# Ionian University

**Department of Informatics**



IONIAN UNIVERSITY

-- Semester Project --

Portable Intrusion Detection System for Security Information and Event Management


**Avgoustis Andreas**

**Supervisor:** Stylianos Karagiannis , Christoforos Ntandoyan

# Summary

Wireless network attacks are actions of infiltration and intrusion targeting wireless networks and are serious threats. Wireless network attacks aim to capture information sent through the network and/or infiltrate information traffic. Security Information and Event Management (SIEM) is a security solution that helps organizations identify potential threats and vulnerabilities before they have the opportunity to disrupt business operations. It detects anomalies in user behavior and uses artificial intelligence to automate many manual processes related to threat detection and incident response. It has become a key component in modern Security Operation Centers for security management and compliance use cases.

An intrusion detection system (IDS) is a monitoring system that detects suspicious activities and generates alerts when detected. Based on these alerts, a security operations center analyst or incident responder can investigate the issue and take appropriate actions to mitigate the threat.

# Contents

# List of Figures

# Chapter A

# Introduction

Wireless network attacks are actions of infiltration and intrusion targeting wireless networks and constitute serious threats. Wireless network attacks aim to capture information sent through the network or penetrate information traffic. SIEM is a security solution that helps organizations recognize potential threats and vulnerabilities to security before they have a chance to disrupt business operations.
It detects user behavior anomalies and uses artificial intelligence to automate many manual processes related to threat detection and incident response. It is a core element in modern security operation centers used for security management and compliance.
This project aims to deploy a portable intrusion detection system for security information and event management in a real network environment to search for vulnerabilities.

## A.1 Related Work

Many studies have been conducted for system security. One such study by Yousef Hashem et al.[1] αξιολογούν τα evaluates Wazuh, AuditD, and Falco tools based on a set of requirements defined by Ericsson, including flexibility, scalability, and reliability, setting performance evaluation metrics with active background operations.
Results indicate that with proper configuration, Wazuh can be used as an intrusion detection system in embedded systems with limited hardware. AuditD and Falco serve as excellent additions to detect compromise indicators. The solution used a minimal set of intrusion detection rules and activates more modules upon suspicious activity, increasing

threat detection at the cost of CPU and runtime overhead.
Unlike previous studies, this work uses Wazuh to log events for vulnerability identification in the network. Besides general security research, vulnerability detection has also been investigated. A study by Fuad Mat Isa et al. [2] aimed to provide a comprehensive analysis of two products' security and performance indicators on two different operating systems. Experiments evaluated the impact of open-source intrusion detection and prevention systems. Snort and Suricata run on Windows and Linux. Types of attacks and system resource use such as drop rate and intrusion detection capability served as benchmarks. Results showed Snort performs better on Linux for intrusion detection compared to Suricata. On Windows platforms, Snort shows lower detection than its Linux execution. The opposite occurred with Suricata, where Linux configuration failed to detect any attacks. Linux executions also consume more system resources than Windows based ones. This study uses only Suricata combined with Wazuh for log collection.
Other research uses systems for event management through machine learning. One study [3] creates SIEM based on live analysis using machine learning on an IDS. Implementing such live analysis techniques in IDS with machine learning embedded in SIEM requires a combined system with multiple orchestrated processes and services.
Particle choice is required for components creating live analysis in IDS using ML embedded in SIEM. An open-source system is required for easy industrial deployment. This research builds the system using common open-source components for live analysis, detection, and monitoring of cyberattacks.
This research uses a combination of Elastic (ELK) Stack, Splunk, and Zeek IDS for system construction. Measurement ensures chosen components are robust and reliable, focusing on CPU and RAM consumption, measuring performance with a Denial of Service (DoS) test packet rate of 344,1/sec.Elasticsearch consumes the most CPU (78%) and 2300 Mb RAM, while Zeek uses the least (3.5% CPU and 225 Mb RAM). The system detected DoS network attacks during tests.
In contrast to earlier studies, this research focuses on network protocol attacks by using tools listed in that study.

