Information & Security

G. Stevanoski et al.
vol. 55, no. 1 (2024): 81-94
https://doi.org/10.11610/isij.5550

Published since 1998
ISSN 0861-5160 (print), ISSN 1314-2119 (online)

Research Article

# Testbed of an Integrated Network Operations Center and a Security Operations Center Based on Open-Source Tools

***Goce Stevanoski*** [1] (✉), ***Marko Porjazoski*** [2], ***Aleksandar Risteski*** [2], ***and Mitko Bogdanoski*** [1]

[1] *Military Academy "General Mihailo Apostolski," Skopje, Republic of North Macedonia, https://ma.edu.mk/en*

[2] *Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University in Skopje, Republic of North Macedonia, https://feit.ukim.edu.mk*

ABSTRACT:

This article proposes a testbed IT environment that includes an Integrated Network Operations Center and a Security Operations Center based on open-source tools for conducting cybersecurity research. The testbed is capable of monitoring and configuring network devices and systems. The design includes physical devices, virtual machines, and strategically deployed sensors for performance and security-related data collection. It enables the study of network traffic, anomaly detection, and cybersecurity threats. The framework serves as a foundation for cybersecurity testing, offering real-time insights into the network's behavior, detecting faults, and identifying potential vulnerabilities.

ARTICLE INFO:

RECEIVED: 20 SEP 2024

REVISED: 23 OCT 2024

ONLINE: 29 OCT 2024

KEYWORDS:

cybersecurity testbed, network operations center, security operations center, NOC, SOC, integration, network traffic analysis, open-source tools



✉ Corresponding Author: goce.stevanoski@ugd.edu.mk

## Introduction

Cybersecurity breaches have become a significant threat to IT network infrastructure, leading to substantial financial losses and serious negative impacts on organizational reputation. This should serve as a strong incentive for every organization to implement measures to protect its infrastructure by constantly monitoring, evaluating, and upgrading its network systems. One approach to achieving this goal is by creating a separate IT network environment—a testbed environment—that mirrors the needs and parameters of the real production IT network infrastructure.

This testbed network should act as a "safe haven" for the unhindered evaluation of an organization's IT network security posture. The necessity of this approach was demonstrated in the recent CrowdStrike incident, where faulty updates that were not properly evaluated caused significant disruptions to IT networks worldwide, affecting many organizations across various industries.[1]

In addition to facilitating safer distribution of network updates, having a testbed network provides a secure environment for testing various network threat scenarios and evaluating mitigation strategies. These environments can be developed on a small scale but should closely reflect the real production IT network setup. The skills learned in testbeds can be directly applied to large-scale or production networks. This approach helps establish concepts, methodologies, and best practices that can be implemented in real IT environments, providing users with the necessary experience to work in the field of cybersecurity. Key approaches for building testbed networks include resource virtualization, environment simulation, network emulation, sandboxing, hardware-in-the-loop, and so on.

One of the main benefits of a safe environment for network threat testing is the ability to analyze network traffic to detect anomalies and malicious activities. The introduction of a testbed IT environment can help in understanding normal network traffic and identifying unexpected, potentially dangerous behaviors. As a critical cybersecurity skill, network traffic analysis involves monitoring and analyzing data flows to detect security vulnerabilities. Furthermore, it provides in-depth insights into all types of network protocols and trends within the IT environment. It enhances the utilization of monitoring tools, log analysis, and security problem identification. All these aspects can later be applied in real-world scenarios to counter cybersecurity attacks on large-scale organizational networks.

In this paper, we present an open-source-based testbed IT infrastructure that ensures network traffic and log collection for future analytics in intrusion detection, using Machine Learning (ML) and Artificial Intelligence (AI) algorithms to recognize anomalies. The testbed is capable of monitoring, manipulating, and configuring network devices and systems. Our approach follows the state-of-the-art guidance for developing an integrated Network Operations Center (NOC) and Security Operation Center (SOC), as proposed by Shahjee and Ware.[2]

## Related Work

The foundations for building testbed IT networks were introduced in the 1980s and 1990s with the research on the first computer-based simulations. At that time, Keshav [3] and Doner [4] proposed the development of network simulators for research purposes. This was followed by the works of Bishop, [5] Hill et al.,[6] Mullins et al., [7] and Volynkin and Skormin,[8] who identified the need for additional tools and techniques for teaching, testing, and research in the field of network security. By the end of the 1990s, this research led to the development of network simulators like NS-2 and OPNET Modeler (now Riverbed Modeler).

