



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

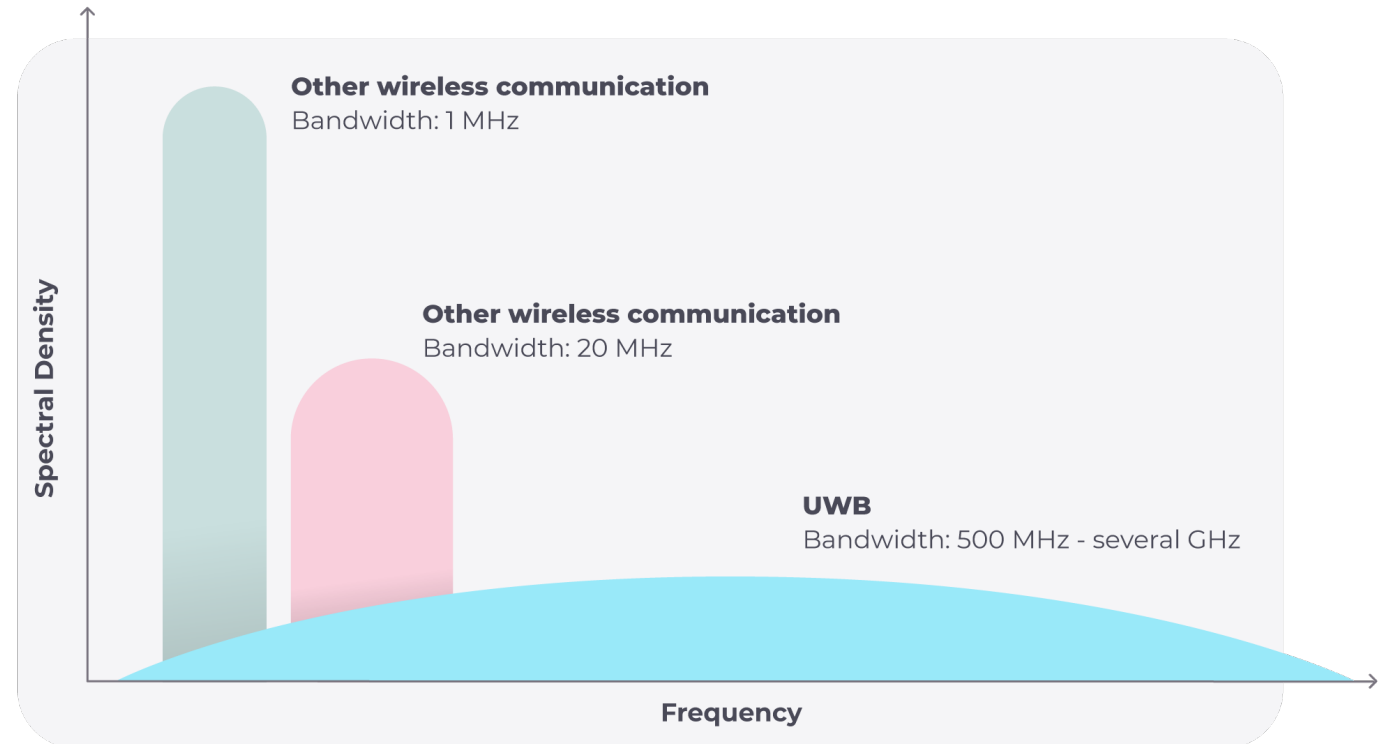


Introduction

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

UWB

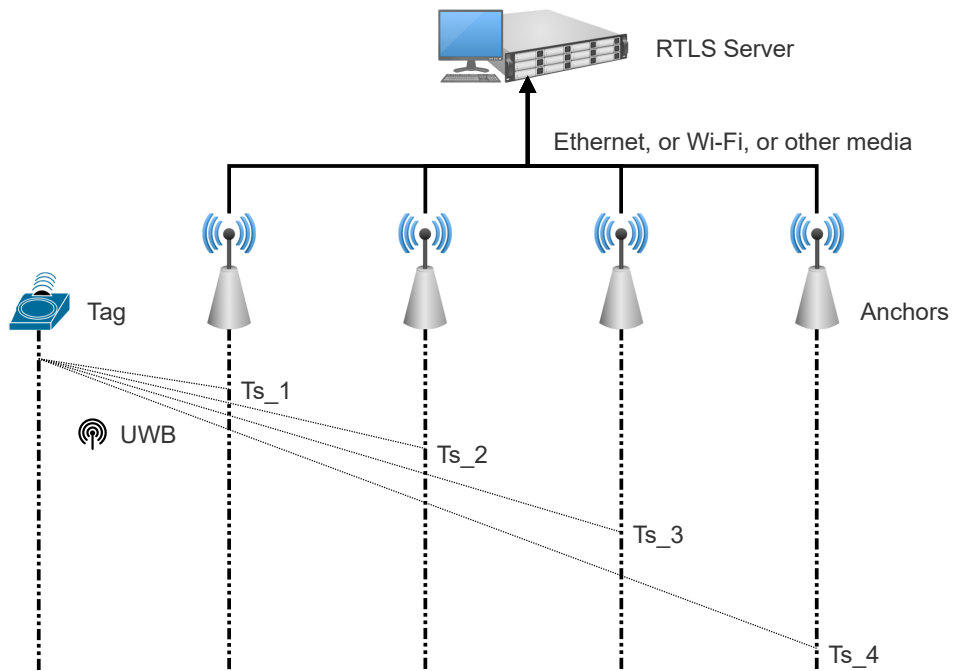
- Ideal for **short-range devices**
- Can transmit information quickly over short distances
- Ability to **send data through solid objects** like walls and other barriers
- UWB is the preferred communication protocol for RTLS