## A.2 Contribution

This research, beyond technical results, aims to provide a fundamental tool for system and information security. It will be used in a real network to ensure the integrity of results. The Alpha Educational Institute gave permission to use its systems, computers, and network for the research.

## A.3 Indicative Structure

Remaining research covers technical methodologies, attack scenarios executed, and final conclusions. Chapter 2 concerns IDPS and SIEM implementation, section 2.1 covers system architecture and 2.2 covers tools used. Chapter 3 presents execution of a real network attack scenario including commands used, and chapter 4 contains final conclusions and future prospects.

# Chapter B

# Implementation of IDPS and SIEM

This chapter covers network architecture and tools used for experimentation.

## B.1 Topology - Architecture

The system architecture (Figure B.1) consists of 5 Windows systems with different versions as victims, the Wazuh[5] system for log collection, Suricata[4] generating log rules, Slack for threat notifications above a certain level, and Kali Linux as the attacker.

## B.2 Εργαλεία

Tools used include:

- Wazuh VM
- Suricata
- Agents

### B.2.1 Wazuh VM

Wazuh[6] is an open-source security platform providing intrusion detection, threat hunting, log management, and compliance monitoring. It consists of components like Agents,

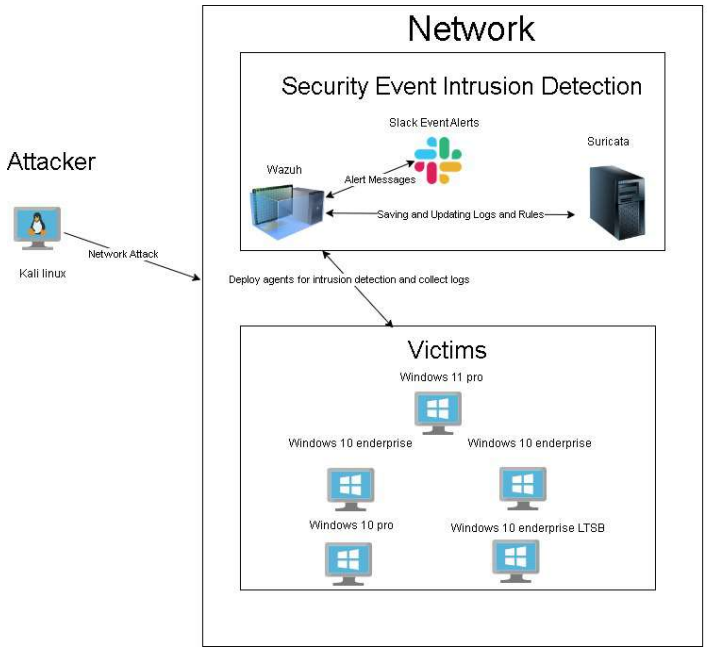


**Figure B.1:** *Schematic Illustration of Network and Attack*

the Wazuh server, Indexer, and Dashboard.

Το Wazuh αποτελείται από τα ακόλουθα στοιχεία όπως φαίνεται και στην εικόνα B.2': Agents: Lightweight software packages installed on monitored infrastructure and endpoints, collecting system logs, configuration files, and events specified in /var/ossec/etc/ossec.conf. Agents send collected security events continuously to the Wazuh server for analysis via secure TLS channels. Agents can be configured for active response on threats.

Διακομιστής: Monitors all agents, configures and updates them, collects and analyzes data, checks if security events trigger pre-configured threat rules, and can be clustered for load balancing.

Indexer: Scalable full-text search and analysis engine storing security events from the server as JSON documents, enabling near real-time search and visualization through a dashboard or REST API.

Dashboard: Central GUI presenting collected data clearly, monitoring Wazuh status and summarized security events, and providing API console for interaction with Wazuh REST API.