With the rise of virtualization technology, opportunities for creating testbeds expanded, making it possible to introduce real environments into testbed infrastructures, primarily due to the lower cost of implementation. Many researchers recognized the advantages of developing network security testbeds for testing and analyzing new concepts. In 2007, Volynkin and Skormin proposed the use of software virtualization for designing a virtual network testbed capable of containing the execution of dangerous code during research and development experiments.[8] Similar approaches were later proposed by van Heerden et al.,[9] Uramova and co-authors,[10] and Bălan et al.[11]

These opportunities, combined with real-world devices and services, have brought testbed environments into the research field of detecting anomalies in IT networks for countering cyber-related threats. Various testbeds have been created specifically to generate logs and extract network traffic data supporting the development and evaluation of Intrusion Detection Systems (IDS). For example, Sharafaldin et al. [12]] created a network with real devices, introducing various services such as firewalls, servers, user devices, and switches. This setup allowed them to launch different attacks and classify the extracted network traffic. In contrast to previous approaches, our approach uses real-world infrastructure but also incorporates a virtual environment. Ring et al.[13] proposed a small business infrastructure that includes web, email, file, and backup servers. They simulated normal user behavior using scripts and conducted various attacks. Their network is connected to the internet, but the reproducibility of the tests is limited. In our approach, the testbed is connected to the internet, but the firewall filters the traffic, securing a managed testing environment.

In 2022, Collins, Hussain, and Schwab [14] introduced a systematic approach for incorporating SOCs into cybersecurity experiments, including both evaluation and testing. They proposed a reference SOC model, and for the implementation of that model, they provided various software distributions suitable for deployment on cyber ranges, along with guidance and methodology for rigorous experiments, including those involving human cyber operators. In our paper, we broaden their scope by implementing an integrated NOC and SOC focused on defining open-source tools for the testbed environment.

## Open-Source Based Integrated NOC and SOC

The introduction of NOC and SOC elements into organizational IT networks should provide centralized performance and security monitoring of the deployed resources. Integrating NOC and SOC leverages the benefits of both elements while reducing deployment and maintenance costs. In this paper, we propose a testbed IT architecture that follows the state-of-the-art directions,[2] as shown in Fig. 1. This architecture consists of three layers: the Data Source Layer, the System Management Layer, and the Service Management Layer, each with its own dedicated functions that manage the performance and security of the network.

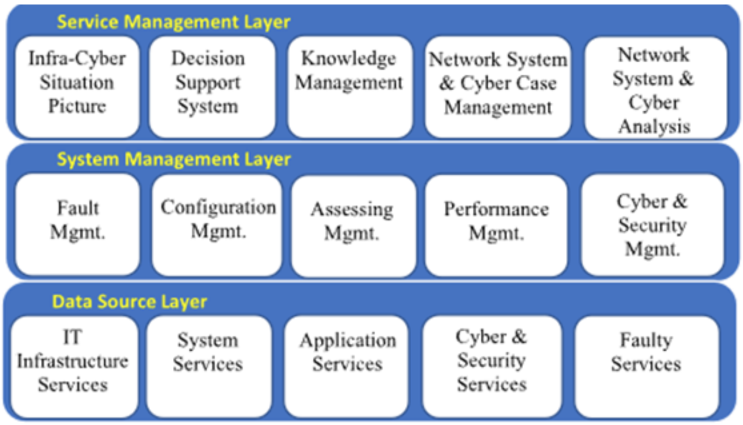


**Figure 1: Integrated NOC and SOC architecture.**[2]

The Data Source Layer integrates all network elements and security endpoints within the infrastructure. This layer generates data points for various measurements and sends them to the System Management Layer in the form of logs and events.

At the System Management Layer, real-time monitoring is conducted according to the FCAPS model,[15] which includes processing the collection of logs and events to detect faults, configuration, administration, performance, and security issues within the network. This layer produces security-related alert events and passes them to the Service Management Layer.

