ISTANBUL TECHNICAL UNIVERSITY
Faculty of Computer Science and Informatics

CYBER SECURITY ENGINEERING

INTERNSHIP PROGRAM REPORT

HAKAN DURAN
150200091

FALL / 2025

Table of Contents

1 INFORMATION ABOUT THE INSTITUTION

Turkcell is a leading information technology and communications company, primarily focused on delivering innovative technology solutions to its customers. Its clients includes both individual and institutional consumers of GSM and internet services [1]. As one of the dominant players in Turkey's GSM and internet sectors, Turkcell has earned a strong reputation. The company operates two key branches in Istanbul: one in Taksim and the other in Küçükyalı, where I had the opportunity to complete my internship.

Turkcell, various teams work together seamlessly, including software developers, business analysts, HR professionals, system administrators, network specialists, cloud engineers, and cybersecurity experts. The company is led by CEO Ali Taha Koç [2], with Şenol Kazancı serving as the Chairman of the Board [3].

I completed my internship in the Defensive Cybersecurity Operations Department at Turkcell. The primary objective of this department is to provide defensive security for Turkcell, its subsidiary companies, and its customers. The department consists of three teams, each with distinct responsibilities: Digital Forensics and Incident Responders, System Administrators, and Cybersecurity Analysts. I was part of the Cybersecurity Analysts team, and working with this group taught me a great deal about cybersecurity, the engineering of cybersecurity solutions, and the development of essential cybersecurity software.

As part of my role, my department provided me with a VLAN, which included a switch and servers within the network. I created a virtual environment for detection engineering by collecting and decoding logs, as well as developing rulesets to identify potential threat factors in my internship.

2 INTRODUCTION

During my internship, I had the chance to gain practical experience with Security Information and Event Management (SIEM) [4], Security Orchestration, Automation, and Response (SOAR) technologies, as well as EDRs, IDS/IPSs, and DLPs. I was involved in analyzing and investigating security incidents, as well as developing playbooks for alarm analysis. I also created algorithms to efficiently store malicious IP addresses and built a virtual environment for detection engineering.

3 DESCRIPTION AND ANALYSIS OF THE INTERNSHIP PROJECT

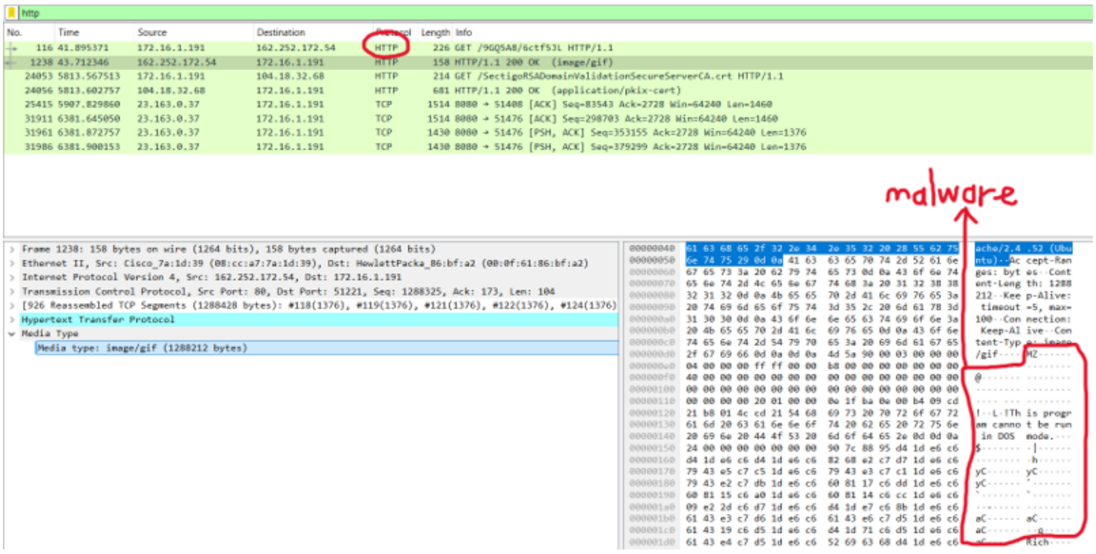
3.1 Incident Review

During my internship, my primary responsibility was handling incident reviews, a significant task in ensuring the security of Turkcell's network. My workday began at 7:00 AM and ended at 3:00 PM, with a meeting with the manager following the shift. One of the key tools I used was XSOAR, which brought in alerts from ArcSight ESM. These alerts provided valuable information, including fields such as source IP, destination IP, source and destination ports, device hostname, username, executed commands, request URLs, and more. Each alert provided essential data that I had to carefully analyze in order to assess its severity.

