Security of Business Intelligence Systems in the Face of Cyber Attacks

Mario-Tudor CHIRIAC
Bucharest University of Economic Studies
Faculty of Cybernetics, Statistics and Economic Informatics
Bucharest, Romania
chiriacmario21@stud.ase.ro

This article aims to examine the effects of various types of cyberattacks on Business Intelligence (BI) systems, as well as the protective measures required to prevent such incidents. The simulated attacks include Denial of Service (DoS), Man-in-the-Middle (MITM), and Evil Twin attacks. These represent some of the most frequently employed techniques used by cyber attackers to compromise an organization's infrastructure, either to steal critical data for malicious purposes or to disrupt operations, potentially leading to severe financial losses or even bankruptcy. Therefore, it is essential that Business Intelligence systems be supported by a robust and well-secured infrastructure in order to mitigate risks and prevent catastrophic outcomes.
Keywords: *Business Intelligence system security; BI systems; Denial of Service; Man-in-the-Middle; Evil Twin; security measures*

1 Introduction

A Business Intelligence (BI) system represents an integrated combination of technologies, applications, and processes that enable organizations to collect, store, analyze, and visualize data in order to make better, faster, and more informed decisions.

In practice, a BI system aggregates and centralizes all relevant organizational data originating from diverse sources, such as databases, Excel files, enterprise management applications, or even online services. These data are subsequently processed, cleaned, and structured to facilitate efficient analysis. BI systems allow users to perform detailed analytical operations, generate reports, and develop interactive dashboards that provide a clear and comprehensive overview of organizational performance. Through these tools, decision-making structures within the company can monitor key performance indicators in real time, rapidly identify trends or emerging issues, and conduct comparative analyses across time periods, products, or departments [1], [2].

Moreover, BI systems automate reporting and analytical processes, thereby saving time and reducing the risk of human error. They also enable forecasting based on historical data and analytical models, assisting management in making data-driven decisions and responding promptly to market fluctuations or internal operational changes.

Consequently, the security of a BI system is critical to business continuity. System compromise or large-scale data breaches may lead to severe consequences, including the leakage or loss of confidential information, such as financial records, business strategies, or customer and supplier data; the erosion of competitive advantage, as sensitive information may reach competitors or black markets; the disruption of ongoing operations through data deletion, manipulation, or restricted system access; and the loss of trust among customers, partners, or investors, particularly in cases of publicly disclosed security incidents.

In the context of accelerated digitalization and the exponential growth in data volumes managed by organizations, BI systems have become increasingly attractive targets for

cyber attackers. Due to their integration of sensitive data from across the organization's functional domains, these systems are susceptible to sophisticated attack vectors. Common methods employed by cyber attackers include unauthorized access through the exploitation of application vulnerabilities or misconfigurations, data interception via Man-in-the-Middle (MITM) or Evil Twin attacks, and Denial of Service (DoS) attacks that may disrupt access to critical reports and datasets [3], [4], [5].

In the following sections, we will examine several of the most commonly employed techniques used by cyber attackers to target a Business Intelligence (BI) system, as well as the consequences associated with such attacks. For the purpose of simulation, Power BI will be used to emulate the client environment on one laptop, MariaDB will simulate the database server on a separate laptop, and Kali Linux will be deployed on a third device acting as the attacker.

2 Denial Of Service (DOS)

A Denial of Service (DoS) attack represents a malicious action through which an attacker attempts to render an information system, server, application, or online service unavailable to legitimate users by overwhelming it with an excessive volume of requests or by exploiting vulnerabilities that disrupt its normal operation [6].

In the context of a DoS attack, system resources such as CPU, memory, bandwidth, or request-processing capacity are rapidly and artificially exhausted, preventing the targeted service from responding to legitimate user requests. The result is a degradation of performance that may escalate to a complete service outage, thereby severely affecting service availability and the operational continuity of an organization.

This type of attack does not typically aim at data exfiltration, but rather at disrupting access to resources, generating system blockages, delays, and temporary or prolonged unavailability of the targeted infrastructure. The consequences of a DoS attack on a BI system may include:

- The blockage of access to reports and analyses critical to the organization;
- Significant delays or the inability to generate reports in a timely manner;
- Impairment of the decision-making process, as management no longer has access to the necessary information;
- Potential financial losses and missed business opportunities resulting from information unavailability.