The Service Management Layer is the top layer, and it is responsible for decision-making. It receives input and creates a holistic overview of the network situation. Here, logs and events are collected, mapped, correlated, and indexed to assist with incident management, detection, auditing, impact analysis, and forecasting of future conditions and events.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Service Management Layer** | OTRS (Open Ticket Request System) Zammad The Hive | MediaWiki Zammad | GLPI / iTop | iTop | GLPI | iTop | OTRS / Zammad | |
| | **Incident Management** | **Knowledge Management** | **Asset and Configuration Management** | **Service Level Management** | **Service Request Management** | **Change Management** | **Problem Management** | |
| **System Management Layer** | | Prometheus Zabbix | Cockpit phpIPAM | Ansible / Puppet RANCID | Nagios Core / Icinga Zabbix | Shuffle OSSEC Wazuh The Hive | | |
| | | **Performance Management** | **Administration** | **Configuration Management** | **Fault Management** | **Security Management** | | |
| **Data Source Layer** | Elasticsearch Logstash Fluentd Graylog | nProbe ntopng Zeek Suricata | Prometheus Collectd Telegraf | PostgreSQL MySQL/Maria MongoDB InfluxDB | Kafka RabbitMQ NATS | Ceph MinIO GlusterFS | Beats Flume Vector | Wireshark tcpdump pcap-ng |
| | **Log Collection and Management Tools** | **Network Data Collection Tools** | **Metric Colletion Tools** | **Database Collection and Management Tools** | **Event Collection Tools** | **File and Object Storage Tools** | **Data Aggregation Tools** | **Packet Capture Tools** |
| | **Network Elements** | **Network Elements** | **Network Elements** | **Network Elements** | **Network Elements** | **Network Elements** | **Network Elements** | **Network Elements** |
| | **Switches, Routers, Security Endpoints, Computers, Laptops, Telephones, Monitoring Systems/Devices, Firewalls, Network and** | | | | | | | |

**Figure 2: Open-source based tools for integrated NOC-SOC.**

This layered approach to designing integrated NOC and SOC shows a clear path for implementing appropriate tools in each of the layers. In our testbed we are proposing the use of open-source tools since the IT community over the years has developed a robust and reliable products which can fit in every part of the layers. Fig. 2. shows the open-source tools in each of the layers and presets the difference between the layers and the tools that can be used. Some of the tools are developed to fit widely in the structure, in-between different layers but they serve the layer's intention as they are deployed.

## Testbed IT Infrastructure

Our proposed testbed is designed to support cybersecurity-related research. It is deliberately planned to accurately represent an organizational environment, composed of a production network and an integrated NOC and SOC built with open-source tools, as presented in the previous section. A high-level framework is shown in Fig. 3. This testbed oversees the devices in the production segment of the network by collecting logs and events and storing them in the integrated NOC and SOC. This enables the integrated NOC and SOC to create an overall situational awareness picture and make decisions based on various implemented analytics. These decisions can later be implemented by distributing configuration commands and files. The testbed anticipates the implementation of various types of network sensors.

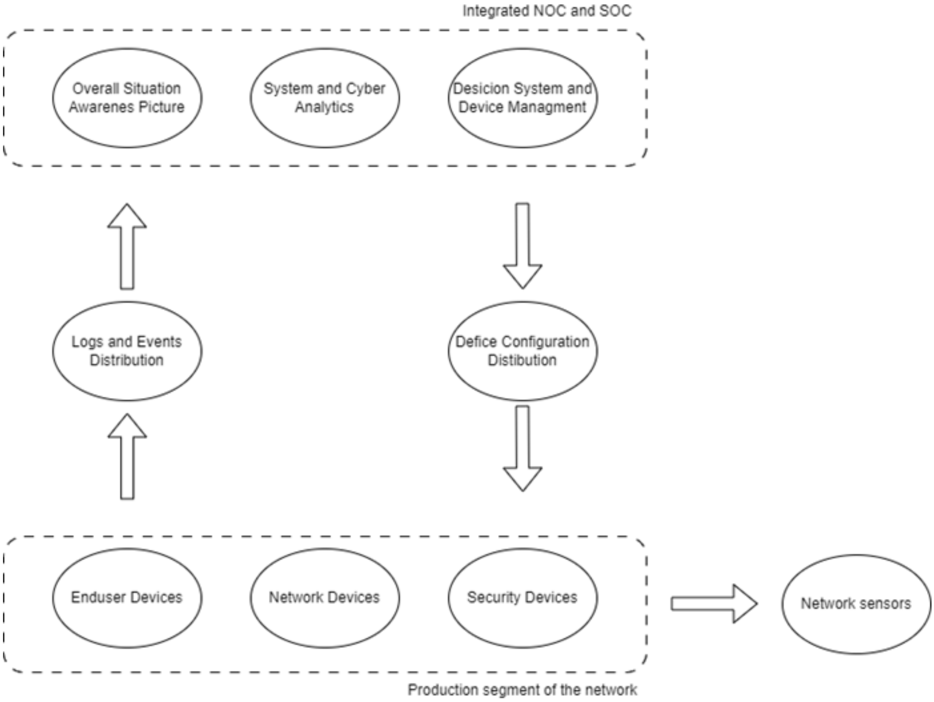


**Figure 3: High-level design of the proposed testbed network.**

To demonstrate the proof of concept for the proposed IT network infrastructure, we have deployed the testbed according to the architecture presented in Fig. 4. According Krishnamoorthi and Carleton,[16] 90 % of the organizations use Windows Active Directory in their infrastructure; therefore, our architecture is set on Windows Domain Services. This domain network represents the baseline of a production network environment that connects to the integrated NOC and SOC for performance and security monitoring and network resource management. The testbed is constructed using a combination of physical devices and virtual machines (VMs), with the VMs distributed across two dedicated physical servers and open-source software solutions. Additional sensors have been strategically deployed throughout the network to facilitate in-depth research and analysis. These sensors play a critical role in monitoring and collecting detailed data points related to network activities, providing valuable insights for cybersecurity research.

