

* Correspondence: Pawel.Dymora@prz.edu.pl

Received: 11 April 2020; Accepted: 6 May 2020; Published: 8 May 2020

Abstract: Fractal and multifractal analysis can help to discover the structure of the communication system, and in particular the pattern and characteristics of traffic, in order to understand the threats better and detect anomalies in network operation. The massive increase in the amount of data transmitted by different devices makes these systems the target of various types of attacks by cybercriminals. This article presents the use of fractal analysis in detecting threats and anomalies. The issues related to the construction and functioning of the Security Operations Centre (SOC) are presented. To examine the correctness of SOC, several attacks on virtual systems located in the network were carried out, such as Denial of Service (DoS) attack, brute force, malware infections, exploits. Based on data collected from monitoring and devices, the response to the event was analyzed, and multifractal spectra of network traffic before and during the incident were created. The collected information allows us to verify the theses and confirm the effectiveness of multifractal methods in detecting anomalies in the operation of any Information and Communication Technology (ICT) network. Such solutions will contribute to the development of advanced intrusion detection systems (IDS).

Keywords: anomaly detection; computer network traffic; Hurst exponent; multifractal spectrums; TCP/IP; IoT; communication security; Industry 4.0; Security Operation Center (SOC)

1. Introduction

The security and privacy of online services have many challenges and problems that are still open to research. Security and privacy issues are still a significant challenge. Work anomalies can cause severe privacy and security issues when providing a service. In the era of rapid technological development related to information technology, the number and complexity of threats to the security of computer systems or networks are also growing. Preventing all security threats becomes impossible due to the constant adaptation of hackers to emerging information protection technologies. Organizations that have information technology (IT) systems take the necessary steps to counteract any threats to computer and network resources efficiently. Reactive action in the event of an incident is often not enough [1]. Therefore, in many cases, it becomes necessary to use external services or create a Security Operation Center (SOC), which is the security management center in the organization. The primary mission of an SOC is to actively counteract all threats and analyze them in case of a security incident [2,3].

As an anomaly, we consider a specific traffic pattern identified/detected throughout the whole range of analyzed data, indicating some deviation from the accepted standard. Such a deviation may be caused by various factors, both hardware and software, including increased activity caused by
security threads or denial of access attacks, as well as other types of destructive actions of cyberattacks. There are many tools and methods for testing traffic generated during packet transmission in computer networks. However, testing long-term traffic relationships using detection methods has an advantage over standard methods, as classical ones can detect known types of attacks that have already been previously identified and denoted. They operate on a specific traffic pattern, which is discovered by fittings. The types and methods of attacks are still evolving, so these methods do not keep up with the rapid changes. Therefore, detecting anomalies in traffic is much more flexible and prospective. That is why we base our research on it.

The performance and correctness of the proposed method are verified based on the examination of captured network traffic from a simulated network infrastructure in case of the occurrence and absence of different types of attacks. The presented results of the research with the use of multifractal analysis show significant possibilities to detect both short-term attacks and more intense ones. The research covered different types of attacks, and the analysis was carried out for the entire network traffic as well as from the perspective of different services and protocols. As the main method of our research, we use self-similarity and long-range dependency (LRD) of the network traffic nature. Self-similarity and scaling phenomena have dominated internet traffic analysis. With the identification of the LRD in network traffic, the research community has undergone a mental shift from Poisson and memory-less processes to LRD and bursty processes. The LRD phenomenon and self-similarity can be analyzed using the Hurst (H) parameter and multifractal spectrum analysis. The analysis of the multifractal dependencies of a given set of data allows for determining the characteristics of the tested data. The multifractal decomposition can be used to analyze the processes taking place in small time scales. A recent analysis of the traffic measurements from different communication networks showed that traffic is long-range dependent, or fractal (self-similar). Our article aims to emphasize the superiority of the Hurst parameter estimator based on the multifractal spectrum [4].

Network traffic can be saved both as an increasing number and as a sequence of gaps between adjacent events. The oldest stream model is the Poisson process. As a result of real network traffic research, internal stream correlation was observed. The model based on the Poisson process did not take this feature of generated traffic into account. In this way, further models were developed that more accurately reflect the real nature of network traffic. However, many of them are characterized by autocorrelation, which disappears very quickly (exponentially). These are called short memory processes. Meanwhile, the development of new network services has changed the characteristics of the traffic pattern and capacity. Currently, network traffic, especially in the Internet of Things (IoT) infrastructure, is characterized by a much stronger correlation over long distances. Most real processes, as shown by scientific research, are multifractal. Multifractal analysis and estimation of the Legendre’s spectral characteristics can detect the violation of multifractal properties caused by cyber-attacks. Self-similarity is a phenomenon that retains the statistical properties of the model despite changes in the applied time scale. The most visible feature in the characteristics of network traffic indicating the occurrence of self-similarity is the occurrence of densities and dilutions at intervals between events, and the lack of blurring of this feature despite the use of several different time scales (e.g., seconds, minutes, hours) [4]. Statistical data analysis enables quantification of network behavior, and thus a description of network operation anomalies. However, the most common methods of Hurst estimation include R/S statistics as a function of time scale, variance of the scaled process as a function of time scale, and absolute value method. These methods are qualitative and are used to determine the similarity of the examined stream of events.

The article presents issues related to the construction and operation of the SOC. It proposes the implementation of selected tools for the security of the IT system and assesses the effectiveness of their service in detecting threats. For the validation of the anomaly detection process, using multifractal analysis, selected attacks on virtual systems located in the network were carried out. The study of responses to a given security incident helped to create a multifractal spectra of network traffic before and during the attack based on data collected from monitoring and security devices.
The collected information could verify this thesis and confirm the effectiveness of multifractal methods in detecting anomalies in the operation of any Information and Communication Technology (ICT) network, including IoT infrastructure [4].

2. Literature Review

Fractal and multifractal analysis can help to uncover the structure of all kinds of systems in order to have a better understanding of such systems. In particular, both approaches have many different, exciting applications in the security of IT and IoT systems. The massive increase in data sent through various devices and communication protocols has raised serious security concerns that have increased the importance of developing advanced Intrusion Detection Systems (IDS).

