Informatyka techniczna i sztuczna inteligencja
Technical informatics and Artificial Intelligence
Engineer of XXI Century'2024

Wydział
Budowy Maszyn
i Informatyki
Uniwersytet Bielsko-Bialski

MONOGRAFIA

INFORMATYKA TECHNICZNA
I SZTUCZNA INTELIGENCJA

2024

Identifying attacks on the Bluetooth protocol using Wireshark and the Splunk Siem system

Olha Partyka [1]

[2] *Lviv Polytechnic National University, Associate professor of Information Protection Department, olha.o.mykhailova@lpnu.ua*

Abstract: This article explores the relevance of the security issues surrounding Bluetooth devices, used daily by billions of users. The Bluetooth protocol remains vulnerable to various attacks, such as brute force attacks, man-in-the-middle attacks (MITM), MAC address spoofing, PIN code-breaking, and DoS attacks. The paper presents methods for monitoring and identifying such attacks using the Wireshark network analyzer and the Splunk SIEM system. This study aims to improve the protection of Bluetooth traffic by combining Wireshark traffic analysis with Splunk's real-time monitoring and alerting capabilities through correlation rules. An algorithm is developed for detecting attacks by analyzing anomalous Bluetooth protocol behavior and identifying the use of malicious files. The research demonstrates how typical Bluetooth attacks such as DoS, Spoofing, and malware transmission can be detected using these tools. The research results confirm that applying these methods ensures reliable Bluetooth traffic monitoring and enables prompt responses to potential threats.

Keywords: Bluetooth security, Wireshark, Splunk SIEM, Bluetooth attacks, DoS attack, Man-in-the-Middle (MITM) attack, MAC address spoofing, PIN code-breaking, malware detection, traffic monitoring, real-time security analysis, correlation rules.

Identyfikacja ataków na protokół Bluetooth z wykorzystaniem Wireshark i systemu Splunk Siem

Olha Partyka [1]

[1] *Narodowy Uniwersytet Politechnika Lwowska, Docent Katedry Ochrony Informacji, olha.o.mykhailova@lpnu.ua*

Streszczenie: W artykule zbadano istotność kwestii bezpieczeństwa urządzeń Bluetooth, z których codziennie korzystają miliardy użytkowników. Protokół Bluetooth pozostaje podatny na różne ataki, takie jak ataki siłowe, ataki typu man-in-the-middle (MITM), podszywanie się pod adres MAC, łamanie kodu PIN i ataki DoS. W artykule przedstawiono metody monitorowania i identyfikowania takich ataków przy użyciu analizatora sieci Wireshark i systemu Splunk SIEM. Celem tego badania jest poprawa ochrony ruchu Bluetooth poprzez połączenie analizy ruchu Wireshark z możliwościami monitorowania w czasie rzeczywistym i ostrzegania Splunk za pomocą reguł korelacji. Opracowano algorytm wykrywania ataków poprzez analizę nieprawidłowego zachowania protokołu Bluetooth i identyfikację użycia złośliwych plików. Badania pokazują, w jaki sposób typowe ataki Bluetooth, takie jak DoS, Spoofing i transmisja złośliwego oprogramowania, mogą być wykrywane przy użyciu tych narzędzi. Wyniki badań potwierdzają, że stosowanie tych metod zapewnia niezawodne monitorowanie ruchu Bluetooth i umożliwia szybką reakcję na potencjalne zagrożenia.

Słowa kluczowe: bezpieczeństwo Bluetooth, Wireshark, Splunk SIEM, ataki Bluetooth, atak DoS, atak typu Man-in-the-Middle (MITM), podszywanie się pod adres MAC, łamanie kodów PIN, wykrywanie złośliwego oprogramowania, monitorowanie ruchu, analiza bezpieczeństwa w czasie rzeczywistym, reguły korelacji.

1. Introduction

Bluetooth technology, introduced in the 1990s, has become a vital component in the modern digital ecosystem, facilitating wireless communication for billions of devices worldwide. Despite its widespread use, the Bluetooth

DOI: https://doi.org/10.53052/9788367652292.16

protocol remains susceptible to various security vulnerabilities. These include brute force attacks aimed at password cracking, man-in-the-middle (MITM) attacks, data interception, and MAC address spoofing. Such threats pose significant risks, potentially leading to data breaches, device malfunctions, and other adverse consequences. Both personal and corporate devices are at risk of these attacks, highlighting the importance of this research [1,2].

Given the increasing prevalence of Bluetooth-enabled devices, there is a pressing need for enhanced monitoring and protection methods to mitigate these vulnerabilities. The primary focus of this study is to develop an improved methodology for monitoring Bluetooth traffic using Wireshark and Splunk SIEM systems. These tools allow for real-time detection and response to Bluetooth-based attacks.

This research aims to improve the detection methods for Bluetooth attacks through the combined use of the Wireshark network analyzer and the Splunk SIEM system. The significance of this study lies in developing a robust traffic monitoring system that enables the timely identification of threats and supports SOC analysts in their quick response.

The current state of Bluetooth security research has explored many vulnerabilities. Still, there remains a gap in practical, real-time detection methods that apply to personal and corporate environments. Key works in this field have highlighted the limitations of existing security mechanisms and the need for enhanced protection [3–10]. This study addresses these gaps by proposing a comprehensive solution that integrates traffic analysis and correlation rules to detect and mitigate attacks on Bluetooth protocols.

The main aim of this work is to enhance the security of Bluetooth communications by integrating Wireshark's traffic analysis capabilities with Splunk's real-time monitoring and alerting features. The principal conclusion is that this combined approach offers a reliable and scalable solution for monitoring Bluetooth traffic and responding to security threats promptly.

The methodology employed in this research is designed to be easily replicable, providing detailed steps for capturing and analyzing Bluetooth traffic. Wireshark is used for packet capture, while Splunk's SIEM capabilities allow for real-time monitoring and correlation of security events.

2. Bluetooth Protocol Basics

Bluetooth uses low-power radio waves in the 2400 to 2483.5 GHz range, part of the Industrial and Scientific ISM band. This range is also used by other wireless devices, such as baby monitors or door openers, which can cause interference. However, Bluetooth technology dynamically selects channels using adaptive frequency hopping to minimize this interference.[11]

Bluetooth comes in two primary flavors: Bluetooth Low Energy (LE) and Bluetooth Classic (BR/EDR). LE uses minimal power, is suitable for low-power devices such as activity trackers or smartwatches, and supports point-to-point and mesh networks. Bluetooth Classic offers faster data transfer rates but requires direct pairing between devices.

In Bluetooth BR/EDR, devices connect automatically when they come within range of each other. Once paired, devices exchange encrypted data, ensuring the security of the information. Bluetooth LE can also pair, but it is not required. In this case, the device broadcasts advertising packets provided by another device, allowing the device to select a connection without manual interaction.[13,14]

Bluetooth supports small personal area networks (PANs), where peripheral devices connect to a single central device, such as a smartphone, to ensure constant connectivity. Bluetooth dynamically adapts frequencies to avoid interference between different PANs within the exact location or other wireless technologies.[15]

The range of Bluetooth depends on the following factors:

1. Radio spectrum - determines the frequency of wireless communication.
2. Physical layer - determines the characteristics of the signal, such as transmission speed and error protection.
3. Receiver sensitivity - the minimum signal level at which the receiver can correctly process data.
4. Transmitter power - higher signal power allows for a more extended range but drains the battery faster.
5. Antenna gain - determines the efficiency of transmitting and receiving signals.
6. Path loss - signal attenuation due to distance or interference, such as walls or moisture.

The latest Bluetooth updates include Forward Error Correction (FEC) technology, which allows for increased range by correcting data transmission errors without increasing signal strength.[16-18]

Bluetooth provides various security mechanisms, including encryption and FIPS-compliant authentication. When devices connect, unique security keys are created to protect data from eavesdropping or interference. Users can also customize the visibility of devices to minimize the risk of unwanted connections by making them invisible or turning off Bluetooth when it is not needed. Bluetooth combines flexibility, reliability, and security, making it an indispensable technology for short-range wireless communications in many areas, from consumer devices to the IoT

3. Description of Wireshark functionality and use of the Splunk SIEM system

Wireshark is a powerful tool for analyzing network traffic, allowing you to identify malicious operations and security violations. Although Wireshark is not a classic intrusion detection tool (IDS), it will enable experts to identify suspicious activity through detailed analysis of network packets. Network administrators use this tool for diagnostics, security engineers to identify problems, and developers to debug protocols.

The main features of Wireshark include:

1. Capture packet data from different network interfaces (Ethernet, Bluetooth, Wi-Fi);
2. Detailed analysis of network protocols;
3. Filtering, searching and exporting packets;
4. Creation of statistical reports on traffic;
5. Ability to integrate with other data analysis tools.

Wireshark allows network administrators to view all traffic packets, detecting deviations and anomalies that may indicate network attacks, such as man-in-the-middle attacks, ICMP attacks, or port scanning. The tool allows you to analyze packets collected by other interception programs and save the results in different formats, making it a versatile solution for analyzing network data.

Splunk is an SIEM solution for collecting, analyzing, and visualizing data from various sources, particularly network logs. Using Splunk, organizations can monitor user activity, authentication attempts, and other security-related events. Due to the vast amount of data generated in modern networks, manual analysis becomes impractical. Splunk automates this process by integrating data from different sources to create reports and identify suspicious activity.

Key benefits of Splunk for security analysts:

1. Automate the collection and analysis of security logs;
2. Detect and assess incidents in real-time;
3. Visualize risks with built-in dashboards;
4. Create correlation rules to monitor suspicious activity.

Splunk provides effective security log management, facilitating threat detection with risk-based alerts (RBA) and reducing false positives. This enables Security Operations Center (SOC) analysts to respond to incidents and improve productivity quickly. Additionally, integration with MITRE ATT&CK helps create situational awareness that allows you to analyze threats in the context of known vulnerabilities.

Conclusions Wireshark and Splunk provide a complete approach to traffic analysis and network security monitoring. Wireshark helps you deeply analyze traffic, and Splunk provides management and real-time monitoring, making these tools indispensable for modern cybersecurity systems.