### *Domain Network*

The domain network within our testbed is structured around a Windows domain managed by Windows Server 2022 (WinSRV2022), which functions as the Domain Controller (DC). The DC is the central authority responsible for managing the network, ensuring security, and providing essential services to the domain. Several physical client computers are integrated into this domain network, allowing for a realistic simulation of an organizational environment.

Within this domain, users are created in the Active Directory of the DC, each assigned specific roles that dictate their level of access to the network infrastructure. This role-based access control is a key feature of organizational networks, ensuring that users have appropriate permissions aligned with their job functions. The centralized management of users, devices, and services within the Windows domain network effectively mirrors the complexities of a real-world organizational setup, providing a robust platform for cybersecurity testing and research.

This testbed environment enables researchers to explore and experiment with various cybersecurity scenarios, simulating real-world conditions and threats in a controlled and secure setting.

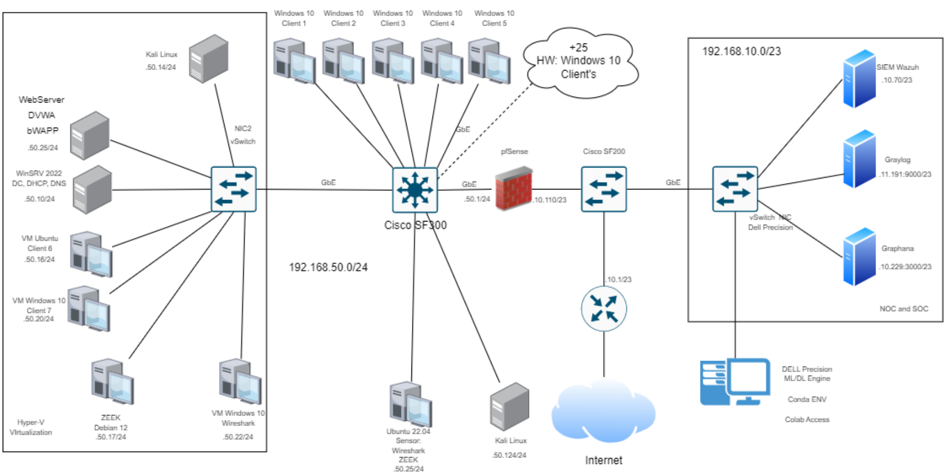


**Figure 4: Architecture of a deployed testbed IT infrastructure.**

### *Services*

The testbed network is supported by several different services. These services include Active Directory Domain Services (AD DS), Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), Group Policy, Hyper-V, Network Policy and Access Services (NPAS), web services, and others. These services can be removed, and additional services that this testbed can support can also be added, depending on the planned attacks that will be generated.

### *Security*

Monitoring and reporting on the production network play a significant role in this setup. This is achieved through the implementation of an integrated NOC and SOC that utilizes various tools, such as Security Information and Event Management (SIEM) via Wazuh, the log management solution Graylog Open 5.0, and the visualization tool Grafana.

- Wazuh is an open-source SIEM solution that provides centralized aggregation and real-time analysis of telemetry for detecting threats and assessing compliance. It uses Endpoint Detection and Response (EDR) agents and collects event data logs from various sources in the network, such as network devices, endpoints, cloud workloads, and applications, to ensure enhanced security. Some of its capabilities include security log analysis, vulnerability detection, security configuration assessment, regulatory compliance, reporting insights from endpoint events, alerting, and notifications.
- Wazuh integrates with Wazuh XDR, an Extended Detection and Response (XDR) platform that stores telemetry data points from endpoints, network devices, cloud workloads, third-party APIs, and other sources for a unified approach to security monitoring and protection.
- The event data is stored in OpenSearch, a distributed search and analytics engine. OpenSearch has the unique ability to store data in multiple locations on the network. Since logs can be voluminous, distributing them across multiple locations allows for faster searching. Regardless of the type of data, OpenSearch enables storage and analysis.
- Graylog Open 5.0 is an open-source centralized Log Management System (LMS) that aggregates, organizes, and analyzes data collected from various devices, applications, and operating systems. Graylog Open parses the received data by adding relevant information or extracting unnecessary details. It is highly efficient, even when handling petabytes of data, and is useful for forensic investigations, threat hunting, and business analytics.