In [5], the authors propose the use of multifractal analysis and Kalman filter for analyzing the security of Cyber-Physical Systems (CPS). The first stage of the proposed approach is data reduction due to time series processing. The second stage is the detection of anomalies, and the third stage is the prediction of the attack. For the detection of anomalies, it was proposed to use multifractal analysis and time series forecasting. The Kalman filter is used to detect cyber-attacks at an early stage because it accurately predicts the state of the system. The effectiveness of the proposed approach is confirmed by the results of the conducted experiments and analysis presented in the article.

The authors in [6] propose the use of multifractal analysis to detect anomalies of backbone network traffic that indicate network failures or attacks. Multifractal spectrum characteristics are used as safety indicators. The effectiveness of the proposed approach is confirmed by experimental results on the detection of Denial of Service (DoS) attacks.

Host-Based Intrusion Detection Systems (HIDS) are gaining traction in discovering malicious software inside a host operating system. In [7], the authors have developed a new cognitive host-based anomaly detection system based on a supervised AdaBoost machine learning algorithm. Notably, an information fractal dimension-based approach is incorporated in the original AdaBoost machine learning algorithm to assign a higher weight to the classifier that estimates the wrong hypothesis. The result of the empirical dataset showed that the malware detection performance of the proposed algorithm outperforms the original AdaBoost algorithm in detecting positives, including the reduction in false negatives.

Another solution for anomaly detection is shown in [8]. The authors present a new anomaly detector for Sketch and MultiScale (SMS) data traffic, based on combining random projections (sketches) with multi-scale analysis, which has low computational complexity. Sketches allow the automatic and reliable extraction of “normal” movement and the detection of anomalies without the need for training data. Multi-scale analysis extracts statistical descriptors using the wavelet leader tools recently developed for multi-fractional analysis, without having to select a time scale first. The proposed detector is illustrated using a sizeable recent set of backbone traffic data from the Measurement and Analysis on the WIDE Internet (MAWI) archive and compared with existing indicators.

In [9], anomaly detection used a combination of fuzzy C-means clustering and fuzzy interpolation scheme to analyze network traffic and detect malicious network activity. Once malicious activity is detected, IoT-Keeper automatically enforces network access restrictions against IoT device, generating this activity and prevents it from attacking other devices or services. This method shows that IoT-Keeper has a low resource footprint, and it can detect and mitigate various network attacks, without requiring explicit attack signatures or sophisticated hardware.

Other techniques also realize the detection of anomalies. In [10], the authors created a network intrusion detection mechanism using machine learning techniques. The model was constructed using a supervised learning methodology, using historical network data and data collected from the Internet.

According to [11], the authors proposed a method for detecting false attacks, as well as for locating the position of opponents performing attacks using a wireless forgery attack detector, which uses K-means cluster analysis. Presented studies using both 802.11 (WiFi) and 802.15.4 (ZigBee) networks showed that it is possible to detect wireless forgery with both a high detection frequency and a low
false-positive rate, thus providing strong evidence for the effectiveness of the K-means forgery detector and the attack locator. Similar results are presented in [12]; their experimental results show that the proposed methods can achieve over 90 percent hit rate and precision when determining the number of attackers. Their localization results, using a representative set of algorithms, provide strong evidence of the high accuracy of localizing multiple adversaries.

The Physical Layer (PHY) authentication test is presented in [13]. The authors, using information about the radio channel (such as received signal strength indicators), assessed the effectiveness of detecting forgery attacks in wireless networks. Both simulation and experimental results have validated the efficiency of the proposed strategies.

An interesting and innovative approach to the protection of wireless networks is presented in [14]. The authors examined the influence of piloting falsification attacks on the estimation of the reverse link channel, assuming that base station (BS) knows the angle of arrival (AoAs) of legal channels. To compare the performance of the estimators, theoretical analysis and numerical simulations were used, which showed that even imperfect knowledge of AoA and the averaging power of forgery signals allows maximum likelihood estimator (MLE) and minimum mean square error estimator (MMSEE) to achieve a significant increase in performance compared to least squares estimator (LSE).

The article [15] presents non-cryptographic mechanisms that are complementary to authentication and may detect forgery of devices with little or no dependence on cryptographic keys. Counterfeit-resistant relationships associated with transmitted packets and forgery-proof consistency checks have been introduced, which enable other network units to detect abnormal activity.

The cyber-attack technologies are constantly being developed, and the number of new threats is steadily growing. In the article [4,16], an innovative method of anomaly detection based on the estimation of self-similarity of systems and networks has been presented. Proposed in this article, the methods, in particular multifractal analysis, show sensitivity to any deviation in network traffic properties resulting from anomalies. The presented analysis allowed us to verify the theses and confirm the effectiveness of multifractal methods in detecting anomalies in the operation of any communication network. This shows that anomaly detection based on multifractality can be ideal for protecting critical data and maintaining the continuity of internet services, including IoT.

3. Self-Similarity of Processes

The usefulness of multifractal analysis in quantifying network traffic in supporting anomaly detection and in developing other applications to protect the security of transmission is presented. Multifractality is commonly observed in complex natural and socioeconomic systems. The multifractal analysis provides powerful tools for understanding the complex non-linear nature of time series in various fields. The approaches used to solve the problem of anomaly detection depend on the nature of the data collected. Network data can be obtained at many levels of detail, e.g., based on the end-user or based on the network. End-user data contains information that characterizes the end application (refers to the TCP: transmission control protocol and the UDP: user datagram protocol). Network data describes the operation of network devices and includes information collected from network devices (routers, switches). Traffic numbers obtained from both types of data can be used to generate the time series to which statistical signal processing techniques can be applied.

Self-similarity and fractals are concepts initiated by Benoit B. Mandelbrot. Self-similarity can be associated with “fractals”, which are objects with an unchanged appearance at various scales. Self-similarity means that network traffic displays shape similarity on an extended time scale. Hurst parameter (abbreviated as H) is an important index to measure the similarity, which is often used in traffic congestion control and access control. Therefore, estimating the Hurst parameter accurately and rapidly has significance in network management and control [4,16].

Stochastic processes are considered as sequences of variables, which can be characterized by using average value, variance, process probability distribution value, and higher stochastic moments. A stationary process, on the other hand, is a stochastic process for which the process probability
distribution value does not change. A characteristic element of a part of stochastic processes is the fact that the values are mutually dependent on time. However, such a process can be a stationary process, because a process’ probability distribution value in a moment \( t \) is determined without any premises about the time preceding the process. A process with long-range dependencies is a process for which time is infinite. The autocorrelation function for such a process is a slowly vanishing function [17,18].

