UWB Real Time Locating Systems

How Secure Radio Communications May Fail in Practice

Andrea Palanca | Luca Cremona | Roya Gordon

Overview

01 Introduction

02 Methodology of Research

03 Attack Demos

04 Remediations

05 Summary & Key Takeaways

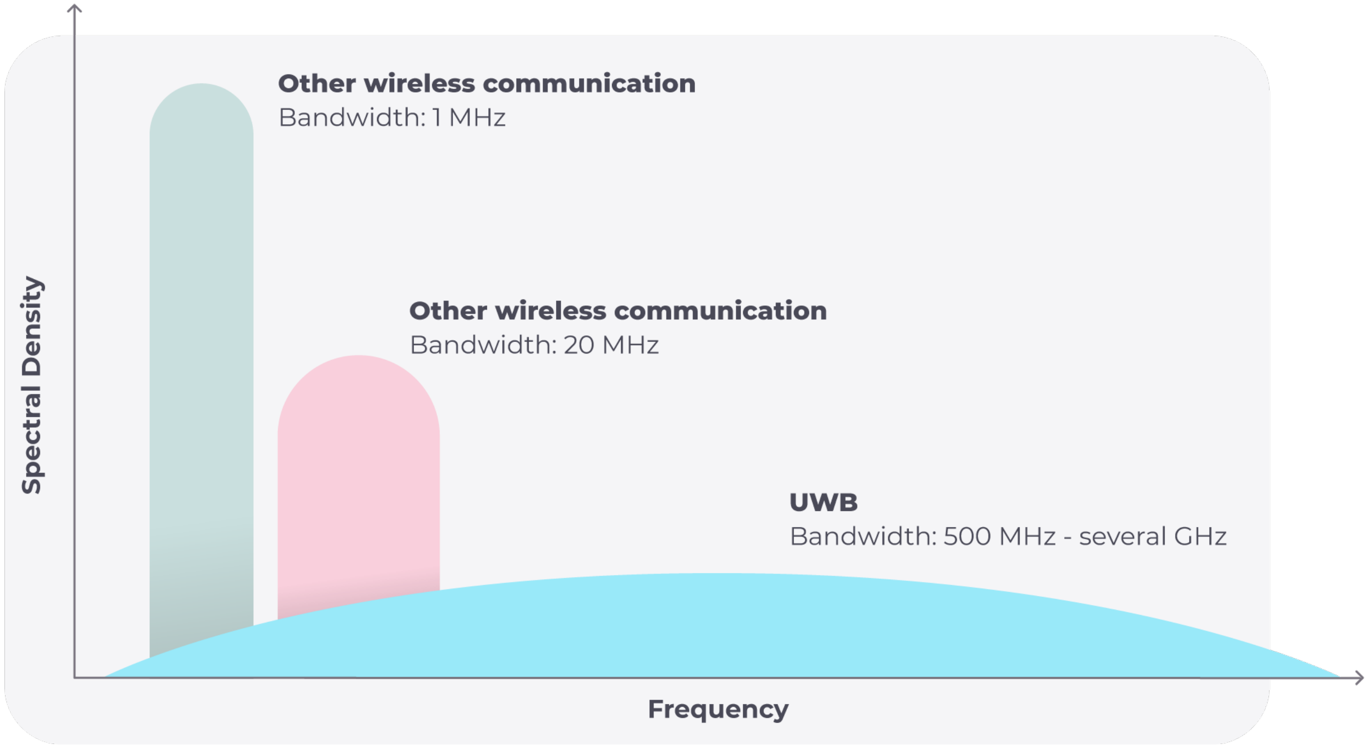
Introduction

Introduction

Wireless communication systems are susceptible to various security threats that can compromise their reliability and impact production operations

UWB

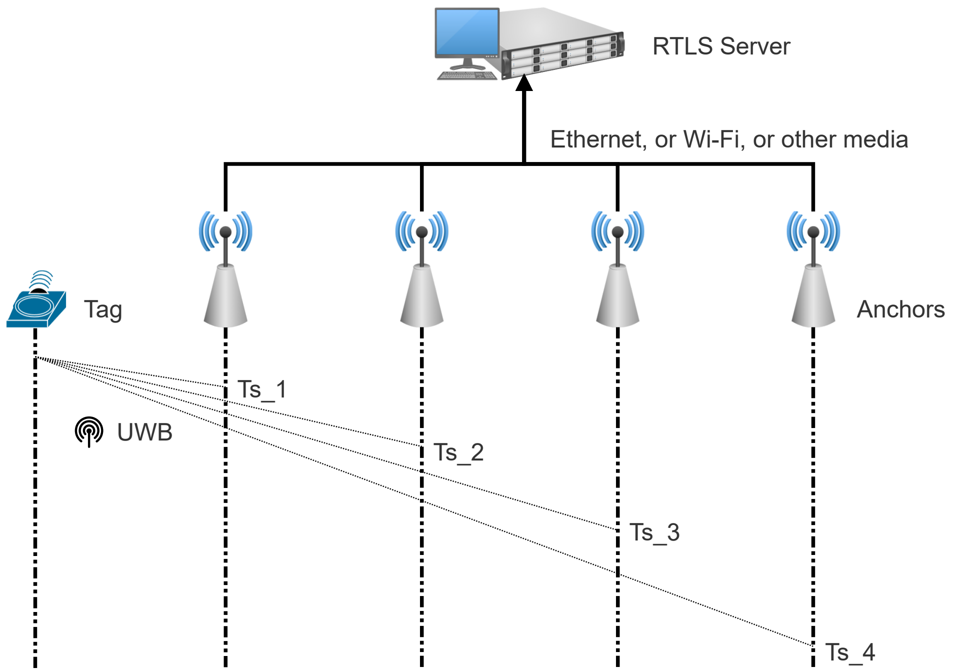
- Ideal for **short-range devices**
- Can transmit information quickly over short distances
- Ability to **send data through solid objects** like walls and other barriers
- UWB is the preferred communication protocol for RTLS



Spectral density for UWB and narrowband - Source: [FiRa Consortium]

Introduction – Cont'd

Wireless communication systems are susceptible to various security threats that can compromise their reliability and impact production operations



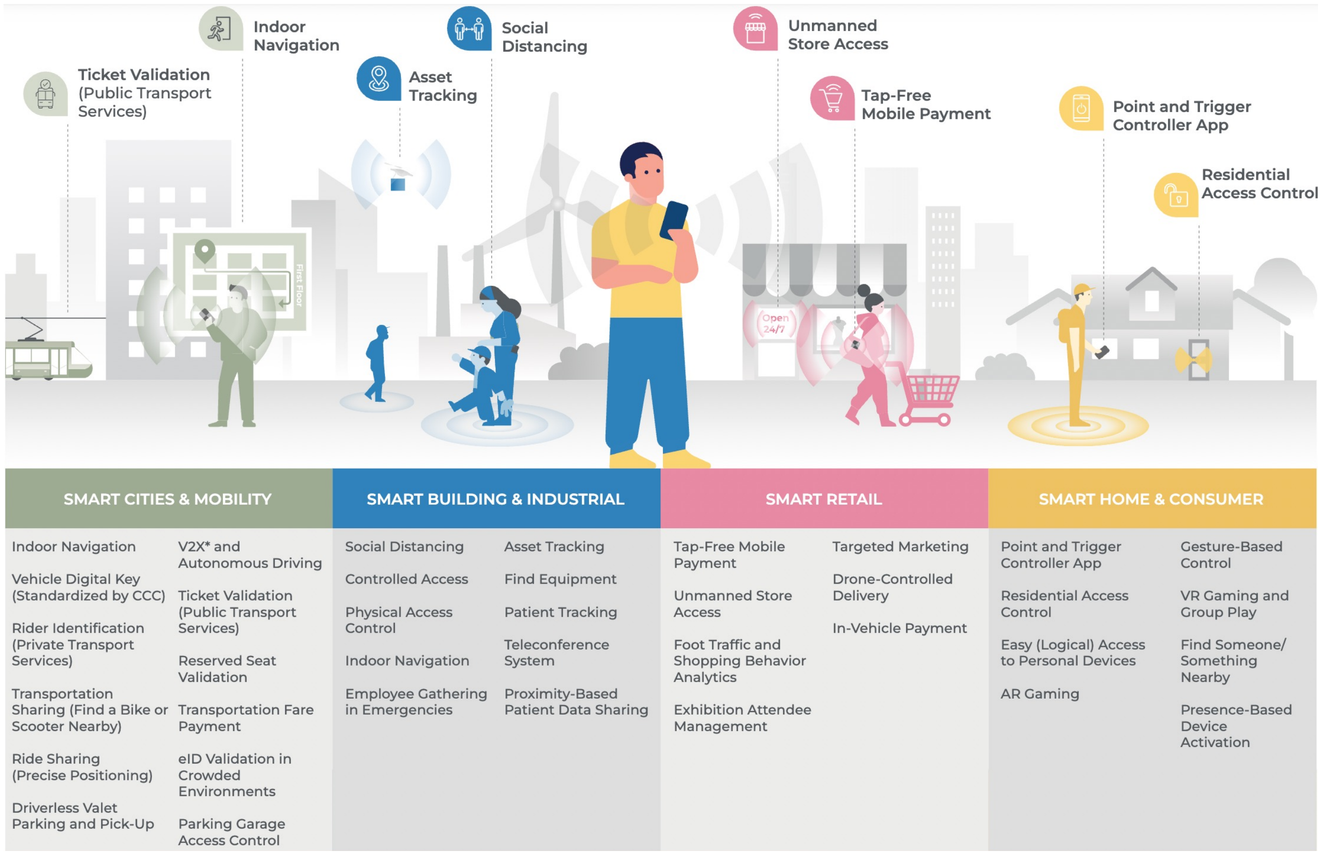
Operation of an UWB TDoA RTLS

RTLS

- Uses UWB signals to **locate stationary/mobile objects**
- 3 **components**:
  - Tags
  - Anchors
  - Server that computes, shows and stores tag positions
- The time of arrival is analyzed to **determine the position of a tag**

Introduction – Cont’d

Wireless communication systems are susceptible to various security threats that can compromise their reliability and impact production operations