Grafana is an open-source visualization tool that provides querying capabilities, data visualization, alerting on specific data points, and investigation of metrics, logs, and traces stored in databases. It helps transform time-series data into insightful graphs and visualizations. Grafana also supports a versatile plugin framework that enables connections to different types of data sources, such as NoSQL/SQL databases, ticketing tools, or OpenSearch.

The flow of data is presented in Fig. 5. EDRs are deployed on the endpoint machines and managed by the Wazuh SIEM manager, from which they receive configuration updates. All event logs collected from the endpoints by the EDRs are sent to Graylog Open LMS for parsing. Graylog Open parses the data, removes unnecessary information, and adds relevant external data (e.g., IP address location coordinates). This filtered and parsed data is sent to the Wazuh OpenSearch distributed system for ingestion. From this point, all relevant data is available in OpenSearch, and Wazuh SIEM and Grafana can be used for querying, defining alerts, conducting analytics, generating reports, and creating visualizations for better understanding.

From a security perspective, a pfSense firewall is introduced. Positioned between the networks in the testbed environment, this firewall is configured to

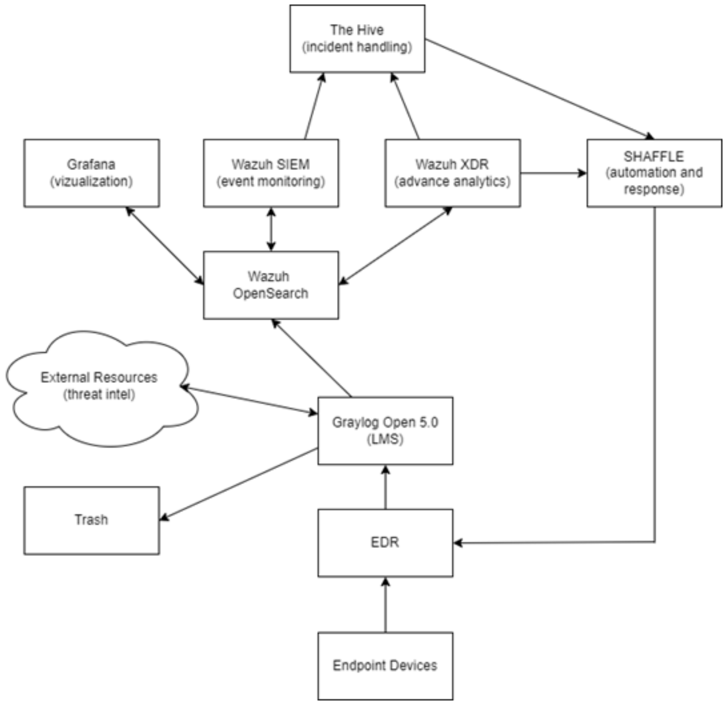


**Figure 5: Flow of security-related data in the deployed testbed architecture.**

manage network traffic and enforce control, ensuring that malicious traffic is blocked from reaching other parts of the network.

- pfSense is an open-source FreeBSD distribution that can be installed on commodity hardware to act as a firewall and router. It has a user-friendly web interface for management and includes a software package system to extend its capabilities. Some pfSense features include Stateful Packet Inspection, IP/DNS-based filtering, anti-spoofing, captive portal guest networks, time-based rules, connection limits, NAT mapping (inbound/ outbound), IDS/IPS, Snort-based packet analysis, Layer 7 application detection, and access to emerging threats and IP block lists.

### *Sensors*

In addition to the implemented monitoring and reporting via SIEM and XDR solutions, sensor devices are integrated at various network locations. These sensors register activities and collect logs from network traffic for future reference. For this purpose, the Wireshark packet analyzer and Zeek are installed on these sensors.

- Wireshark is an open-source network packet analyzer that presents captured packet data in as much detail as possible. It provides an in-depth overview of what happens in network traffic. Wireshark has many features, some of which include: capturing live packet data from a network interface, opening files containing packet data captured with tcpdump/WinDump, Wireshark, and many other packet capture programs, importing packets from text files containing hex dumps of packet data, displaying packets with very detailed protocol information, saving captured packet data, exporting some or all packets in a number of capture file formats, filtering packets on many criteria, searching for packets on many criteria, colorizing packet display based on filters, and creating various statistics.