The multifractal spectrum curves represent the distribution of intersection densities between the different areas. Multifractal spectrums can be determined in many different ways. The two primary methods are: Legendre transformation of the split function and establishing spot metering histogram boundaries. In particular, we have presented the details of these methods at work [4].

4. The Methodology of the Study on Network Traffic Anomaly Detection

An advanced simulation environment has been created for the research. The sophisticated infrastructure allowed us to realize many research scenarios and to measure traffic parameters, including the possibility of its analysis with division into individual network protocols. In order to create and efficiently simulate real network infrastructure, Graphical Network Simulator (GNS3) software was used. This is a free, graphical network emulator, allowing for comprehensive creation and testing of networks built from virtual equipment (a rich library of network devices from leading suppliers such as Cisco, Juniper, and virtual machines). GNS3 can create complex network topologies based on hardware virtualization. The model of designed and tested sophisticated network infrastructure used in the research is shown in Figure 1.

![Network model](image_url)

**Figure 1.** Network model used for simulation research.
Figure 1 shows a detailed diagram of the entire infrastructure under investigation. In the proposed simulation model, one can distinguish the following network infrastructure components:

- **Firewall pfSense**: In the presented model, it served as a firewall to filter incoming traffic from the Internet. It was directly connected to an R1 router. The Internet was connected to the network using the network card of the computer on which the network project was created;

- **Router R1**: vRouter solution was used, which replicates the functionality of hardware routing of internet protocol in the third layer of the ISO/OSI model. In this model, it served as a network gateway for devices in the local network. The network was divided into segments called VRF_A, VRF_B, VRF_C, VRF_SOC, and De-Militarized Zone (DMZ) using Virtual Routing and Forwarding (VRF). VRF is a technology that can partition the router into logical routers, each with a separate routing table. To divide the network into VRF instances for end devices, G0/2 interface has been used. For each subnetwork connected to this interface, a gateway to the LAN has been added, and the dynamic Internet Protocol (IP) address was configured. The VRF_SOC subnetwork was handled by the G0/1 interface. Routing in the network was static;

- **Network switches**: Two Ethernet switches available in the GNS3 library were used, whose function was limited to transmitting network traffic further. Additionally, the Open vSwitch was used to divide the traffic into individual segments, defined previously in the R1 router. For this purpose, virtual local area networks (VLANs), corresponding to VRF instances, were created. Additionally, a mirroring port has been enabled to intercept traffic from port G0/0 and send it entirely to port G1/1 to monitor network traffic between R1 router and devices in the local network;

- **Hosts**: A lightweight Linux TinyCore distribution with a Mozilla Firefox preinstalled browser was used to connect to the web interface of network security devices. The VRF_A subnetwork contains a host with Ubuntu Xenial installed, which was used as a platform for installing and testing security. In the VRF_B subnet, there is a host with KaliLinux system, which has many tools to simulate attacks on computer systems. In the VRF_C subnet, there was a VPC, a virtual computer that used minimal resources and had no graphical interface;

- **Servers-SRV1**: a virtual server located in the DMZ zone, similarly to VPC, had no graphical interface and allowed only the simplest ping operations.

In addition, free software like Wireshark [19], OpenVAS [20], Snort [21], Kali Linux [22], and Bro [23] were used for the data collection. Snort is a Network Intrusion Detection and Prevention System (IDS/IPS), available under a free license. This system offers a wide range of attack detection mechanisms and enables automatic traffic analysis and the registration of packets passing through networks based on IP/TCP/UDP/Internet Control Messages Protocol (ICMP) protocols. Snort can analyze packet streams, and search for and match suspicious content, as well as detect many attacks and anomalies. OpenVAS is an entirely free and easily accessible framework that acts as a vulnerability scanner. Wireshark can capture and record data packets, as well as their decoding. Thanks to a large number of additions, it can recognize and decode many communication protocols. It is mainly used by network administrators, exclusive services, and hackers to track packages. Kali Linux is a Debian-based Linux operating system distribution based on Debian, intended mainly for security breaches and penetration tests, as well as security audits. Bro software (an Open Source Network Security Monitoring Tool) is not an active security device, like a firewall or intrusion prevention system. Bro is a sensor, hardware, software, virtual, or cloud platform that quietly and unobtrusively observes network traffic. This software interprets what it sees and creates compact, high-fidelity transaction logs, file content, and fully customized output, suitable for manual review on disk or in a more analyst-friendly tool like a security and information event management (SIEM) system.

Devices in subnetworks VRF_A, VRF_B, and VRF_C were not able to connect to each other. However, each of these subnetworks could connect to the Internet, DMZ, and VRF_SOC. The DMZ zone could not connect to the Internet. The devices located in the VRF_SOC subnetwork were used to secure and monitor the IT system. They had unlimited access to all resources of the created network.
Their characteristics would be presented in the further part of this work. Kali Linux was used as a tool to carry out attacks. The system has a wide range of tools used for penetration tests and security breaches. In order to carry out tests, Kali Linux machine was placed in the VRF_B subnet and allowed to connect to other subnets. After booting, the network interface sent an IP address query to the Dynamic Host Configuration Protocol (DHCP) server, which was captured by IDS and presented as a high-priority alert.

The article presents a detailed analysis of network traffic, both in regular operation and during several selected types of network attacks on the security of selected network components and services provided.

The first group of the tested attacks will include Denial of Service (DoS) attacks. These attacks are aimed at blocking a computer or network, thus preventing users from accessing their resources. They do this by flooding the target with traffic or sending it information that causes a failure. In both cases, a DoS attack deprives authorized users of the expected service or resource [24]. PC1_Ubuntu was used as the target of the attack, using three types of flood type attacks. The hping3 tool installed on PC2_Kali_Linux host was used to test responses to three types of DoS attacks.

The first type of DoS attack is the UDP DoS flood. When a victim receives a UDP packet on a specific port, he first checks if any programs are running that are currently listening to the requests. If no application is receiving packets through this port, replies are sent with an ICMP packet to inform the sender that the destination is unreachable. This type of attack uses a false source IP address, making it difficult to reveal the location of the attacker and potentially saturate them with response packets sent from the victim. As a result of using resources to check and respond to each packet, the attacked target may quickly become overloaded enough to deny service to regular traffic.