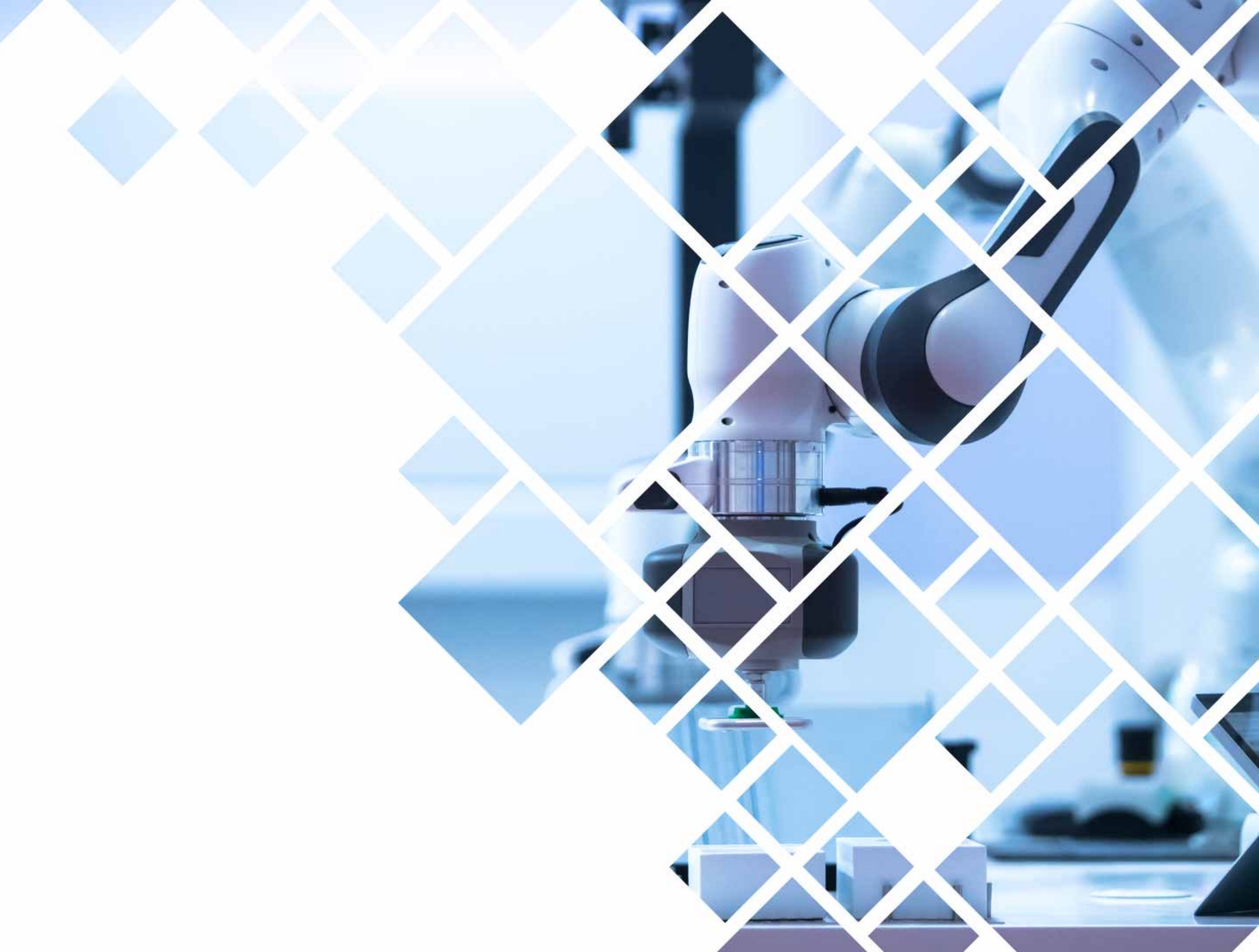
Source: *FiRa Consortium*

Cyber Threats

Networks will be **vulnerable to attacks** by cyber criminals who are seeking to exploit vulnerabilities in order to gain access to sensitive data or **disrupt operations**.

black hat USA 2022

Motivation

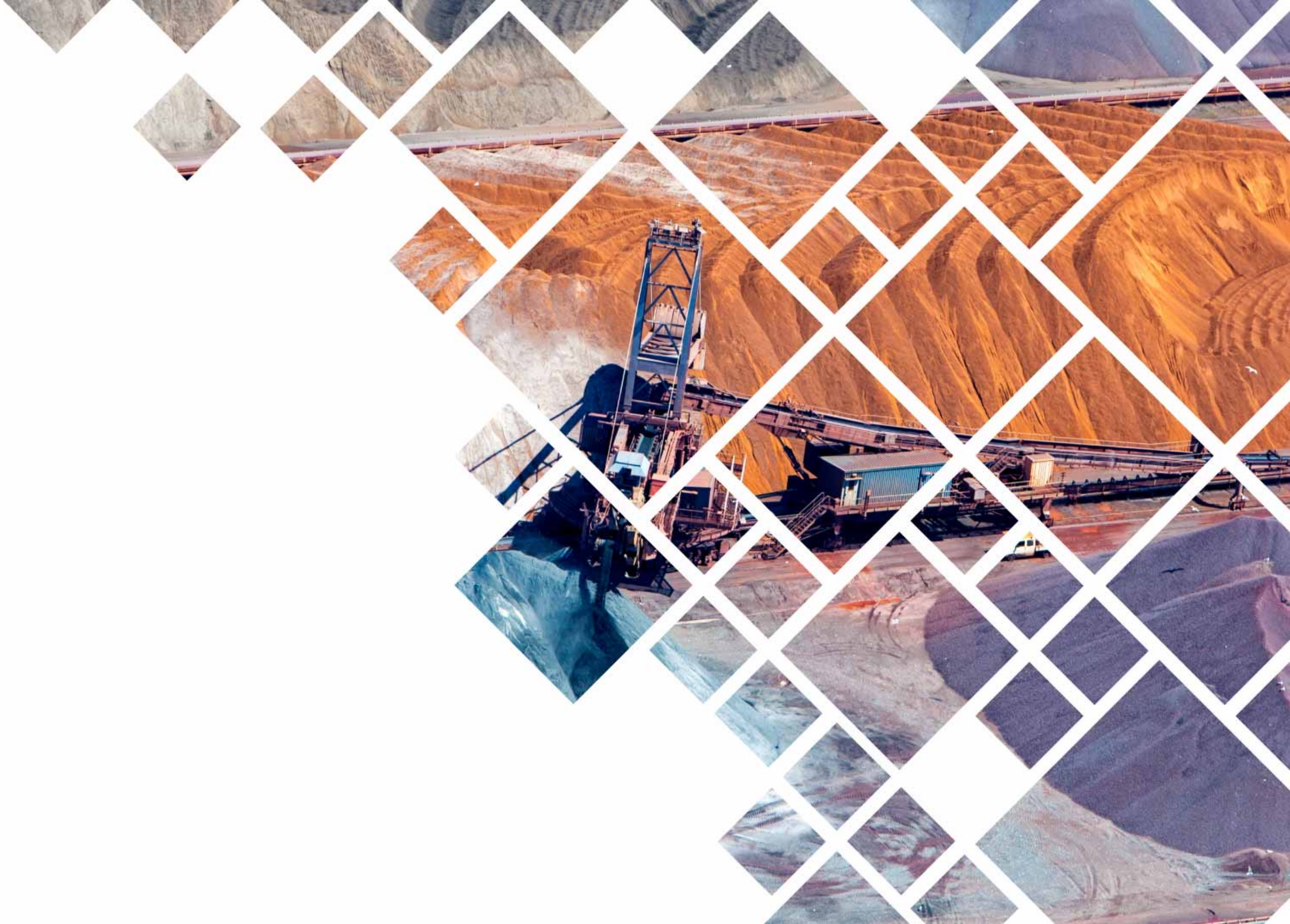


Motivation

According to the FiRa consortium, in 2018 there was an increased demand for "improvements to existing modulations to increase the integrity and accuracy of ranging measurements"

- In 2020, the Institute of Electrical and Electronic Engineers (IEEE) released **standard 802.15.4**
- IEEE quickly followed up with the **802.15.4z amendment**, also released in 2020
- **Synchronization** and **exchange of location** data are considered "out-of-scope" by the standard
- These communications, whose design is left entirely to vendors, are critical aspects for the **overall posture of TDoA RTLS**
- Additionally, there has not been any research **on UWB focusing on this specific problem**

This is what motivated us to take a deeper look into how threat actors can exploit this vulnerability and disrupt environments utilizing UWB RTLS.



Scope

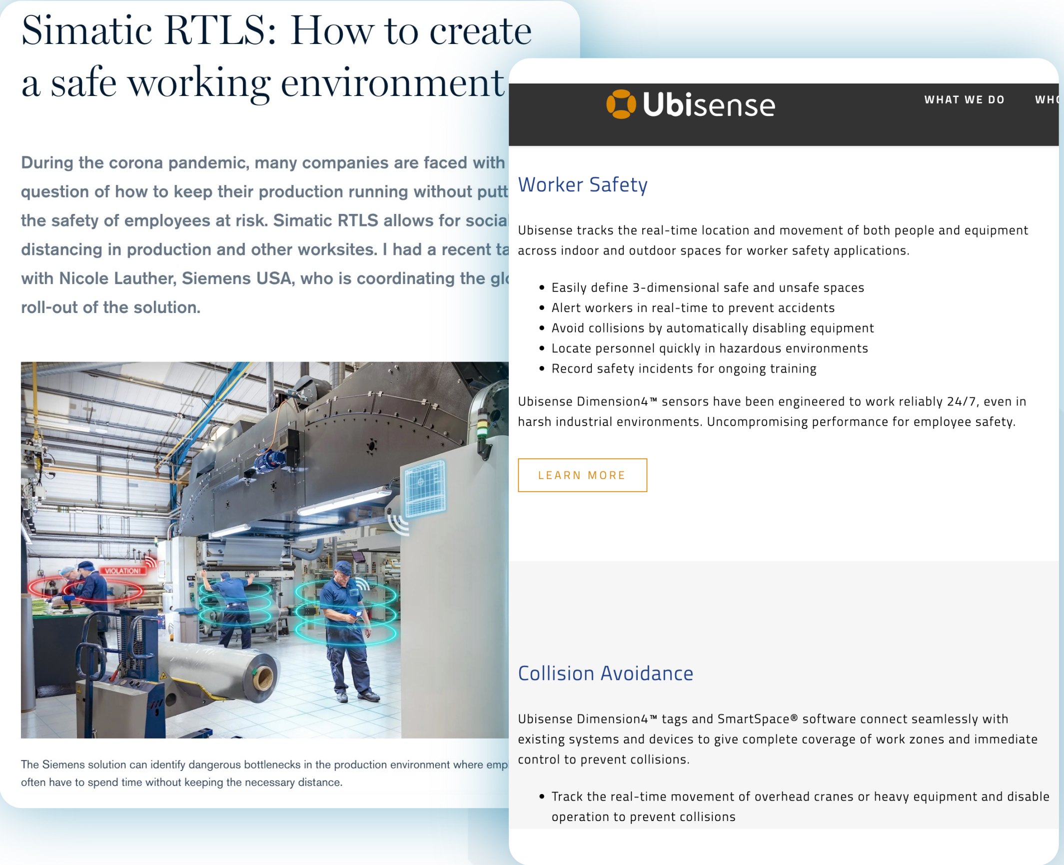
Industry Scope

Focus: industrial and healthcare sectors

- Highly targeted
- UWB RTLS widely used in critical applications

Examples: use cases where UWB is protecting people's lives

- Employee and patient tracking
- Geofencing
- Contact tracing



Simatic RTLS: How to create a safe working environment

During the corona pandemic, many companies are faced with question of how to keep their production running without putt the safety of employees at risk. Simatic RTLS allows for socia distancing in production and other worksites. I had a recent ta with Nicole Lauther, Siemens USA, who is coordinating the gl roll-out of the solution.