Moreover, such attacks may serve as a cover for additional malicious activities, including data exfiltration or the compromise of other internal systems, capitalizing on the disruption and confusion generated by the unavailability of the BI platform. The figure below illustrates a DoS attack targeting a Business Intelligence infrastructure.

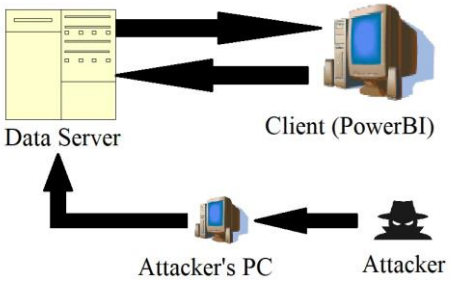


Figure 1 – Illustration of a DoS Attack on a Business Intelligence Infrastructure

The legitimate client application (Power BI) normally communicates with the database server; however, the attacker generates a massive volume of malicious traffic directed at the same server. This artificial flood of requests exhausts the server's resources, preventing it from responding to legitimate queries. As a result, Power BI becomes unable to access data or generate reports, thereby illustrating the direct impact of a DoS

attack on a critical analytics and reporting service.

DoS attacks are typically executed using specialized software tools, often developed with malicious intent. Examples of tools that have been employed in DoS attacks include LOIC (Low Orbit Ion Cannon), which became widely known during coordinated hacktivist campaigns such as those conducted by Anonymous between 2010 and 2012, and HOIC (High Orbit Ion Cannon), an enhanced version of LOIC capable of launching more complex HTTP-based attacks and bypassing certain basic protection mechanisms. Additionally, attackers may develop and deploy custom-designed software tailored to the specific characteristics and vulnerabilities of the targeted system [7], [8].

In practice, in order to maximize the impact on the targeted system, an attacker conducting a DoS attack may employ the following techniques:

1. High-concurrency request generation. On a single machine, the attacker may use dozens or even hundreds of concurrent threads to generate thousands of requests per second, thereby simulating the behavior of multiple legitimate users. In the case of Distributed Denial of Service (DDoS) attacks, this logic is massively scaled: the attacker controls a network of compromised devices, commonly referred to as a botnet, which may range from several thousand to tens of millions of devices, including routers, IP cameras, servers, and personal computers. Each compromised device executes the same malicious code in parallel, directing traffic toward the same target.
2. Combined attack techniques. To overload the server more effectively, attackers may simultaneously launch HTTP requests, SQL queries, and TCP/IP or UDP packets. These packets may contain modified or malformed structures designed to exploit specific vulnerabilities and to complicate the filtering and mitigation of malicious traffic.

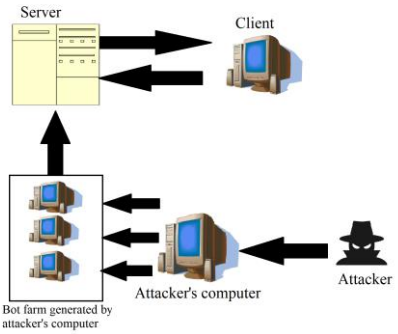


Figure 2 – Illustration of a Thread-Based DoS Attack on a Business Intelligence Infrastructure

We will exemplify a Denial of Service (DoS) attack targeting a Business Intelligence server that relies on a database structured according to the snowflake schema model, as illustrated in Figure 3.

The dataset used in the experiment is specific to an e-commerce environment and includes detailed information regarding sales transactions, products, customers, employees, and retail locations. This structure enables the tracking of transactions by day, geographical region, product category, or employee and store performance, thereby providing a granular and multidimensional perspective on business activity. For instance, sales over a specific period, regional customer preferences, or the efficiency of various product categories can be analyzed rapidly and comprehensively.