The second attack is a DoS attack of the TCP Synchronize Sequence Numbers (SYN) type. It consists of sending a large number of SYN packets to a target, to which it responds with SYN/ACK packets, and then leaving the port open, ready to receive an ACK packet from the sender to complete the call reconciliation process. While waiting for an ACK packet that never arrives, the attacker continues to send SYN packets. The arrival of each new packet causes the victim to temporarily maintain a new connection to the open port for a specified period of time. After using all available ports, it stops responding to regular network traffic.

The third DoS attack is ICMP DoS. It consists of flooding the victim with ping packets. The same number of packets is sent to the sender in response. Additionally, an attacker may use custom codes or send fragmented packets, which the recipient must reunite. This burdens both incoming and outgoing network channels, consuming considerable bandwidth and causing a denial of service. This attack floods the target system with ICMP packets, changing the source address.

The second group of tested attacks includes brute force attacks. These attacks involve breaking the passwords or usernames of network resources or services using trial and error methods. Depending on the complexity, the process can take from a few seconds to many years. Secure Shell (SSH) and File Transfer Protocol (FTP) servers were installed on the PC1_Ubuntu host to perform tests. Then, using the hydra program available in Kali Linux, dictionary attacks were carried out by substituting passwords from [25] supplied with the system.

The last group of tested attacks was a malware infection. Malware can identify different types of viruses, spyware, ransomware, rootkits, or Trojan horses. They consist of code developed by cybercriminals, designed to cause significant damage to data and systems or to gain unauthorized access to the network. A standard method of spreading malicious programs is to provide them in the form of a link or attachment in emails that require the user to click on the link or open a file to run malicious processes. Diamorphine and Reptile were used to simulate rootkit activity on the system. These tools can hide any processes and files. Additionally, a test malware file was downloaded from [26] and malware samples from [27].
5. Multifractal Analysis of Performed Tests Results

The network traffic (transmitted packets) in the network infrastructure under study was saved in real-time in files by Security Onion sensors [28]. The collected data in the form of time series of transmitted packets allowed us to assess the network traffic between individual devices in the local network and R1 managing router. In order to provide a more accurate picture, the collected packets were exported using the Wireshark program v. 3.0.10 (License GPLv2), then Matlab 2019b software (MATLAB and Simulink. Natick, Massachusetts, United States) and Fraclab 2.2 toolkit (Copyright of the HIDS system client and Google Rapid Response (GRR) software to the server. The relatively small length of the spectrum in the X-axis resulted from a small number of UDP packets generated per second by devices during normal operation.

![Multifractal spectrum for network traffic during normal operation.](image1)

**Figure 2.** Multifractal spectrum for the whole network traffic during normal operation.

The spectrum analysis shows no anomalies. The presented flow does not show any abnormalities. In the case of a sample containing abnormal traffic associated with a network attack, the spectrum should significantly differ from the trajectory determined by the standard data flow, which will allow the detection of anomalies or threats.

Figure 3 shows the time period of network operation in which UDP packets generate the traffic. This type of packet in the examined model was used mainly to send information about the status of the HIDS system client and Google Rapid Response (GRR) software to the server. The relatively small length of the spectrum in the X-axis resulted from a small number of UDP packets generated per second by devices during normal operation.

![Multifractal spectrum for network traffic with UDP packets during normal operation.](image2)

**Figure 3.** Multifractal spectrum for network traffic with user datagram protocol (UDP) packets during normal operation.
5.1. Simulation of a DoS Attack and Its Multifactorial Analysis

The first simulation of unusual phenomena in the modeled network infrastructure was to carry out DoS attacks. Attacks related to UDP flood were captured as anomalies in network operation by Bro software. Two attacks of this type were successfully detected and identified during testing. The information gathered shows that the number of packets sent through ports supported by UDP protocol in comparison to other connections increased significantly. The collected data can identify the source and target of the attack. A repeated attempt to respond by the target host has failed, which may indicate a DoS attack. Default Snort rules were not able to detect unwanted activity, so new security rules were added, to send alerts in case of a UDP packet overload. The operation, with new security rules and identified attacks, is shown in Figure 4.

![UDP flood attack detected in Snort.](image)

Figure 4. UDP flood attack detected in Snort.

Figure 5 shows the multifractal spectrum of network traffic, during which one of the UDP flood attacks was detected. This is clearly indicated by the deviation from the standard, presented during normal network operation.

![The multifractal spectrum of network traffic generated by UDP packets during a UDP flood attack.](image)

Figure 5. The multifractal spectrum of network traffic generated by UDP packets during a UDP flood attack.

Figure 6 shows the multifractal spectrum of traffic generated by TCP packets during a UDP flood attack. In the normal operation, the number of TCP packets transmitted over the network under investigation was higher than in the case of UDP. This may have been due to the need for the authenticated connection of tools with external servers and the use of terminals using SSH protocol or graphical interface using Hyper Text Transfer Protocol Secure (HTTPS).
Another attack analyzed was a TCP SYN attack. It was also detected during simulation by Bro software. Additional alerts in response to this event were also generated by the Snort software, as shown in Figure 7.

![Figure 6](image-url) The multifractal spectrum of network traffic generated by transmission control protocol (TCP) packets during a UDP flood attack.

The number of packets transmitted in the network via port 80 significantly exceeded the other connections. There were many sources of network connections, which could indicate that the attacker was changing the IP address after each connection attempt or reconciliation process left. The rule “ET DROP Spamhaus DROP Listed Traffic Inbound group”, informs about attempts to establish a connection, with the host being the target of the attack, which were rejected by system firewalls.

Because the number of TCP packets during the normal operation was higher than in the case of UDP, the deviations shown in Figure 8 are small. Nevertheless, they indicate some anomalies, which may indicate two TCP SYN flood attacks detected by network security.
Another attack analyzed was one from the group of ICMP DoS attacks. Snort detected an ICMP flood attack. Alarms generated by the system are shown in Figure 9.

![Figure 9. Alerts to Internet Control Messages Protocol (ICMP) flood in Snort.](image)

For more detailed analysis and to increase the level of detection of this type of threat, Snort has added rules to detect a DoS ICMP attack. For additional testing, the IDS engine was changed to Suricata, and the attack was performed again. As you can see in Figure 10, another engine also detected an incorrect packet size and ICMP packet code.

![Figure 10. Suricata Intrusion Detection Systems (IDS) alerts for ICMP flood attack.](image)

Due to the insignificant number of ICMP packets generated during normal conditions, only one multifractal spectrum has been created, showing all the network traffic during an attack, as shown in Figure 11. In the following case, the anomalies of network traffic caused by flooding the device with ICMP packets are presented by an apparent deviation from the norm, which indicates the high intensity of the attack.