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



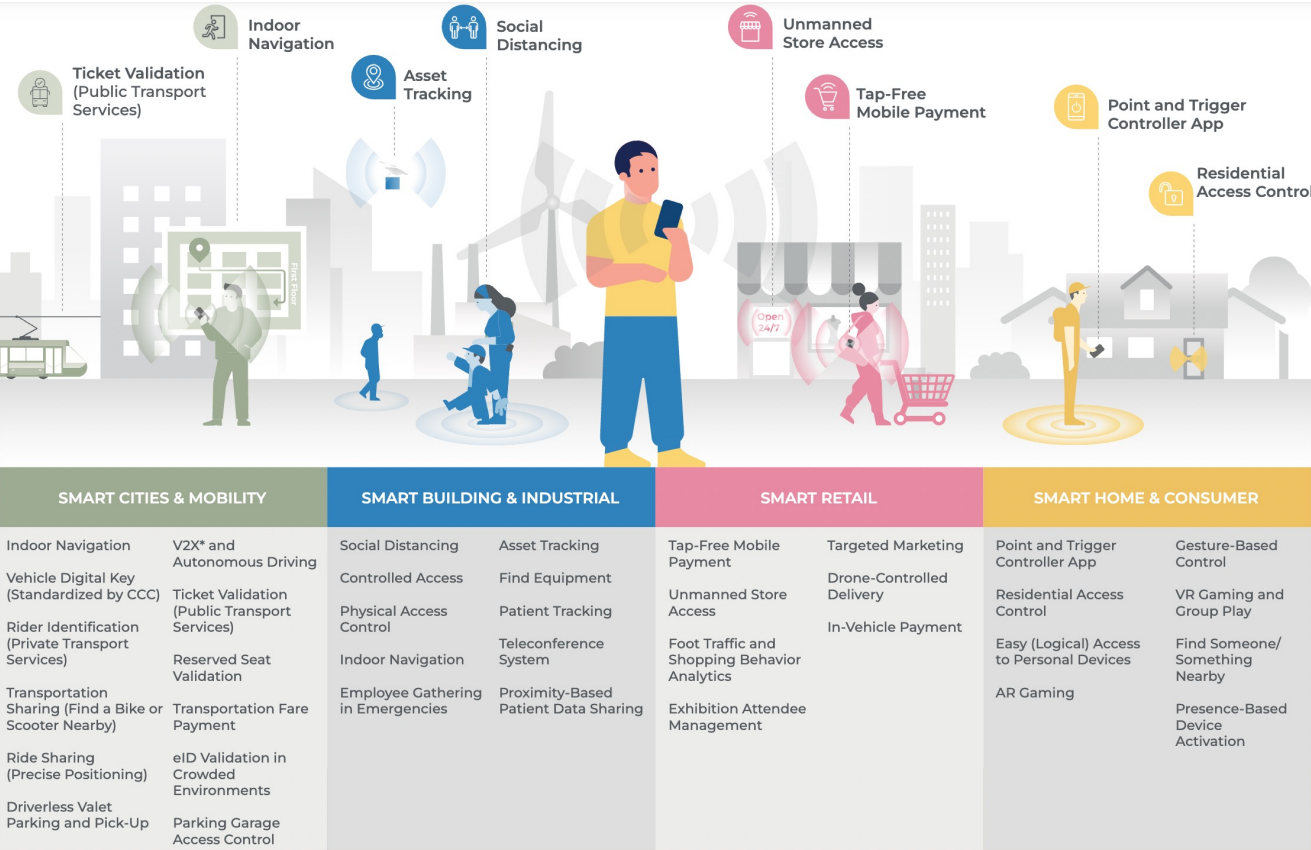
Operation of an UWB TDoA RTLS

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

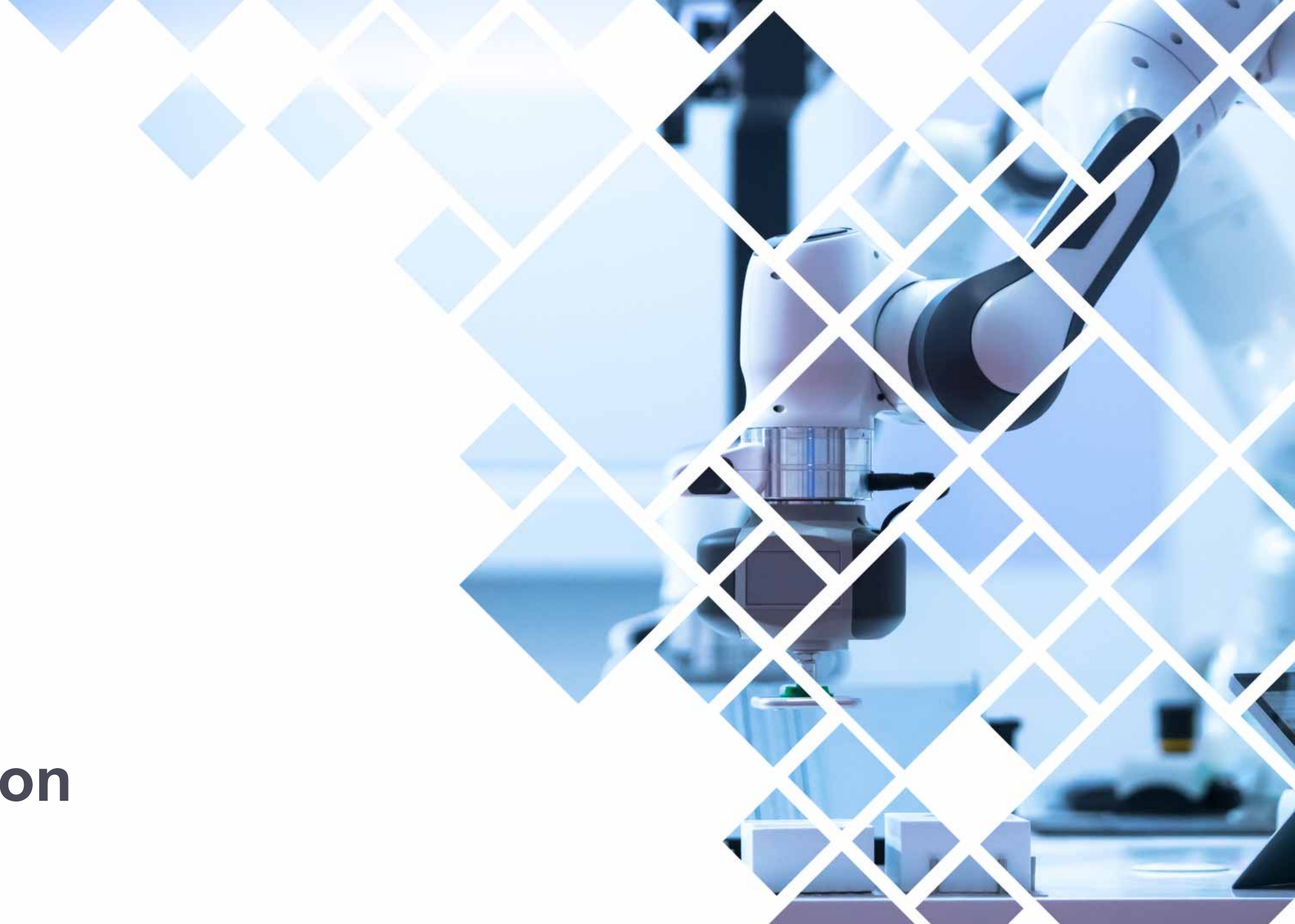


Source: [FiRa Consortium](#)

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**.

Motivation



Motivation

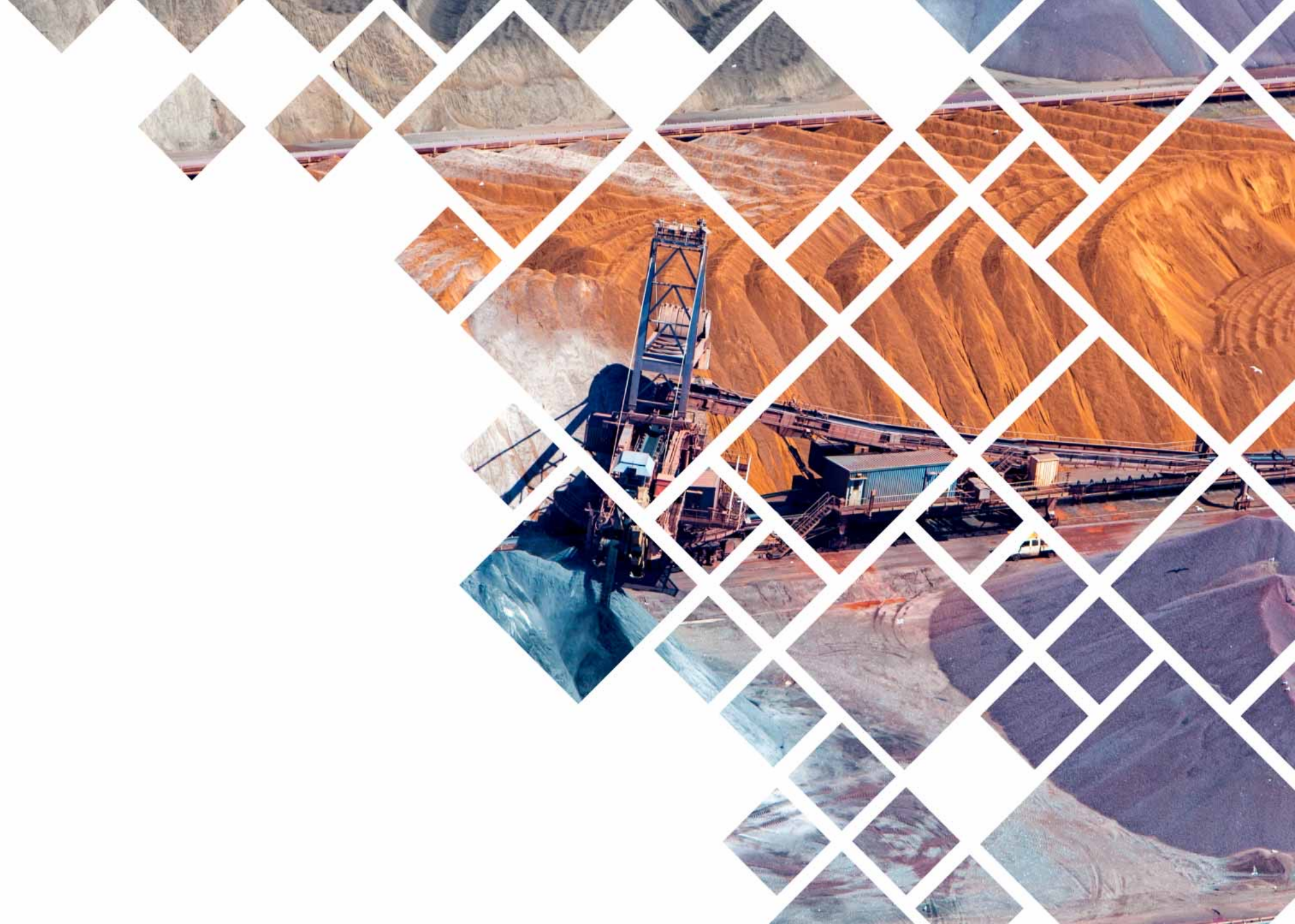
According to the FiRa consortium, in 2018 there was an increased demand for “improvements to existing modulations to increase the integrity and accuracy of ranging measurements”

- 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 the question of how to keep their production running without putting the safety of employees at risk. Simatic RTLS allows for social distancing in production and other worksites. I had a recent talk with Nicole Lauther, Siemens USA, who is coordinating the global roll-out of the solution.