- Zeek is an open-source passive network traffic analyzer used as a network security monitor (NSM) for detecting and investigating suspicious or malicious activity. Besides its usage in the security domain, Zeek supports a wide range of traffic analysis tasks, including performance measurement and troubleshooting. Zeek creates an extensive set of logs that describe network activity in a comprehensive way, where every connection seen on the wire is presented. Zeek also captures application-layer transcripts, including all HTTP sessions with their requested URIs, MIME types, key headers, and server responses; key content of SMTP sessions; SSL certificates; DNS requests with replies; and much more. Zeek writes all this information into structured tab-separated or JSON log files suitable for post-processing with additional software.

Zeek is optimized for interpreting network traffic and generating logs based on that traffic. It is not optimized for byte matching and is not a protocol analyzer like Wireshark, which depicts every element of network traffic at the frame level, or a system for storing traffic in packet capture (PCAP) form. Instead, Zeek produces compact and high-fidelity network logs, which contribute to a better understanding of network traffic and usage.

## Testbed Network Monitoring Potential Demonstration

To demonstrate the potential of the deployed testbed in cybersecurity, we conducted a test on monitoring network traffic during a Command Line Injection Attack on the web server. The purpose of this test is to showcase the traffic monitoring capabilities of the proposed testbed environment, and it does not delve into the process of anomaly detection for intrusion, as that is not the primary focus of our paper.

In this scenario, the attacker machine performs Command Line Injection Attacks on our test web page. The implemented sensors monitor network activities and collect data points. In this example, data points are collected from three locations: the Zeek sensor for transactional logs, the Wireshark sensor for network packet data, and the Wazuh SIEM for event logging.

Fig. 6 shows the transactional logs collected from the Zeek sensor during our research. Zeek's ability to mark streams with UIDs provides a comprehensive way to analyze connections between different protocols in the same event.


**Figure 6: Zeek logs captured from the command line injection attack.**

Fig. 7 presents the Wireshark network packet capture capabilities. This capture is rich with internal data points that, when used with tools like CICFlowMeter, can be extracted and utilized for future analysis.

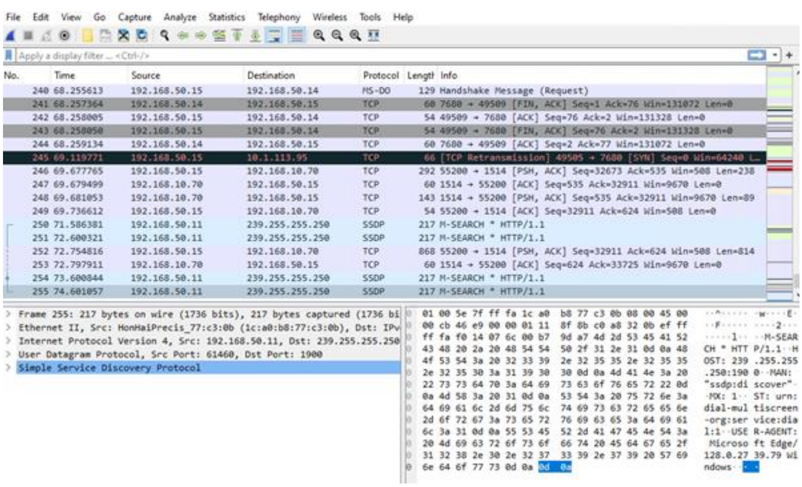


**Figure 7: Wireshark capture from Command line injection attack.**

Fig. 8 clearly displays the event generation from the web server during the attack. Event monitoring, performed here using the Sysmon tool, expands the possibilities for monitoring events on endpoint devices and significantly aids in both security monitoring and performance monitoring of end-user devices.

The captured IT network traffic from the deployed testbed environment demonstrates the wealth of data that can be extracted from this setup. Furthermore, it can support additional research on that data using various tools and approaches, such as Machine Learning (ML) and Artificial Intelligence (AI) algorithms.