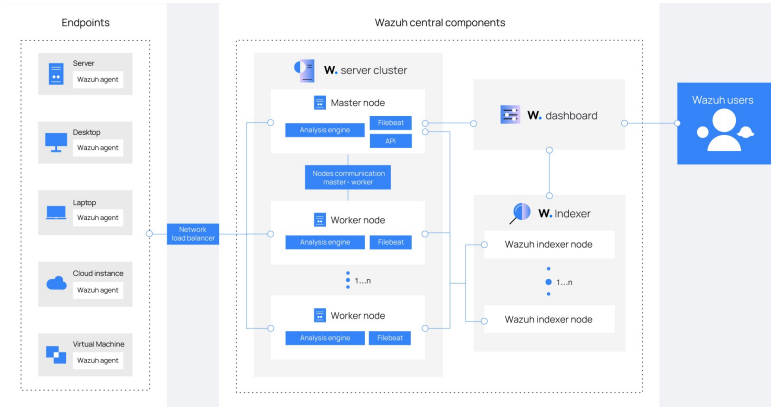


**Figure B.2:** *Schematic Illustration of Wazuh Architecture*

### B.2.2 Suricata

Suricata is a free and open-source solution providing IDS, IPS, Network Security Monitoring (NSM), and offline packet capture processing. Suricata outputs in JSON or YAML format, facilitating integration with Wazuh and ELK stack. It supports logging, sniffer, and intrusion detection modes and features automatic protocol detection on ports minimizing configuration workload.

### B.2.3 Agents

Agents must be installed on victim systems to properly log events. Wazuh supports endpoints on systems like Windows. Agents provide features such as:

- Log collector
- Command execution
- File integrity monitoring (FIM)
- Security configuration assessment (SCA)
- System inventory

- Malware detection
- Active response
- Container security
- Cloud security

# Chapter C

# Scenario Execution

This chapter executes network attack scenarios to detect vulnerabilities and open ports, monitor data transfer, and observe Slack notifications. Kali Linux virtual machine is used as the attacker.

## C.1 Step 1 - Scan of IP Address

Identifying IP addresses connected to the network and information on gateways using the netdiscover and nmap tools.

```
6 Captured ARP Req/Rep packets, from 6 hosts.   Total size: 360
-------------------------------------------------------------------------------

  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-------------------------------------------------------------------------------

192.168.1.1     08:aa:89:41:da:d0      1      60  zte corporation
192.168.1.13    50:3e:aa:4c:6e:da      1      60  TP-LINK TECHNOLOGIES CO.,
192.168.1.41    34:e6:ad:07:66:a6      1      60  Intel Corporate
192.168.1.79    c8:60:00:05:49:10      1      60  ASUSTek COMPUTER INC.
192.168.1.69    10:0b:a9:24:cd:44      1      60  Intel Corporate
```

```
192.168.1.142    08:00:27:54:a5:c4      1      60  PCS Systemtechnik ↩
    GmbH
```

Several IPs detected with open TCP ports, facilitating network attacks such as packet sniffing.

```
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-15 09:22 EDT
Nmap scan report for H1600V7.home (192.168.1.1)
Host is up (0.0034s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
52869/tcp open  unknown
MAC Address: 08:AA:89:41:DA:D0 (zte)

Nmap scan report for 192.168.1.13 (192.168.1.13)
Host is up (0.00086s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT     STATE SERVICE
7070/tcp open  realserver
MAC Address: 50:3E:AA:4C:6E:DA (Tp-link Technologies)

Nmap scan report for 192.168.1.41 (192.168.1.41)
Host is up (0.0042s latency).
All 1000 scanned ports on 192.168.1.41 (192.168.1.41) are in ignored ↩
    states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 34:E6:AD:07:66:A6 (Intel Corporate)

Nmap scan report for 192.168.1.69 (192.168.1.69)
Host is up (0.0084s latency).
All 1000 scanned ports on 192.168.1.69 (192.168.1.69) are in ignored ↩
    states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 10:0B:A9:24:CD:44 (Intel Corporate)
```

```
Nmap scan report for 192.168.1.79 (192.168.1.79)
Host is up (0.0042s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
2869/tcp open  icslap
7070/tcp open  realserver
MAC Address: C8:60:00:05:49:10 (Asustek Computer)

Nmap scan report for 192.168.1.142 (192.168.1.142)
Host is up (0.0076s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE
22/tcp  open  ssh
111/tcp open  rpcbind
443/tcp open  https
MAC Address: 08:00:27:54:A5:C4 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.1.135 (192.168.1.135)
Host is up (0.000022s latency).
All 1000 scanned ports on 192.168.1.135 (192.168.1.135) are in ↩
   ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 256 IP addresses (7 hosts up) scanned in 17.67 seconds
```