By simulating a DoS attack on this infrastructure, it becomes possible to observe concretely how the availability of access to these critical analytical data decreases. Managers, business analysts, and support teams are no longer able to access real-time reports, extract key performance indicators, or make data-driven decisions, resulting in operational bottlenecks and potential financial losses. Moreover, due to the interconnected nature of the snowflake schema, any service disruption propagates rapidly across multiple dimensions, thereby amplifying the negative

impact of a DoS attack on the Business Intelligence system.

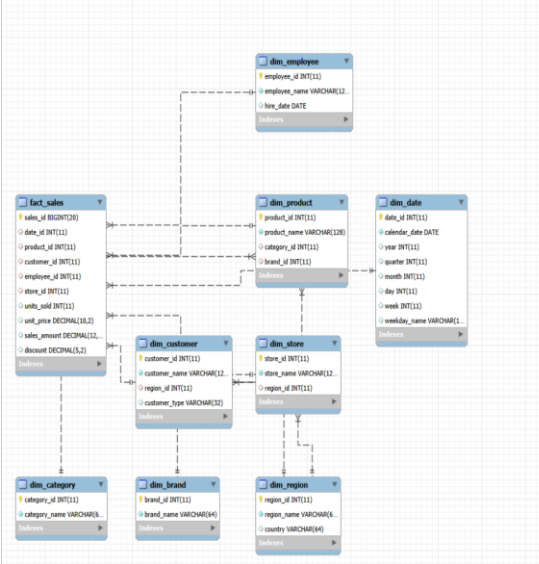


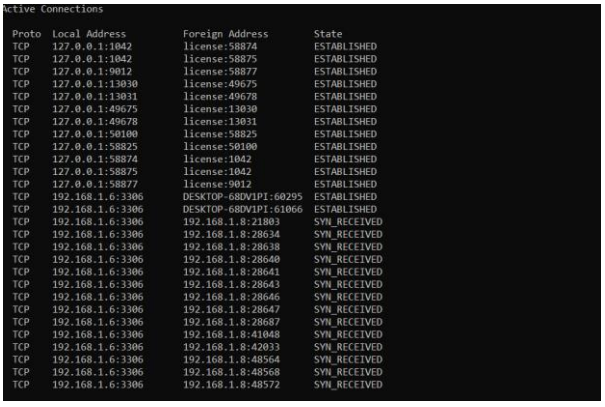
Figure 3 – Illustration of the Snowflake Schema of the Database

The DoS attack manifests through the repeated generation and transmission, at very short intervals, of a high volume of TCP/IP packets directed toward the port dedicated to the database server. By default, the MariaDB service runs on port 3306, which consequently becomes the primary target of the overload. For this simulation, the **hping3** utility available on Linux will be used. From the attacker's laptop, the following command is executed in the terminal: **`sudo hping3 -S --flood -p 3306 192.168.1.6.`** The parameters of this command represent the following:

- **`sudo`** – executes the command with elevated (root) privileges;
- **`hping3`** – the network packet generation and analysis tool;
- **`-S`** – sets the SYN flag, generating TCP SYN packets (commonly used in SYN flood attacks);
- **`--flood`** – ends packets as fast as possible, without waiting for replies ;
- **`-p 3306`** – specifies the target port (MariaDB's default port);
- **`192.168.1.6`** – the IP address of the targeted database server.

By executing the netstat command on the server, a significant number of connections in the SYN-RECEIVED state can be observed. These represent incomplete TCP connections, for which the server has received the initial SYN packet but has not received the final ACK response required to complete the three-way handshake. Under normal conditions, this state is transient and lasts only a few milliseconds. However, during a SYN flood–based DoS attack, the number of such connections increases rapidly. The server allocates resources for each entry in the SYN-RECEIVED state, thereby exhausting its connection tables and reducing its capacity to respond to legitimate requests.

Moreover, in a real attack scenario, the number of connections in the SYN-RECEIVED state would be extremely high, as a SYN flood attack is characterized by the transmission of a large volume of SYN packets within a very short time frame. The greater the number of such packets, the more quickly the server's queue of half-open connections becomes saturated, ultimately leading to performance degradation and, eventually, the inability to process legitimate requests[9],[10],[11].