4. Overview of Bluetooth attacks

As the number of Bluetooth devices used worldwide grows, so do the threats associated with their security. Various types of attacks on Bluetooth systems require constant adaptation of security mechanisms. Like any other wireless technology, attackers can intercept, intentionally jam or compromise Bluetooth transmissions. The main threats can be divided into three categories:

1. Disclosure threat - when unauthorized persons can intercept information transmitted via Bluetooth.
2. Integrity threat - the threat that the transmitted data can be modified, misleading the recipient.
3. Denial of service (DoS) threat - attackers can block access to the service, limiting or completely denying the user access.

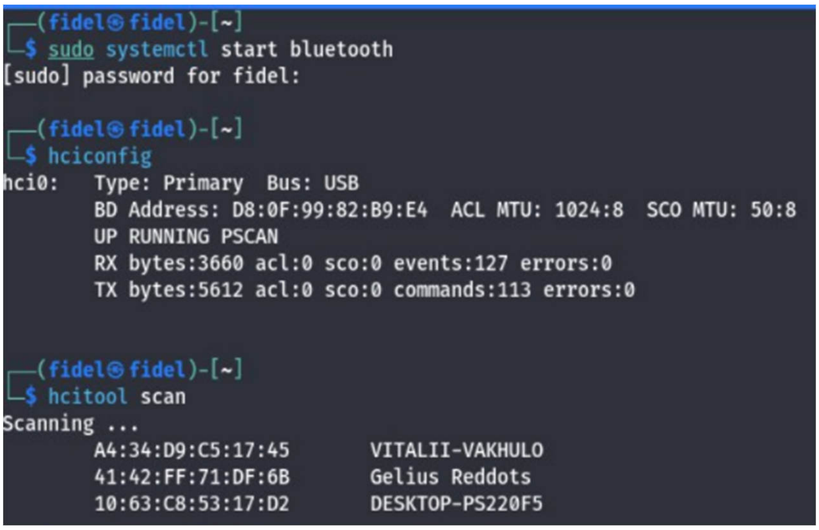
Attacks during the message process. Bluetooth devices are vulnerable to attacks during pairing since this process does not always provide sufficient encryption. Attackers can intercept communication keys and then perform man-in-the-middle (MITM) attacks.

PIN cracking. Bluetooth devices communicate via the exchange of a personal identification number (PIN). An attacker eavesdropping on this process can use a brute-force attack to discover the PIN and gain unauthorized access to the device.

MAC spoofing. When forming a network (piconet), attackers can change the MAC address of a device, posing it as a legitimate device. This allows them to connect to the network or break the connection using special tools to modify data.
Man-in-the-Middle (MITM) attack. This attack allows an attacker to intercept and modify data between two devices using a shared key obtained through eavesdropping or PIN brute force. Such attacks can be launched with Secure Simple Pairing (SSP), especially during the input/output (IO) exchange stage.
BlueJacking Attack: This attack uses Bluetooth to transmit unwanted messages on a device without the owner knowing who the sender is. The message itself is not harmful but can be used to mislead the user or add a malicious contact to the address book.
BlueSnarfing Attack: This attack allows attackers to gain unauthorized access to device data via Bluetooth, including contacts and files. Defenses against this attack include turning off the visibility of the device.
These attacks are just a part of a more extensive range of threats to Bluetooth devices. Protection against such threats is provided by encryption, such as AES.
It is important to remember that constantly improving Bluetooth security methods is necessary, as new threats are continually emerging and evolving.

Bluetooth attack methodology

We will simulate a Bluetooth attack using Bluetooth DOS, Bluetooth spoofing, and sending a malicious file. The attacker will operate Kali Linux, and the victim will be Windows 11 connected to wireless headphones. To implement Bluetooth DOS, we will use a script developed in the Python scripting language, which will transmit large connection packets to the target device (Fig. 1).
To get started, you need to start the Bluetooth service with the systemctl start bluetooth command. Next, we view the adapter configuration with the hiconfig command. The next step is to determine the targets, namely their MAC addresses and names.



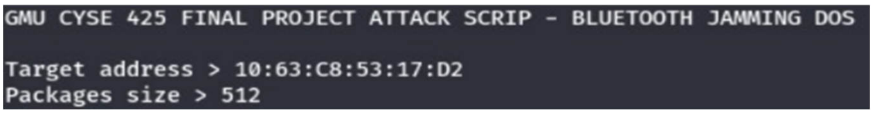
```
┌──(fidel㉿fidel)-[~]
└─$ sudo systemctl start bluetooth
[sudo] password for fidel:

┌──(fidel㉿fidel)-[~]
└─$ hciconfig
hci0:   Type: Primary  Bus: USB
        BD Address: D8:0F:99:82:B9:E4  ACL MTU: 1024:8  SCO MTU: 50:8
        UP RUNNING PSCAN
        RX bytes:3660 acl:0 sco:0 events:127 errors:0
        TX bytes:5612 acl:0 sco:0 commands:113 errors:0


┌──(fidel㉿fidel)-[~]
└─$ hcitool scan
Scanning ...
        A4:34:D9:C5:17:45       VITALII-VAKHULO
        41:42:FF:71:DF:6B       Gelius Reddots
        10:63:C8:53:17:D2       DESKTOP-PS220F5
```

Figure 1. Implementing scanning to identify targets

We have identified the targets. The victim (DESKTOP-PS220F5) is connected to wireless headphones (Gelius Reddots). We need to interrupt the connection between them. To do this, run the script and pass the target MAC address (10:63:C8:53:17:D2) and the packet size (let's take the maximum size supported by the Bluetooth protocol - 512) to its parameters (Fig. 2).

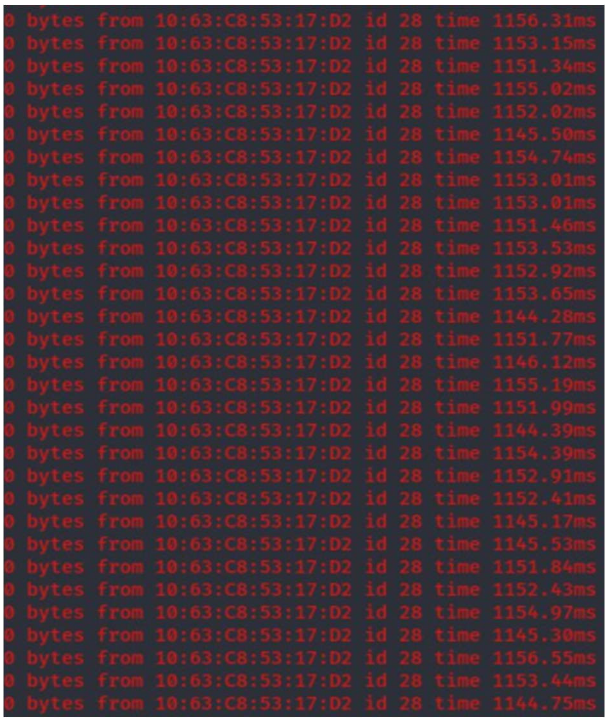


```
GMU CYSE 425 FINAL PROJECT ATTACK SCRIP - BLUETOOTH JAMMING DOS

Target address > 10:63:C8:53:17:D2
Packages size > 512
```

Figure 2. Initiating a DoS attack on Bluetooth

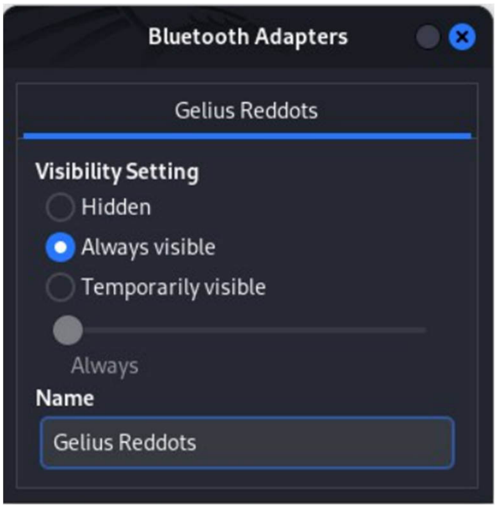
All that remains is to press “Enter” and wait until the target device stops responding (0 bytes will be transmitted in the response) (Fig. 3).



```
0 bytes from 10:63:C8:53:17:D2 id 28 time 1156.31ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1153.15ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1151.34ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1155.02ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1152.02ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1145.50ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1154.74ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1153.01ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1153.01ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1151.46ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1153.53ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1152.92ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1153.65ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1144.28ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1151.77ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1146.12ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1155.19ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1151.99ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1144.39ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1154.39ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1152.91ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1152.41ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1145.17ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1145.53ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1151.84ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1152.43ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1154.97ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1145.30ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1156.55ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1153.44ms
0 bytes from 10:63:C8:53:17:D2 id 28 time 1144.75ms
```

Figure 3. Result a DoS attack on Bluetooth

Now that the connection with the wireless headphones is broken, we will implement a Bluetooth Spoofing attack. To do this, we need to disguise the name and class of the device on the attacker's device as wireless headphones. We can replace the visible name directly in the Bluetooth adapter settings (Fig. 4).