![Figure 11. The multifractal spectrum of the total network traffic during an ICMP flood attack.](image)
DoS-attacks are performed by devices located outside the network and are intended to load sensitive resources such as servers. The devices used in the above test were placed inside the computer network in order to check the reaction of the tools used in a situation where a firewall will pass DoS attacks. They were unable to take effective action. The results can be used to correctly configure firewall rules filtering incoming traffic from the Internet to prevent similar attacks in the future. Figure 12 shows the SSH session conducted with an OpenSSH server installed on a PC1-Ubuntu machine. Visible deviations may be related to the process of establishing an SSH session during DoS-attacks.

![Image](image_url)

**Figure 12.** The multifractal spectrum of Secure Shell (SSH) traffic during Denial of Service (DoS) attacks.

5.2. *Simulation of a Brute Force Attack and Its Multifactorial Analysis*

Another group simulated on the presented network infrastructure was brute force attacks. A host-based intruder detection system detected brute force attacks on SSH and FTP servers. The Hydra program started multiple connections to the SSH server using different authentication data, changing them each time until a password could be found. The brute force blocking script consists of two rules. The first one is a command to call a brute force blocking script with a given IP address if the condition is met. The second rule is responsible for effective action in case of detecting a threat in this case of a brute force attack by running a specific command. In the case of the SSH server, the triggered rule searches for any information related to the illegal user, which will appear in every log event. If it is called eight times in one hundred and twenty seconds, the next rule is activated, which informs about the attack and blocks any movement from the attacker for the time specified in the rule, which was set for three minutes to carry out many attempts of the attack. Attack alerts shown in Figure 13 inform about multiple attempts to log into the SSH server by a non-existent user. Additionally, the NIDS system detected a scan associated with the attack. After developing the details of the alert, one can see the source and destination address of the attack.
Figure 13. Detected brute force attack attempts on the SSH server in Snort.

Figure 14 illustrates the traffic related to the SSH protocol during a brute force attack. The increased length of the spectrum in the point dimension compared to the case of an ordinary session indicates the high activity of the attacker trying to access the server.

![Multifractal spectrum of SSH protocol during brute force attack](image1)

Figure 14. The multifractal spectrum of network traffic generated for SSH protocol during a brute force attack on an SSH server.

Figure 15 shows the multifractal spectrum of FTP and TCP traffic generated when establishing a correct connection with the vsftpd server installed on the PC1 host. The characteristics of this spectrum do not show any symptoms associated with a network attack. Figure 16 shows the reaction of the system to an attack on the FTP server.

![Multifractal spectrum of FTP protocol during brute force attack](image2)

Figure 15. The multifractal spectrum of network traffic generated for FTP protocol during a brute force attack.

Figure 16. Detected brute force attack attempts on the FTP server in Snort.
Brute force on the FTP server was associated with PAM, a Linux module responsible for the dynamic authentication of applications and services. As before, multiple login attempts were detected, and then scripts blocking connection with the attack source were launched. NIDS also captured packets containing information about the attack and classified them as high priority alerts. Figure 17 illustrates the multifractal spectrum of TCP and FTP packet flow during a brute force attack. The apparent deviation in the fractal dimension may indicate anomalous network behavior related to the attack.

![Figure 17. The multifractal spectrum during brute force attacks on the File Transfer Protocol (FTP) server.](image)

The information obtained for both attacks allows further action to be taken to protect against brute force. If an attack never fails, but the attacker makes further attempts after being unblocked by the system, the scripts allow the IP address to be placed on a blacklist, which will effectively prevent similar activity in the future. To avoid false alarms and to avoid blocking authorized users, the HIDS system has a whitelist, where trusted IP addresses can be placed.

5.3. Simulation of Malware Infections

For malware infections, no multifractal spectrum graphs were created, due to the fact that the tests were carried out on local machines and did not cause the spread of infections, which did not create anomalous traffic inside the tested network. The detection of malicious programs and processes was the responsibility of the Rootcheck scanner installed together with Wazuh agent software. The scanning frequency was set every five minutes. Using Diamorphine, a harmless system process was hidden, which Rootcheck then managed to detect as a potential threat in the form of kernel rootkit. With Reptile software, files were hidden in system folders. The scan was also able to find and classify one of the files as an anomaly due to incorrect file size. The results are shown in Figure 18.
The multifractal spectrum during brute force attacks on the File Transfer Protocol (FTP) server. The information obtained for both attacks allows further action to be taken to protect against brute force. If an attack never fails, but the attacker makes further attempts after being unblocked by the system, the scripts allow the IP address to be placed on a blacklist, which will effectively prevent similar activity in the future. To avoid false alarms and to avoid blocking authorized users, the HIDS system has a whitelist, where trusted IP addresses can be placed.

5.3. Simulation of Malware Infections

For malware infections, no multifractal spectrum graphs were created, due to the fact that the tests were carried out on local machines and did not cause the spread of infections, which did not create anomalous traffic inside the tested network. The detection of malicious programs and processes was the responsibility of the Rootcheck scanner installed together with Wazuh agent software. The scanning frequency was set every five minutes. Using Diamorphine, a harmless system process was hidden, which Rootcheck then managed to detect as a potential threat in the form of kernel rootkit. With Reptile software, files were hidden in system folders. The scan was also able to find and classify one of the files as an anomaly due to incorrect file size. The results are shown in Figure 18.

The VirusTotal tool integrated with software agent Wazuh was used to scan the folders. The monitored folder was the default folder for the Mozilla Firefox browser on Ubuntu. Both the malware samples and the eicar text file triggered the alarm shown in Figure 19. The activity related to the exe file downloading was reported by the NIDS system as a probable malicious file.

The overview of the report provides information on the location of the virus, the date of infection, and the checksums. One of the detected files that triggered the alerts was downloaded using Google Rapid Response, which was used for further threat analysis. The malicious file was scanned, and essential information on size and hash code was obtained. The file system was then examined for viruses with similar properties. The system returned five results. Figure 20 shows the results of the scan of one of the infections found.
samples and the eicar text file triggered the alarm shown in Figure 19. The activity related to the exe file downloading was reported by the NIDS system as a probable malicious file.

Figure 19. Malware software detected in the monitored directory in Snort.