The Siemens solution can identify dangerous bottlenecks in the production environment where emp often have to spend time without keeping the necessary distance.

Ubisense WHAT WE DO WHO

Worker Safety

Ubisense tracks the real-time location and movement of both people and equipment across indoor and outdoor spaces for worker safety applications.

- Easily define 3-dimensional safe and unsafe spaces
- Alert workers in real-time to prevent accidents
- Avoid collisions by automatically disabling equipment
- Locate personnel quickly in hazardous environments
- Record safety incidents for ongoing training

Ubisense Dimension4™ sensors have been engineered to work reliably 24/7, even in harsh industrial environments. Uncompromising performance for employee safety.

LEARN MORE

Collision Avoidance

Ubisense Dimension4™ tags and SmartSpace® software connect seamlessly with existing systems and devices to give complete coverage of work zones and immediate control to prevent collisions.

- Track the real-time movement of overhead cranes or heavy equipment and disable operation to prevent collisions

Examples of UWB RTLS use cases advertised by vendors

Analyzed Solutions

- Our research was performed on the following solutions, which **target the industrial and healthcare sectors**:

Sewio Indoor Tracking
RTLS UWB Wi-Fi Kit

Avalue Renity
Artemis Enterprise Kit

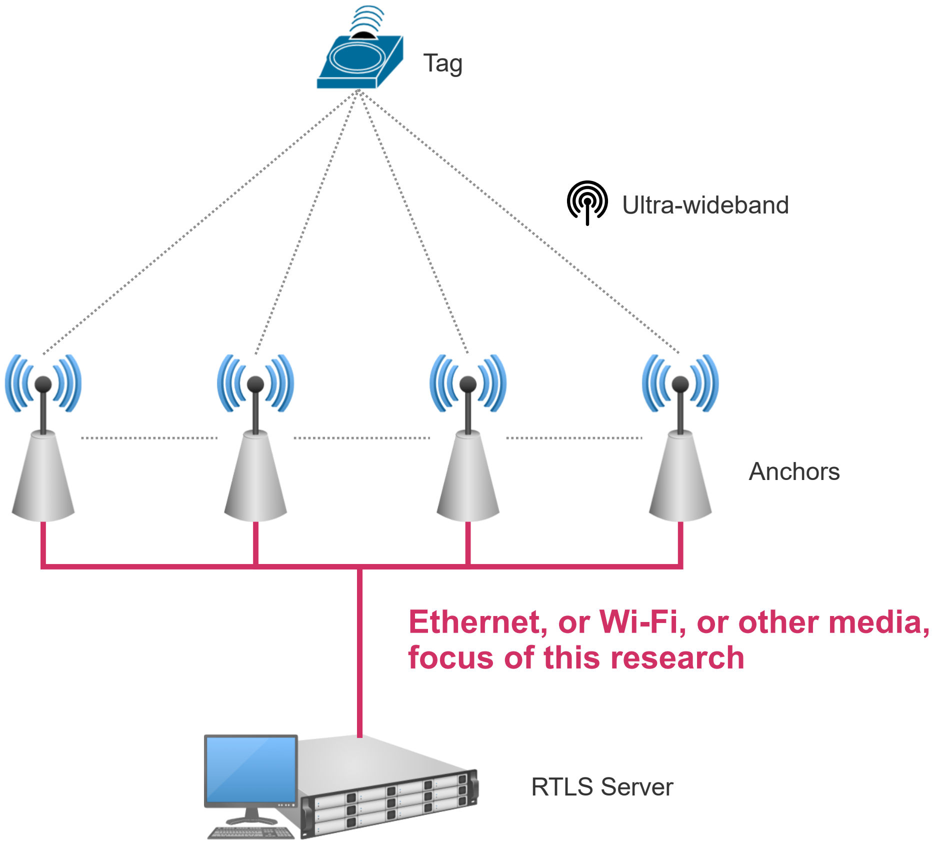
- Both these UWB RTLS kits come equipped with a set of tags, anchors, and a server software that provides the **aforementioned safety features**

Technical Scope

- Network communications occurring in a **normal RTLS infrastructure**:
    - UWB
        - Tags to anchors
        - Anchors to anchors
    - Ethernet/Wi-Fi/other
        - Anchors to RTLS server

As in a chain, a flaw in **any** of these communications may lead to a compromise of the **entire** infrastructure

- Up to now, security research has exclusively focused on the analysis of UWB signals. **This is the first research analyzing the communications on Ethernet/Wi-Fi/etc.**



Architecture of an UWB TDoA RTLS

TDoA Background and Theory

Packet Taxonomy

In a TDoA RTLS there are normally two kinds of packets exchanged among anchors and server

Synchronization packets

- Anchors' clocks are usually not in-sync, (different boot times, clock drifts, etc.)
- A reference anchor continuously sends an UWB signal that is received by all non-reference anchors
- The reference anchor sends a synchronization packet containing the transmission timestamp, the non-reference anchors a synchronization packet containing the receiving timestamp
- The server uses this information to build a common notion of time

Positioning packets

- A tag emits an UWB signal that is received by all anchors
- All anchors send the timestamps at which they received the UWB signal from the tag to the central positioning server, inside positioning packets

Algorithm Details

The routine implemented in TDoA RTLS can usually be divided in two major phases

Clock Synchronization

- There are many synchronization algorithms in literature. This work used the Linear Interpolation algorithm
- The basic idea is to convert all anchors' timestamps to a common clock domain, so that they can be compared. These converted timestamps are called Global Times (GT)

Position Estimation

- We cannot immediately derive the distances from the GTs as we are missing the transmission instants
- We can, however, correlate the difference of GTs to the difference of distances. This is why the algorithm is called Time Difference of Arrival

$$Delta(i,j,t) = \left(GT(reference,j,t) - GT(i,j,t)\right) * c = GT(reference,j,t) * c - GT(i,j,t) * c$$
$$= d(reference,j,t) - d(i,j,t)$$

*All details in our whitepaper! Download it from the **briefing page**, or from the **nozominetworks.com** website*

Algorithm Details – Cont'd

The routine implemented in TDoA RTLS can usually be divided in two major phases

Position Estimation – Cont'd

- Eventually, a non-linear system of equations can be set up to solve for Xj,t, Yj,t, and Zj,t, i.e., the position of tag j at the instant t:

$$
\begin{cases}
\big(pTs(reference,j,t) - sTS(reference,t) - (CS(1,t) * \big(pTs(1,j,t) - sTS(1,t)\big) + ToF(1))\big) * c = \\
\sqrt{(Xj,t - Xreference)^2 + (Yj,t - Yreference)^2 + (Zj,t - Zreference)^2} - \sqrt{(Xj,t - X1)^2 + (Yj,t - Y1)^2 + (Zj,t - Z1)^2} \\
\dots \\
\big(pTs(reference,j,t) - sTS(reference,t) - (CS(N,t) * \big(pTs(N,j,t) - sTS(N,t)\big) + ToF(N))\big) * c = \\
\sqrt{(Xj,t - Xreference)^2 + (Yj,t - Yreference)^2 + (Zj,t - Zreference)^2} - \sqrt{(Xj,t - XN)^2 + (Yj,t - YN)^2 + (Zj,t - ZN)^2}
\end{cases}
$$

Summary

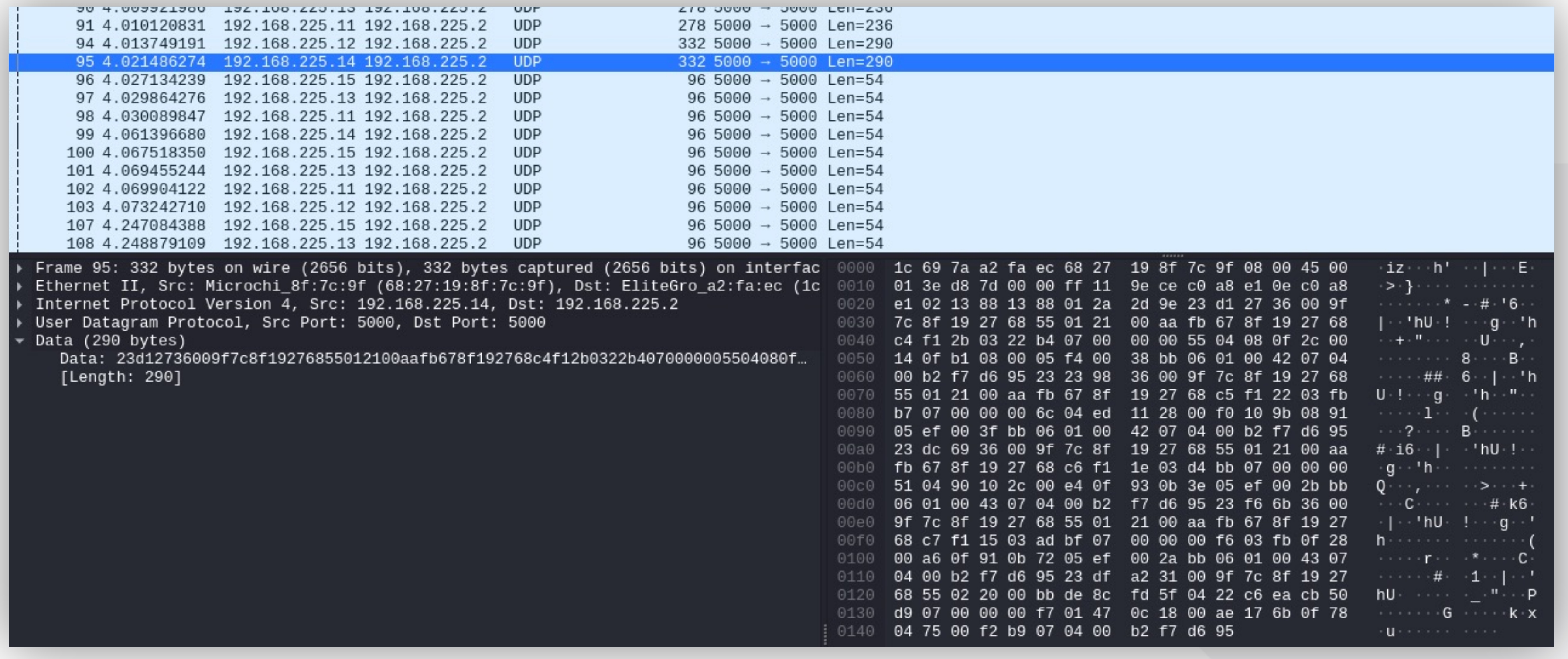
To obtain the position of a tag, the following data need to be known:

- All coordinates of the anchors involved
- Synchronization timestamps
- Positioning timestamps

Reverse Engineering of Devices Network Traffic

Network Traffic

- Both solutions use custom, unknown binary network protocols for the communications among anchors and server. No standard data structures are immediately recognizable



Example of Ethernet network packet in Sewio RTLS

Packet Dissection

- By reverse engineering the server software, full packet structure was reconstructed

```
reportUniversalObj.syncArrival = binary.parse(oneOptionBuffer)
    .word8("option")
    .word16lu("option_len")
    .word8("fcode")
    .word8("device_id6")
    .word8("device_id5")
    .word8("device_id4")
    .word8("device_id3")
    .word8("device_id2")
    .word8("device_id1")
    .word8("seq_num")
    .word8("sync_group_seq_num")
    .word64lu("uwb_timestamp")
    .word16lu("max_noise")
    .word16lu("first_path_amp1")
    .word16lu("std_noise")
    .word16lu("first_path_amp2")
    .word16lu("first_path_amp3")
    .word16lu("max_growth_cir")
```

```
ted AbstractUwbPacket processBody(ByteBuffer byteBuffer) {
turn (new Builder()).setOrder(UwbLibUtils.toUnsignedShort(byteBuffer.getShort()))
      .setSlaveId(byteBuffer.getLong()).setMasterId(byteBuffer.getLong()).setStatus(byteBuffer.get())
      .setTxsTimestamp(byteBuffer.getDouble()).setRxsTimestamp(byteBuffer.getDouble())
      .setSignalData(this.generalDataPacketParser == null
              ? null
              : this.generalDataPacketParser.parseSignal(byteBuffer))
      .setExtData(this.generalDataPacketParser == null
              ? null
              : this.generalDataPacketParser.parseExtData(byteBuffer))
      .build();
```

Example of code snippets from Sewio and Avalue RTLS servers

```
Sewio UWB Protocol
    Separator: 0x23
    Data CRC: 0x27d1
    Report Length: 0x0036
    Anchor Mac: 68:27:19:8f:
    Report Type: U
    Options
        SyncArrival
            Option Length: 0x00
            Function Code: 0xaa
            Device ID: 68:27:19
            Sequence Number: 0x
            Sync Group Sequence
            UWB Timestamp: 3308
            Maximum Noise: 0x04
            First Path Amp 1: 0
            Standard Noise: 0x0
            First Path Amp 2: 0
            First Path Amp 3: 0
            Maximum Growth CIR:
            Rx Pream Count: 0x0
            First Path Index: 0
        DWT Temp
            Option Length: 0x00
            UWB Temp: 0x42
        Barometer
            Option Length: 0x00
            Barometer Data: -1781073998
    Anchor Mac: 68:27:19:8f:7c:9f
```

```
Avalue UWB Protocol
    Separator: 0x5758
    Packet Type: TDoA (17)
    Body Length: 0x28
    Body
        Order: 245
        Tag ID: 0x0000001300000020
        Anchor ID: 0x000000090000a05e
        Sync Timestamp: 1.92853446141952
        Event Data: 0
        Battery Data: 68
        First Path Amp 1: 3224
        First Path Amp 2: 8721
        First Path Amp 3: 15906
        Maximum Growth CIR: 10558
        Rx Pream Count: 237
        Extra Data Type: 0
        Extra Data Length: 0
    Checksum: 0x0ab2
```

Sewio and Avalue RTLS dissectors

*We are freely releasing PCAPs and dissectors for both Sewio and Avalue RTLS! Download them from the **briefing page**, or from **github.com/NozomiNetworks***

Security Considerations

Confidentiality

- No confidentiality in the anchors-server communications
  - The synchronization and positioning timestamps are sent in cleartext

Integrity

- No secure integrity mechanisms either
  - Sewio RTLS uses CRC-16
  - Avalue RTLS performs a byte per byte sum of all packets

```
function convertRawTimestampToString(byte1, byte2, byte3, byte4, byte5) {
    const ticksString = (("0" + byte1.toString(16)).slice(-2)) +
        (("0" + byte2.toString(16)).slice(-2)) +
        (("0" + byte3.toString(16)).slice(-2)) +
        (("0" + byte4.toString(16)).slice(-2)) +
        (("0" + byte5.toString(16)).slice(-2));
    return parseInt(ticksString, 16) * (1 / (128 * 499.2E6));
}
```

Sewio RTLS – Extraction of timestamps directly from the network traffic

```
public boolean check(byte[] byteArray) {
    boolean result = false;
    if (ArrayUtils.isNotEmpty(byteArray) && byteArray.length >= this.checkSumLength) {
        int checkSumAnswer = UwbLibUtils.toUnsignedShort(UwbLibUtils
                .getShort(Arrays.copyOfRange(byteArray, byteArray.length - this.checkSumLength, byteArray.length)));
        int sum = 0;

        for (int idx = 0; idx < byteArray.length - this.checkSumLength; ++idx) {
            sum += UwbLibUtils.toUnsignedByte(byteArray[idx]);
        }

        result = checkSumAnswer == sum;
    }

    return result;
}
```

Avalue RTLS – Checksum computation

Anchors Coordinates Prerequisite

Anchor Coordinates Prerequisite

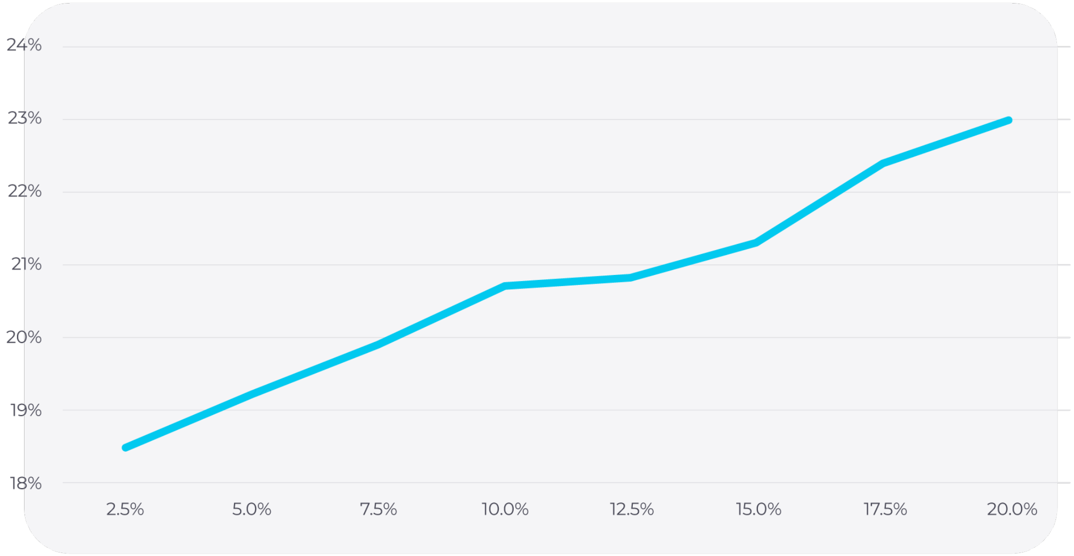
Obtaining the anchor coordinates is the most challenging requirement. They are manually input at the first installation and never transmitted through the network

Attacker with Physical Access

- If the anchors are visible, obtaining their coordinates is simple
- If not, an attacker can still produce an estimation by measuring the power levels of the anchors’ wireless signals (UWB, Wi-Fi, etc.)
- In fact, tag coordinates can be estimated even with imperfect anchor coordinates

If anchor coordinates are estimated with a <10% error, the tag coordinates are computed with an average error of <20%, i.e., **~50 cm** in a 6m x 5m room

Tag Coordinates Average Error with respect to Anchor Coordinates Error



Anchor Coordinates Prerequisite – Cont’d

Obtaining the anchor coordinates is the most challenging requirement. They are manually input at the first installation and never transmitted through the network

Attacker with Remote Access

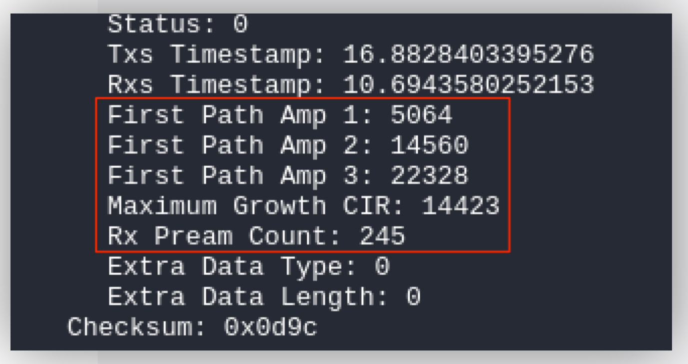
- Besides timestamps, anchors transmit on the wire the power level information of the received UWB signal. We can compute two different metrics:
  - First Path Power Level (FPPL)

$$FPPL = 10 * \log_{10}\left(\frac{FP1^2+FP2^2+FP3\hat{}2}{PAC^2}\right) - A \quad dBm$$

  - Receive Power Level (RPL)

$$RPL = 10 * \log_{10}\left(\frac{MGC * 2^{17}}{PAC^2}\right) - A \quad dBm$$

We **devised** and **present** a technique that remote attackers can apply to circumvent this obstacle



Power level information in Avalue RTLS packets

Anchor Coordinates Prerequisite – Cont'd

Obtaining the anchor coordinates is the most challenging requirement. They are manually input at the first installation and never transmitted through the network

Attacker with Remote Access – Cont'd

- It is not possible to directly estimate the absolute distance, due to evolving temporary conditions
- However, if **in a given moment** t0 the power level information is identical, the tag j0 that triggered those packets must be positioned about exactly at the same distance from all anchors

$$Delta(i0, j0, t0) = \left(GT(reference, j0, t0) - GT(i0, j0, t0)\right) * c = 0$$

- Considering that CS(reference, t0) = 1 and ToF(reference) = 0, we can exploit this information and estimate the time of flights, thus the distances of the other anchors from the reference

$$ToF(i0) = CS(i0, t0) * (pTs(i0, j0, t0) - sTS(i0, t0)) - pTs(reference, j0, t0) + sTS(reference, t0)$$

Anchor Coordinates Prerequisite – Cont’d

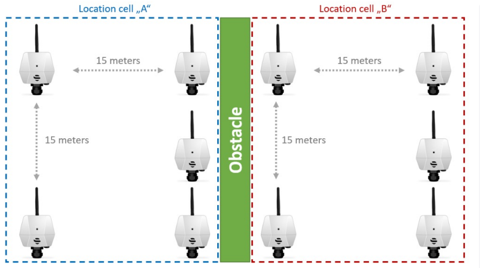
Obtaining the anchor coordinates is the most challenging requirement. They are manually input at the first installation and never transmitted through the network

Attacker with Remote Access – Cont’d

- Finally, to obtain the coordinates, we can leverage the following installation constraint:

Keep a square geometry during the deployment.