Bluetooth Adapters

Gelius Reddots

Visibility Setting

○ Hidden

● Always visible

○ Temporarily visible

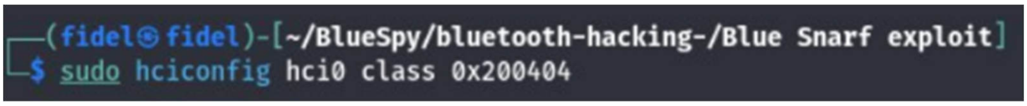
Always

Name

Gelius Reddots

Figure 4. Changing the visible name of the Bluetooth adapter

Next, we replace the Bluetooth adapter class with the headphone class (0x200404) using the command hciconfig <Bluettoth adapter> class <class to be installed>.\ (Fig. 5)

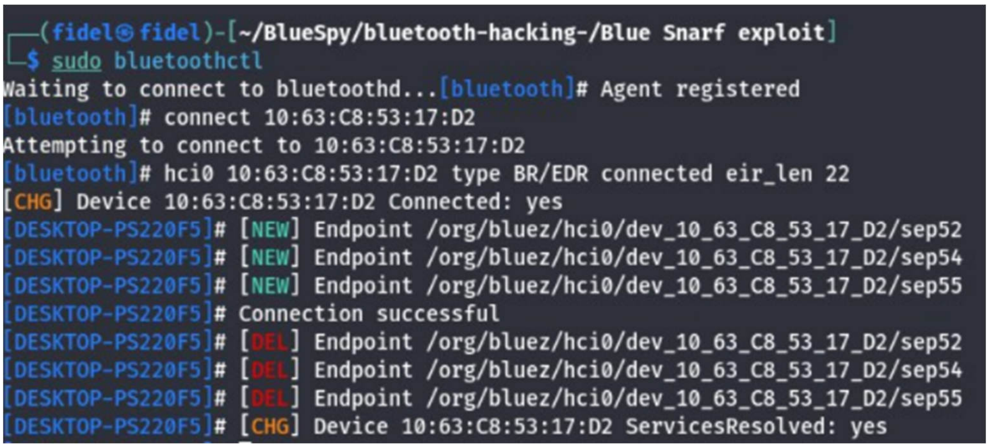


```
┌──(fidel㉿fidel)-[~/BlueSpy/bluetooth-hacking-/Blue Snarf exploit]
└─$ sudo hciconfig hci0 class 0x200404
```

Figure 5. Changing Bluetooth adapter class

Now we need to connect to the target device so that the victim thinks that the headphones have reconnected automatically. To do this, enter the Bluetooth connection management command bluetoothctl and connect to the target device command connect <MAC address>. The message "Device <MAC address> connected: yes (Fig. 6).

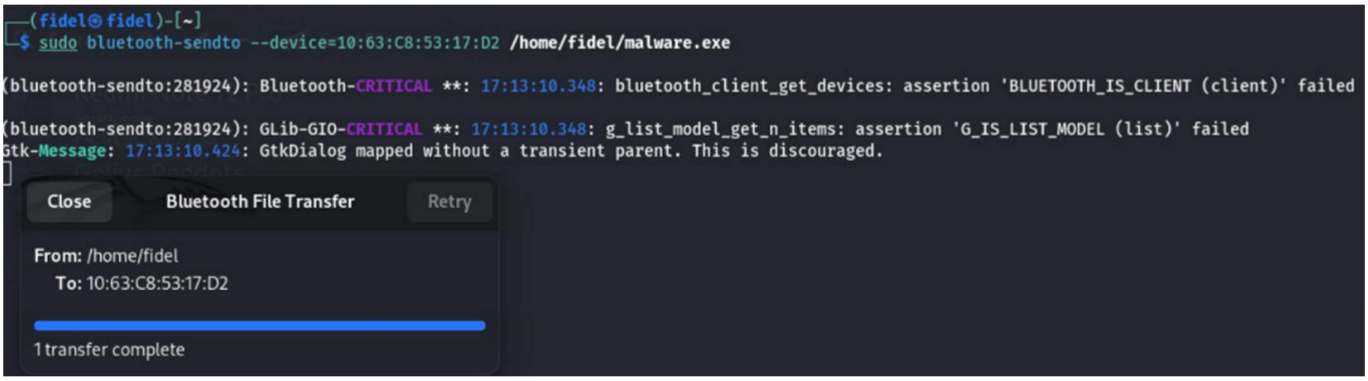
DOI: https://doi.org/10.53052/9788367652292.16



```
┌──(fidel㉿fidel)-[~/BlueSpy/bluetooth-hacking-/Blue Snarf exploit]
└─$ sudo bluetoothctl
Waiting to connect to bluetoothd...[bluetooth]# Agent registered
[bluetooth]# connect 10:63:C8:53:17:D2
Attempting to connect to 10:63:C8:53:17:D2
[bluetooth]# hci0 10:63:C8:53:17:D2 type BR/EDR connected eir_len 22
[CHG] Device 10:63:C8:53:17:D2 Connected: yes
[DESKTOP-PS220F5]# [NEW] Endpoint /org/bluez/hci0/dev_10_63_C8_53_17_D2/sep52
[DESKTOP-PS220F5]# [NEW] Endpoint /org/bluez/hci0/dev_10_63_C8_53_17_D2/sep54
[DESKTOP-PS220F5]# [NEW] Endpoint /org/bluez/hci0/dev_10_63_C8_53_17_D2/sep55
[DESKTOP-PS220F5]# Connection successful
[DESKTOP-PS220F5]# [DEL] Endpoint /org/bluez/hci0/dev_10_63_C8_53_17_D2/sep52
[DESKTOP-PS220F5]# [DEL] Endpoint /org/bluez/hci0/dev_10_63_C8_53_17_D2/sep54
[DESKTOP-PS220F5]# [DEL] Endpoint /org/bluez/hci0/dev_10_63_C8_53_17_D2/sep55
[DESKTOP-PS220F5]# [CHG] Device 10:63:C8:53:17:D2 ServicesResolved: yes
```

Figure 6. Successfully connected to the target device

All that remains is to quickly transfer the malware so that the attack is fully implemented. To do this without confirmation on the recipient's side, we use the BlueZ Tools software. This can be done with the command bluetooth-sendto --device=<MAC address> <path to the file to be transferred> (Fig. 7).



```
┌──(fidel㉿fidel)-[~]
└─$ sudo bluetooth-sendto --device=10:63:C8:53:17:D2 /home/fidel/malware.exe

(bluetooth-sendto:281924): Bluetooth-CRITICAL **: 17:13:10.348: bluetooth_client_get_devices: assertion 'BLUETOOTH_IS_CLIENT (client)' failed

(bluetooth-sendto:281924): GLib-GIO-CRITICAL **: 17:13:10.348: g_list_model_get_n_items: assertion 'G_IS_LIST_MODEL (list)' failed
Gtk-Message: 17:13:10.424: GtkDialog mapped without a transient parent. This is discouraged.
```

From: /home/fidel
To: 10:63:C8:53:17:D2

1 transfer complete

Figure 7. Successfully transfer file to target device

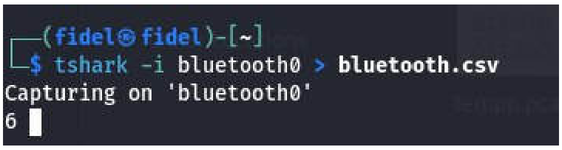
The Bluetooth protocol is not secure. Every system has its vulnerabilities. A large number of attacks have been developed based on them. Some vulnerabilities still exist even in new devices with new operating systems, proven by simulating an incident using only Bluetooth vulnerabilities. This confirms the relevance of protection against attacks on this protocol.

5. Identification and analysis of attacks on Bluetooth

To detect attacks promptly, you must set up real-time traffic monitoring by the Bluetooth SIEM system and write correlation rules to search for traffic according to specified patterns. This can be achieved using the following algorithm:

- Ensuring real-time traffic monitoring:
  It creates a file that will record all Bluetooth traffic the network analyzer intercepts.
  Configuring the SIEM system to monitor this file.
  Checking the receipt and readability of logs.
- Setting up attack notifications:
  Analysis of possible attacks and methods of their implementation.
  Writing correlation rules.
  Testing correlation rules and reducing the number of false positives.
- Analysis of attack notifications:
  Receiving an attack notification.
  Analyze the attack notification and check the accuracy of the response.
  Updating the correlation rule when false wear is detected.

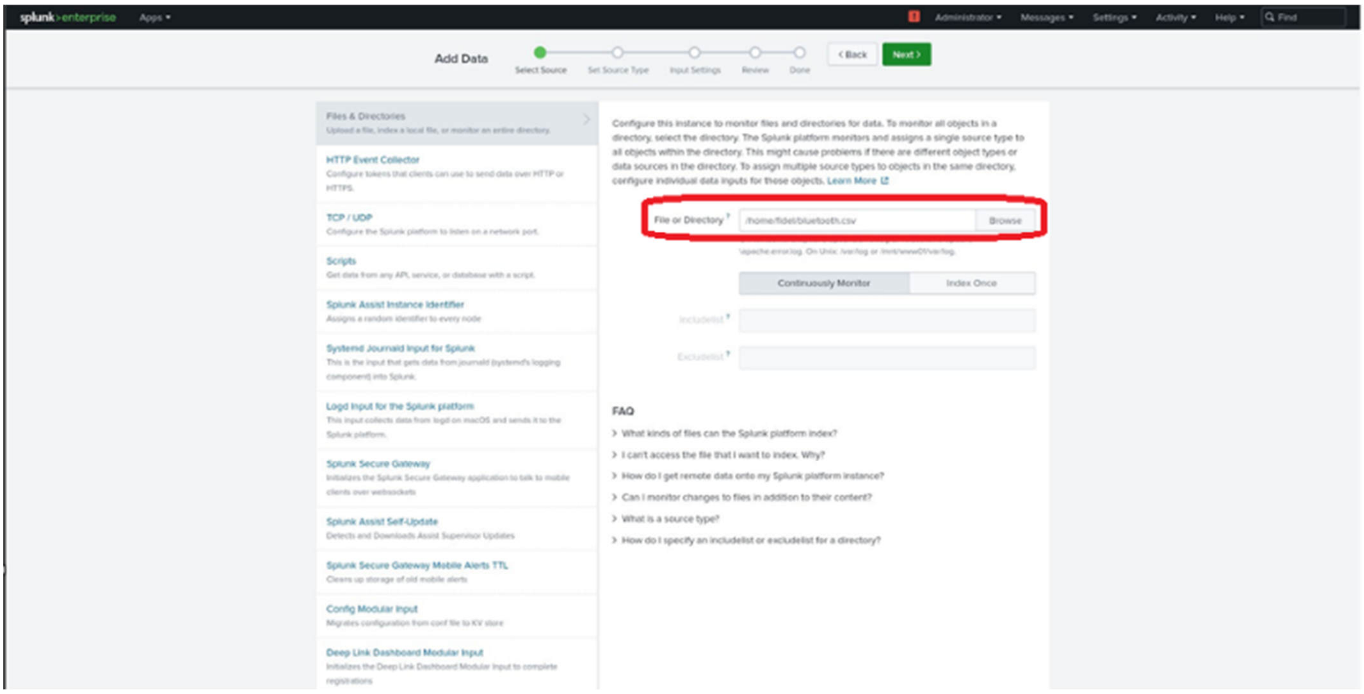
We need a file that will collect traffic and update automatically to conduct real-time monitoring (Fig. 8). This cannot be done directly through the Wireshark interface, but Wireshark has a built-in command line component, tshark, that allows you to write data to a file in real time. To implement this, you just need to run the command tshark –i <interface> > <path to file>.



```
┌──(fidel㉿fidel)-[~]
└─$ tshark -i bluetooth0 > bluetooth.csv
Capturing on 'bluetooth0'
6 
```

Figure 8. Create a file that will update in real time

Now, we have a data source for monitoring. The next step is to enter this data into the Splunk SIEM system. Let's add data. For Splunk to receive data in real time, you need to monitor the file. Therefore, we select Monitor. The next step is to specify the path to the file that will be the data source (Fig. 9).