Upon receiving these alerts, I conducted detailed investigations to determine whether they were false positives or true positives. A false positive would indicate a harmless event, while a true positive signified a real security threat. In some cases, I needed to contact server owners directly, either via email or phone, to confirm whether they had initiated certain actions that triggered an alert.

If I identified malicious addresses scanning our network, I took immediate action by banning those addresses. I would then investigate whether their scanning attempts were successful, checking for signs of data exfiltration or unauthorized access. In cases where I detected Command and Control (C2) addresses, I added them to the C2 database to prevent further compromise.

Another critical aspect of my role involved responding to malware incidents. When a malware alert was raised, I performed antivirus scans, isolated the affected systems, and checked for any lateral movements within the network. This process was essential to ensure that the malware did not spread to other systems and that it was fully contained.



```
No.    Time           Source          Destination     Protocol Length Info
  116  41.895371      172.16.1.191    162.252.172.54  HTTP      226 GET /9GQ5A8/6ctf53L HTTP/1.1
 1238  43.712346      162.252.172.54  172.16.1.191    HTTP      158 HTTP/1.1 200 OK  (image/gif)
24053  5813.567513    172.16.1.191    104.18.32.68    HTTP      214 GET /SectigoRSADomainValidationSecureServerCA.crt HTTP/1.1
24056  5813.602757    104.18.32.68    172.16.1.191    HTTP      681 HTTP/1.1 200 OK  (application/pkix-cert)
25415  5907.829860    23.163.0.37     172.16.1.191    TCP      1514 8080 → 51408 [ACK] Seq=83543 Ack=2728 Win=64240 Len=1460
31911  6381.645050    23.163.0.37     172.16.1.191    TCP      1514 8080 → 51476 [ACK] Seq=298703 Ack=2728 Win=64240 Len=1460
31961  6381.872757    23.163.0.37     172.16.1.191    TCP      1430 8080 → 51476 [PSH, ACK] Seq=353155 Ack=2728 Win=64240 Len=1376
31986  6381.900153    23.163.0.37     172.16.1.191    TCP      1430 8080 → 51476 [PSH, ACK] Seq=379299 Ack=2728 Win=64240 Len=1376
```

```
> Frame 1238: 158 bytes on wire (1264 bits), 158 bytes captured (1264 bits)
> Ethernet II, Src: Cisco_7a:1d:39 (08:cc:a7:7a:1d:39), Dst: HewlettPacka_86:bf:a2 (00:0f:61:86:bf:a2)
> Internet Protocol Version 4, Src: 162.252.172.54, Dst: 172.16.1.191
> Transmission Control Protocol, Src Port: 80, Dst Port: 51221, Seq: 1288325, Ack: 173, Len: 104
> [926 Reassembled TCP Segments (1288428 bytes): #118(1376), #119(1376), #121(1376), #122(1376), #124(1376)
> Hypertext Transfer Protocol
v Media Type
    Media type: image/gif (1288212 bytes)
```

```
00000040  61 63 68 65 2f 32 2e 34  2e 35 32 20 28 55 62 75   ache/2.4 .52 (Ubu
00000050  6e 74 75 29 0d 0a 41 63  63 65 70 74 2d 52 61 6e   ntu)··Ac cept-Ran
00000060  67 65 73 3a 20 62 79 74  65 73 0d 0a 43 6f 6e 74   ges: byt es··Cont
00000070  65 6e 74 2d 4c 65 6e 67  74 68 3a 20 31 32 38 38   ent-Leng th: 1288
00000080  32 31 32 0d 0a 4b 65 65  70 2d 41 6c 69 76 65 3a   212··Kee p-Alive:
00000090  20 74 69 6d 65 6f 75 74  3d 35 2c 20 6d 61 78 3d    timeout =5, max=
000000a0  31 30 30 0d 0a 43 6f 6e  6e 65 63 74 69 6f 6e 3a   100··Con nection:
000000b0  20 4b 65 65 70 2d 41 6c  69 76 65 0d 0a 43 6f 6e    Keep-Al ive··Con
000000c0  74 65 6e 74 2d 54 79 70  65 3a 20 69 6d 61 67 65   tent-Typ e: image
000000d0  2f 67 69 66 0d 0a 0d 0a  4d 5a 90 00 03 00 00 00   /gif···· MZ······
000000e0  04 00 00 00 ff ff 00 00  b8 00 00 00 00 00 00 00   ········ ········
000000f0  40 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   @······· ········
00000100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00000110  00 00 00 00 20 01 00 00  0e 1f ba 0e 00 b4 09 cd   ········ ········
00000120  21 b8 01 4c cd 21 54 68  69 73 20 70 72 6f 67 72   !··L·!Th is progr
00000130  61 6d 20 63 61 6e 6e 6f  74 20 62 65 20 72 75 6e   am canno t be run
00000140  20 69 6e 20 44 4f 53 20  6d 6f 64 65 2e 0d 0d 0a    in DOS  mode.···
00000150  24 00 00 00 00 00 00 00  90 7c 88 95 d4 1d e6 c6   $······· ·|······
00000160  d4 1d e6 c6 d4 1d e6 c6  82 68 e2 c7 d7 1d e6 c6   ········ ·h······
00000170  79 43 e5 c7 c5 1d e6 c6  79 43 e3 c7 c1 1d e6 c6   yC······ yC······
00000180  79 43 e2 c7 db 1d e6 c6  60 81 17 c6 dd 1d e6 c6   yC······ `·······
00000190  60 81 15 c6 a0 1d e6 c6  60 81 14 c6 cc 1d e6 c6   `······· `·······
000001a0  09 e2 2d c6 d7 1d e6 c6  d4 1d e7 c6 8b 1d e6 c6   ··-····· ········
000001b0  61 43 e3 c7 d6 1d e6 c6  61 43 e6 c7 d5 1d e6 c6   aC······ aC······
000001c0  61 43 19 c6 d5 1d e6 c6  d4 1d 71 c6 d5 1d e6 c6   aC······ ··q·····
000001d0  61 43 e4 c7 d5 1d e6 c6  52 69 63 68 d4 1d e6 c6   aC······ Rich····
```