Active Connections

Proto	Local Address	Foreign Address	State
TCP	127.0.0.1:1042	license:58874	ESTABLISHED
TCP	127.0.0.1:1042	license:58875	ESTABLISHED
TCP	127.0.0.1:9012	license:58877	ESTABLISHED
TCP	127.0.0.1:13030	license:49675	ESTABLISHED
TCP	127.0.0.1:13031	license:49678	ESTABLISHED
TCP	127.0.0.1:49675	license:13030	ESTABLISHED
TCP	127.0.0.1:49678	license:13031	ESTABLISHED
TCP	127.0.0.1:50100	license:58825	ESTABLISHED
TCP	127.0.0.1:58825	license:50100	ESTABLISHED
TCP	127.0.0.1:58874	license:1042	ESTABLISHED
TCP	127.0.0.1:58875	license:1042	ESTABLISHED
TCP	127.0.0.1:58877	license:9012	ESTABLISHED
TCP	192.168.1.6:3306	DESKTOP-68DVIPI:60295	ESTABLISHED
TCP	192.168.1.6:3306	DESKTOP-68DVIPI:61066	ESTABLISHED
TCP	192.168.1.6:3306	192.168.1.8:21803	SYN_RECEIVED
TCP	192.168.1.6:3306	192.168.1.8:28634	SYN_RECEIVED
TCP	192.168.1.6:3306	192.168.1.8:28638	SYN_RECEIVED
TCP	192.168.1.6:3306	192.168.1.8:28640	SYN_RECEIVED
TCP	192.168.1.6:3306	192.168.1.8:28641	SYN_RECEIVED
TCP	192.168.1.6:3306	192.168.1.8:28643	SYN_RECEIVED
TCP	192.168.1.6:3306	192.168.1.8:28646	SYN_RECEIVED
TCP	192.168.1.6:3306	192.168.1.8:28647	SYN_RECEIVED
TCP	192.168.1.6:3306	192.168.1.8:28687	SYN_RECEIVED
TCP	192.168.1.6:3306	192.168.1.8:41048	SYN_RECEIVED
TCP	192.168.1.6:3306	192.168.1.8:42033	SYN_RECEIVED
TCP	192.168.1.6:3306	192.168.1.8:48564	SYN_RECEIVED
TCP	192.168.1.6:3306	192.168.1.8:48568	SYN_RECEIVED
TCP	192.168.1.6:3306	192.168.1.8:48572	SYN_RECEIVED

Figure 4 – Illustration of connections on the BI Server

In this particular scenario, the attacker did not have access to modern equipment or a sufficiently powerful infrastructure capable of fully saturating the server's connection table. The attack was launched from a single laptop with limited hardware resources, including CPU capacity, memory, and network bandwidth, which significantly constrained the maximum number of SYN packets that could be transmitted per second.

Consequently, although connections in the SYN-RECEIVED state are observable on the server, their number is not sufficiently high to cause a complete denial of service of the database system. The server remains capable of processing a portion of legitimate requests, albeit with noticeably degraded performance.

3 Man-In-The-Middle (MITM)

A Man-in-the-Middle (MITM) attack represents a type of cyberattack in which an attacker covertly positions themselves between two communicating entities, such as a client and a server, without the knowledge of either party. In this scenario, the attacker is able to intercept, monitor, modify, or redirect the data traffic exchanged between the two entities, while making them believe that they are communicating directly with one another. Within the context of an MITM attack, the attacker may obtain sensitive information, including authentication credentials, financial data, personal information, or confidential communication content. Furthermore, the attacker may alter transmitted data, inject malicious content, or redirect users to compromised resources, often without the victims' immediate awareness [4].

Man-in-the-Middle attacks are particularly dangerous in unsecured or improperly configured networks, such as public Wi-Fi environments, but they may also occur within internal infrastructures if adequate security mechanisms are not implemented, including encrypted communication channels, proper mutual authentication, and rigorous validation of digital certificates.

A fundamental characteristic of an MITM attack is that data traffic is effectively forced to pass through the attacker's system, for example, the attacker's laptop. The attacker thus becomes an intermediary node within the communication process, gaining control over the data flow in both directions. This position enables the attacker not only to observe the content of the communication, but also to manipulate it in real time before it reaches its intended destination.