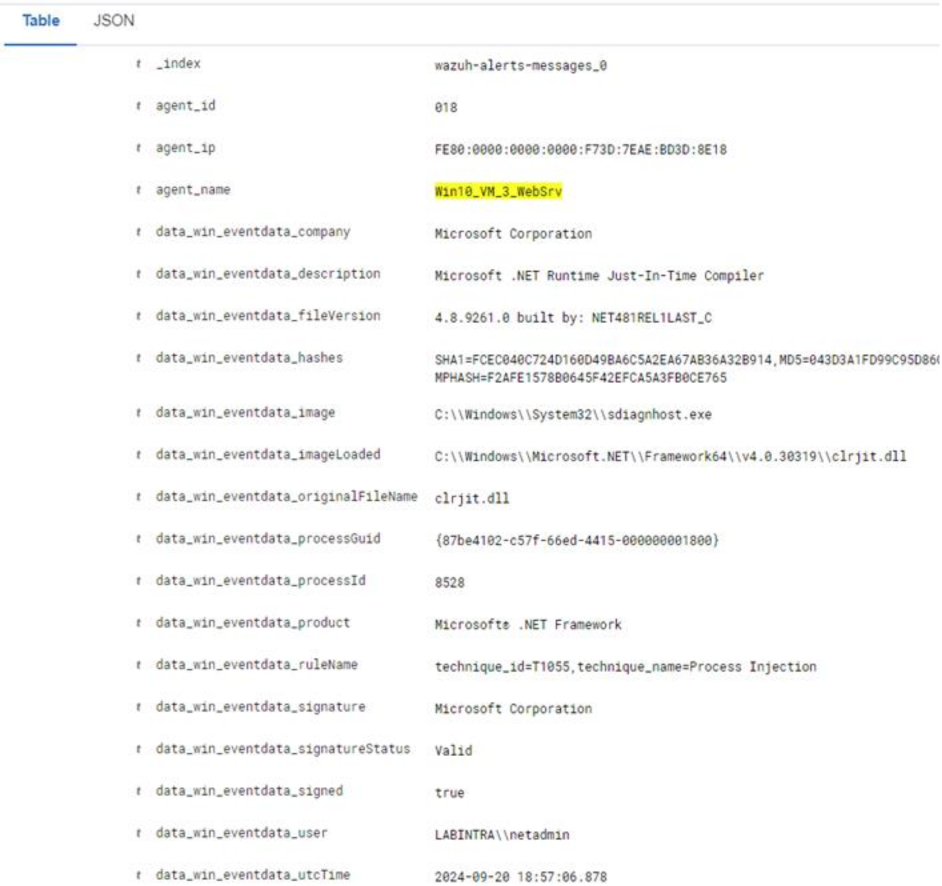
Table JSON

| | |
|---|---|
| t _index | wazuh-alerts-messages_0 |
| t agent_id | 018 |
| t agent_ip | FE80:0000:0000:0000:F73D:7EAE:BD3D:8E18 |
| t agent_name | Win10_VM_3_WebSrv |
| t data_win_eventdata_company | Microsoft Corporation |
| t data_win_eventdata_description | Microsoft .NET Runtime Just-In-Time Compiler |
| t data_win_eventdata_fileVersion | 4.8.9261.0 built by: NET481REL1LAST_C |
| t data_win_eventdata_hashes | SHA1=FCEC040C724D160D49BA6C5A2EA67AB36A32B914,MD5=043D3A1FD99C95D86<br>MPHASH=F2AFE1578B0645F42EFCA5A3FB0CE765 |
| t data_win_eventdata_image | C:\\Windows\\System32\\sdiagnhost.exe |
| t data_win_eventdata_imageLoaded | C:\\Windows\\Microsoft.NET\\Framework64\\v4.0.30319\\clrjit.dll |
| t data_win_eventdata_originalFileName | clrjit.dll |
| t data_win_eventdata_processGuid | {87be4102-c57f-66ed-4415-000000001800} |
| t data_win_eventdata_processId | 8528 |
| t data_win_eventdata_product | Microsoft® .NET Framework |
| t data_win_eventdata_ruleName | technique_id=T1055,technique_name=Process Injection |
| t data_win_eventdata_signature | Microsoft Corporation |
| t data_win_eventdata_signatureStatus | Valid |
| t data_win_eventdata_signed | true |
| t data_win_eventdata_user | LABINTRA\\netadmin |
| t data_win_eventdata_utcTime | 2024-09-20 18:57:06.878 |

**Figure 8: Event Generation on Wazuh SIEM.**

## Conclusions

In this paper, we presented a testbed for an organizational Windows domain network managed by an integrated NOC and SOC based on open-source tools.

The conducted monitoring of network activities has demonstrated that the testbed can support the collection of various types of data points, which provide significant value for research in the cybersecurity domain. With monitoring implemented at multiple points within the network, the testbed enables reliable collection of network traffic and system logs necessary for detecting anomalies using ML and AI algorithms. Additionally, this highlights the strengths of open-source software as a reliable tool that can be successfully utilized for developing new security solutions and countering cybersecurity threats.

Overall, the work presented in this paper establishes a solid foundation for conducting studies on anomaly detection for intrusion detection and provides guidance for future research in this area.