## C.2 Step 2 - Specific IP Scanning

Targeted scanning on specific IPs using advanced nmap scanning, detecting open ports, running services, and operating system details. Wazuh detected threats and sent Slack notifications accordingly.

```
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 11:54
```

```
Completed NSE at 11:54, 0.00s elapsed
Initiating NSE at 11:54
Completed NSE at 11:54, 0.00s elapsed
Initiating NSE at 11:54
Completed NSE at 11:54, 0.00s elapsed
Initiating ARP Ping Scan at 11:54
Scanning 192.168.1.142 [1 port]
Completed ARP Ping Scan at 11:54, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:54
Completed Parallel DNS resolution of 1 host. at 11:54, 0.00s elapsed
Initiating SYN Stealth Scan at 11:54
Scanning 192.168.1.142 (192.168.1.142) [1000 ports]
Discovered open port 443/tcp on 192.168.1.142
Discovered open port 111/tcp on 192.168.1.142
Discovered open port 22/tcp on 192.168.1.142
Completed SYN Stealth Scan at 11:54, 0.96s elapsed (1000 total ports)
Initiating Service scan at 11:54
Scanning 3 services on 192.168.1.142 (192.168.1.142)
Completed Service scan at 11:54, 18.34s elapsed (3 services on 1 host↩
    )
Initiating OS detection (try #1) against 192.168.1.142 ↩
    (192.168.1.142)
NSE: Script scanning 192.168.1.142.
Initiating NSE at 11:54
Completed NSE at 11:54, 7.69s elapsed
Initiating NSE at 11:54
Completed NSE at 11:54, 1.27s elapsed
Initiating NSE at 11:54
Completed NSE at 11:54, 0.00s elapsed
Nmap scan report for 192.168.1.142 (192.168.1.142)
Host is up (0.0032s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE   VERSION
22/tcp  open  ssh       OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 e760c09009962f2e95fc3c2d5b3d371f (RSA)
|   256 d0417d2e2a98aa9a9b94ecf071af2bbc (ECDSA)
|_  256 454d3abbdd18951ce0870b5d4dabe9c7 (ED25519)
```

```
111/tcp open  rpcbind   2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|_  100000  3,4          111/udp6  rpcbind
443/tcp open  ssl/https
| tls-alpn:
|_  http/1.1
| ssl-cert: Subject: commonName=wazuh-dashboard/organizationName=↩
   Wazuh/countryName=US
| Subject Alternative Name: IP Address:127.0.0.1
| Issuer: organizationName=Wazuh
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2023-05-15T11:58:28
| Not valid after:  2033-05-12T11:58:28
| MD5:   24021789f79aaefcee00be3fe82b873f
|_SHA-1: 48c332400df1b3dbb6cb9eb51354a9a36b1c561a
|_ssl-date: TLS randomness does not represent time
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