Figure 1: A malware can be found at PCAP file in our virtual training environment.

As it can be seen from the figure 1, there is a GET request made from 172.16.1.191 address to 162.252.172.54 address. Upon investigation, it was identified that the request contained a malware payload disguised as an image with the MIME type image/gif, demonstrating a defense evasion technique commonly employed by threat actors. The first step was to confirm whether the malware had been successfully delivered to the target system. This involved analyzing the response from the server and checking for signs of successful execution or persistence on the infected machine. It was critical to determine if any sensitive data had been exfiltrated during the attack. The network traffic was monitored for any signs of unusual outbound data transfers, which could indicate that the attacker was attempting to steal sensitive information from the compromised system. Catching and responding to such types of malware is significant because these evasion techniques can cause significant harm to the network. They can lead to data breaches, system compromise, and even further lateral movement within the network, endangering the integrity and confidentiality of sensitive information.

Phishing attacks were also a significant concern. Company employees reported phishing emails that they suspected might be malicious. If these suspicions were confirmed, I immediately took action by deleting the phishing email from the email repositories of all users to prevent anyone from clicking the malicious link by mistake. Additionally, I added the sender's email address to the email gateway blacklist and the phishing URL to the IPS blacklist. This ensured that no one could access the phishing site, even if they clicked on the link, and prevented the sender from sending future phishing emails.

Another part of my responsibilities involved investigating high web requests and unusual or bad requests targeting our web services. These could indicate attempts by malicious actors to discover vulnerabilities. By reviewing these requests, I was able to identify potential threats and mitigate any risks to the integrity of our web services.

Overall, the incident review process required thorough attention to detail, fast decision-making, and effective communication with other teams within the company. By efficiently managing these alerts and incidents, I played an integral role in maintaining the security and stability of Turkcell's network.

3.2 Sustaining Stability of System

In addition to managing security incidents, I was also responsible for ensuring the stability and functionality of Turkcell's defensive systems. Our systems collected logs from various sources, including servers, Active Directory (AD), EDRs, IPSs, firewalls, and DLP systems. Among these logs, key ones like audit logs and syslog were essential for identifying potential issues and maintaining system integrity [5].