Files & Directories
Upload a file, index a local file, or monitor an entire directory.

HTTP Event Collector
Configure tokens that clients can use to send data over HTTP or HTTPS.

TCP / UDP
Configure the Splunk platform to listen on a network port.

Scripts
Get data from any API, service, or database with a script.

Splunk Assist Instance Identifier
Assigns a random identifier to every node

Systemd Journald Input for Splunk
This is the input that gets data from journald (systemd's logging component) into Splunk.

Logd Input for the Splunk platform
This input collects data from logd on macOS and sends it to the Splunk platform.

Splunk Secure Gateway
Initializes the Splunk Secure Gateway application to talk to mobile clients over websockets

Splunk Assist Self-Update
Detects and Downloads Assist Supervisor Updates

Splunk Secure Gateway Mobile Alerts TTL
Clears up storage of old mobile alerts

Config Modular Input
Migrates configuration from conf file to KV store

Deep Link Dashboard Modular Input
Initializes the Deep Link Dashboard Modular Input to complete registrations

Configure this instance to monitor files and directories for data. To monitor all objects in a directory, select the directory. The Splunk platform monitors and assigns a single source type to all objects within the directory. This might cause problems if there are different object types or data sources in the directory. To assign multiple source types to objects in the same directory, configure individual data inputs for those objects. Learn More

File or Directory ? /home/fidel/bluetooth.csv Browse

\apache\error.log. On Unix: /var/log or /mnt/www01/var/log.

Continuously Monitor | Index Once

Includelist ?

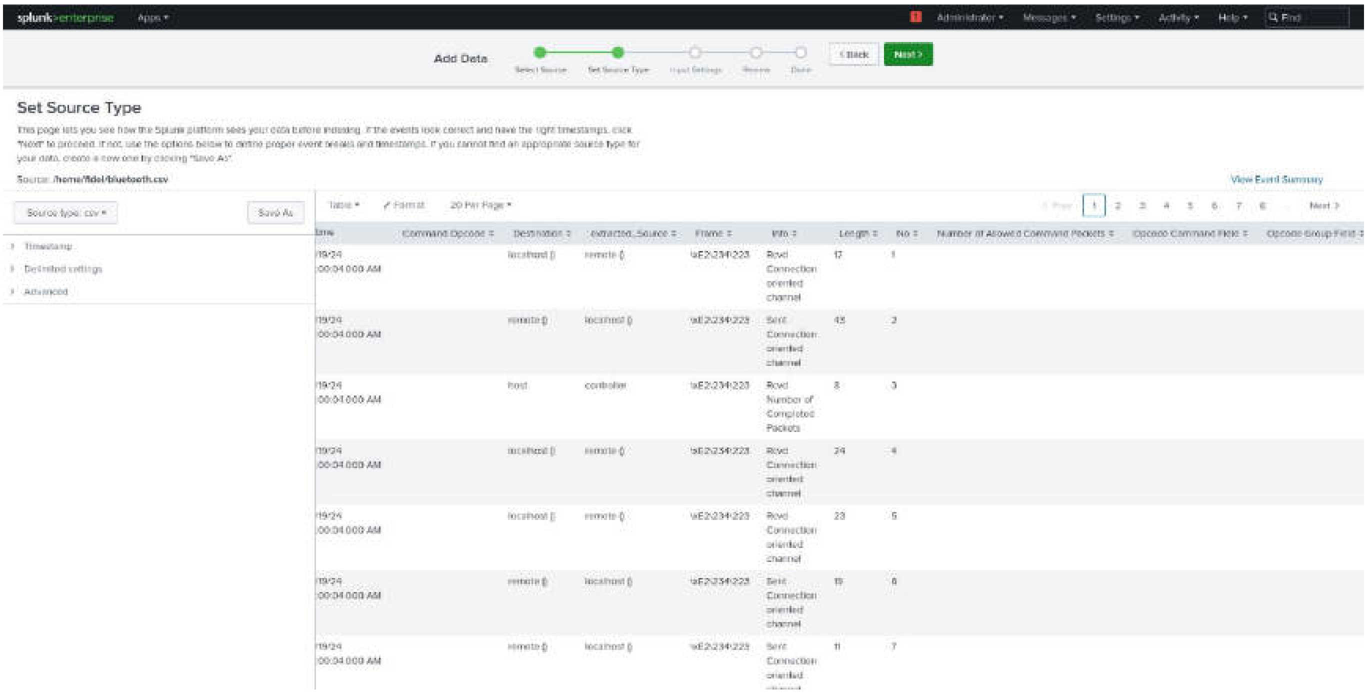
Excludelist ?

FAQ

› What kinds of files can the Splunk platform index?
› I can't access the file that I want to index. Why?
› How do I get remote data onto my Splunk platform instance?
› Can I monitor changes to files in addition to their content?
› What is a source type?
› How do I specify an includelist or excludelist for a directory?

Figure 9. Data source settings

At this stage, you can view the data as a table and check the correctness of the names and output of the fields. If necessary, you can change the view and specify which logs the source will send (if Splunk has a built-in configuration for them). In our case, we left the CSV file since Splunk does not support pure Bluetooth traffic (Fig.10).



Set Source Type

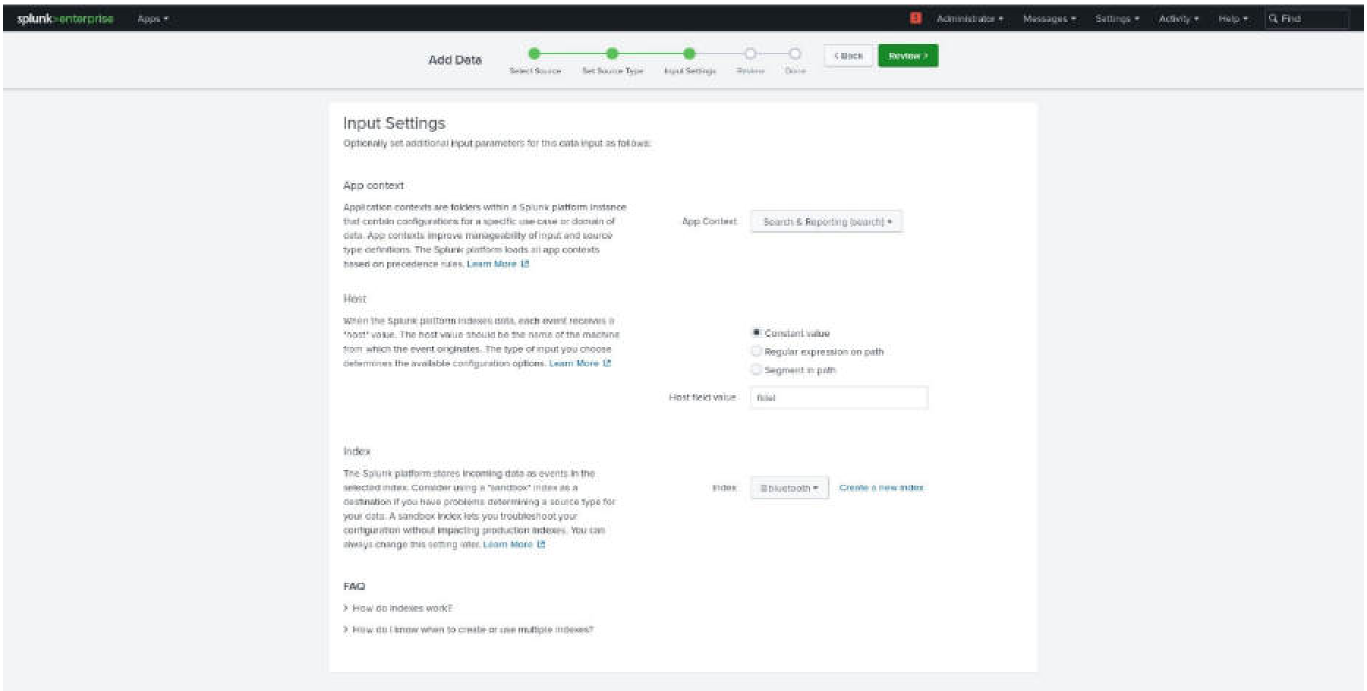
This page lets you see how the Splunk platform sees your data before indexing. If the events look correct and have the right timestamps, click "Next" to proceed. If not, use the options below to define proper event breaks and timestamps. If you cannot find an appropriate source type for your data, create a new one by clicking "Save As".