| fingerprint-strings:
|   DNSStatusRequestTCP, DNSVersionBindReqTCP, Kerberos, RPCCheck, ↩
   RTSPRequest, TLSSessionReq, TerminalServerCookie, tor-versions:
|     HTTP/1.1 400 Bad Request
|   FourOhFourRequest:
|     HTTP/1.1 401 Unauthorized
|     osd-name: wazuh-server
|     x-frame-options: sameorigin
|     content-type: application/json; charset=utf-8
|     cache-control: private, no-cache, no-store, must-revalidate
|     set-cookie: security_authentication=; Max-Age=0; Expires=Thu, ↩
   01 Jan 1970 00:00:00 GMT; Secure; HttpOnly; Path=/
|     content-length: 77
|     Date: Wed, 14 Jun 2023 15:54:23 GMT
```

```
|     Connection: close
|     { statusCode :401, error : Unauthorized , message : ↩
   Authentication required }
|   GetRequest:
|     HTTP/1.1 302 Found
|     location: /app/login?
|     osd-name: wazuh-server
|     x-frame-options: sameorigin
|     cache-control: private, no-cache, no-store, must-revalidate
|     set-cookie: security_authentication=; Max-Age=0; Expires=Thu, ↩
   01 Jan 1970 00:00:00 GMT; Secure; HttpOnly; Path=/
|     content-length: 0
|     Date: Wed, 14 Jun 2023 15:54:23 GMT
|     Connection: close
|   HTTPOptions:
|     HTTP/1.1 404 Not Found
|     osd-name: wazuh-server
|     x-frame-options: sameorigin
|     content-type: application/json; charset=utf-8
|     cache-control: private, no-cache, no-store, must-revalidate
|     content-length: 60
|     Date: Wed, 14 Jun 2023 15:54:23 GMT
|     Connection: close
|_    { statusCode :404, error : Not Found , message : Not Found }
| http-title: Wazuh
|_Requested resource was /app/login?
1 service unrecognized despite returning data. If you know the ↩
   service/version, please submit the following fingerprint at https↩
   ://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port443-TCP:V=7.93%T=SSL%I=7%D=6/14%Time=6489E2A9%P=x86_64-pc-↩
   linux-gnu
SF:%r(GetRequest,15C, HTTP/1\.1\x20302\x20Found\r\nlocation:\x20/app/↩
   login
SF:\?\r\nosd-name:\x20wazuh-server\r\nx-frame-options:\x20sameorigin\↩
   r\nca
SF:che-control:\x20private,\x20no-cache,\x20no-store,\x20must-↩
   revalidate\r
SF:\nset-cookie:\x20security_authentication=;\x20Max-Age=0;\↩
```

```
    x20Expires=Thu
SF:,\x2001\x20Jan\x201970\x2000:00:00\x20GMT;\x20Secure;\x20HttpOnly↩
    ;\x20P
SF:ath=/\r\ncontent-length:\x200\r\nDate:\x20Wed,\x2014\x20Jun\↩
    x202023\x20
SF:15:54:23\x20GMT\r\nConnection:\x20close\r\n\r\n )%r(HTTPOptions↩
    ,143, HT