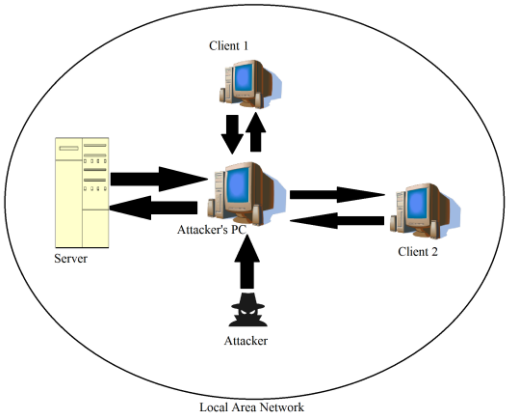


Figure 5 – Illustration of the Man-in-the-Middle (MITM) Concept between Two Client Computers

Figure 5 clearly illustrates the operational mechanism of a Man-in-the-Middle (MITM) attack within a Local Area Network (LAN). Under normal conditions, Client 1 and Client 2 would communicate directly with the server, exchanging data packets without the involvement of any third party. In the depicted scenario, this legitimate communication flow is disrupted by introducing the attacker's PC as an intermediary node within the network. All communications between the clients and the server are redirected to pass through the attacker's system first, without the legitimate parties being aware of this interception.

At the LAN level, such an attack is feasible because devices rely on local addressing and routing mechanisms, such as the resolution

of IP addresses into MAC addresses or the identification of the appropriate network gateway. The attacker exploits the absence of strict verification mechanisms within these processes and successfully impersonates both parties simultaneously:

• To the clients, the attacker presents themselves as the legitimate server;

• To the server, the attacker presents themselves as a legitimate client.

Thus, packets sent by Client 1 or Client 2 first reach the Attacker's PC, where they can be inspected, modified, or recorded, and are subsequently forwarded to the server. The server's responses follow the same reverse path, once again passing through the attacker before reaching the clients. From the victims' perspective, the communication appears normal, as the data flow is not interrupted but merely intercepted.

To experiment with the MITM concept, the Bettercap tool on Linux will be used. After identifying the IPv4 address of the Power BI client using the **netstat** command on Windows, the following command is executed: **sudo bettercap –iface eth0,** where:

• **sudo** – is required to run Bettercap with administrative privileges, as the application directly interacts with the network interface and processes packets at a low level (Layer 2 / Layer 3);

• **bettercap** – is the application used for traffic analysis and manipulation within the Man-in-the-Middle experimental setup;

• **-iface eth0** – specifies the network interface on which Bettercap will operate. In this case, eth0 represents the Ethernet interface of the Linux system, through which the attacker is connected to the same local network as the Power BI client and the database server [12], [13].

Subsequently, within the same Bettercap interface, the following commands are executed: **set arp.spoof.targets 192.168.1.8, arp.spoof on, net.sniff on,** where:

- The command **set arp.spoof.targets 192.168.1.8** specifies the IP address of the target system, in this case the Power BI client, against which the ARP spoofing technique will be applied. Through this configuration, the attacker explicitly designates the device within the local network that will be targeted.
- The command **arp.spoof on** activates the ARP spoofing module. Its purpose is to manipulate the ARP tables of the devices within the network, causing both the client and the server to route their traffic through the attacker's system. In this manner, the attacker positions themselves as an intermediary in the communication process, effectively implementing a Man-in-the-Middle scenario at the Local Area Network (LAN) level.
- The command **net.sniff on** activates the network traffic interception and analysis module. This module enables the capture of packets and the display of relevant information, such as DNS requests, HTTP connections, or other types of unencrypted traffic, thereby demonstrating full visibility into the communication exchanged between the client and the server [14].

Figure 6 – Capture Showing the Interception of Power BI Client Traffic

At this stage, the attacker intercepts the traffic by redirecting it through their own device, which becomes an intermediary node between the client and the server. Although the traffic is rerouted through the attacker's system, the communication itself is not disrupted, making the attack difficult to detect by legitimate users.