Source: /home/fidel/bluetooth.csv

ime	Command Opcode ≑	Destination ≑	extracted_Source ≑	Frame ≑	Info ≑	Length ≑	No ≑	Number of Allowed Command Packets ≑	Opcode Command Field ≑	Opcode Group Field ≑
19/24 00:04.000 AM		localhost ()	remote ()	\xE2\234\223	Rcvd Connection oriented channel	17	1			
19/24 00:04.000 AM		remote ()	localhost ()	\xE2\234\223	Sent Connection oriented channel	45	2			
19/24 00:04.000 AM		host	controller	\xE2\234\223	Rcvd Number of Completed Packets	8	3			
19/24 00:04.000 AM		localhost ()	remote ()	\xE2\234\223	Rcvd Connection oriented channel	24	4			
19/24 00:04.000 AM		localhost ()	remote ()	\xE2\234\223	Rcvd Connection oriented channel	23	5			
19/24 00:04.000 AM		remote ()	localhost ()	\xE2\234\223	Sent Connection oriented channel	19	6			
19/24 00:04.000 AM		remote ()	localhost ()	\xE2\234\223	Sent Connection oriented	11	7			

Figure 10. Type of downloaded data

At the next stage, it is necessary to bind the data to the index to display all tangential data by its name (Fig.11).



Input Settings

Optionally set additional input parameters for this data input as follows:

App context

Application contexts are folders within a Splunk platform instance that contain configurations for a specific use case or domain of data. App contexts improve manageability of input and source type definitions. The Splunk platform loads all app contexts based on precedence rules. Learn More

App Context: Search & Reporting (search) ▾

Host

When the Splunk platform indexes data, each event receives a "host" value. The host value should be the name of the machine from which the event originates. The type of input you choose determines the available configuration options. Learn More

● Constant value
○ Regular expression on path
○ Segment in path

Host field value: fidel

Index

The Splunk platform stores incoming data as events in the selected index. Consider using a "sandbox" index as a destination if you have problems determining a source type for your data. A sandbox index lets you troubleshoot your configuration without impacting production indexes. You can always change this setting later. Learn More

Index: bluetooth ▾ Create a new index

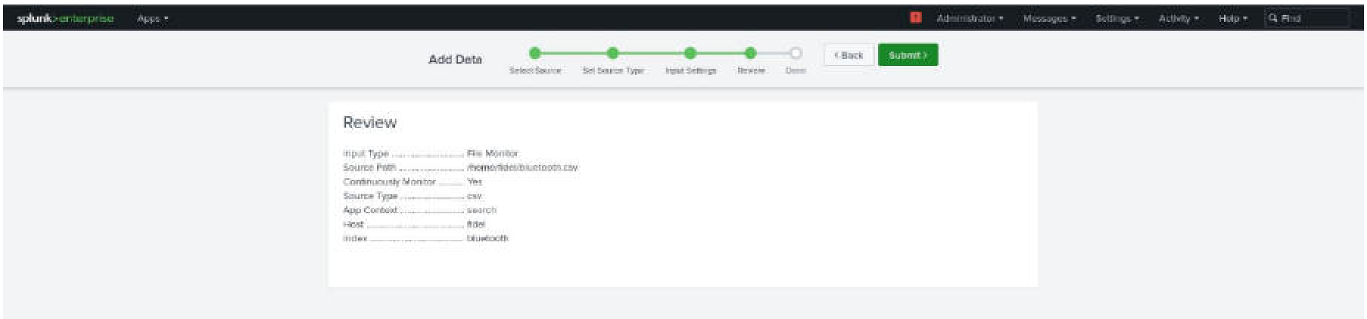
FAQ

› How do indexes work?

› How do I know when to create or use multiple indexes?

Figure 11. Assigning an index to the loaded data

All that remains is to check the correctness of the configuration and confirm the actions (Fig.12).

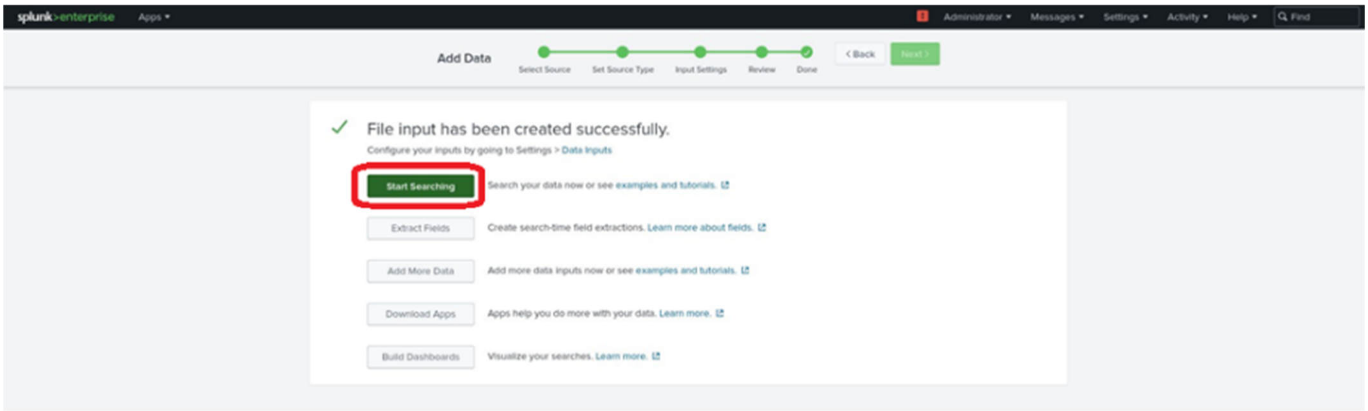


Review

Input Type .... File Monitor
Source Path .... /home/fidel/bluetooth.csv
Continuously Monitor .... Yes
Source Type .... csv
App Context .... search
Host .... fidel
Index .... bluetooth

Figure 12. Checking the settings

Once the source is configured and the data is loaded, you should review it to ensure that the settings are correct and the data is being received (Fig. 13).



✓ File input has been created successfully.

Configure your inputs by going to Settings > Data Inputs

Start Searching — Search your data now or see examples and tutorials.

Extract Fields — Create search-time field extractions. Learn more about fields.

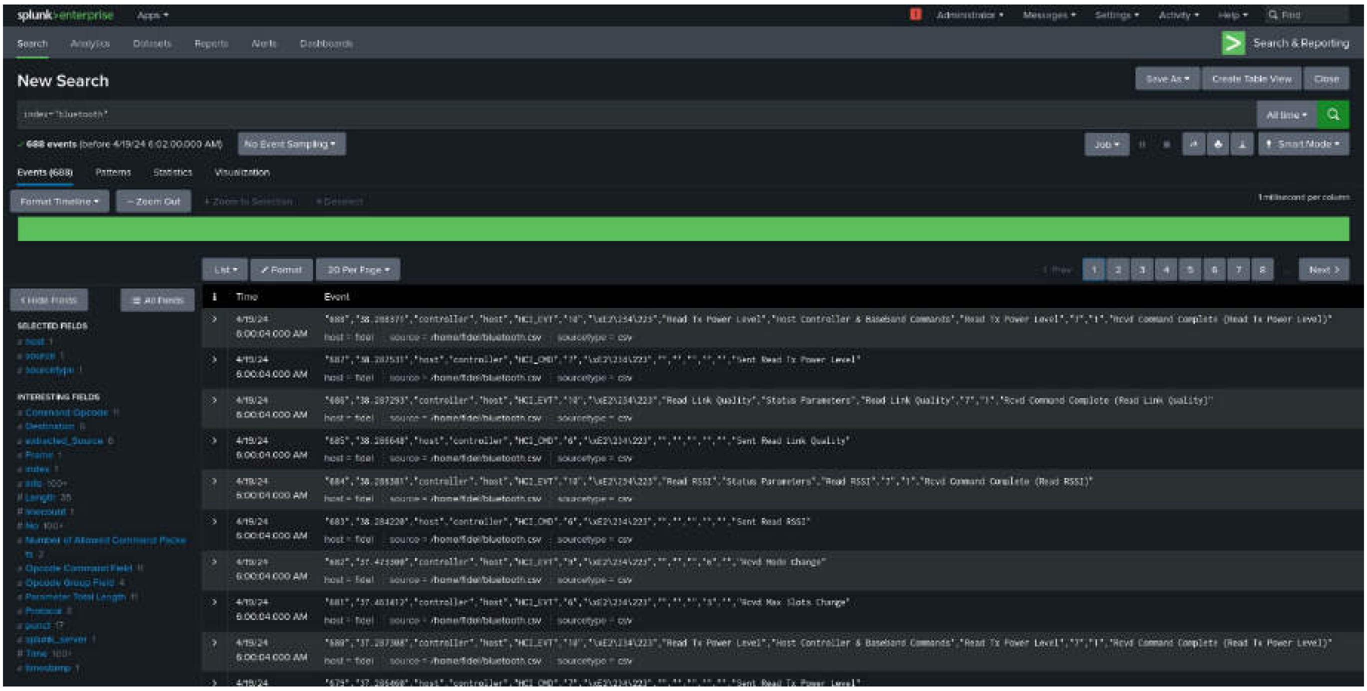
Add More Data — Add more data inputs now or see examples and tutorials.

Download Apps — Apps help you do more with your data. Learn more.

Build Dashboards — Visualize your searches. Learn more.

Figure 13. Completing the data source setup

Now we enter the query index=”bluetooth”. We see that the data is displayed correctly, all useful fields are displayed on the left and the data transfer is stable (Fig. 14).