Once the logs were collected, they were sent to our Security Information and Event Management (SIEM) system for further processing. Parsing these logs involved breaking down the data into manageable and readable components, often converting complex entries into structured formats that could be easily analyzed. For example, syslog logs provided information about system events, while auditd logs captured security-relevant events, and journald logs helped track system processes and their statuses. Parsing these logs accurately was critical for identifying patterns or anomalies.

Following parsing, the next step was log correlation and aggregation. This process involved linking related events from different sources to provide a comprehensive view of system activity. Log correlation helped to group similar alerts together, reducing noise and highlighting potential threats that required attention. Aggregation further helped by combining multiple alerts into a single event, streamlining the analysis and response process. This made it easier to identify trends, such as repeated malicious IP addresses or recurring abnormal activities, across various systems and devices.

Despite having a robust system in place, log-related issues occasionally arose. The most significant challenges were log receiving problems. These issues could stem from various causes, such as firewall misconfigurations, port mismatches, streaming failures, or heavy log loads that put strain on the system. As an engineer, my responsibility was to troubleshoot and resolve these problems efficiently.

My approach to troubleshooting began with identifying the root cause of the issue. If the problem was related to server load, I would increase the server's computational resources to handle the larger volume of incoming logs. In some cases, problems arose due to Docker cache issues, which could also be resolved by clearing or managing the cache more effectively. However, if the problem was system-related and required root privileges to fix, I would escalate the issue to the server administrator for assistance.

At times, the issue was outside of our system, such as problems with the firewall or the customer's network. In these cases, it was important to communicate with the relevant teams or customers to troubleshoot the problem together and find a solution.

Another common cause of log receiving problems was suppression. When there were too many alerts triggering without a way to stop them, a suppression list would be added to the rule. This helped reduce the volume of alerts, especially when known false positives were causing unnecessary noise in the system. Suppression was useful for preventing the SIEM from becoming overwhelmed with repetitive alerts. However, if not managed properly, this suppression could lead to serious issues over time. If the suppression list was not regularly reviewed or adjusted, it could result in critical alerts being missed or ignored. To prevent this, it was essential to monitor and, if necessary, adjust or remove the suppression list to ensure that important security events were not overlooked.

Maintaining the stability of the system required a balance between ensuring that the logs were properly collected and processed, and troubleshooting any issues that arose. My role in this process helped ensure that Turkcell's defensive systems remained reliable and capable of detecting and responding to security threats in a timely manner.

3.3 Playbook Management

A playbook, in the context of cybersecurity and Security Orchestration, Automation, and Response (SOAR) systems, is a predefined, automated process designed to guide analysts through a series of steps to respond to specific security incidents. It provides a structured approach to handling alerts, ensuring consistency, efficiency, and speed in the response process. Playbooks are an essential part of SOAR systems because they not only automate repetitive tasks but also help link related events together, adding an additional layer of aggregation to enhance threat detection and response.

The role of playbooks is important in managing security incidents and ensuring the smooth operation of security teams. They help shift tasks from one team member to another, especially during shift changes such as 15:00, 23:00, and 07:00. Playbooks can automate the transfer of responsibility for cases between analysts, ensuring that no incident is overlooked and that response times are consistent, regardless of the time of day.

Another key function of playbooks is the automation of email communications. Since we provide Security Operations Center (SOC) services to our customers, it is important to send them updates about alerts in their networks. Playbooks streamline this process by generating pre-configured emails, which significantly reduce the time analysts spend drafting emails to customers. By automating this aspect of the communication, playbooks allow analysts to focus on more critical tasks, improving overall efficiency.

As part of my internship, I also developed playbooks using Python. One example of this was creating a playbook that automatically fetched critical information, such as IP addresses, from the alerts and included them in the body of emails sent to customers. This automation made the process much more efficient and ensured that important details were always included in the communications. However, due to company policy, I am unable to share the code I wrote for these playbooks.

In addition to developing playbooks, I was also responsible for analyzing and evaluating their functionality. Occasionally, playbooks would encounter errors that prevented us from viewing important alarm details, which could delay the response process. When this happened, I would create a ticket for the playbook developers through OneDesk, the ticketing system used by our company, to resolve the issue. This ensured that any technical problems with the playbooks were promptly addressed and that the analysts had the tools they needed to perform their duties effectively.

For documentation purposes, we used Confluence to track project progress, document solutions to problems, and share knowledge across teams. This platform was essential for maintaining clear records of playbook management tasks, as well as providing a repository of useful information that could be referenced by the team when troubleshooting issues or developing new playbooks.