Upon executing the commands **`set dns.spoof.domains ms-pbi.pbi.microsoft.com`**, **`set dns.spoof.address 192.168.1.10`**, **`dns.spoof on`** the following effects occur:

The first effect becomes visible upon executing the command **`set dns.spoof.domains ms-pbi.pbi.microsoft.com`**, whereby the attacker explicitly specifies the domain that will be targeted for DNS manipulation. Consequently, Bettercap no longer responds to all DNS requests within the network, but only to those attempting to resolve the domain associated with the Power BI service. This selective filtering renders the attack more precise and less conspicuous, as it does not interfere with overall DNS traffic, but instead focuses exclusively on a critical service.

The second effect is achieved through the command **`set dns.spoof.address 192.168.1.10`**, which defines the forged IP address that will be returned as the DNS response. Instead of resolving the domain ms-pbi.pbi.microsoft.com to the legitimate Microsoft infrastructure addresses, the client will receive the attacker-specified IP address. In practical terms, the attacker gains control over the destination to which Power BI traffic is directed.

Upon execution of the **`dns.spoof on`** command, the DNS spoofing module is activated**.** From this point onward, any DNS request for the specified domain, intercepted through the already established MITM mechanism (via ARP spoofing), receives a falsified response generated by the attacker. This forged response is delivered before, or in place of, the legitimate one, causing the client to accept the malicious IP address as authentic.

Figure 7 – Combination of ARP Spoofing and DNS Spoofing Attacks

The combined effect of these commands is the compromise of the DNS resolution process: the Power BI client is transparently redirected to a system controlled by the attacker, without displaying obvious errors and without apparently interrupting the application's functionality. From the user's perspective, the connection appears normal; however, the transmitted data may be intercepted, analyzed, modified, or blocked by the attacker.

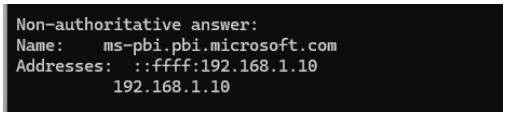


Figure 8 – Result of Combining ARP and DNS Spoofing Attacks from the Client's Perspective

The final capture, displaying a "Non-authoritative" DNS response, confirms the impact of the attack from the client's perspective: the Power BI domain is resolved to an internal IP address (192.168.1.10), demonstrating that the DNS resolution process has been compromised. From the perspective of the Power BI application, the communication appears normal; however, in reality, the data are being redirected and may be monitored, analyzed, or manipulated by the attacker.

4 Evil Twin

An Evil Twin attack represents a type of cyberattack targeting wireless networks in which an attacker creates a malicious Wi-Fi access point that impersonates a legitimate

one by replicating its SSID, security configuration, and in some cases even its MAC address. The objective of this attack is to deceive users into connecting, unknowingly, to the attacker-controlled network while believing they are accessing the authentic infrastructure. Once victims connect to the Evil Twin access point, the attacker gains control over the entirety of their network traffic. This enables the interception of communications, the collection of sensitive data such as authentication credentials, session tokens, and personal information, as well as the modification or redirection of traffic. Furthermore, the attacker may launch additional attacks, including Man-in-the-Middle attacks, DNS redirections, or the injection of malicious content. The Evil Twin attack is particularly dangerous because it exploits users' trust in familiar Wi-Fi networks and does not require the direct compromise of the legitimate infrastructure. It is commonly encountered in public or poorly secured wireless networks, but it may also be deployed within internal environments where adequate mechanisms for authentication, encryption, and access point validation are not properly implemented [5].

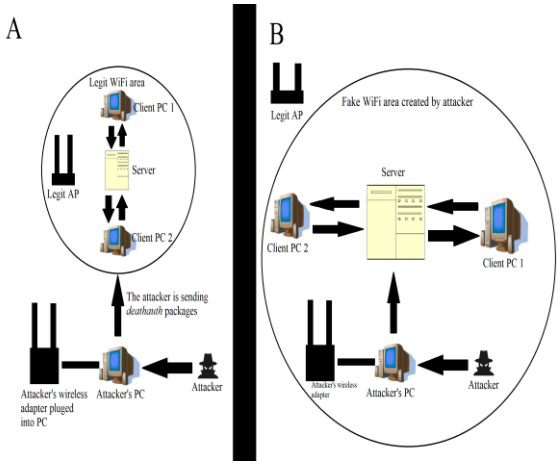


Figure 9 - Illustration of the Man-in-the-Middle (MITM) Concept Between Two Client Computers

Figure 9 illustrates the mechanism of an Evil Twin attack, conducted in two distinct stages, labeled as scenarios A and B.