## References


1. A. Shaji George, "When Trust Fails: Examining Systemic Risk in the Digital Economy from the 2024 CrowdStrike Outage," *Partners Universal Multidisciplinary Research Journal* 1, no. 2 (2024): 134–152.
2. Deepesh Shahjee and Nilesh R. Ware, "Integrated Network and Security Operation Center: A Systematic Analysis," *IEEE Access* 10 (2022): 27881–27898, https://doi.org/10.1109/ACCESS.2022.3157738.
3. Srinivasan Keshav, *REAL: A Network Simulator* (USA: University of California Berkeley, 1988).
4. John R. Doner, "GENESIM (Generic Network Simulator)," *IEEE Journal on Selected Areas in Communications* 6, no. 1 (1988): 172–179, https://doi.org/10.1109/49.192740.
5. Matt Bishop, "The State of Infosec Education in Academia: Present and Future Directions," in *Proceedings of the National Colloquium on Information System Security Education*, 1997, pp. 19–33.
6. JJohn M. D. Hill, Curtis A. Carver, Jeffrey W. Humphries, and Udo W. Pooch, "Using an Isolated Network Laboratory to Teach Advanced Networks and Security," *ACM SIGcSe Bulletin* 33, no. 1 (2001): 36–40.
7. Paul Mullins *et al.*, "Panel on integrating security concepts into existing computer courses," *ACM SIGCSE Bulletin* 34, no. 1 (2002): 365–366.
8. Alexander Volynkin and Victor Skormin, "Large-scale Reconfigurable Virtual Testbed for Information Security Experiments," in *2007 3rd International Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities*, IEEE, 2007, pp. 1–9, https://doi.org/10.1109/TRIDENTCOM.2007.4444663.
9. Renier van Heerden, Heloise Pieterse, Ivan Burke, and Barry Irwin, "Developing a Virtualised Testbed Environment in Preparation for Testing of Network Based Attacks," in *2013 International Conference on Adaptive Science and Technology*, IEEE, Nov. 2013, pp. 1–8, https://doi.org/10.1109/ICASTech.2013.6707509.

[10] Jana Uramova, Pavel Segec, Jozef Papan, and Ivana Bridova, "Management of Cybersecurity Incidents in Virtual Lab," in *2020 18th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, IEEE, Nov. 2020, pp. 724–729, https://doi.org/10.1109/ICETA51985.2020.9379159.

[11] Titus Bălan, Dan Robu, Florin Sandu, and Alexandra Bălan, "Building a Virtualized Cybersecurity Lab: Using Industry Support, Academic Programs and Open Source Solution for Setting-Up a Virtualized Cybersecurity Lab," in *Internet of Things, Infrastructures and Mobile Applications: Proceedings of the 13th IMCL Conference 13*, Springer, 2021, pp. 1024–1032.

[12] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSp*, vol. 1, 2018, pp. 108–116.

[13] Markus Ring, Sarah Wunderlich, Dominik Grüdl, Dieter Landes, and Andreas Hotho, "Flow-based benchmark data sets for intrusion detection," in *Proceedings of the 16th European Conference on Cyber Warfare and Security, ACPI*, 2017, pp. 361–369.

[14] Michael Collins, Alefiya Hussain, and Stephen Schwab, "Towards an Operations-Aware Experimentation Methodology," in *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, IEEE, Jun. 2022, pp. 384–393, https://doi.org/10.1109/EuroSPW55150.2022.00046.

[15] ISO – International Organization of Standards, "ISO/IEC 10040: Information Technology – Open Systems Interconnection – Systems Management Overview," *ISO/IEC*, October 15, 1998.

[16] Swetha Krishnamoorthi and Jarad Carleton, *Active Directory Holds the Keys to your Kingdom, but is it Secure?* (Frost and Sullivan, 2020).

## About the Authors

Goce **Stevanoski** – see the CV on p. 43 of this volume, https://doi.org/10.11610/isij.5539

Marko **Porjazoski**, D.Sc., is a Full Professor with 24 years of experience in the field of computer networks. He has worked in private organizations and in universities as a research and teaching assistant, assistant professor, associate professor, and full professor. He is currently a Full Professor in the Faculty of Electrical Engineering and Information Technologies, Saints Cyril and Methodius University, Skopje, Republic of North Macedonia.
https://orcid.org/0000-0002-8542-3242

Prof. Aleksandar **Risteski** – see the CV on p. 43 of this volume, https://doi.org/10.11610/isij.5539

Prof. Mitko **Bogdanoski** – see the CV on p. 43 of this volume, https://doi.org/10.11610/isij.5539