Due to dilution of precision phenomena, the best approach is the "squaring" the location cell. The ratio between the two sides should not be higher than 3:1 to achieve highest possible accuracy.



- Given that the anchor map is most times a rectangle, by arbitrarily setting the reference anchor in position (0;0), the coordinates of all other anchors can be easily estimated (they are given by the two shortest distances)
- An attacker can adapt the expected shape on the basis of the number of anchors detected in the communications

Anchor Coordinates Prerequisite – Cont’d

Obtaining the anchor coordinates is the most challenging requirement. They are manually input at the first installation and never transmitted through the network

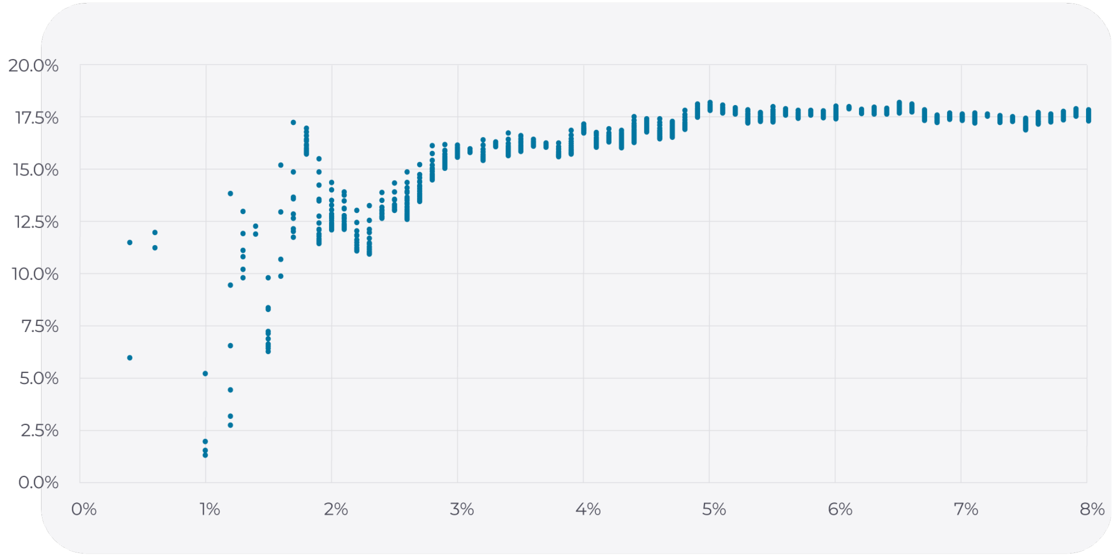
Attacker with Remote Access – Cont’d

- This was actually tested in the Avalue RTLS, using both the First Path Power Level (FPPL) as well as the Receive Power Level (RPL)
- The best results are obtained using FPPL with a threshold of ~1% between the lowest power level and the highest

It was possible to estimate the anchors coordinates with an error of **less than 10%** with respect to the real values

- This can be accurate enough for attack scenarios where cm-level precision is not required

Anchors Coordinates Average Error wrt First Path Power Level (FPPL) Acceptance Threshold



Adversary Tactics, Techniques, and Procedures (TTPs)

Traffic Interception

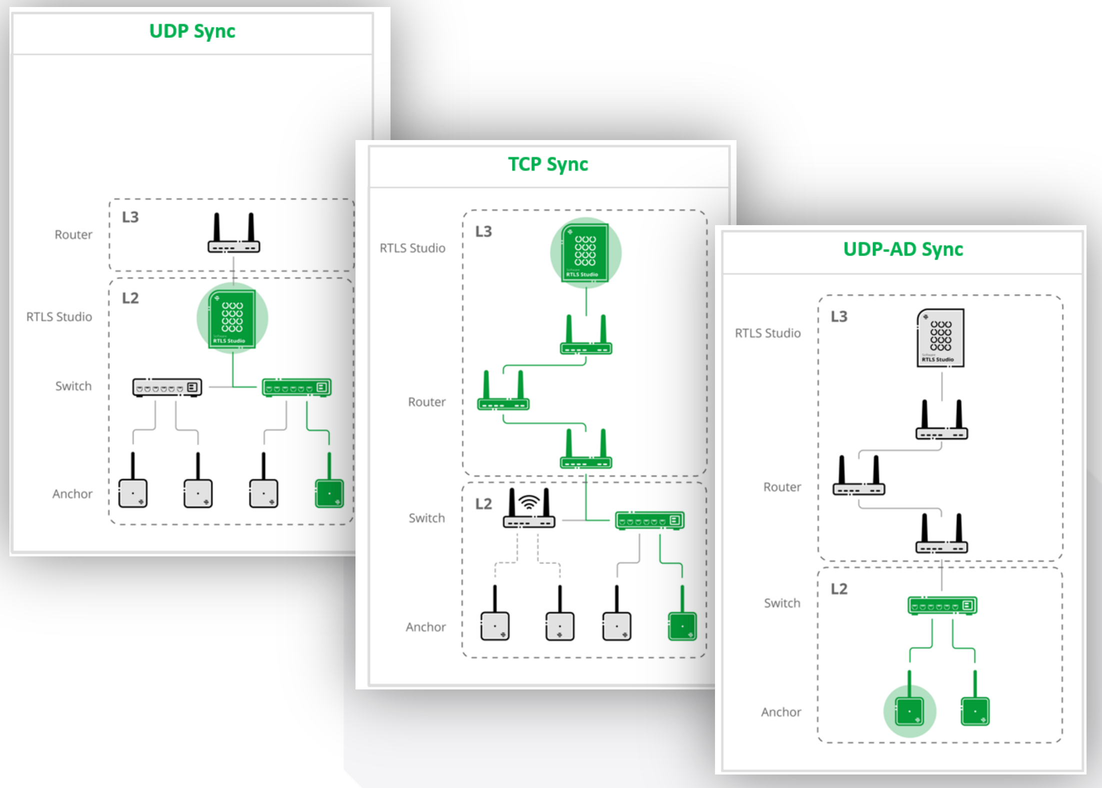
To perform any meaningful attacks against RTLSs, an attacker first needs to intercept all network packets

Intercepting traffic requires two steps:

1. gaining a foothold inside the anchors-server backhaul network
2. executing a Man in the Middle (MitM) attack

Network Access

- Both Sewio and Avalue RTLS allow either Ethernet or Wi-Fi to be used for the network backhaul
- Gaining access to an Ethernet network requires that an attacker:
  - either compromises a computer in that network
  - or surreptitiously adds a rogue device
- The complexity of these attacks varies on the basis of the RTLS deployment configuration



Deployment configurations available on Sewio RTLS

Traffic Interception – Cont’d

To perform any meaningful attacks against RTLSs, an attacker first needs to intercept all network packets

Network Access – Cont’d

- As for Wi-Fi, both solutions support WPA2-PSK
- Gaining access to a Wi-Fi network requires:
  - either the knowledge of the WPA2 password
  - or the exploitation (if any) of vulnerabilities in the wireless appliances
- As for the first point, out of the box, both solutions use a static password that can be found in the public documentation
- In case an asset owner does not change it, obtaining access to the backhaul network is simple

Avalue have completed all the pre-settings before shipping, please refer below information for login.

http://tplinkwifi.net

Password: 7[illegible]

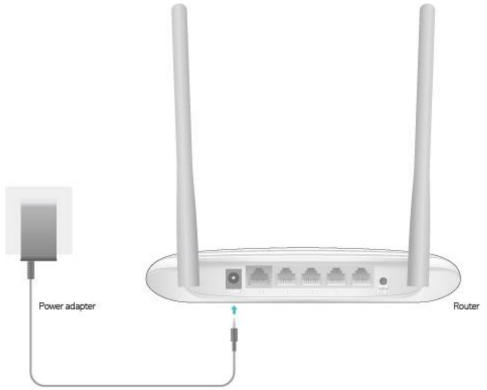
Wi-Fi credentials for admin access are:

SSID: Artemis

Password: 7[illegible]

Avalue also provide a configuration file that can be simply re-loaded into the router.