The overview of the report provides information on the location of the virus, the date of infection, and the checksums. One of the detected files that triggered the alerts was downloaded using Google Rapid Response, which was used for further threat analysis. The malicious file was scanned, and essential information on size and hash code was obtained. The file system was then examined for viruses with similar properties. Figure 20 shows the results of the scan of one of the infections found.

Figure 20. Information about the scanned file in Google Rapid Response (GRR).

There are no antivirus programs installed on the network’s end terminals, whose main task is to eliminate the threats mentioned above. The purpose of the analysis in this section was to show the tools for centralized detection and response to network threats resulting from malware infection.

5.4. Simulation of Exploiting System Vulnerability and Its Multifactorial Analysis

The first simulation of exploiting system vulnerability is an attack using a Shellshock vulnerability. This type of attack on a system with the current version of the bash system shell has not produced any results and raises the alarm, indicating an attempted system tampering, visible in Figure 21.

Figure 21. Shellshock vulnerability attack detected in Snort.

Figure 22 shows the multifractal spectrum of network traffic related to the HTTP protocol, which is used during a failed attack during the simulation of exploiting Shellshock vulnerability. Multiple unsuccessful attempts have left more packets behind to detect the attack.

Figure 22. The multifractal spectrum of network traffic generated for Hyper Text Transfer Protocol (HTTP) during a failed Shellshock attack.

The attacks were carried out using a sensitive version of bash and the Metasploitable system. They were successful, which resulted in unauthorized access to the system resources. Besides this, intrusion detection systems did not detect any threat, which can be seen in the analysis of the spectrum presented in Figure 23. This does not allow us to determine whether there has been a breach of security measures due to its similarity to the spectra, representing network traffic during normal operation.

Figure 23. The multifractal spectrum of network traffic generated for HTTP protocol during a successful Shellshock attack.
The attacks were carried out using a sensitive version of bash and the Metasploitable system. They were successful, which resulted in unauthorized access to the system resources. Besides this, intrusion detection systems did not detect any threat, which can be seen in the analysis of the spectrum presented in Figure 23. This does not allow us to determine whether there has been a breach of security measures due to its similarity to the spectra, representing network traffic during normal operation.

In the case of Structured Query Language (SQL) Injection attack, incorrectly formulated union select queries executed by a web application returned the error and systems responsible for intrusion detection classified the event as an attempted attack, as shown in Figure 24.

The multifractal spectrum shown in Figure 25 applies to all HTTP traffic generated during SQL Injection attacks. In its characteristics, it is very similar to the spectrum showing a successful Shellshock attack, which also does not allow assessment of whether and when a security breach occurred.
The scan results shown in Figure 26 show vulnerabilities in the index.php file, the use of which may lead to the release of sensitive data. The OpenVAS software was used to prevent the vulnerabilities described above. It was used to perform a thorough scan of systems and applications on the PC1-Ubuntu and Metasploitable hosts. The scan results shown in Figure 26 show vulnerabilities in the index.php file, the use of which may lead to the release of sensitive data.

The detected system vulnerabilities have been classified as potential threats of Medium and Hard levels. These are mainly due to the use of outdated software versions or errors in the application configuration. Recommended solutions to the problems concerned updating outdated applications and removing or restricting access to phpinfo file. The scanner did not detect a vulnerability associated with Shellshock type vulnerability because the sensitive version of bash shell was run from a separate script. In contrast, the default bash on hosts does not have this vulnerability.

6. Effectiveness of Multifractal Analysis in Identifying Network Traffic Anomalies and Security Issues

To correlate the moment of detection of the abnormality with the deviations appearing in the graphs, a comparative analysis of multifractal spectra was performed for selected types of attacks. Comparative charts with two spectra were added to each of the protocols under study, based on all-day network traffic without incidents and on the day when security tests were conducted. Figure 27 shows a comparison of network traffic related to UDP protocol during regular operation and during an attack. The polynomial approximation was used to generate reference curves. Figure 28
concerns the comparison of all network traffic (all protocols) with that caused only by UDP packets. The flow of UDP packets during regular operation is more predictable, as evidenced by the determination of factor $R^2$ equal to 0.9785. In contrast, the low value of this factor during a UDP flood attack indicates high unpredictability of network behavior. A clear difference in the approximation lines suggests the occurrence of an attack, most probably at the point (0.39;0.88) marked in the graph. Spectra of incidents take similar shapes in the fractal dimension.

![Figure 27](image1.png)

**Figure 27.** Comparison of the multifractal spectrums of network traffic generated for UDP protocol during normal work and UDP flood attack.

![Figure 28](image2.png)

**Figure 28.** Comparison of the multifractal spectrums of the total network traffic (all protocols) and traffic generated for UDP protocol during regular work and UDP flood attack.
Figure 29 shows a comparison of TCP-related traffic during normal operation and flooding with TCP SYN packets. Figure 30 concerns the comparison of the entire network traffic with that generated only by TCP packets. In this attack, the determining factor in both cases is not high. Spectra already at the starting point differ, which indicates that the attacker is sending SYN packets and attempts to respond to the victim.

![Figure 29](image1.png)

**Figure 29.** Comparison of the multifractal spectrums of network traffic generated for TCP protocol during regular work and TCP SYN flood attack.

It can be seen that both spectra concerning network traffic during the occurrence of the anomaly take a similar length in the point dimension, which may indicate an increased number of packets per second appearing in the tested flows. Figure 31 shows a comparison of traffic during a regular SSH session and a brute force attack. Figure 32 refers to the full network (all protocols) traffic and SSH traffic during a daily work and brute force attacks.

![Figure 30](image2.png)

**Figure 30.** Comparison of the multifractal spectrums of the total network traffic (all protocols) and traffic generated for TCP protocol during regular work and TCP SYN flood attack.
Figure 31. Comparison of the multifractal spectrums of network traffic generated for SSH protocol during normal work and brute force attacks.

Figure 32. Comparison of the multifractal spectrums of the total network traffic (all protocols) and traffic generated for SSH protocol during normal work and brute force attacks.

The adjustment in both cases was high. The brute force attack is indicated by a significantly longer spectrum length of the SSH brute force characteristics on the graph, which may indicate a large number of incorrect login attempts. Similarly, as in the case of a TCP SYN flood attack, the spectra differ in the starting points, which indicates the occurrence of an attack in the early period of packet traffic. Similarly, as in the previously tested protocols, the length of the spectra concerning network traffic in which incidents occurred is more extended than during regular operation.