Overall, playbook management was a vital part of my role, helping to streamline processes, enhance efficiency, and ensure that security incidents were handled in a consistent and timely manner. The experience I gained in developing, analyzing, and troubleshooting playbooks gave me valuable insights into how automation can improve cybersecurity operations.

3.4 Software Development

During my internship, I had the opportunity to contribute to the software development efforts within the cybersecurity team. One of the key projects I worked on was creating a database system with API integration to manage malicious IP addresses. For this project, I chose MySQL as the database management system due to its stability, ease of use, and flexibility. MySQL is a widely used relational database that supports complex queries and large amounts of data, which was essential for our needs. It is also highly compatible with other technologies, such as APIs, and offers efficient data retrieval and storage capabilities, making it an ideal choice for handling a growing database of malicious IP addresses.

The system I developed was designed to collect malicious IP addresses from various sources, including public threat intelligence services, our service providers, and directly from attacks against our network. These IP addresses were then provided to firewalls and IPS systems, enabling us to proactively defend against potential threats. The system allowed for CRUD (Create, Read, Update, Delete) operations, which meant that new addresses could be added to the database, and if an address was later determined to be harmless, it could be removed from the blacklist. This flexibility was crucial for maintaining an accurate and up-to-date database of malicious addresses.

In addition to managing the database, I was also responsible for integrating the backend with the frontend through an API. This integration allowed users to interact with the system easily, whether they needed to add new addresses, review existing entries, or remove addresses that were no longer considered a threat. The database also tracked additional information, such as the reason for blacklisting an address, who added the address to the blacklist, and the duration for which the address would be blocked. This level of detail ensured that the system could be effectively managed and audited.

Our system also utilized Firebase for authentication purposes. Firebase is a platform developed by Google that provides a range of backend services, including authentication, real-time databases, and cloud storage [6]. In this project, Firebase was used to handle user authentication, ensuring that only authorized personnel could access and manage the database. This added an extra layer of security to the system and helped maintain control over who could make changes to the blacklist.

In addition to the malicious address database project, I was also involved in a deep learning project focused on phishing detection. This was a collaborative effort with the artificial intelligence team, where we used historical data of phishing emails collected over the years in our email repository. Analysts had previously identified these emails as either false positives or true positives, which gave us a rich dataset for training the deep learning model. We used LLAMA, a deep learning framework, to train the model, and by leveraging this dataset, we were able to achieve a 95% accuracy rate in detecting phishing emails. This project demonstrated the power of machine learning in enhancing cybersecurity by automating the detection of phishing attempts, which are a common threat in email-based attacks.

Throughout these projects, I gained valuable experience in both software development and the practical application of cybersecurity measures. I was able to deepen my understanding of database management, API integration, and authentication systems, while also contributing to the development of cutting-edge solutions like the deep learning model for phishing detection. These experiences provided me with a strong foundation in both the technical and collaborative aspects of software development in the cybersecurity field.

3.5 Creating a Training Environment

As part of my internship, I was tasked with creating a training environment to simulate real-world security scenarios. The company provided our team with a dedicated VLAN, which consisted of servers and a switch. This environment was crucial for testing and developing security tools, as well as providing a controlled space for learning and experimentation.

One of the key components in this environment was a server that hosted Wazuh, an open-source security monitoring platform. Wazuh is a powerful tool used for log analysis, intrusion detection, vulnerability detection, and compliance monitoring [7]. It consists of several components that work together to provide comprehensive security monitoring capabilities. The main components of Wazuh include:

- Wazuh Manager: The central component of the system, responsible for processing and analyzing security events. It collects logs from agents installed on other servers and generates alerts based on predefined rules.

- Wazuh Indexer: A component that stores and indexes the security data collected by the Wazuh Manager. This enables fast querying and retrieval of security events and logs.

- Wazuh Dashboard: A web interface that provides an intuitive and visual representation of the security data, allowing analysts to monitor alerts, manage configurations, and run queries.

Setting up Wazuh in the training environment proved to be a challenging task. The server I was working on had no communication with the internet, which made it difficult to perform an online installation. I attempted to set up Wazuh using offline installation methods; however, I encountered issues during the process, and even found related discussions on GitHub where Wazuh developers mentioned they were working on resolving the problem. Despite these setbacks, I still needed to have Wazuh set up for the project, so I decided to try a different approach using Docker.