To load the file into the router, follow these steps:



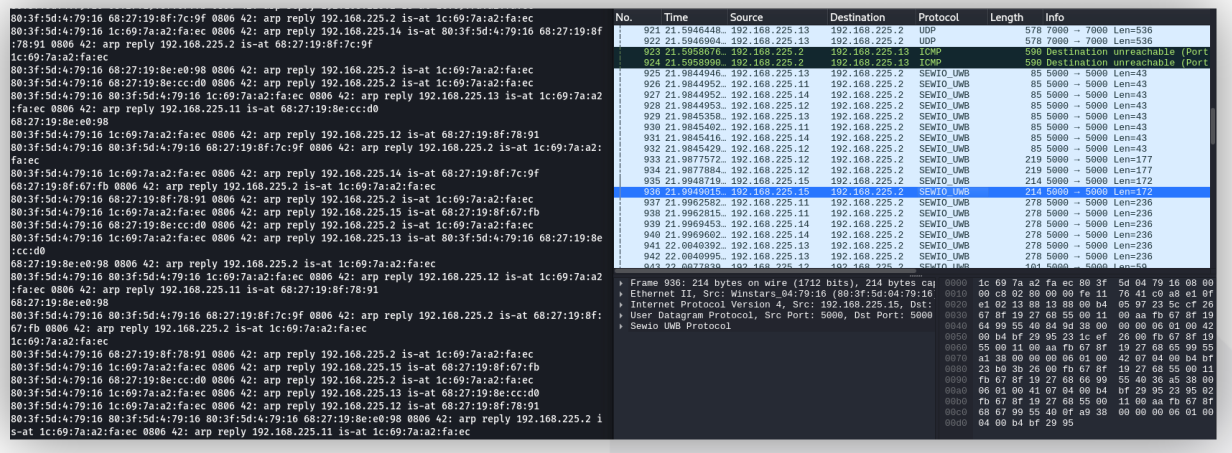
Default WPA2-PSK password on Avalue RTLS

Traffic Interception – Cont’d

To perform any meaningful attacks against RTLSs, an attacker first needs to intercept all network packets

Man in the Middle

- In the tests executed, it was possible to MitM both solutions via standard ARP spoofing attacks



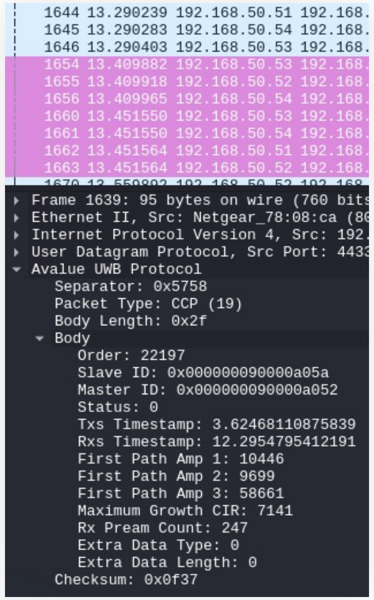
MitM attack against Sewio RTLS

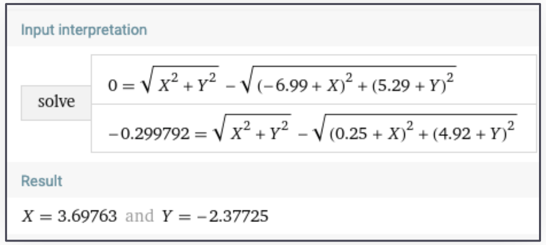
$$arpspoof\ -i\ attacker\_eth\ -t\ server\_ip\ anchor1\_ip\ \&\ arpspoof\ -i\ attacker\_eth\ -t\ anchor1\_ip\ server\_ip$$

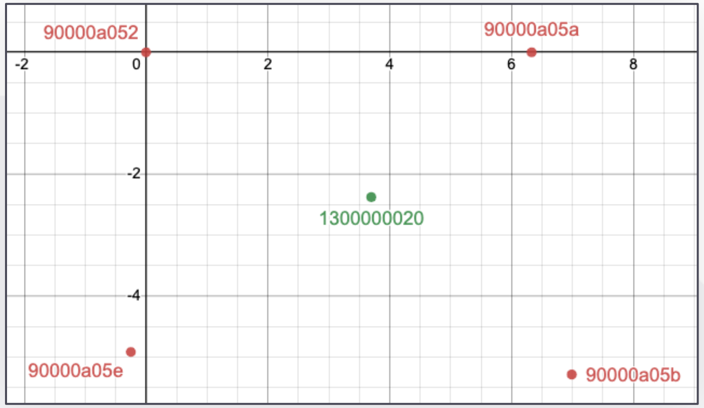
The attacks were completely **undetected** by the RTLS. No warnings or abnormal behavior that may alert an operator were shown.

Passive Eavesdropping Attacks

After obtaining access to an RTLS network and launching the MitM attack, an attacker can reconstruct the position of tags by executing one of the TDoA algorithms known in literature







01 Position of target shown in RTLS

02 Traffic is intercepted, anchor coordinates are estimated, timestamps are extracted

03 A TDoA algorithm is applied

04 Attacker obtains the position of target

Active Traffic Manipulation Attacks

To accomplish an active attack, an attacker first needs to do a target reconnaissance and add traffic filtering routines to the attack algorithm

Target Reconnaissance

- To successfully deceive an operator, it is important that the tag movements appear natural

If the target is a human being, faking its position with harsh, sudden movements would **warn an operator** and make them think that, at the very least, a malfunctioning is occurring

- This phase can be accomplished by simply performing a passive eavesdropping attack against the target

Active Traffic Filtering

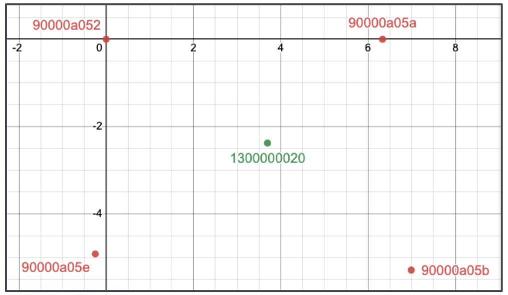
- If the packet is a synchronization packet, it must be automatically forwarded to the destination
- If the packet is a positioning packet of a target, its timestamp must be modified (and the checksum updated). If not a target one, it must be forwarded unaltered
- Many techniques are available. Notably, we leveraged iptables NFQUEUE, a flexible userspace packet handler

Active Traffic Manipulation Attacks – Cont'd

Finally, an attacker can alter the timestamps by simply inverting all the equations previously described

Packet Information Manipulation

- In a manipulation attack, the tag coordinates are known (they are the target coordinates that an attacker wants to fake for a given tag) and the positioning timestamps are unknown
- First, the attacker derives the modified positioning timestamps according to the target coordinates
- Finally, the attacker re-computes the packet checksums and then sends the modified packets



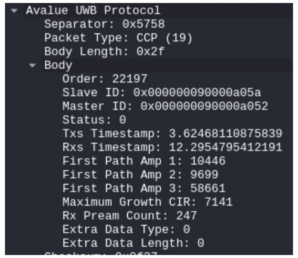
01 The attacker defines a target coordinate for a given tag

$$\sqrt{(Xj0,t0 - Xreference)^2 + (Yj0,t0 - Yreference)^2 + (Zj0,t0 - Zreference)^2} - \sqrt{(Xj0,t0 - X1)^2 + (Yj0,t0 - Y1)^2 + (Zj0,t0 - Z1)^2} = Delta(1,j0,t0)$$

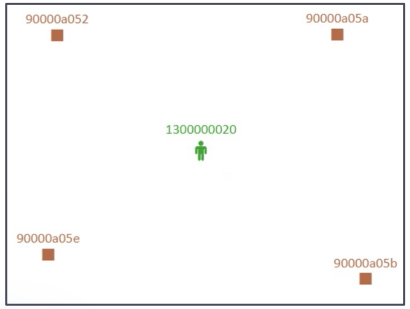
$$\ldots$$

$$\sqrt{(Xj0,t0 - Xreference)^2 + (Yj0,t0 - Yreference)^2 + (Zj0,t0 - Zreference)^2} - \sqrt{(Xj0,t0 - XN)^2 + (Yj0,t0 - YN)^2 + (Zj0,t0 - ZN)^2} = Delta(N,j0,t0)$$