SF:TP/1\.1\x20404\x20Not\x20Found\r\nosd-name:\x20wazuh-server\r\nx-↩
    frame-
SF:options:\x20sameorigin\r\ncontent-type:\x20application/json;\↩
    x20charset
SF:=utf-8\r\ncache-control:\x20private,\x20no-cache,\x20no-store,\↩
    x20must-
SF:revalidate\r\ncontent-length:\x2060\r\nDate:\x20Wed,\x2014\x20Jun\↩
    x2020
SF:23\x2015:54:23\x20GMT\r\nConnection:\x20close\r\n\r\n{\ statusCode↩
    \ :40
SF:4,\ error\ :\ Not\x20Found\ ,\ message\ :\ Not\x20Found\ } )%r(↩
    FourOhFo
SF:urRequest,1C9, HTTP/1\.1\x20401\x20Unauthorized\r\nosd-name:\↩
    x20wazuh-s
SF:erver\r\nx-frame-options:\x20sameorigin\r\ncontent-type:\↩
    x20application
SF:/json;\x20charset=utf-8\r\ncache-control:\x20private,\x20no-cache↩
    ,\x20n
SF:o-store,\x20must-revalidate\r\nset-cookie:\↩
    x20security_authentication=;
SF:\x20Max-Age=0;\x20Expires=Thu,\x2001\x20Jan\x201970\x2000:00:00\↩
    x20GMT;
SF:\x20Secure;\x20HttpOnly;\x20Path=/\r\ncontent-length:\x2077\r\↩
    nDate:\x2
SF:0Wed,\x2014\x20Jun\x202023\x2015:54:23\x20GMT\r\nConnection:\↩
    x20close\r
SF:\n\r\n{\ statusCode\ :401,\ error\ :\ Unauthorized\ ,\ message↩
    \ :\ Auth
SF:entication\x20required\ } )%r(tor-versions,1C, HTTP/1\.1\x20400\↩
    x20Bad\
SF:x20Request\r\n\r\n )%r(RTSPRequest,1C, HTTP/1\.1\x20400\x20Bad\↩
```

```
    x20Reque
SF:st\r\n\r\n )%r(RPCCheck,1C, HTTP/1\.1\x20400\x20Bad\x20Request\r\n↩
    \r\n
SF:)%r(DNSVersionBindReqTCP,1C, HTTP/1\.1\x20400\x20Bad\x20Request\r\↩
    n\r\n
SF: )%r(DNSStatusRequestTCP,1C, HTTP/1\.1\x20400\x20Bad\x20Request\r\↩
    n\r\n
SF: )%r(TerminalServerCookie,1C, HTTP/1\.1\x20400\x20Bad\x20Request\r↩
    \n\r\
SF:n )%r(TLSSessionReq,1C, HTTP/1\.1\x20400\x20Bad\x20Request\r\n\r\n↩
     )%r(
SF:Kerberos,1C, HTTP/1\.1\x20400\x20Bad\x20Request\r\n\r\n );
MAC Address: 08:00:27:54:A5:C4 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Uptime guess: 0.127 days (since Wed Jun 14 08:52:17 2023)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=261 (Good luck!)
IP ID Sequence Generation: All zeros

TRACEROUTE
HOP RTT     ADDRESS
1   3.19 ms 192.168.1.142 (192.168.1.142)

NSE: Script Post-scanning.
Initiating NSE at 11:54
Completed NSE at 11:54, 0.00s elapsed
Initiating NSE at 11:54
Completed NSE at 11:54, 0.00s elapsed
Initiating NSE at 11:54
Completed NSE at 11:54, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect ↩
    results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 31.31 seconds
           Raw packets sent: 1023 (45.806KB) | Rcvd: 1015 (41.290KB)
```