Docker, a containerization platform, was essential in this situation. It allowed me to package Wazuh and its components into containers, ensuring that they could run in isolated environments without the need for a full installation on the server. Docker made it possible to deploy Wazuh efficiently and consistently, even in the absence of an internet connection. The use of Docker also provided flexibility, as I could easily manage the deployment and ensure that all components of Wazuh were running in separate containers, which made the overall system more manageable and scalable.

To begin the process, I first communicated with the firewall team, as I needed to fetch the Docker image required for the Wazuh installation. After successfully obtaining the Docker image, I used docker-compose to build and deploy Wazuh. Docker Compose is a tool that allows you to define and run multi-container Docker applications, making it easier to set up and manage the various components of Wazuh in the training environment. After completing the installation of Wazuh using Docker, I proceeded to set up Wazuh agents on other servers within the VLAN to collect and send logs to the Wazuh Manager.

Once the agents were running and logs were being collected, I used Wazuh Query Language (WQL) to query and analyze the logs. WQL is a powerful querying tool that allows you to filter and search through large volumes of log data to identify specific events or anomalies. This was especially useful for training and testing, as it enabled me to efficiently extract the information I needed from the logs, such as security alerts or potential threats.

By setting up this training environment, I gained valuable experience in working with security monitoring tools like Wazuh, as well as Docker, which is widely used for containerization and application deployment. This project not only improved my technical skills but also gave me a deeper understanding of how security monitoring systems can be implemented and used to protect networks and systems.

3.6 Technical Challenges

Throughout my internship, I encountered several technical challenges that not only tested my problem-solving skills but also deepened my understanding of system administration, security operations, and the intricacies of managing IT infrastructure.

One of the first challenges I faced involved rebooting a server. I was unaware at the time that rebooting a server can sometimes trigger unexpected problems, especially if the server is not properly prepared for it. After rebooting, I noticed a series of issues arise. I soon learned that rebooting is a critical process, and it's important to notify the team in advance and ensure that all necessary precautions are taken beforehand. I should have reported my intention to reboot the server and made sure that there was no significant impact on the system. Fortunately, the server I rebooted did not contain critical data or functions, so the issues were relatively minor. However, I learned the importance of careful planning and communication before performing such operations.

Another problem occurred when I needed to restart the wazuh.manager service within a Docker container. Unfortunately, I quickly discovered that the usual system commands like systemctl or service were not available inside the container. Without these commands, I had no choice but to restart the entire Docker container. While this action initially seemed to resolve the issue, it later led to a more significant problem: the Wazuh system stopped working after the restart. It turned out that the restart had disrupted the connectivity between Wazuh and its components, leading to failed API connections. This experience highlighted the importance of thoroughly understanding the environment in which applications are running and the impact of restarting services in containerized environments.

A third issue I encountered involved a repository problem during installation. One of the installation scripts I ran added a new repository file in the /etc/yum/ directory for RPM-based package management. Unfortunately, this new file replaced the existing repository configuration, causing the package manager to fail when attempting to install or update packages. I wasn't able to install anything using yum or dnf until this issue was resolved. I reached out to the system administration team, who were able to restore the original repository file and fix the issue. This experience taught me the importance of carefully reviewing installation scripts and configurations to avoid conflicts, especially when dealing with package managers.

Another challenge I faced was with the Offensive Security (OffSec) team, who regularly conducted penetration testing and security scans on Turkcell's network. At times, it was difficult to distinguish between a legitimate threat actor and the OffSec team's activities. Early in my internship, I mistakenly blocked an IP address that appeared to exhibit suspicious behavior. Unfortunately, it turned out to be a legitimate address, and I unintentionally disrupted the OffSec team's testing. They were understanding, recognizing that I was new to the role and that mistakes like this can happen. However, this incident underscored the importance of clear communication with other teams, especially when it comes to identifying potential threats versus authorized security activities.

Lastly, I encountered some issues with the database schema in one of the projects I worked on. The database schema had poor design, and there were inconsistent foreign key relationships that led to errors and data integrity problems. It took some time to identify the root cause of these issues, and even more time to resolve them. I had to carefully analyze the structure of the database and make adjustments to the schema to ensure that the foreign key relationships were properly defined. This process helped me appreciate the importance of good database design and the challenges that arise when working with poorly structured data.

Despite these challenges, each problem was an opportunity for learning and growth. They taught me valuable lessons about system administration, database management, and the complexities of working in a cybersecurity environment. These experiences reinforced the importance of thorough planning, clear communication with colleagues, and the need to remain adaptable when solving technical problems.