02 The TDoA algorithm is applied backwards



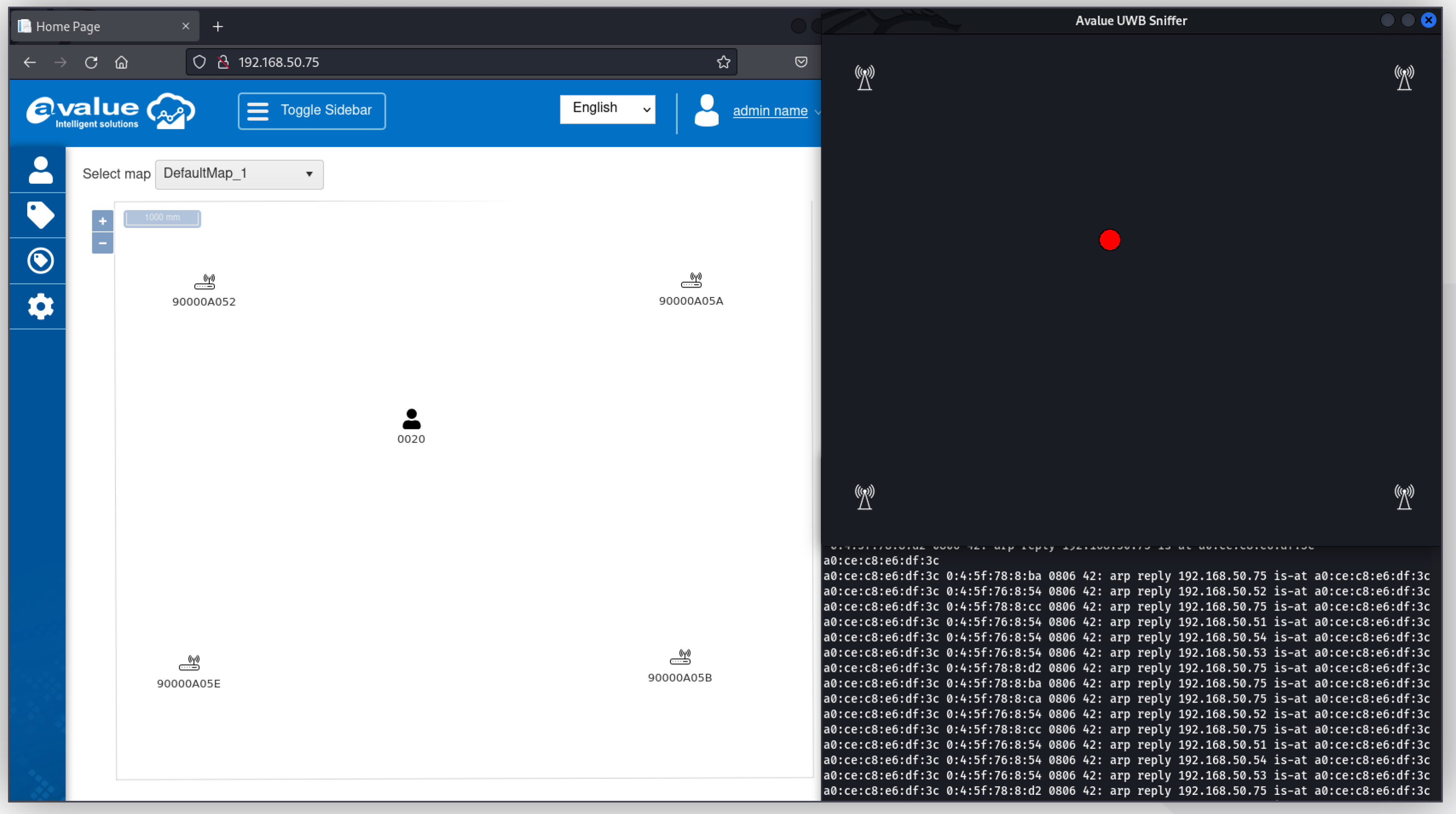
03 Packet is updated



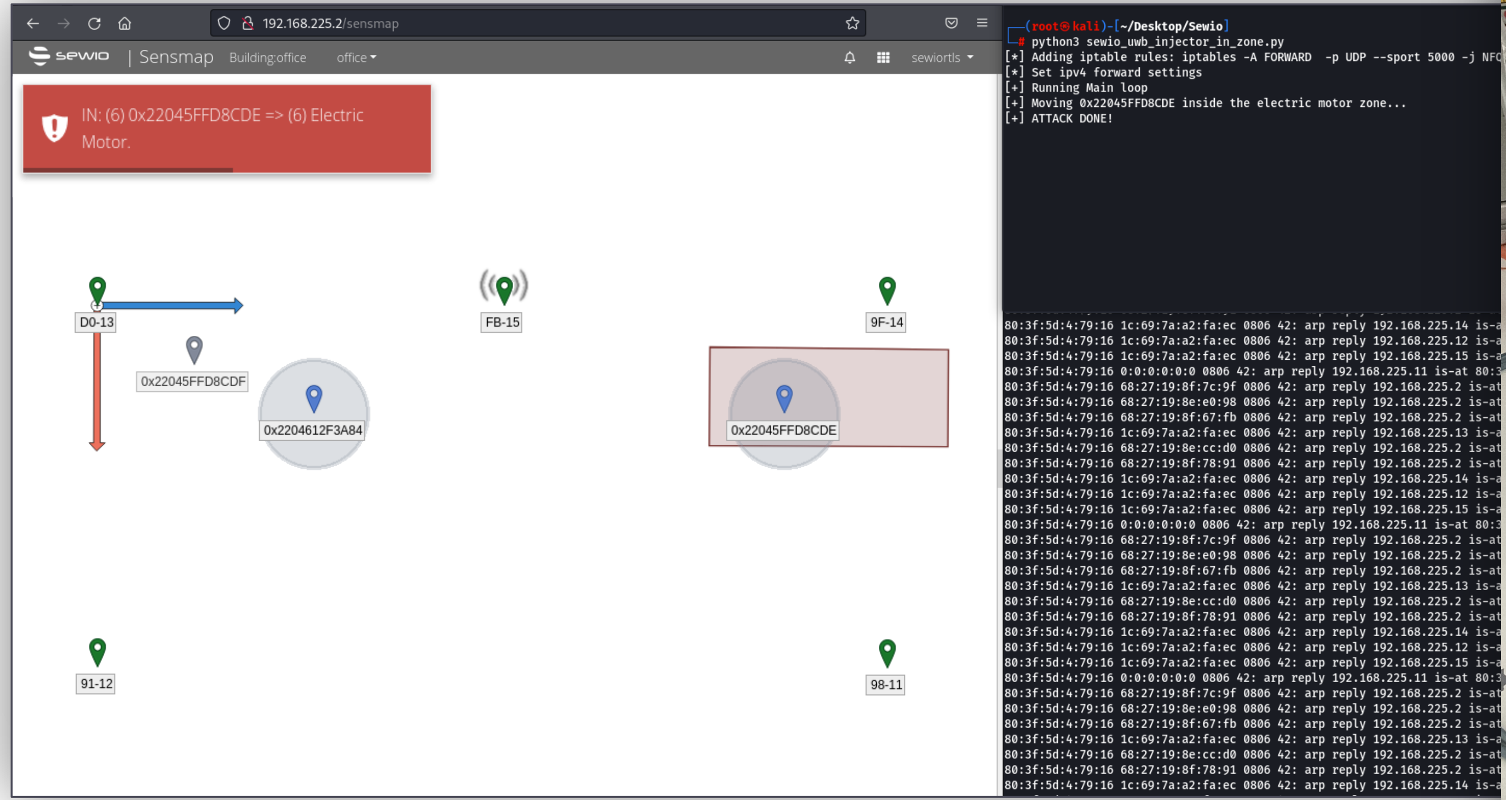
04 Position of target is altered in RTLS