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



WHAT WE DO WHO WE SERVE

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**

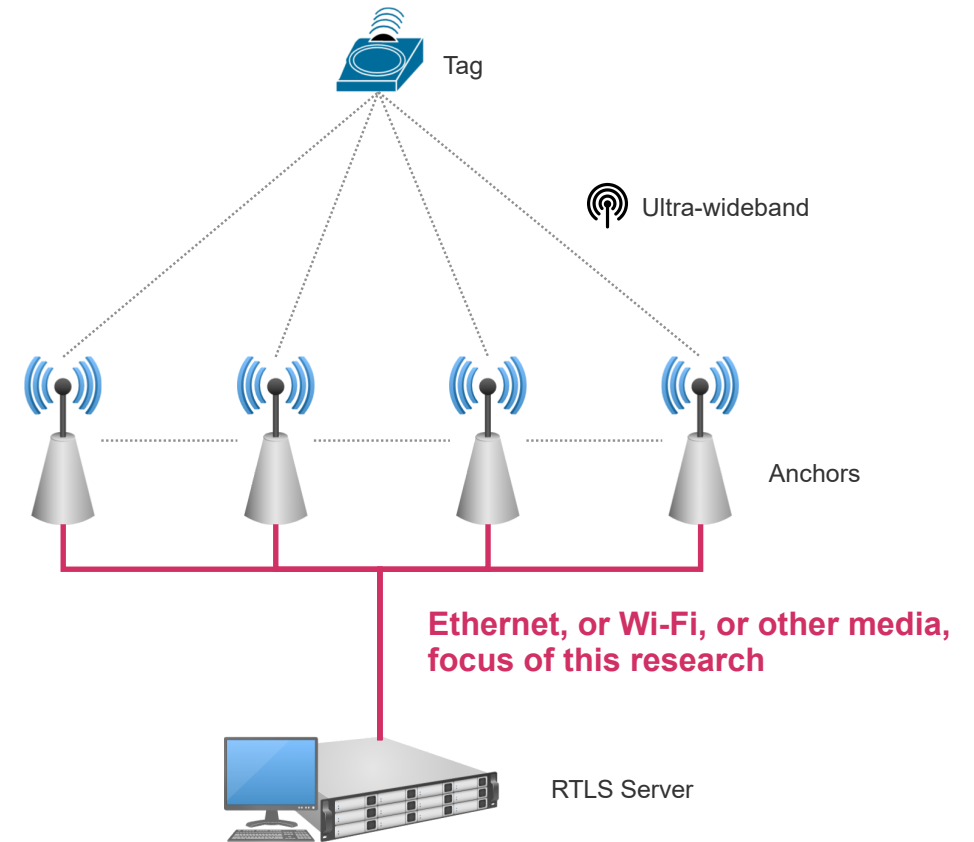
Technical Scope

- 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

$$\begin{aligned} \Delta(i, j, t) &= (GT(\text{reference}, j, t) - GT(i, j, t)) * c = GT(\text{reference}, j, t) * c - GT(i, j, t) * c \\ &= d(\text{reference}, j, t) - d(i, j, t) \end{aligned}$$



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 $X_{j,t}$, $Y_{j,t}$, and $Z_{j,t}$, i.e., the position of tag j at the instant t :

$$\left\{ \begin{array}{l} (pTs(reference, j, t) - sTS(reference, t) - (CS(1, t) * (pTs(1, j, t) - sTS(1, t)) + ToF(1))) * c = \\ \sqrt{(X_{j,t} - X_{reference})^2 + (Y_{j,t} - Y_{reference})^2 + (Z_{j,t} - Z_{reference})^2} - \sqrt{(X_{j,t} - X_1)^2 + (Y_{j,t} - Y_1)^2 + (Z_{j,t} - Z_1)^2} \\ \dots \\ (pTs(reference, j, t) - sTS(reference, t) - (CS(N, t) * (pTs(N, j, t) - sTS(N, t)) + ToF(N))) * c = \\ \sqrt{(X_{j,t} - X_{reference})^2 + (Y_{j,t} - Y_{reference})^2 + (Z_{j,t} - Z_{reference})^2} - \sqrt{(X_{j,t} - X_N)^2 + (Y_{j,t} - Y_N)^2 + (Z_{j,t} - Z_N)^2} \end{array} \right.$$



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

90	4.009921980	192.168.225.13	192.168.225.2	UDP	278	5000 → 5000	Len=230
91	4.010120831	192.168.225.11	192.168.225.2	UDP	278	5000 → 5000	Len=236
94	4.013749191	192.168.225.12	192.168.225.2	UDP	332	5000 → 5000	Len=290
95	4.021486274	192.168.225.14	192.168.225.2	UDP	332	5000 → 5000	Len=290
96	4.027134239	192.168.225.15	192.168.225.2	UDP	96	5000 → 5000	Len=54
97	4.029864276	192.168.225.13	192.168.225.2	UDP	96	5000 → 5000	Len=54
98	4.030089847	192.168.225.11	192.168.225.2	UDP	96	5000 → 5000	Len=54
99	4.061396680	192.168.225.14	192.168.225.2	UDP	96	5000 → 5000	Len=54
100	4.067518350	192.168.225.15	192.168.225.2	UDP	96	5000 → 5000	Len=54
101	4.069455244	192.168.225.13	192.168.225.2	UDP	96	5000 → 5000	Len=54
102	4.069904122	192.168.225.11	192.168.225.2	UDP	96	5000 → 5000	Len=54
103	4.073242710	192.168.225.12	192.168.225.2	UDP	96	5000 → 5000	Len=54
107	4.247084388	192.168.225.15	192.168.225.2	UDP	96	5000 → 5000	Len=54
108	4.248879109	192.168.225.13	192.168.225.2	UDP	96	5000 → 5000	Len=54