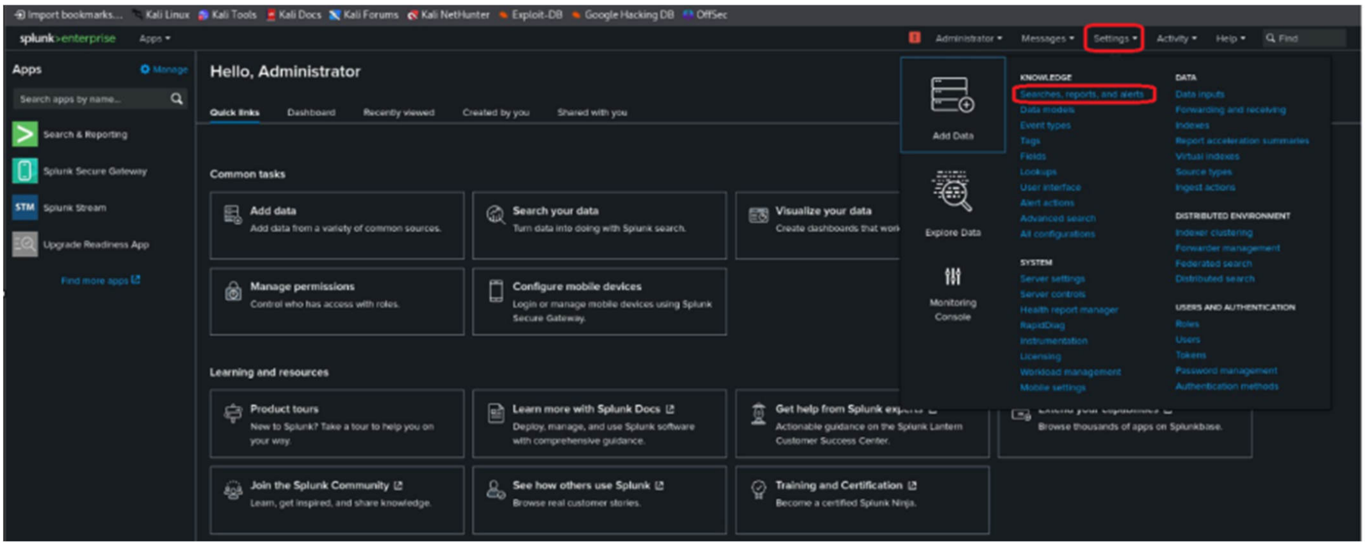
```
index="bluetooth"
✓ 688 events (before 4/19/24 6:02:00.000 AM)

i  Time                     Event
>  4/19/24 6:00:04.000 AM   "688","38.288371","controller","host","HCI_EVT","10","\xE2\234\223","Read Tx Power Level","Host Controller & Baseband Commands","Read Tx Power Level","7","1","Rcvd Command Complete (Read Tx Power Level)"
                            host = fidel   source = /home/fidel/bluetooth.csv   sourcetype = csv
>  4/19/24 6:00:04.000 AM   "687","38.287531","host","controller","HCI_CMD","7","\xE2\234\223","","","","","","Sent Read Tx Power Level"
                            host = fidel   source = /home/fidel/bluetooth.csv   sourcetype = csv
>  4/19/24 6:00:04.000 AM   "686","38.287293","controller","host","HCI_EVT","10","\xE2\234\223","Read Link Quality","Status Parameters","Read Link Quality","7","1","Rcvd Command Complete (Read Link Quality)"
                            host = fidel   source = /home/fidel/bluetooth.csv   sourcetype = csv
>  4/19/24 6:00:04.000 AM   "685","38.286648","host","controller","HCI_CMD","6","\xE2\234\223","","","","","","Sent Read Link Quality"
                            host = fidel   source = /home/fidel/bluetooth.csv   sourcetype = csv
>  4/19/24 6:00:04.000 AM   "684","38.288381","controller","host","HCI_EVT","10","\xE2\234\223","Read RSSI","Status Parameters","Read RSSI","7","1","Rcvd Command Complete (Read RSSI)"
                            host = fidel   source = /home/fidel/bluetooth.csv   sourcetype = csv
>  4/19/24 6:00:04.000 AM   "683","38.284220","host","controller","HCI_CMD","6","\xE2\234\223","","","","","","Sent Read RSSI"
                            host = fidel   source = /home/fidel/bluetooth.csv   sourcetype = csv
>  4/19/24 6:00:04.000 AM   "682","37.473309","controller","host","HCI_EVT","9","\xE2\234\223","","","","6","","Rcvd Mode Change"
                            host = fidel   source = /home/fidel/bluetooth.csv   sourcetype = csv
>  4/19/24 6:00:04.000 AM   "681","37.483412","controller","host","HCI_EVT","6","\xE2\234\223","","","","5","","Rcvd Max Slots Change"
                            host = fidel   source = /home/fidel/bluetooth.csv   sourcetype = csv
>  4/19/24 6:00:04.000 AM   "680","37.287388","controller","host","HCI_EVT","10","\xE2\234\223","Read Tx Power Level","Host Controller & Baseband Commands","Read Tx Power Level","7","1","Rcvd Command Complete (Read Tx Power Level)"
                            host = fidel   source = /home/fidel/bluetooth.csv   sourcetype = csv
>  4/19/24                  "679","37.286468","host","controller","HCI_CMD","7","\xE2\234\223","","","","","","Sent Read Tx Power Level"
```

Figure 14. Displaying data in the Splunk platform

5.1. Developing correlation rules to detect ongoing attacks

We need to develop correlation rules that automatically search for data according to a given pattern and generate notifications to detect attacks. To do this, go to settings on the main page and select the category "Searches, reports, and alerts" (Fig.1 5).



Add data from a variety of common sources.

Turn data into doing with Splunk search.

Create dashboards that wor

Control who has access with roles.

Login or manage mobile devices using Splunk Secure Gateway.

New to Splunk? Take a tour to help you on your way.

Deploy, manage, and use Splunk software with comprehensive guidance.

Actionable guidance on the Splunk Lantern Customer Success Center.

Browse thousands of apps on Splunkbase.

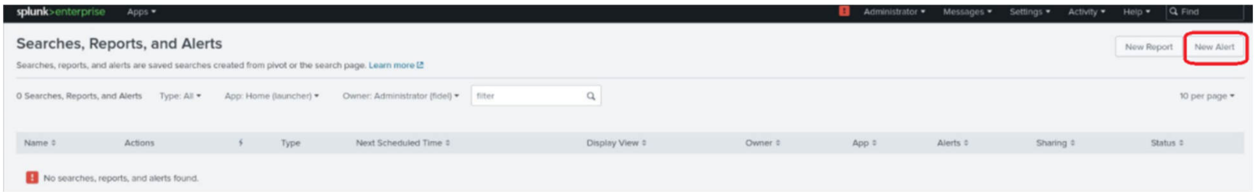
Learn, get inspired, and share knowledge.

Browse real customer stories.

Become a certified Splunk Ninja.

Figure 15. Navigate from the main page to create notifications.

There are currently no alert rules. Let's create one by clicking the 'New Alert button'.



Searches, reports, and alerts are saved searches created from pivot or the search page. Learn more

No searches, reports, and alerts found.

Figure 16. Notification management menu

Now you need to create a notification. To do this, be sure to specify the rule name in the Title field. Not necessarily, but to provide SOC analysts with more context, specify a description in the Description field. In the Search field, directly specify the search parameter that will be used to search for the required data. The App field is automatically assigned to the part of the SIEM solution currently in use. In Permissions, you need to specify "Shared in app" so that the alert is visible to all SOC analysts. In order not to overload the system and at the same time maintain the speed of reaction, we specify that the rule should be triggered every hour and search for information for the previous hour. In the Trigger Conditions section, you should leave the recommended default parameters. And finally, in the Trigger action section, specify what action Splunk should perform. In our case, Splunk will highlight the notification and save it locally. In the same field, we assign criticality to the created notification. Splunk uses its own query language, Splunk Processing Language (SPL). It is very similar to the SQL query language and is easy to use. Queries to detect attacks will look like this:

1. The DOS attack we used uses the Bluetooth authentication protocol – L2CAP. We can detect it by the large number of these packets, which is more than expected.

```
index="bluetooth" Protocol=L2CAP
| stats values(extracted_Source) as Source values(Destination) count by Protocol
| where count > 1000
```

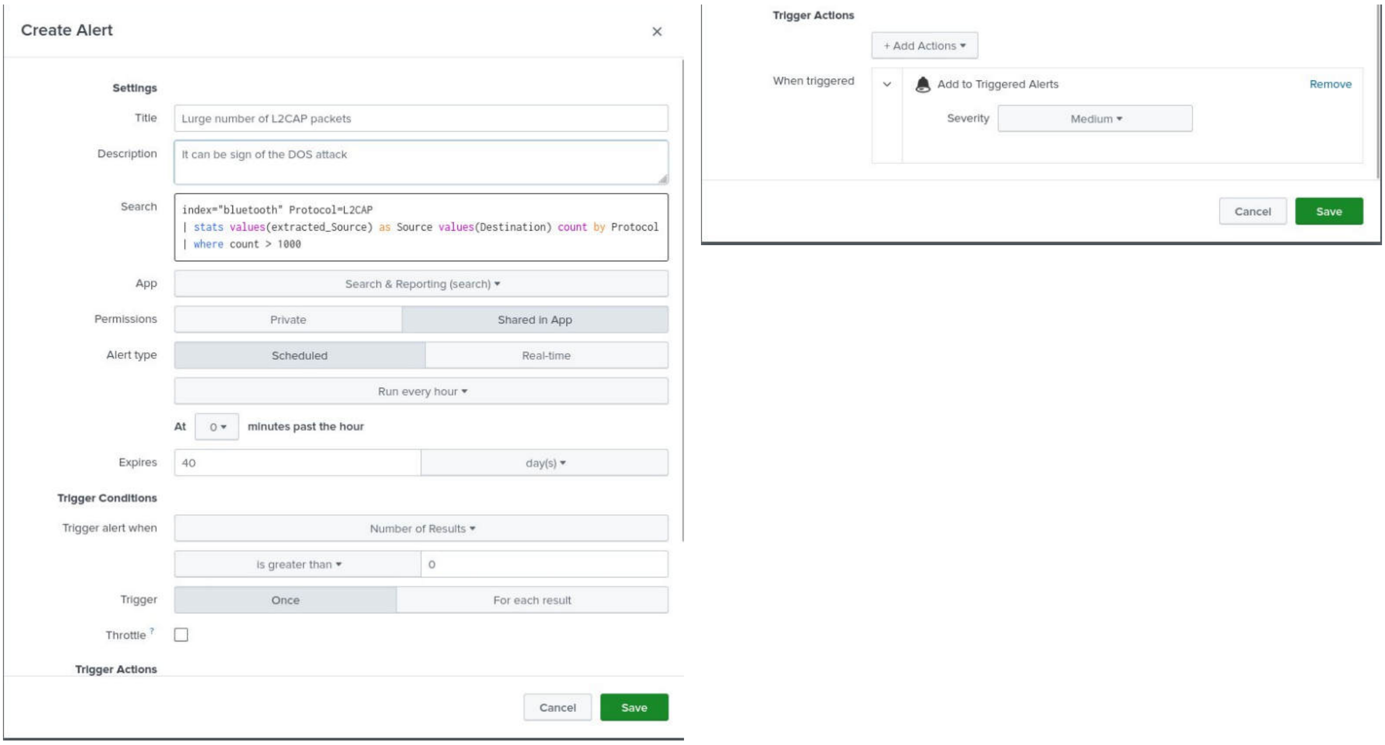
2. Bluetooth Spoofing can be detected as a connection from an unexpected MAC address.

```
index="bluetooth" Protocol="SMP" AND NOT (extracted_Source IN ("localhost()"))
| table Destination extracted_Source Info
```