In Scenario A, the normal operation of a legitimate Wi-Fi network is depicted. The clients (Client PC 1 and Client PC 2) are connected to a legitimate Access Point, through which they communicate directly with the server. Data traffic flows normally, without the involvement of any third party, and the network infrastructure operates according to its standard configuration. At the same time, the attacker is positioned within proximity of the network and is equipped with a wireless adapter connected to their own system. The attacker begins transmitting deauthentication frames at the IEEE 802.11 level to the clients, with the objective of forcing them to disconnect from the legitimate Access Point. This action does not directly compromise the server or the legitimate Access Point; however, it disrupts and destabilizes the users connection. Scenario B illustrates the actual exploitation phase of the Evil Twin attack. The attacker creates a rogue access point configured with the same SSID as the legitimate network, thereby emulating the original infrastructure. Due to the forced disconnection and the visual similarity between the two networks, the clients automatically or inadvertently reconnect to the attacker-controlled rogue network.

At this stage, all communications between the clients and the server are routed through the Attacker's PC, which becomes an intermediary node within the network. Although the server appears to continue communicating normally with the clients, the traffic is in fact intercepted and controlled by the attacker. The attacker is thus able to monitor transmitted data, modify packet contents, or redirect traffic to alternative destinations.

To simulate an Evil Twin attack, the **`hostapd`** utility will be used. On the attacker's

device, a rogue wireless network is configured, as illustrated in Figure 10. Subsequently, the attacker executes the command: **sudo hostapd -dd /etc/hostapd/hostapd.conf.**

Figure 10 – Configuration of **Hostapd**

The configuration presented in Figure 10 highlights how the attacker can replicate the essential parameters of the legitimate network, including the SSID, channel, and security type, thereby increasing the likelihood that client stations will automatically reconnect to the rogue access point.

After creating the rogue access point using **hostapd**, the attacker must ensure the full operational functionality of the network infrastructure for clients connecting to the Evil Twin network. To achieve this, the attacker configures a local DNS service using the **dnsmasq** application.

Dnsmasq is a lightweight service that provides DNS forwarding, controlled DNS spoofing, and DHCP server functionalities. It is frequently used in testing environments, laboratory setups, or Evil Twin attack scenarios.

The attacker creates a configuration file named **dnsmasq.conf**, which contains the following directives:

- **interface=wlan0**
  This directive specifies the network interface on which the dnsmasq service will operate. In this case, `wlan0` represents the attacker's wireless interface associated with the rogue access point previously created using hostapd. Consequently, dnsmasq will respond exclusively to requests originating from devices connected to the malicious Wi-Fi network and will not interfere with other network interfaces on the system.
- **dhcp-range=192.168.50.2,192.168.50.100,12h** This directive defines the range of IP addresses that will be dynamically assigned to clients connected to the Evil Twin network via the DHCP protocol. In the example above, dnsmasq distributes IP addresses between 192.168.50.2 and 192.168.50.100, with a lease time of 12 hours for each allocation.

Through this configuration, the attacker establishes a separate subnet fully controlled by them, within which clients automatically receive:

- a valid IP address;
- a subnet mask;
- a default gateway;
- a DNS server, which by default is the attacker's system.

Subsequently, the attacker executes the command: **sudo dnsmasq -C dnsmasq.conf**

After configuring and activating the rogue Evil Twin access point, the attacker enters the exploitation phase of the clients' connections. At this stage, victim devices connect to the malicious Wi-Fi network under the assumption that it represents the legitimate infrastructure. From this moment onward, the attacker gains full control over the network traffic generated by the connected clients. In practice, the attacker employs a wireless adapter capable of packet monitoring and frame injection, typically by switching the interface into monitor mode. This capability enables the attacker to:

- Observe all Wi-Fi frames transmitted by the clients;
- Force repeated disconnections and reconnections;
- Ensure that victims remain connected exclusively to the rogue network;
- Prevent devices from reconnecting to the legitimate Access Point [15], [16], [17].