▶ Frame 95: 332 bytes on wire (2656 bits), 332 bytes captured (2656 bits) on interface	0000	1c 69 7a a2 fa ec 68 27	19 8f 7c 9f 08 00 45 00	.iz...h'E
▶ Ethernet II, Src: Microchi_8f:7c:9f (68:27:19:8f:7c:9f), Dst: EliteGro_a2:fa:ec (1c	0010	01 3e d8 7d 00 00 ff 11	9e ce c0 a8 e1 0e c0 a8	>...}... ..
▶ Internet Protocol Version 4, Src: 192.168.225.14, Dst: 192.168.225.2	0020	e1 02 13 88 13 88 01 2a	2d 9e 23 d1 27 36 00 9f	...*...-#...'6...
▶ User Datagram Protocol, Src Port: 5000, Dst Port: 5000	0030	7c 8f 19 27 68 55 01 21	00 aa fb 67 8f 19 27 68	...'hU! ...g...'h
▼ Data (290 bytes)	0040	c4 f1 2b 03 22 b4 07 00	00 00 55 04 08 0f 2c 00	+...'...U... ..
Data: 23d12736009f7c8f19276855012100aafb678f192768c4f12b0322b4070000005504080f...	0050	14 0f b1 08 00 05 f4 00	38 bb 06 01 00 42 07 04	...8...B...
[Length: 290]	0060	00 b2 f7 d6 95 23 23 98	36 00 9f 7c 8f 19 27 68	...##...6... ...'h
	0070	55 01 21 00 aa fb 67 8f	19 27 68 c5 f1 22 03 fb	U!...g...'h..."
	0080	b7 07 00 00 00 6c 04 ed	11 28 00 f0 10 9b 08 91	...1... (...
	0090	05 ef 00 3f bb 06 01 00	42 07 04 00 b2 f7 d6 95	...?.... B ...
	00a0	23 dc 69 36 00 9f 7c 8f	19 27 68 55 01 21 00 aa	#.i6... ...'hU!...
	00b0	fb 67 8f 19 27 68 c6 f1	1e 03 d4 bb 07 00 00 00	-g...'h'... ..
	00c0	51 04 90 10 2c 00 e4 0f	93 0b 3e 05 ef 00 2b bb	Q... ..>...+
	00d0	06 01 00 43 07 04 00 b2	f7 d6 95 23 f6 6b 36 00	...C... ..#..k6..
	00e0	9f 7c 8f 19 27 68 55 01	21 00 aa fb 67 8f 19 27'hU! ...g...'
	00f0	68 c7 f1 15 03 ad bf 07	00 00 00 f6 03 fb 0f 28	h'... .. (...
	0100	00 a6 0f 91 0b 72 05 ef	00 2a bb 06 01 00 43 07	...r...*...C...
	0110	04 00 b2 f7 d6 95 23 df	a2 31 00 9f 7c 8f 19 27	...#...1... ...'
	0120	68 55 02 20 00 bb de 8c	fd 5f 04 22 c6 ea cb 50	hU... .._..."..P
	0130	d9 07 00 00 00 f7 01 47	0c 18 00 ae 17 6b 0f 78	...G... ..k..x
	0140	04 75 00 f2 b9 07 04 00	b2 f7 d6 95	..u... ..

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")

def AbstractUwbPacket processBody(ByteBuffer byteBuffer) {
    return (new Builder()).setOrder(UwbLibUtils.toUnsignedShort(byteBuffer.getShort()))
        .setSlaveId(byteBuffer.getLong()).setMasterId(byteBuffer.getLong()).setStatus(byteBuffer.get())
        .setTxTimestamp(byteBuffer.getDouble()).setRxTimestamp(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:7c:9f
  Report Type: U
  ▼ Options
    ▼ SyncArrival
      Option Length: 0x00
      Function Code: 0xaa
      Device ID: 68:27:19:8f:7c:9f
      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: 0x00000001300000020
      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

```
38 function convertRawTimestampToString(byte1, byte2, byte3, byte4, byte5) {
39     const ticksString = (("0" + byte1.toString(16)).slice(-2)) +
40         (("0" + byte2.toString(16)).slice(-2)) +
41         (("0" + byte3.toString(16)).slice(-2)) +
42         (("0" + byte4.toString(16)).slice(-2)) +
43         (("0" + byte5.toString(16)).slice(-2));
44     return parseInt(ticksString, 16) * (1 / (128 * 499.2E6));
45 }
```

Sewio RTLS – Extraction of timestamps directly from the network traffic

```
public boolean check(byte[] byteArray) {
    boolean result = false;
    if (ArrayUtils.isEmpty(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



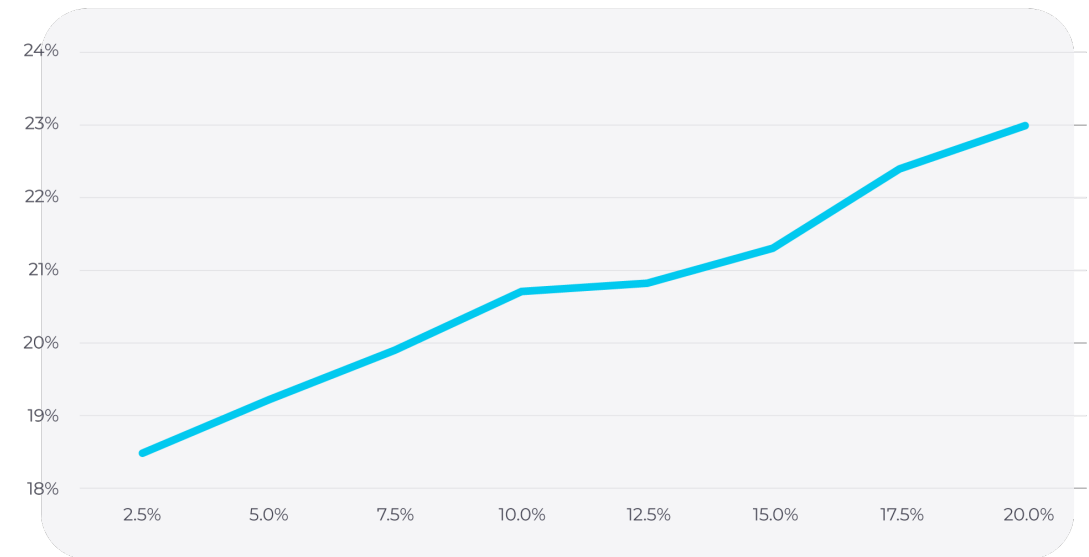
Attacker with Physical Access

- 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

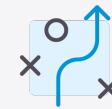
- Besides timestamps, anchors transmit on the wire the power level information of the received UWB signal. We can compute two different metrics:

- First Path Power Level (FPPL)

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

- 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