DOI: https://doi.org/10.53052/9788367652292.16

3. Malware transmission can be detected as a file transmission with an executable extension.

```
index="bluetooth" Protocol="OBEX" Info IN (*.exe*, *.ps1*, *.sh*, *.app*, *.bat*)
| table Destination extracted_Source Info
```



Create Alert

Settings

Title: Lurge number of L2CAP packets

Description: It can be sign of the DOS attack

Search:

```
index="bluetooth" Protocol=L2CAP
| stats values(extracted_Source) as Source values(Destination) count by Protocol
| where count > 1000
```

App: Search & Reporting (search)

Permissions: Private / Shared in App

Alert type: Scheduled / Real-time

Run every hour

At 0 minutes past the hour

Expires: 40 day(s)

Trigger Conditions

Trigger alert when: Number of Results

is greater than 0

Trigger: Once / For each result

Throttle ? □

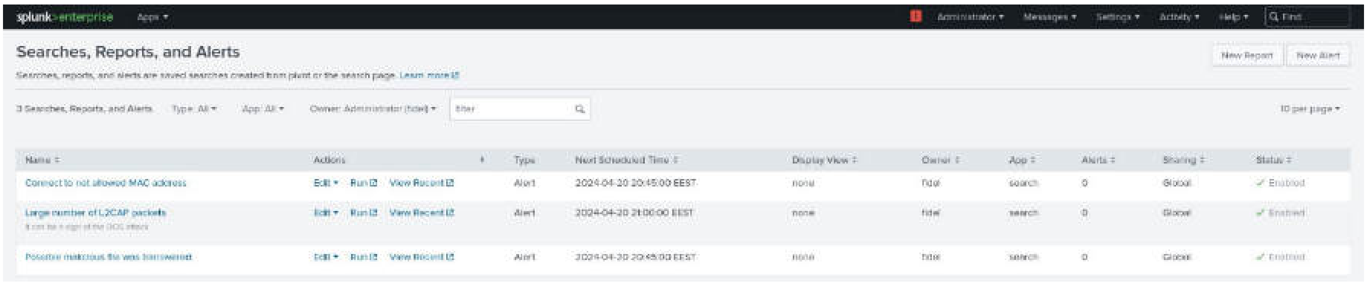
Trigger Actions

When triggered: Add to Triggered Alerts

Severity: Medium

Figure 17. Create a new alert

Using this method, we will create three rules that will detect three conducted attack vectors (Fig. 18).



Searches, Reports, and Alerts

Searches, reports, and alerts are saved searches created from pivot or the search page. Learn more

3 Searches, Reports, and Alerts Type: All App: All Owner: Administrator (fidel)

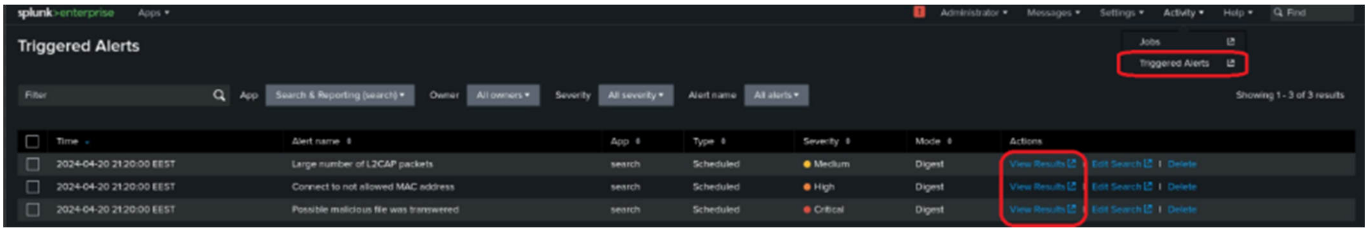
Name	Actions	Type	Next Scheduled Time	Display View	Owner	App	Alerts	Sharing	Status
Connect to not allowed MAC address	Edit Run View Recent	Alert	2024-04-20 20:45:00 EEST	none	fidel	search	0	Global	✓ Enabled
Large number of L2CAP packets It can be a sign of the DOS attack	Edit Run View Recent	Alert	2024-04-20 21:00:00 EEST	none	fidel	search	0	Global	✓ Enabled
Possible malicious file was transmitted	Edit Run View Recent	Alert	2024-04-20 20:45:00 EEST	none	fidel	search	0	Global	✓ Enabled

Figure 18. Notification rules have been created

Developing and implementing these correlation rules significantly improves our ability to detect and mitigate Bluetooth-based attacks in real time. The rules provide a robust mechanism for detecting suspicious traffic patterns, ensuring that security teams can quickly and effectively respond to potential threats. These detection methods can be customized and expanded as Bluetooth technology evolves, providing a scalable approach to secure wireless communications.

5.2. Assessing and analyzing attack notifications

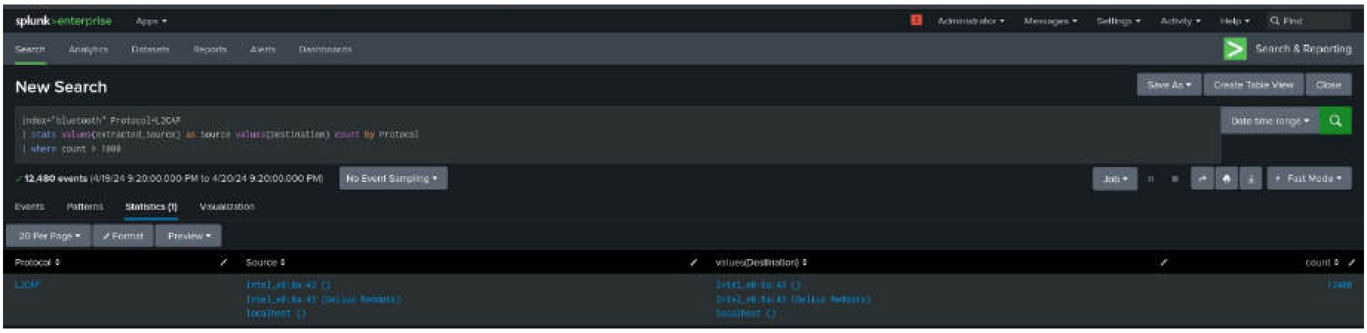
Now we can check the worn-out notifications by going to Activity, then Triggered alerts. Here, you can see the time of the notification, its name, and its criticality. To go directly to the notification analysis, you need to click "View Results".



□	Time	Alert name	App	Type	Severity	Mode	Actions
□	2024-04-20 21:20:00 EEST	Large number of L2CAP packets	search	Scheduled	● Medium	Digest	View Results | Edit Search | Delete
□	2024-04-20 21:20:00 EEST	Connect to not allowed MAC address	search	Scheduled	● High	Digest	View Results | Edit Search | Delete
□	2024-04-20 21:20:00 EEST	Possible malicious file was transwered	search	Scheduled	● Critical	Digest	View Results | Edit Search | Delete

Figure 19. Checking triggered notifications

This is a link to the search added to the notifications in section 3.2. In the first case, we can see the protocol used for filtering (L2CAP), the devices that communicated (Intel_e0:8a:43 (disguised as Gelius Reddots wireless headphones) and localhost () – the current device sending data to Splunk) and the number of packets transmitted. 12480 is significantly more than a regular connection initialization – therefore, this is a DOS attack (Fig. 20).



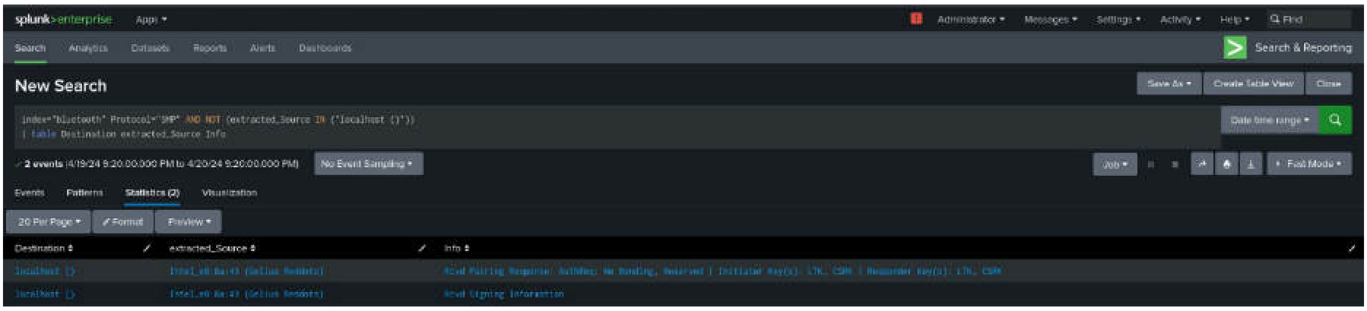
```
index="bluetooth" Protocol=L2CAP
| stats values(extracted_Source) as Source values(Destination) count by Protocol
| where count > 1000
```

✓ 12,480 events (4/19/24 9:20:00.000 PM to 4/20/24 9:20:00.000 PM)

Protocol	Source	values(Destination)	count
L2CAP	Intel_e0:8a:43 () Intel_e0:8a:43 (Gelius Reddots) localhost ()	Intel_e0:8a:43 () Intel_e0:8a:43 (Gelius Reddots) localhost ()	12480

Figure 20. The result of the analysis of the notification about the detection of a DOS attack

In the second case, Intel_e0:8a:43 (Gelius Reddots) successfully connected to localhost (the device that sends data to Splunk). The connection of these wireless headphones is expected. However, this MAC address is not on the list of allowed ones. From this, we can conclude that this is Bluetooth Spoofing (Fig. 21).