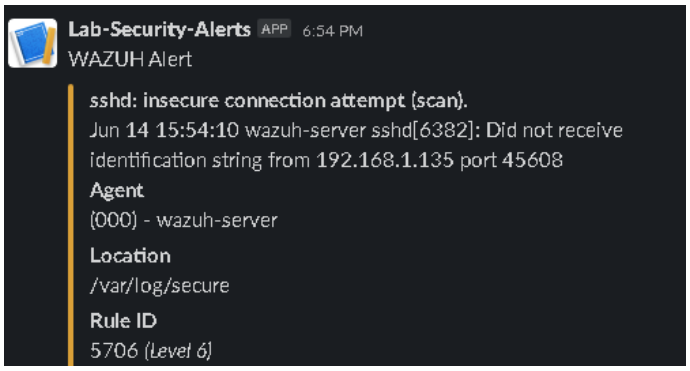


**Figure C.3:** *Threat Notification from Wazuh Server via Slack*

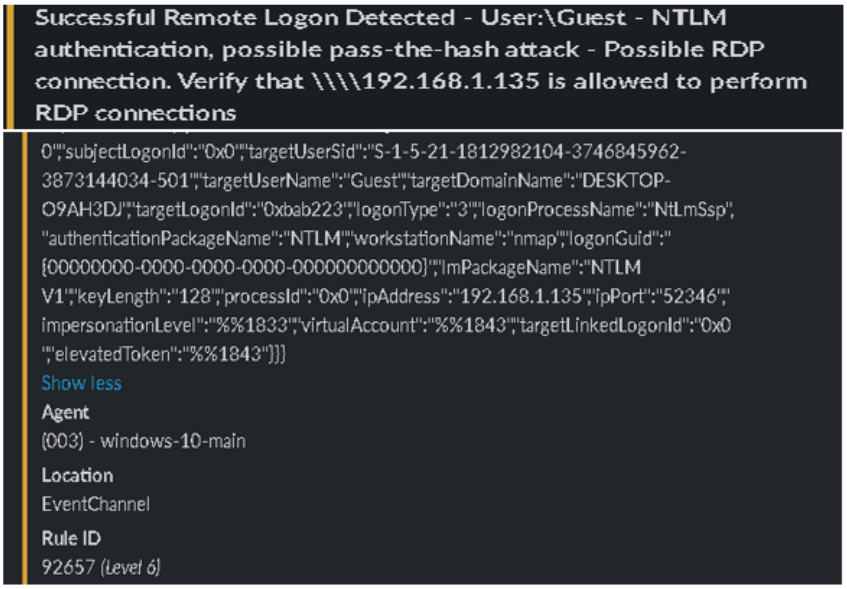


**Figure C.4:** *Threat Notification from Victim Computer via Slack*

## C.3 Step 3 - SSH-Brute Force Attack

Attempted brute force attack on Wazuh server's SSH to simulate a distributed denial of service (DDoS) attack. Wazuh classified the attack at level 10 severity and notified via Slack. Commands and attack details are logged.

```
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-14 12:25 EDT
NSE: [ssh-brute] Trying username/password pair: root:root
```

```
NSE: [ssh-brute] Trying username/password pair: admin:admin
NSE: [ssh-brute] Trying username/password pair: administrator:↩
    administrator
NSE: [ssh-brute] Trying username/password pair: webadmin:webadmin
NSE: [ssh-brute] Trying username/password pair: sysadmin:sysadmin
NSE: [ssh-brute] Trying username/password pair: netadmin:netadmin
NSE: [ssh-brute] Trying username/password pair: guest:guest
NSE: [ssh-brute] Trying username/password pair: user:user
NSE: [ssh-brute] Trying username/password pair: web:web
NSE: [ssh-brute] Trying username/password pair: test:test
NSE: [ssh-brute] Trying username/password pair: root:
NSE: [ssh-brute] Trying username/password pair: admin:
NSE: [ssh-brute] Trying username/password pair: administrator:
NSE: [ssh-brute] Trying username/password pair: webadmin:
NSE: [ssh-brute] Trying username/password pair: sysadmin:
NSE: [ssh-brute] Trying username/password pair: netadmin:
NSE: [ssh-brute] Trying username/password pair: guest:
NSE: [ssh-brute] Trying username/password pair: user:
NSE: [ssh-brute] Trying username/password pair: web:
NSE: [ssh-brute] Trying username/password pair: test:
NSE: [ssh-brute] Trying username/password pair: root:123456
NSE: [ssh-brute] Trying username/password pair: admin:123456
NSE: [ssh-brute] Trying username/password pair: administrator:123456
NSE: [ssh-brute] Trying username/password pair: webadmin:123456
```