```
Status: 0
TxS Timestamp: 16.8828403395276
RxS Timestamp: 10.6943580252153
First Path Amp 1: 5064
First Path Amp 2: 14560
First Path Amp 3: 22328
Maximum Growth CIR: 14423
Rx Pream Count: 245
Extra Data Type: 0
Extra Data Length: 0
Checksum: 0xd9c
```

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** t_0 the power level information is identical, the tag j_0 that triggered those packets must be positioned about exactly at the same distance from all anchors

$$\Delta(i_0, j_0, t_0) = (GT(\text{reference}, j_0, t_0) - GT(i_0, j_0, t_0)) * c = 0$$

- Considering that $CS(\text{reference}, t_0) = 1$ and $ToF(\text{reference}) = 0$, we can exploit this information and estimate the time of flights, thus the distances of the other anchors from the reference

$$ToF(i_0) = CS(i_0, t_0) * (pTs(i_0, j_0, t_0) - sTS(i_0, t_0)) - pTs(\text{reference}, j_0, t_0) + sTS(\text{reference}, t_0)$$

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

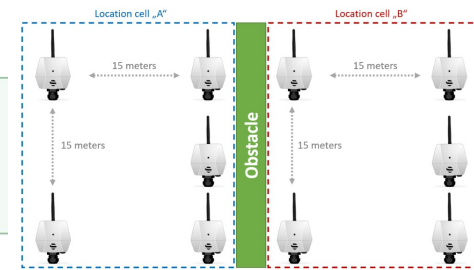


Attacker with Remote Access – Cont'd

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

- ✓ **Keep a square geometry during the deployment.**

Due to [dilation 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

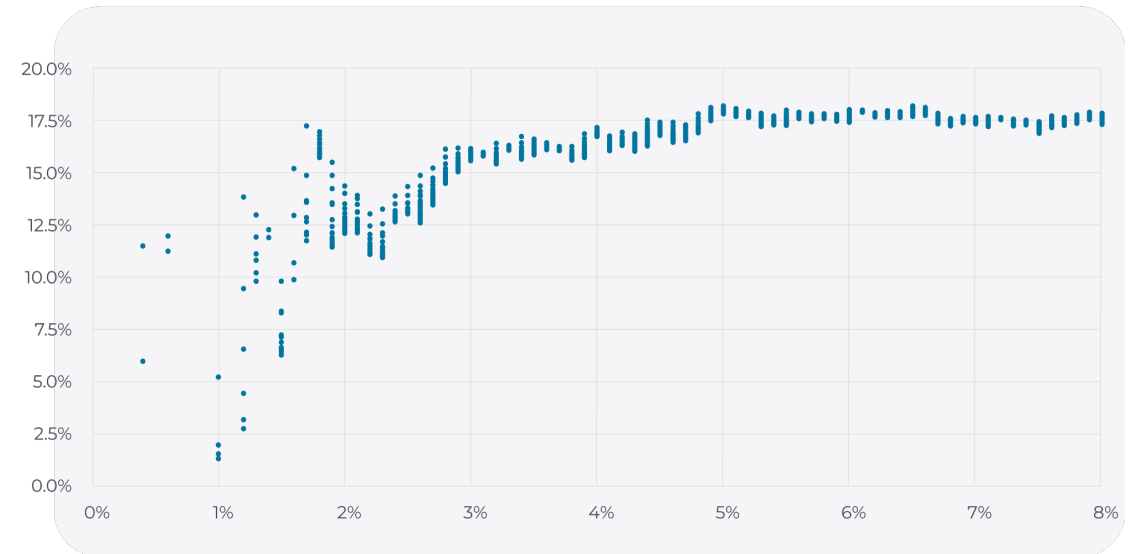
- This was actually tested in the Avalue RTLS, using both the First Path Power Level (FPPL) as well as the Receive Power Level (RPL)
- The best results are obtained using FPPL with a threshold of ~1% between the lowest power level and the highest



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

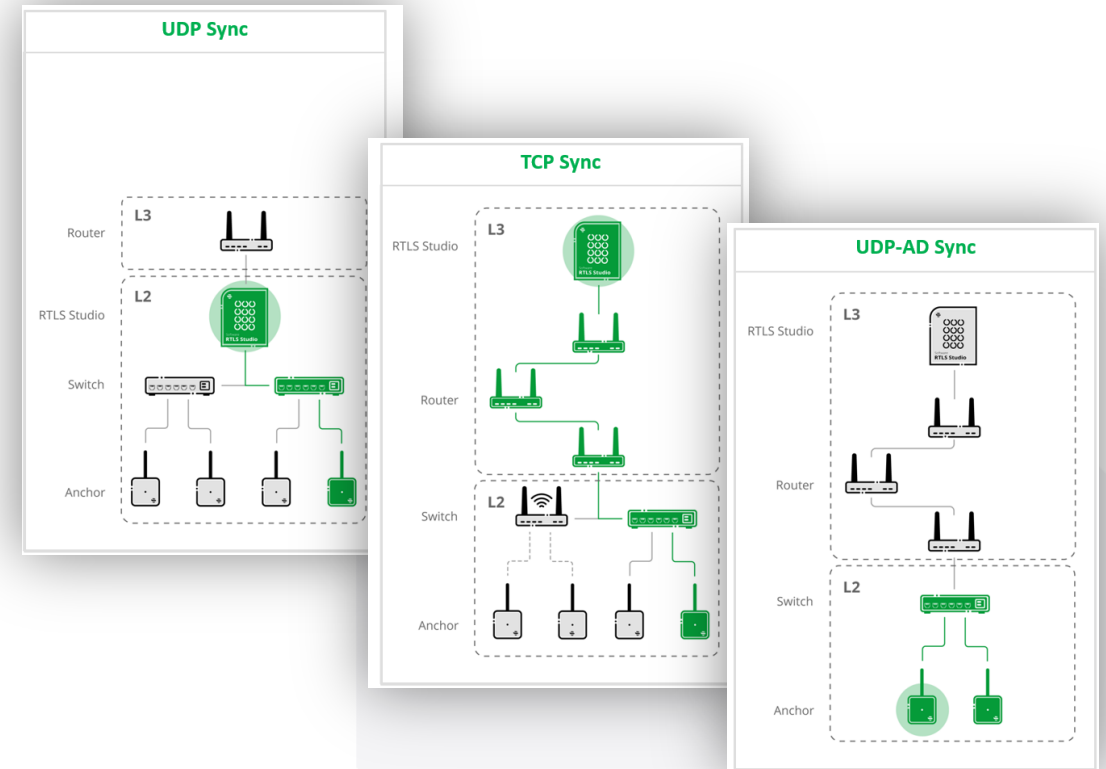
Intercepting traffic requires two steps:

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: 708041019
Wi-Fi credentials for admin access are:
SSID: Artemis
Password: 708041019

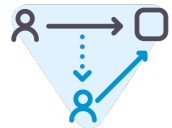
Avalue also provide **a configuration file** that can be simply re-loaded into the router.
To load the file into the router, follow these steps:



Default WPA2-PSK password on Avalue RTLS

Traffic Interception – Cont'd

To perform any meaningful attacks against RTLSSs, 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

No.	Time	Source	Destination	Protocol	Length	Info
921	21.5946448	192.168.225.13	192.168.225.2	UDP	578	7000 - 7000 Len=536
922	21.5946904	192.168.225.13	192.168.225.2	UDP	578	7000 - 7000 Len=536