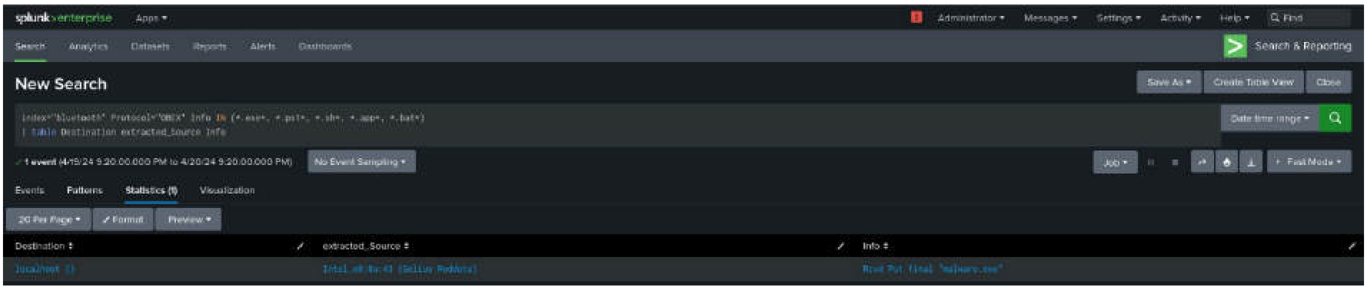
```
index="bluetooth" Protocol="SMP" AND NOT (extracted_Source IN ("localhost ()"))
| table Destination extracted_Source Info
```

✓ 2 events (4/19/24 9:20:00.000 PM to 4/20/24 9:20:00.000 PM)

Destination	extracted_Source	Info
localhost ()	Intel_e0:8a:43 (Gelius Reddots)	Rcvd Pairing Response: AuthReq: No Bonding, Reserved | Initiator Key(s): LTK, CSRK | Responder Key(s): LTK, CSRK
localhost ()	Intel_e0:8a:43 (Gelius Reddots)	Rcvd Signing Information

Figure 21. Analysis result of Bluetooth Spoofing attack detection notification

In the third case, we can see that a malware.exe file was forwarded to localhost (the device that sends data to Splunk) by Intel_e0:8a:43 (Gelius Reddots). This is not normal activity for wireless headphones, and even by the name of the file that was transferred – can guess that malware was sent (Fig. 22).



```
index="bluetooth" Protocol="OBEX" Info IN (*.exe*, *.ps1*, *.sh*, *.app*, *.bat*)
| table Destination extracted_Source Info
```

✓ 1 event (4/19/24 9:20:00.000 PM to 4/20/24 9:20:00.000 PM)

Destination	extracted_Source	Info
localhost ()	Intel_e0:8a:43 (Gelius Reddots)	Rcvd Put final "malware.exe"

Figure 22. The result of the analysis of the notification about the detection of the transfer of the executable file

The alert system created using correlation rules and the Splunk SIEM platform has proven to be an invaluable tool for identifying Bluetooth-related attacks. By analyzing triggered alerts, SOC analysts can assess the severity of an attack, fine-tune the detection system, and ensure rapid response to potential threats. This process not only improves security monitoring but also helps refine and improve the overall cybersecurity posture of Bluetooth-enabled environments.

6. Conclusions

Despite its immense popularity and widespread use in modern digital devices, Bluetooth technology remains vulnerable to several severe security threats. Research has highlighted the risks of attacks such as man-in-the-middle (MITM), brute-force PIN cracking, DoS, and MAC address spoofing, which can disrupt devices, leak sensitive data, and harm both individual users and organizations. As Bluetooth is prevalent in personal and corporate networks, the increasing prevalence of these threats emphasizes the need for enhanced security measures.

This study aimed to develop an effective method for detecting attacks on the Bluetooth protocol by leveraging the Wireshark network analyzer and the Splunk SIEM system. Wireshark enabled deep packet analysis, detecting suspicious patterns and anomalies within the Bluetooth traffic, while Splunk's SIEM capabilities provided real-time monitoring, automatic correlation rules, and notification generation, simplifying the detection and response process.

One of the key contributions of this study is the development and testing of algorithms that accurately detect common Bluetooth attacks, including DoS, Spoofing, and malicious file transfer attempts. The experimental results demonstrated the effectiveness of these algorithms in identifying and mitigating threats in a timely manner, reducing the risk of system compromise. Wireshark’s capabilities, combined with Splunk's real-time monitoring, proved to be a highly efficient approach to monitoring Bluetooth traffic and responding to potential attacks.

In this work, we showcased the practical application of correlation rules and monitoring systems to detect various Bluetooth-based threats. The analysis focused on critical attack vectors like DoS, MITM, and MAC address spoofing, all of which present significant risks to Bluetooth-enabled devices. The integration of Wireshark and Splunk allowed for real-time detection and analysis of malicious traffic patterns, providing crucial insights into ongoing threats.

Overall, the results of this study indicate that the methods employed offer a robust framework for identifying Bluetooth security vulnerabilities. Continuous monitoring and refined correlation rules enable organizations to promptly detect and respond to Bluetooth attacks, significantly enhancing the security of wireless communications.

Further refinement of these systems will be necessary to address new and emerging Bluetooth attack vectors, ensuring continued protection of devices in an ever-evolving digital landscape. This research not only contributes to the current understanding of Bluetooth security but also opens new opportunities for developing advanced methods and technologies to safeguard wireless networks against future attacks.

In conclusion, the proposed methods provide reliable and fast Bluetooth traffic monitoring, enabling timely detection of attacks and swift response to potential threats. This study significantly contributes to the improvement of Bluetooth device security and presents new possibilities for further research in wireless network protection and cybersecurity technologies.

Reference

1. Sairam, K.; Gunasekaran, N.; Reddy, S. Bluetooth in Wireless Communication. IEEE Communications Journal 2020, 7, 34-45. [CrossRef].Bluetooth. Available online: https://bluetooth.com/ (accessed on 29 March 2024).

1. Bluetooth SIG. Understanding Bluetooth Range. Available online: https://www.bluetooth.com/learn-about-bluetooth/key-attributes/range/ (accessed on 29 March 2024).

2. Bluetooth SIG. Learn about Bluetooth: Radio Versions. Available online: https://www.bluetooth.com/learn-about-bluetooth/bluetooth-technology/radio-versions/ (accessed on 29 March 2024).

3. Franklin, C.; Pollette, C. How Bluetooth Works. Available online: https://electronics.howstuffworks.com/bluetooth.htm#pt1 (accessed on 29 March 2024).

4. Ndatiya, V.; Xiao, J.; Meng, K. Network Forensics Analysis Using Wireshark. International Journal of Sensor Networks 2015, 10(2), 75-83. [CrossRef].

5. Banerjee, U.; Vashistha, A.; Saxena, M. Evaluation of Wireshark Capabilities as a Tool for Intrusion Detection Systems. International Journal of Computer Applications 2010, 6(7), 45-50.

6. Hebbar, R.; Mohan, K. Packet Analysis with Network Intrusion Detection Systems. International Journal of Science and Research 2015, 4(2), 120-130.

7. Iqbal, H.; Naaz, S. Wireshark as a Tool for Detecting Various Attacks on Local Area Networks. International Journal of Computer Science and Engineering 2019, 7(5), 78-85. [CrossRef].
8. Wireshark User Guide. Available online: https://www.wireshark.org/docs/wsug_html_chunked/ (accessed on 29 March 2024).
9. Ekelhart, A. Harnessing Logs – A Dictionary for Semantic Security Analysis. ScienceDirect 2019, 110-119. [CrossRef].
10. Moradian, O. S. E. Secure Audit Log Management. ScienceDirect 2018, 1250-1258. [CrossRef].
11. Splunk Enterprise Security Features. Available online: https://www.splunk.com/en_us/products/splunk-enterprise-security-features.html (accessed on 29 March 2024).
12. Minar, N.; Tarique, M. Bluetooth Security Threats and Solutions. International Journal of Distributed and Parallel Systems 2012, 3, 127. [CrossRef].
13. Hassan, A.; Bibon, S.; Hossein, M.; Atikuzzaman, M. Bluetooth Technology Security Threats. Computers & Security 2018, 74, 308-322. [CrossRef].
14. Lonzetta, A.; Cope, P.; Campbell, D.; Mohd, B. Security Vulnerabilities in Bluetooth Used in IoT. Journal of Sensor and Actuator Networks 2018, 7, 28. [CrossRef].
15. Toivanen, P.; Haataja, K. Two Practical Attacks Against Bluetooth Secure Simple Pairing and Countermeasures. IEEE Transactions on Wireless Communications 2010, 9, 384-392. [CrossRef].
16. Sandhya, S.; Devi, S. Contention for Man-in-the-Middle Attacks in Bluetooth Networks. Proceedings of the 4th IEEE International Conference on Computational Intelligence and Communication Networks, Mathura, India, 3-5 November 2012.
17. Kaviyarasu, S.; Mathupandian, P. Bluetooth Jacking Technology: A Review. International Journal of Trend Research and Development 2016, 3. [CrossRef].
18. Bon, M. A Basic Introduction to BLE Security. Available online: https://www.digikey.com/eewiki/display/Wireless/A+Basic+Introduction+to+BLE+Security (accessed on 30 March 2024).

DOI: https://doi.org/10.53052/9788367652292.16