Once clients are connected to the Evil Twin network, the attacker effectively assumes the role of both network gateway and DNS server. The direct consequences for a Business Intelligence system may include:

- Inability to connect to data sources;
- Failure of automated refresh processes;
- Display of incomplete or outdated data;
- Restricted access to reports and dashboards;
- Impairment of data-driven decision-making processes.

From the perspective of a Power BI user, these issues may appear to be ordinary network errors, latency problems, or general instability, without any obvious indication that a cyberattack is taking place.

5 Conclusions

Business Intelligence systems constitute a critical component of the information infrastructure of modern organizations, as they centralize, process, and expose essential data required for strategic decision-making. For this reason, they represent an attractive target for attackers, not only for the purpose of data exfiltration, but also for disrupting availability or manipulating data flows.

Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks primarily target the availability of the BI system. Even if data are not directly compromised, the unavailability of reports, dashboards, and analytical outputs may have significant consequences for decision-making processes and operational continuity.

To mitigate this category of attack, the following measures are essential:

- Implementation of connection limits and request rate-limiting mechanisms;
- Proper configuration of firewalls and Intrusion Detection/Prevention Systems (IDS/IPS);
- Deployment of network-level or cloud-based DDoS protection services;
- Continuous monitoring of server resources and network traffic behavior.

Man-in-the-Middle (MITM) attacks directly affect data confidentiality and integrity. Within the context of a BI system, an MITM attack may enable the interception of authentication credentials, database queries, or analytical results without users being aware of the compromise. Essential protective measures include:

- Mandatory encryption of communications (TLS/SSL) between clients and BI servers;
- Strict validation of digital certificates and avoidance of untrusted certificates;
- Network segmentation and the use of secure VPN connections for BI system access;
- Implementation of strong authentication mechanisms and monitoring of active sessions.

These measures significantly reduce the risk of an attacker positioning themselves within the communication channel and gaining control over the data flow.

Evil Twin attacks highlight vulnerabilities at the wireless network level and are particularly dangerous because they may facilitate MITM attacks without directly exploiting BI servers or applications. Protection against this type of attack requires the use of secure Wi-Fi networks employing

WPA2-Enterprise or WPA3 instead of shared-password networks.

References

[1]https://en.wikipedia.org/wiki/Business_intelligence
[2]https://www.ibm.com/think/topics/business-intelligence
[3]https://en.wikipedia.org/wiki/Denial-of-service_attack#
[4]https://en.wikipedia.org/wiki/Man-in-the-middle_attack
[5]https://en.wikipedia.org/wiki/Evil_twin_(wireless_networks)
[6]https://digitalcommons.sacredheart.edu/cgi/viewcontent.cgi?article=1053&context=computersci_fac
[7]https://en.wikipedia.org/wiki/High_Orbit_Ion_Cannon
[8]https://en.wikipedia.org/wiki/Low_Orbit_Ion_Cannon
[9]https://en.wikipedia.org/wiki/Hping
[10]https://developer.mozilla.org/en-US/docs/Glossary/TCP_handshake
[11]https://en.wikipedia.org/wiki/Transmission_Control_Protocol#
[12]https://en.wikipedia.org/wiki/Ettercap_(software)
[13]https://www.bettercap.org/modules/ethernet/netprobe/
[14]https://www.bettercap.org/modules/ethernet/proxies/dnsproxy/
[15]https://wireless.docs.kernel.org/en/latest/en/users/documentation/hostapd.html#wireless_interface
[16] https://en.wikipedia.org/wiki/Dnsmasq
[17] https://wiki.archlinux.org/title/Dnsmasq

CHIRIAC Mario-Tudor Second year student, in the Master's Programme named: "Databases - business support", within the Faculty of Cybernetics, Statistics and Economic Informatics - the Bucharest University of Economic Studies. In 2024, I graduated from the Bachelor's programme - Economic Informatics in Romanian, within the Faculty of Cybernetics, Statistics and Economic Informatics at the Bucharest University of Economic Studies. Areas of interest: study of SQL and No-SQL databases, programming web applications using Vue.js, React.js.