923	21.5958976	192.168.225.2	192.168.225.13	ICMP	590	Destination unreachable (Port unreachable) (Port unreachable)
924	21.5958990	192.168.225.2	192.168.225.13	ICMP	590	Destination unreachable (Port unreachable)
925	21.5954406	192.168.225.13	192.168.225.2	SEWIO_LMB	85	5000 - 5000 Len=43
926	21.5984952	192.168.225.11	192.168.225.2	SEWIO_LMB	85	5000 - 5000 Len=43
927	21.5984952	192.168.225.14	192.168.225.2	SEWIO_LMB	85	5000 - 5000 Len=43
928	21.5984953	192.168.225.12	192.168.225.2	SEWIO_LMB	85	5000 - 5000 Len=43
929	21.5984958	192.168.225.13	192.168.225.2	SEWIO_LMB	85	5000 - 5000 Len=43
930	21.5984962	192.168.225.11	192.168.225.2	SEWIO_LMB	85	5000 - 5000 Len=43
931	21.5984966	192.168.225.14	192.168.225.2	SEWIO_LMB	85	5000 - 5000 Len=43
932	21.5984969	192.168.225.12	192.168.225.2	SEWIO_LMB	85	5000 - 5000 Len=43
933	21.5987572	192.168.225.12	192.168.225.2	SEWIO_LMB	219	5000 - 5000 Len=177
934	21.5987584	192.168.225.12	192.168.225.2	SEWIO_LMB	219	5000 - 5000 Len=177
935	21.59948719	192.168.225.15	192.168.225.2	SEWIO_LMB	214	5000 - 5000 Len=172
936	21.59949015	192.168.225.15	192.168.225.2	SEWIO_LMB	214	5000 - 5000 Len=172
937	21.5996282	192.168.225.11	192.168.225.2	SEWIO_LMB	278	5000 - 5000 Len=236
938	21.5996285	192.168.225.11	192.168.225.2	SEWIO_LMB	278	5000 - 5000 Len=236
939	21.5996453	192.168.225.14	192.168.225.2	SEWIO_LMB	278	5000 - 5000 Len=236
940	21.5996902	192.168.225.14	192.168.225.2	SEWIO_LMB	278	5000 - 5000 Len=236
941	22.0049392	192.168.225.13	192.168.225.2	SEWIO_LMB	278	5000 - 5000 Len=236
942	22.0049395	192.168.225.13	192.168.225.2	SEWIO_LMB	278	5000 - 5000 Len=236
943	22.0073390	192.168.225.12	192.168.225.2	SEWIO_LMB	404	5000 - 5000 Len=59

Frame 926: 214 bytes on wire (1712 bits), 214 bytes captured on interface 0 (eth0) at 2023-04-07 16:08:00.000000000
Ethernet II, Src: Winstars_04:79:16 (80:3f:5d:04:79:16), Dst: 192.168.225.2 (08:00:27:19:8f:7c:9f)
Internet Protocol Version 4, Src: 192.168.225.15, Dst: 192.168.225.2
User Datagram Protocol, Src Port: 5000, Dst Port: 5000
Sewio LMB Protocol

MitM attack against Sewio RTLSS

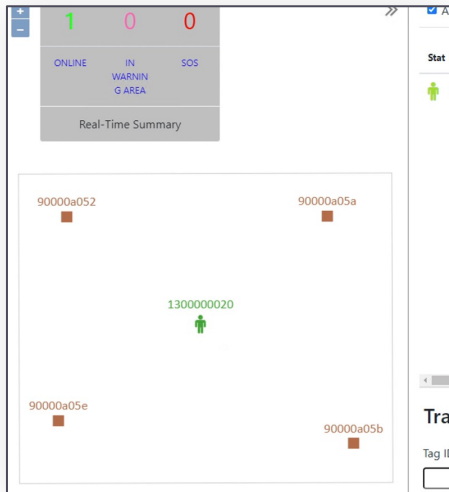
arp spoof -i attacker_eth -t server_ip anchor1_ip & arp spoof -i attacker_eth -t anchor1_ip server_ip



The attacks were completely **undetected** by the RTLSS. 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