Attack Demos

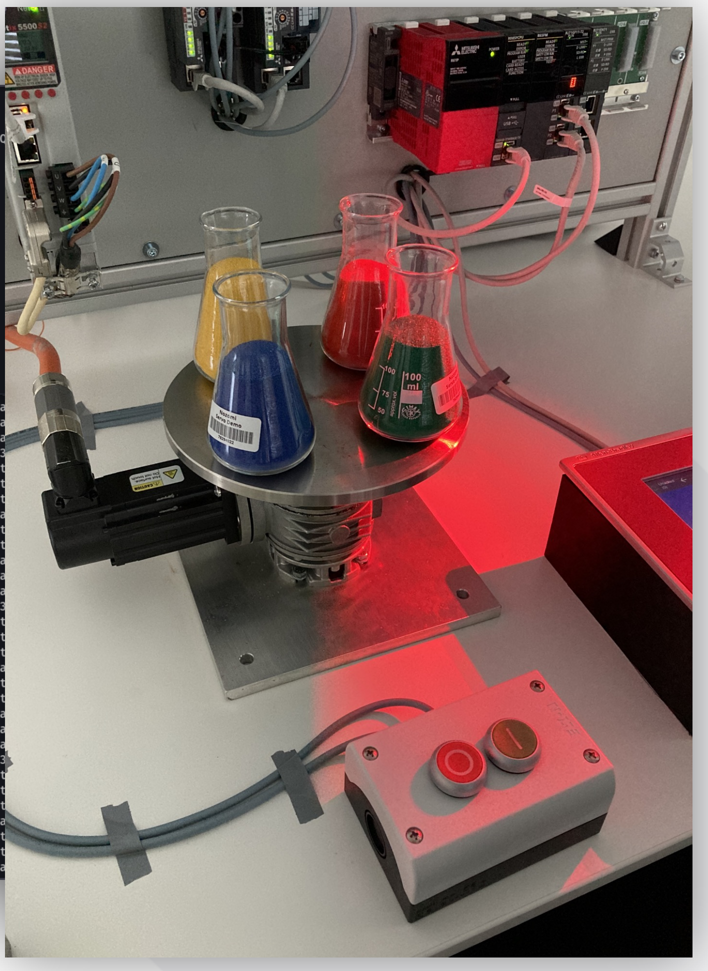
Locating and Targeting People/Assets



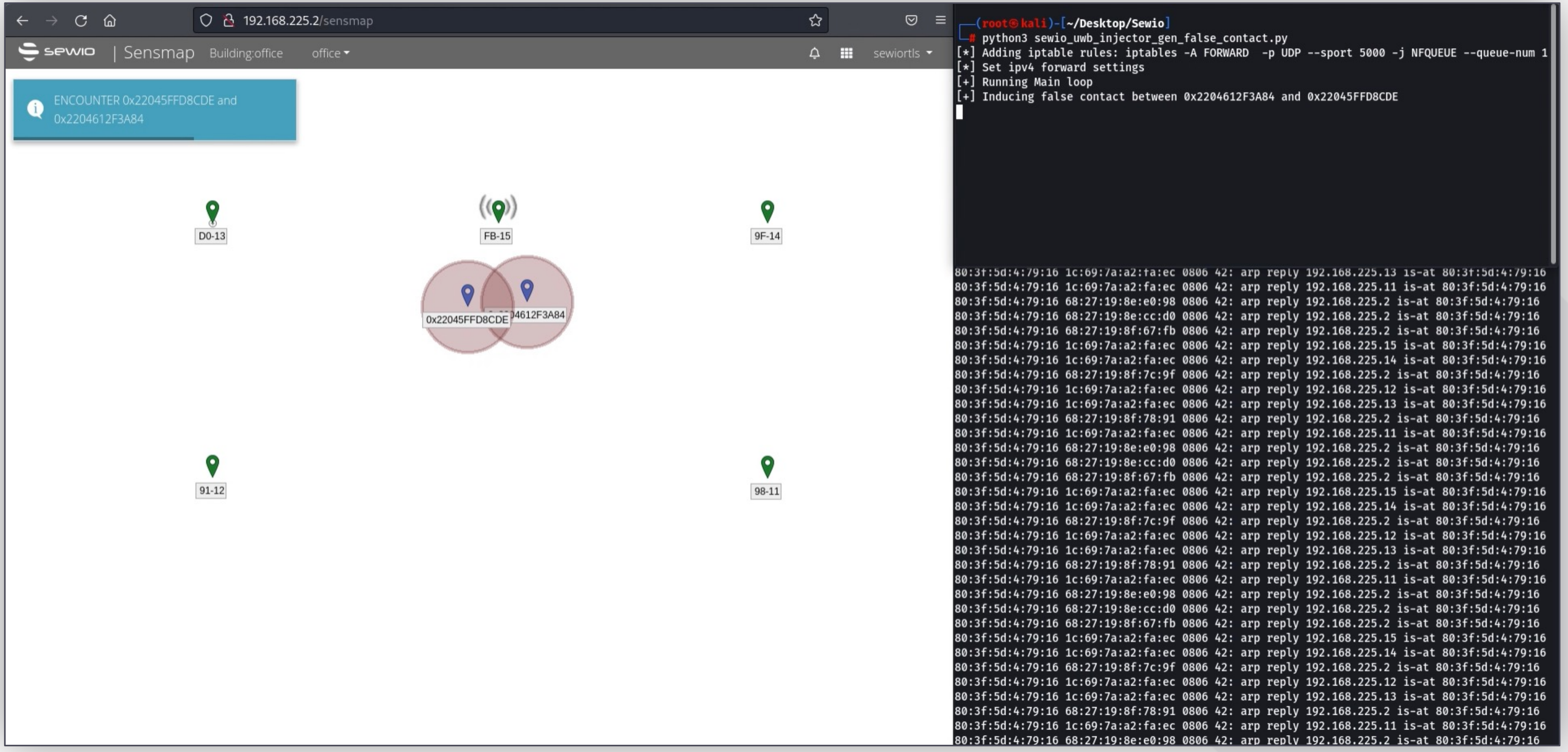
Geofencing



192.168.225.2/sensmap
SEWIO
Sensmap
Building:office
office
sewiortls
IN: (6) 0x22045FFD8CDE => (6) Electric Motor.
D0-13
FB-15
9F-14
0x22045FFD8CDF
0x2204612F3A84
0x22045FFD8CDE
91-12
98-11
(root㉿kali)-[~/Desktop/Sewio]
python3 sewio_uwb_injector_in_zone.py
[*] Adding iptable rules: iptables -A FORWARD -p UDP --sport 5000 -j NFQ
[*] Set ipv4 forward settings
[+] Running Main loop
[+] Moving 0x22045FFD8CDE inside the electric motor zone...
[+] ATTACK DONE!
80:3f:5d:4:79:16 1c:69:7a:a2:fa:ec 0806 42: arp reply 192.168.225.14 is-a
80:3f:5d:4:79:16 1c:69:7a:a2:fa:ec 0806 42: arp reply 192.168.225.12 is-a
80:3f:5d:4:79:16 1c:69:7a:a2:fa:ec 0806 42: arp reply 192.168.225.15 is-a
80:3f:5d:4:79:16 0:0:0:0:0:0 0806 42: arp reply 192.168.225.11 is-at 80:3
80:3f:5d:4:79:16 68:27:19:8f:7c:9f 0806 42: arp reply 192.168.225.2 is-at
80:3f:5d:4:79:16 68:27:19:8e:e0:98 0806 42: arp reply 192.168.225.2 is-at
80:3f:5d:4:79:16 68:27:19:8f:67:fb 0806 42: arp reply 192.168.225.2 is-at
80:3f:5d:4:79:16 1c:69:7a:a2:fa:ec 0806 42: arp reply 192.168.225.13 is-a
80:3f:5d:4:79:16 68:27:19:8e:cc:d0 0806 42: arp reply 192.168.225.2 is-at
80:3f:5d:4:79:16 68:27:19:8f:78:91 0806 42: arp reply 192.168.225.2 is-at
80:3f:5d:4:79:16 1c:69:7a:a2:fa:ec 0806 42: arp reply 192.168.225.14 is-a
80:3f:5d:4:79:16 1c:69:7a:a2:fa:ec 0806 42: arp reply 192.168.225.12 is-a
80:3f:5d:4:79:16 1c:69:7a:a2:fa:ec 0806 42: arp reply 192.168.225.15 is-a
80:3f:5d:4:79:16 0:0:0:0:0:0 0806 42: arp reply 192.168.225.11 is-at 80:3
80:3f:5d:4:79:16 68:27:19:8f:7c:9f 0806 42: arp reply 192.168.225.2 is-at
80:3f:5d:4:79:16 68:27:19:8e:e0:98 0806 42: arp reply 192.168.225.2 is-at
80:3f:5d:4:79:16 68:27:19:8f:67:fb 0806 42: arp reply 192.168.225.2 is-at
80:3f:5d:4:79:16 1c:69:7a:a2:fa:ec 0806 42: arp reply 192.168.225.13 is-a
80:3f:5d:4:79:16 68:27:19:8e:cc:d0 0806 42: arp reply 192.168.225.2 is-at
80:3f:5d:4:79:16 68:27:19:8f:78:91 0806 42: arp reply 192.168.225.2 is-at
80:3f:5d:4:79:16 1c:69:7a:a2:fa:ec 0806 42: arp reply 192.168.225.14 is-a
80:3f:5d:4:79:16 1c:69:7a:a2:fa:ec 0806 42: arp reply 192.168.225.12 is-a
80:3f:5d:4:79:16 1c:69:7a:a2:fa:ec 0806 42: arp reply 192.168.225.15 is-a
80:3f:5d:4:79:16 0:0:0:0:0:0 0806 42: arp reply 192.168.225.11 is-at 80:3
80:3f:5d:4:79:16 68:27:19:8f:7c:9f 0806 42: arp reply 192.168.225.2 is-at
80:3f:5d:4:79:16 68:27:19:8e:e0:98 0806 42: arp reply 192.168.225.2 is-at
80:3f:5d:4:79:16 68:27:19:8f:67:fb 0806 42: arp reply 192.168.225.2 is-at
80:3f:5d:4:79:16 1c:69:7a:a2:fa:ec 0806 42: arp reply 192.168.225.13 is-a
80:3f:5d:4:79:16 68:27:19:8e:cc:d0 0806 42: arp reply 192.168.225.2 is-at
80:3f:5d:4:79:16 68:27:19:8f:78:91 0806 42: arp reply 192.168.225.2 is-at
80:3f:5d:4:79:16 1c:69:7a:a2:fa:ec 0806 42: arp reply 192.168.225.14 is-a



Contact Tracing



Remediations

Segregation and Firewall Rules

Goal

- Move the entire UWB RTLS backhaul network to a segregated network, and secure the access to the network both physically and logically

This is now mandated by some RTLS vendors

Physical access

- Restrict physical access to the device to qualified personnel.
- Disable unused physical interfaces of the device. Unused interfaces could be used to gain access to the operating site.

Software - Safety functions

- Only use protocols that are required to operate the device.
- Restrict access to the device with a firewall or rules in an ACL (Access Control List).
- Using VLANs gives you good protection against DoS attacks. Check whether this is practicable.
- Activate the access logging function (external). Use the central logging function to record changes and access.
- Configure a SysLog server to save all logs to a central location.

SIMATIC RTLS4030G
Operating Instructions, 04/2021, C79000-G8976-C515-06

Siemens RTLS4030G operating instructions

Advantages

- Allows the problem to be mitigated relatively quickly
- Can be enacted by deploying traditional solutions such as VLANs, IEEE 802.1X, firewall rules

Challenges

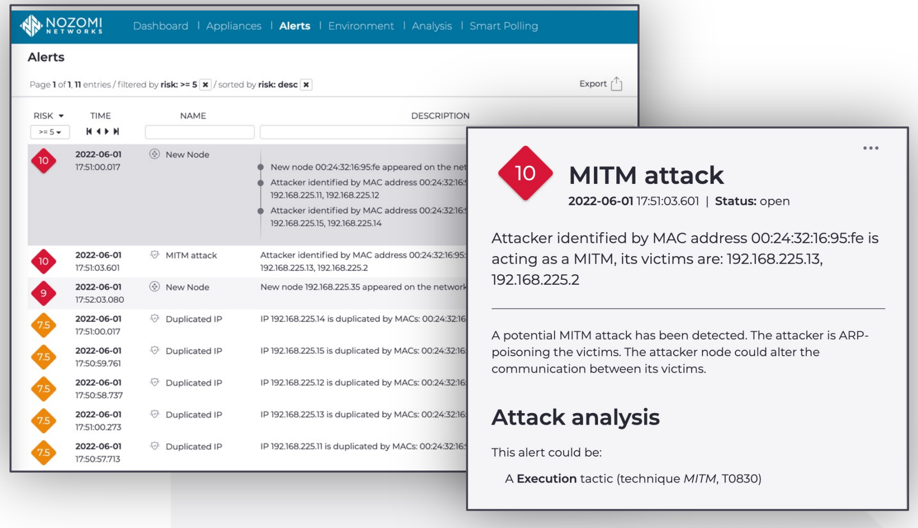
- Some RTLS servers expose core network services on all interfaces. Firewall rules must be set to allow as few services as possible on the management interface
- Does not protect from a physical MitM (either via wire tap, or wireless sniffer if wireless password is compromised)

Intrusion Detection Systems

Goal

- Detect signs of MitM attacks. Leverages the fact that MitM attacks are unavoidable to obtain the timestamps

This option was successfully tested on both Sewio and Avalue RTLS



Detection of a MitM attack by an IDS

Advantages

- Plug-and-play solution
- Allows the problem to be mitigated very quickly

Challenges

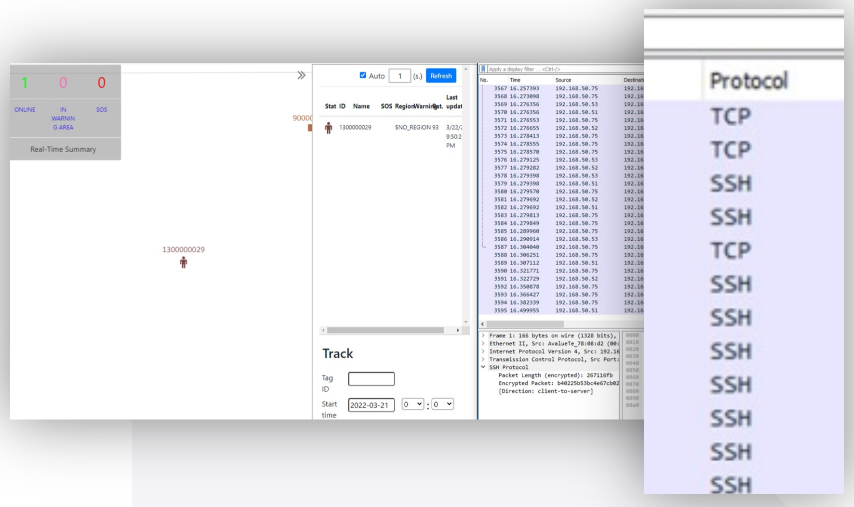
- Does not protect from a physical MitM (either via wire tap, or wireless sniffer if wireless password is compromised)

Traffic Encryption

Goal

- Add a traffic encryption layer on top of the existing communications, to protect even against a physical MitM

This option was successfully tested on the Avalue RTLS for a PoC using standard tools (SSH tunnel and Socat)



SSH tunnel PoC on Avalue RTLS

Advantages

- The closest mitigation to completely solving the problem
- Allows basic RTLS functionalities to remain unaltered

Challenges

- In Avalue RTLS, it was necessary to reduce the number of syncs per second to counteract the higher load, at the expense of a reduced accuracy
- Entirely depends on the accessibility of the RTLS server and anchors from the vendor

Summary & Key Takeaways

Summary

Wireless technology increases efficiency + productivity while reducing unnecessary cabling infrastructure costs

- IEEE 802.15.4z has out of scope areas, creating security loopholes
- Nozomi Networks Labs discovered zero-days in two popular UWB RTLS
- UWB RTLS is used for personnel tracking, geofencing, and contact tracing
- Threat actor TTPs are MitM and eavesdropping or manipulation tactics
- Mitigations include segregation and firewall rules, IDS, and traffic encryption

Black Hat Sound Bytes

Key Takeaways

Weak security requirements in critical software can lead to **safety issues** that cannot be ignored

There are attack surfaces out there that no one is looking at, but **they have significant consequences if compromised**

Exploiting secondary communications in UWB RTLS can be **challenging, but it is doable**

Thank You!

Questions?

labs@nozominetworks.com