Figure 33 presents a comparison of the multifractal spectrums of network traffic generated for FTP protocol during a normal work and brute force attacks. The next one, Figure 34, presents the comparison of the multifractal spectrums of the total network traffic (all protocols) and traffic generated for FTP protocol during a regular work and brute force attacks.
A deficient determination factor of 0.454 regarding the approximation of the multifractal spectrum during an FTP brute force attack indicates a very high unpredictability of network traffic. The starting points of both spectra begin with an approximate point dimension. In contrast, the difference in the fractal dimension changes rapidly, which indicates that the attack quickly began to affect the number of packets present in the network traffic flow. The FTP attack spectrum shows similar characteristics in the fractal dimension as the full traffic spectrum, including security tests.

**Figure 33.** Comparison of the multifractal spectrums of network traffic generated for FTP protocol during normal work and brute force attacks.

**Figure 34.** Comparison of the multifractal spectrums of the total network traffic (all protocols) and traffic generated for FTP protocol during normal work and brute force attacks.
Figure 35 shows the network traffic of HTTP packets during regular operation and a shellshock attack. Figure 36 shows a comparison between full network traffic and HTTP packet flow. The determining factor in both cases takes similar values, but it is smaller in case of incidents that indicate an attack.

**Figure 35.** Comparison of the multifractal spectrums of network traffic generated for HTTP protocol during normal work and shellshock attack.

In Figure 35, one can see the probable point (0.41;1.00) where shellshock attack was detected. Until this point, the spectra presented a similar course. Spectrum associated with the incident shows identical characteristics as the spectra concerning proper network activity, which may indicate a small impact of this attack on network traffic.

**Figure 36.** Comparison of the multifractal spectrums of the total network traffic (all protocols) and traffic generated for HTTP protocol during normal work and shellshock attack.
Figures 37 and 38 show a collective comparison of the spectra both during regular operation and during the detected security incidents.

![Figure 37](image-url) **Figure 37.** Comparison of the multifractal spectrums of network traffic generated for five analyzed protocols during normal work.

![Figure 38](image-url) **Figure 38.** Comparison of the multifractal spectrums of network traffic generated for five analyzed protocols during security incidents.

Analyzing the graphs in Figures 37 and 38, one can see that the starting point of multifractal spectra for all protocols except FTP in regular operation starts in a similar area of the point dimension. A similar relationship also exists for security incidents. Additionally, the graphs have starting points in a further area of the point dimension. Attack spectra show more similar properties than for the actual traffic. Deviations from the norm occur in related areas of the fractal spectrum. Spectra for protocols during standard traffic show less common properties.
7. Conclusions

The problem of characterizing internet traffic is not a challenge that can be solved once and for all. The ever-increasing number of users and the reach of the internet is connected with the constant development of new technologies or their partial change. It is, therefore, necessary to constantly monitor network behavior and to constantly “learn” new patterns of traffic and threats. Only constant verification of the assumptions made will allow adjusting the conceptual models proposed by us and correctly describe the real network and its traffic. Since the internet as we know it today is based on the IP protocol, which is managed by the Internet Engineering Task Force (IETF), the use of IP addresses to identify connected devices and traffic along the way has become a completely natural. The research focuses on IP networks and will be further developed with a special focus on the wireless network.

The presented results of research using multifractal analysis show significant possibilities to detect both short-term and more intense attacks. The research covered different types of attacks, and the analysis was carried out for the entire network traffic as well as in the scope of different services and protocols. This method of detecting traffic anomalies is effective for network traffic of significant volumes, in particular aggregating different types and communication protocols. With small volumes of abnormal data and short-term attacks, such anomaly detection may be difficult due to embedding such traffic in other protocols, and larger aggregated datasets characterized by a normal specification, thus making these attacks undetectable. That is why we also study the entire network traffic. The traffic decomposition and analysis of specific protocols and services can detect anomalies at a low volume and short duration abnormalities, as well as at long-term changes in traffic pattern and attacks.

The fractal analysis helps to calculate and understand the fractal dimension of complex networks, especially anomaly detection. However, it is necessary to describe and characterize many fractal patterns that cannot be described by a single fractal size, which is why we have conducted a multifractal analysis of our network. The multifractal analysis makes it possible to calculate a set of fractal dimensions, especially generalized fractal dimensions.

The estimation of Hurst’s exponent value showed that long-term dependencies characterize network traffic. In each case, the Long Range Dependencies (LRD) were preserved, which emphasizes the general characteristics and nature of the traffic generated at the nodes. The estimation of Hurst’s exponent allows for the analysis of the current course, as well as the prediction of the future trend of data behavior. Thanks to this, it is possible to prepare in advance in an appropriate way to maintain or change the trend. Sudden deviations in the value of the exponent, in relation to the results obtained, may probably be a sign of problems that for network traffic would indicate unusual behavior of the network, possibly associated with a cybercriminal attack.

In this article, the authors present issues related to the implementation and validation of the multifractal analysis for the security incidents’ detection of the communication system in ICT and IoT systems. As shown in the case of the brute force attack (and some others), an anomaly is indicated by a significantly longer spectrum length of the point dimension on graph characteristics, which may indicate a large number of incorrect login attempts or redundant packets. Other tested protocols showed that the length of the spectra concerning network traffic in which incidents occurred is more extended than during regular operation.

Wireless attacks are easy to deploy and can significantly affect network performance. The high vulnerability to forgery attacks allows many other forms of attacks to be implemented. The conventional way to ensure the identity of the person sending the message and to detect the presence of an intruder is through device authentication. Unfortunately, full-scale authentication is not always desirable, as it requires key management and more extensive computations. With the growing use of wireless networks and IoT, this cannot be underestimated so that the authors will focus not only on SOC but also on wireless access networks in future studies.

The study of responses to a given security incident and created multifractal spectra of network traffic before and during the attack based on data collected from monitoring and security devices. The collected information allowed to verify theses and confirm the effectiveness of multifractal methods in
detecting anomalies in the operation of any ICT network, including IoT infrastructure. The solutions studied in this paper may be ideal for analyzing traffic in backbone networks from the point of view of security and attack detection. Tested methods, in particular multifractal analysis, show sensitivity to any deviation in network traffic properties resulting from anomalies. Such traffic analysis methods can be ideal for protecting critical data and maintaining the continuity of internet services, including IoT.