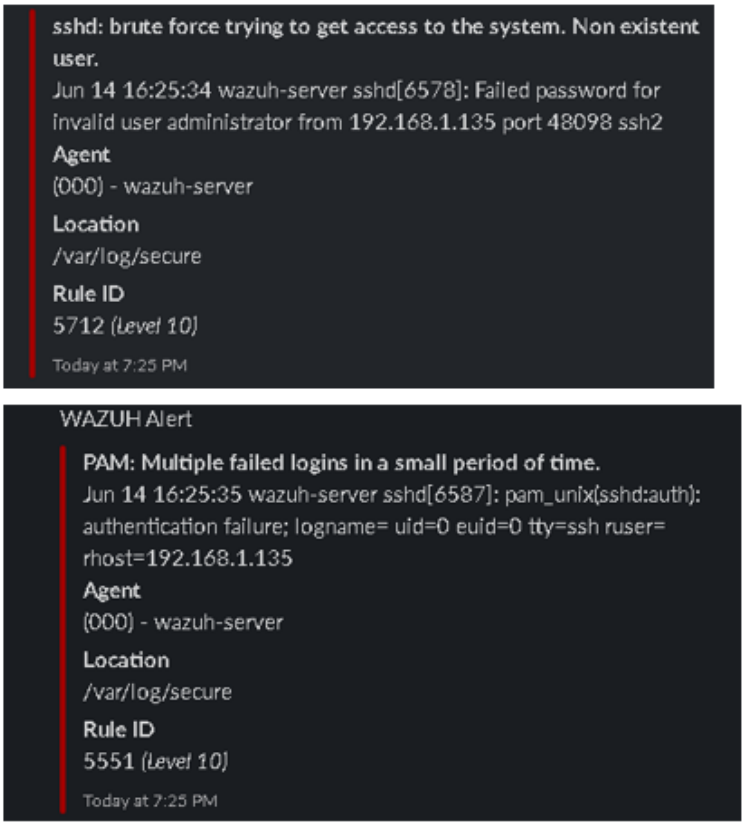


**Figure C.5:** *Level 10 Threat Notification from Wazuh Server via Slack*

# Chapter D

# Conclusions

The attack results show security gaps in the TCP communication protocol allowing easy access to network communication. Similar attacks on UDP communication channels are possible, requiring extensive monitoring and security guides for TCP and UDP packet handling. Measures to protect against hacking attempts aiming to acquire sensitive information are essential to prevent personal and sensitive data leaks.

## D.1 Future Extensions

Based on these results, Alpha Educational Institute must fix communication channel security gaps and other security aspects. Full system deployment is planned for continuous daily security monitoring and fixes. Additionally, monitoring of Microsoft 365 cloud services is requested to enhance service security awareness.

# Bibliography


[1] Yousef Hashem and Elmedin Zildzic. *Endpoint Intrusion Detection and Response Agents in Embedded RAN Products: A suitability and performance evaluation*. 2022.

[2] Fuad Mat Isa et al. "Comprehensive performance assessment on open source intrusion detection system". In: *Proceedings of the Third International Conference on Computing, Mathematics and Statistics (iCMS2017) Transcending Boundaries, Embracing Multidisciplinary Diversities*. Springer. 2019, pp. 45–51.

[3] Adabi Raihan Muhammad, Parman Sukarno, and Aulia Arif Wardana. "Integrated Security Information and Event Management (SIEM) with Intrusion Detection System (IDS) for Live Analysis based on Machine Learning". In: *Procedia Computer Science* 217 (2023), pp. 1406–1415.

[4] Rehan Shams et al. "Comparative Analysis Of Intrusion Detection Systems in SDN". In: *2023 Global Conference on Wireless and Optical Technologies (GCWOT)*. 2023, pp. 1–9. DOI: 10.1109/GCWOT57803.2023.10064664.

[5] Stefan Stanković, Slavko Gajin, and Ranko Petrović. "A Review of Wazuh Tool Capabilities for Detecting Attacks Based on Log Analysis". In: ().

[6] Aamna Tariq et al. "Open source SIEM solutions for an enterprise". In: *Information & Computer Security* 31.1 (2023), pp. 88–107.