4 CONCLUSIONS

During my internship at Turkcell, I gained substantial knowledge and practical experience that enriched both my technical and professional skills. This experience provided me with a deeper understanding of the complexities involved in cybersecurity, system administration, and software development, and allowed me to apply theoretical concepts to real-world situations.

One of the most valuable takeaways from this internship was my exposure to various cybersecurity technologies, including Security Information and Event Management (SIEM), Security Orchestration, Automation, and Response (SOAR), and endpoint detection systems. These tools were pivotal in enabling me to analyze logs, investigate security incidents, and implement defensive measures to protect the company's infrastructure. Through these tasks, I honed my analytical skills and gained a practical understanding of how automated responses and threat detection work in a live environment.

Another key achievement was the development and management of a database system for malicious IP addresses, which improved my skills in backend development, database management, and API integration. By working on this project, I also enhanced my understanding of how proactive defense mechanisms—such as blacklisting malicious addresses—can prevent potential attacks before they occur.

Additionally, the process of creating a training environment using Wazuh and Docker significantly expanded my knowledge of log management, containerization, and system monitoring. The challenge of building the Wazuh system in a restricted offline environment helped me better understand the importance of choosing the right tools and configurations to ensure the stability and effectiveness of security systems. This experience also helped me grasp the complexities involved in configuring and troubleshooting security systems within a constrained infrastructure.

From an engineering perspective, I observed that Turkcell's infrastructure is robust and well-structured, with a strong emphasis on security. However, there were some challenges in the processes and tools, such as issues related to log collection, repository configuration, and communication with other teams. While these challenges were resolved, they revealed areas where improvement could be made to streamline operations. One suggestion would be to enhance the communication protocols between teams, especially in cases where penetration testing activities might interfere with regular security operations. Clearer documentation and better coordination could prevent miscommunication and reduce the risk of blocking legitimate activities.

Another suggestion would be to focus on improving the scalability of the systems used in log aggregation and threat detection. During my internship, I witnessed instances where log processing faced difficulties due to server load or misconfigurations. Introducing more automated systems or optimizing the log management process could further strengthen Turkcell's ability to respond to security incidents efficiently and at scale.

Overall, my internship at Turkcell has provided me with a comprehensive view of the challenges and opportunities within the cybersecurity industry. The skills and insights gained during this period will undoubtedly be valuable in my future career. I am confident that these experiences have prepared me to contribute meaningfully to any engineering or cybersecurity team, and I am grateful for the opportunity to have been part of such an innovative and forward-thinking organization.

5 REFERENCES

[1] Turkcell. (2025, Feb. 20) *Turkcell Genel bakış 2025* [Online]. Available: https://www.turkcell.com.tr/tr/hakkimizda/genel-bakis

[2] Turkcell. (2025, Feb. 20) *Turkcell Yönetim Takımı 2025* [Online]. Available: https://www.turkcell.com.tr/tr/hakkimizda/genel-bakis/yonetim-takimi

[3] Turkcell. (2025, Feb. 20) *Turkcell Kurumsal Yönetim 2025* [Online]. Available: https://www.turkcell.com.tr/tr/hakkimizda/yatirimci-iliskileri/kurumsal-yonetim

[4] G. González-Granadillo, S. González-Zarzosa, and R. Diaz, ”Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures,” *Sensors*, vol. 21, no. 14, pp. 15–16, July, 2021, doi: 10.3390/s21144759

[5] A. Ali, M. Ahmed, and A. Khan, “Audit Logs Management and Security - A Survey,” *Kuwait Journal of Science*, vol. 48, no. 3, Jun. 2021, doi: https://doi.org/10.48129/kjs.v48i3.10624.

[6] L. Moroney, “An Introduction to Firebase,” *The Definitive Guide to Firebase*, pp. 1–24, 2017, doi: https://doi.org/10.1007/978-1-4842-2943-9_1.

[7] S. Moiz, A. Majid, A. Basit, M. Ebrahim, A. A. Abro and M. Naeem, ”Security and Threat Detection through Cloud-Based Wazuh Deployment,” *2024 IEEE 1st Karachi Section Humanitarian Technology Conference (KHI-HTC)*, Tandojam, Pakistan, 2024, pp. 1-5, doi: 10.1109/KHI-HTC60760.2024.10482206.