**Author Contributions:** Conceptualization, P.D. and M.M.; methodology, P.D. and M.M.; software: formal analysis and investigation, P.D. and M.M.; resources, P.D. and M.M.; writing—original draft preparation, P.D. and M.M.; writing—review and editing, P.D. and M.M.; visualization, P.D. and M.M.; supervision, P.D. and M.M.; project administration, P.D. and M.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This project was financed by the Minister of Science and Higher Education of the Republic of Poland within the “Regional Initiative of Excellence” program for years 2019–2022. Project number 027/RID/2018/19: amount granted 11,999,900 PLN.

**Acknowledgments:** We are thankful to Adrian Wolak of Rzeszów University of Technology, for supporting us in the collection of useful information.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Koczkodaj, W.W.; Mazurek, M.; Strzalka, D.; Wolny-Domiak, A.; Woodbury-Smith, M. Electronic health record breaches as social indicators. *Soc. Indic. Res.* 2018, 141, 861–871. [CrossRef]

2. Security Operations Centers against Cybercrime, Top 10 Considerations for Success, Insights on Governance, Risk and Compliance Security Operations Centers against Cybercrime. 2013. Available online: https://www.ey.com/Publication/vwLUAssets/EY-_Security_Operations_Centers_against_cybercrime/$FILE/EY-SOC-Oct-2013.pdf (accessed on 16 October 2019).

3. Winterborn, K. NCC Group Whitepaper: SOC Maturity & Capability. Available online: https://www.nccgroup.trust/globalassets/our-research/uk/whitepapers/2017/ncc-group-whitepaper-soc-maturity-and-capability.pdf (accessed on 14 December 2019).

4. Dymora, P.; Mazurek, M. Anomaly detection in IoT communication network based on spectral analysis and Hurst exponent. *Appl. Sci.* 2019, 9, 5319. [CrossRef]

5. Zegzhda, D.; Lavrova, D.; Poltavtseva, M. Multifractal security analysis of cyberphysical systems. *Nonlinear Phenom. Complex Syst.* 2019, 22, 196–204.

6. Zegzhda, P.D.; Lavrova, D.S.; Shytyrkina, A.A. Multifractal analysis of internet backbone traffic for detecting denial of service attacks. *Autom. Control Comput. Sci.* 2018, 52, 936–944. [CrossRef]

7. Khan, M.S.; Siddiqui, S.; McLeod, R.D.; Ferens, K.; Kinsner, W. Fractal based adaptive boosting algorithm for cognitive detection of computer malware. In Proceedings of the IEEE 15TH International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC), Palo Alto, CA, USA, 22–23 August 2016.

8. Fontugne, R.; Abry, P.; Fukuda, K.; Borgnat, P.; Mazel, J.; Wendt, H.; Veitch, D. Random projection and multiscale wavelet leader based anomaly detection and address identification in Internet traffic. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, 19–24 April 2013.

9. Hafeez, I.; Antikainen, M.; Ding, A.Y.; Tarkoma, S. IoT-KEEPER: Detecting malicious IoT network activity using online traffic analysis at the edge. *IEEE Trans. Netw. Serv. Manag.* 2020, 17, 45–59. [CrossRef]

10. Chakravarti, S.S.; Kannan, R.J. Detection of anomalies in cloud services using network flow data analysis. *Int. J. Electr. Eng. Educ.* 2020. [CrossRef]

11. Chen, Y.; Trappe, W.; Martin, R.P. Detecting and localizing wireless spoofing attacks. In Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, San Diego, CA, USA, 18–21 June 2007.

12. Sobana, J.; Arul, V.; Keerthana, B.; Mekala, R. Detection and localization of multiple spoofing attackers in wireless networks. *IEEE Trans. Parallel Distrib Syst.* 2013, 24, 44–58. [CrossRef]

13. Xiao, L.; Li, Y.; Han, G.; Liu, G.; Zhuang, W. PHY-layer spoofing detection with reinforcement learning in wireless networks. *IEEE Trans. Veh. Technol.* 2016, 65, 10037–10047. [CrossRef]
14. Darsena, D.; Giacinto, G.; Iudice, I.; Verde, F. Design and performance analysis of channel estimators under pilot spoofing attacks in multiple-antenna systems. *IEEE Trans. Inf. Forensics Secur.* 2020, 15, 3255–3269. [CrossRef]

15. Li, Q.; Trappe, W. Detecting spoofing and anomalous traffic in wireless networks via forge-resistant relationships. *IEEE Trans. Inf. Forensics Secur.* 2007, 2, 793–808. [CrossRef]

16. Mazurek, M.; Dymora, P. Network Anomaly Detection Based on the Statistical Self-Similarity Factor, Analysis and Simulation of Electrical and Computer Systems, Lecture Notes in Electrical Engineering; Springer: Berlin, Germany, 2015; Volume 324, pp. 271–287. [CrossRef]

17. Dymora, P.; Mazurek, M.; Strzałka, D. Multifractal analysis of selected multicast routing protocols in IP networks. In Proceedings of the ITM Web Conference, Villeurbanne, France, 5–8 September 2018; Volume 21. [CrossRef]

18. Wójcicki, R. Nowe metody modelowania samopodobnego ruchu w sieciach w oparciu o procesy Poissona z markowską modulacją. *Stud. Inf.* 2005, 26, 23–40.

19. Wireshark. Available online: https://www.wireshark.org/ (accessed on 10 October 2019).

20. Open Vulnerability Assessment Scanner. Available online: https://www.openvas.org/ (accessed on 10 October 2019).

21. Snort. Available online: https://www.snort.org/ (accessed on 11 October 2019).

22. Kali Linux. Available online: https://www.kali.org/ (accessed on 17 June 2018).

23. Zeek. Available online: https://zeek.org/ (accessed on 11 January 2020).

24. Tripathi, N.; Mehtre, B. DoS and DDoS attacks: Impact, analysis and countermeasures. In Proceedings of the TEQIP II National Conference on Advances in Computing, Networking and Security, Cairo, Egypt, 3–5 September 2013.

25. The rockyou.txt.gz File. Available online: https://www.mafia-download.com/file/0lu11NPhba/rockyou-txt-gz.html (accessed on 2 December 2016).

26. European Institute for Computer Anti-Virus Research (EICAR). Available online: https://www.eicar.org/ (accessed on 7 September 2006).

27. TekDefense. Available online: http://www.tekdefense.com/ (accessed on 24 June 2012).

28. Security Onion Documentation. Available online: https://securityonion.readthedocs.io/en/latest/ (accessed on 11 January 2020).

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).