```
1644 13.290239 192.168.50.51 192.168.
1645 13.290283 192.168.50.54 192.168.
1646 13.290403 192.168.50.53 192.168.
1654 13.409882 192.168.50.53 192.168.
1655 13.409918 192.168.50.52 192.168.
1656 13.409965 192.168.50.54 192.168.
1660 13.451550 192.168.50.53 192.168.
1661 13.451550 192.168.50.54 192.168.
1662 13.451564 192.168.50.51 192.168.
1663 13.451564 192.168.50.52 192.168.
1670 13.550802 192.168.50.52 192.168.
> Frame 1639: 95 bytes on wire (760 bits)
> Ethernet II, Src: Netgear_78:08:ca (86
> Internet Protocol Version 4, Src: 192.
> User Datagram Protocol, Src Port: 4433
> Avalue UwB Protocol
> Separator: 0x5758
> Packet Type: CCP (19)
> Body Length: 0x2f
> Body
> Order: 22197
> Slave ID: 0x0000000090000a05a
> Master ID: 0x0000000090000a052
> Status: 0
> Tx Timestamp: 3.62468110875839
> Rx Timestamp: 12.2954795412191
> First Path Amp 1: 10446
> First Path Amp 2: 9699
> First Path Amp 3: 58661
> Maximum Growth CIR: 7141
> Rx Pream Count: 247
> Extra Data Type: 0
> Extra Data Length: 0
> Checksum: 0x0f37
```

02

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

Input interpretation

$$0 = \sqrt{X^2 + Y^2} - \sqrt{(-6.99 + X)^2 + (5.29 + Y)^2}$$
$$-0.299792 = \sqrt{X^2 + Y^2} - \sqrt{(0.25 + X)^2 + (4.92 + Y)^2}$$

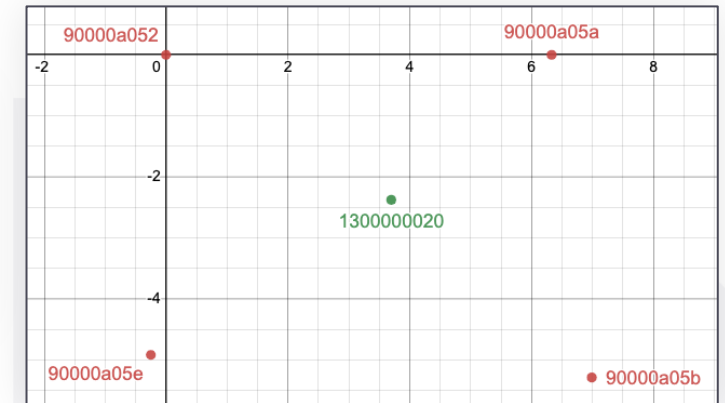
solve

Result

$X = 3.69763$ and $Y = -2.37725$

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

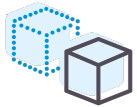


Active Traffic Filtering

- 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

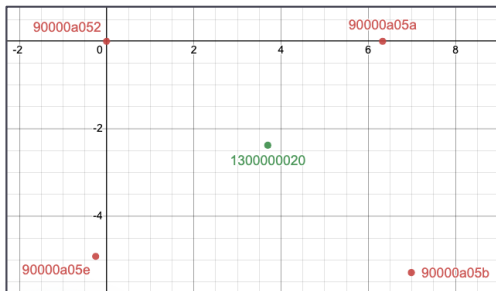
Active Traffic Manipulation Attacks – Cont'd

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

$$\sqrt{(X_{j0}, t0 - X_{reference})^2 + (Y_{j0}, t0 - Y_{reference})^2 + (Z_{j0}, t0 - Z_{reference})^2} - \sqrt{(X_{j0}, t0 - X_1)^2 + (Y_{j0}, t0 - Y_1)^2 + (Z_{j0}, t0 - Z_1)^2} = \text{Delta}(1, j0, t0)$$

...

$$\sqrt{(X_{j0}, t0 - X_{reference})^2 + (Y_{j0}, t0 - Y_{reference})^2 + (Z_{j0}, t0 - Z_{reference})^2} - \sqrt{(X_{j0}, t0 - X_N)^2 + (Y_{j0}, t0 - Y_N)^2 + (Z_{j0}, t0 - Z_N)^2} = \text{Delta}(N, j0, t0)$$

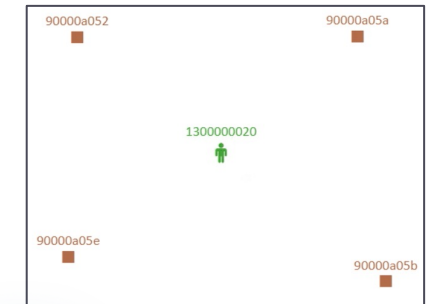
02

The TDoA algorithm is applied backwards

```
▼ Avalue UWB Protocol
Separator: 0x5758
Packet Type: CCP (19)
Body Length: 0x2f
▼ Body
Order: 22197
Slave ID: 0x000000000000a05a
Master ID: 0x000000000000a052
Status: 0
Tx Timestamp: 3.62468110875839
Rx Timestamp: 12.2954795412191
First Path Amp 1: 10446
First Path Amp 2: 9699
First Path Amp 3: 58661
Maximum Growth CIR: 7141
Rx Pream Count: 247
Extra Data Type: 0
Extra Data Length: 0
```

03

Packet is updated



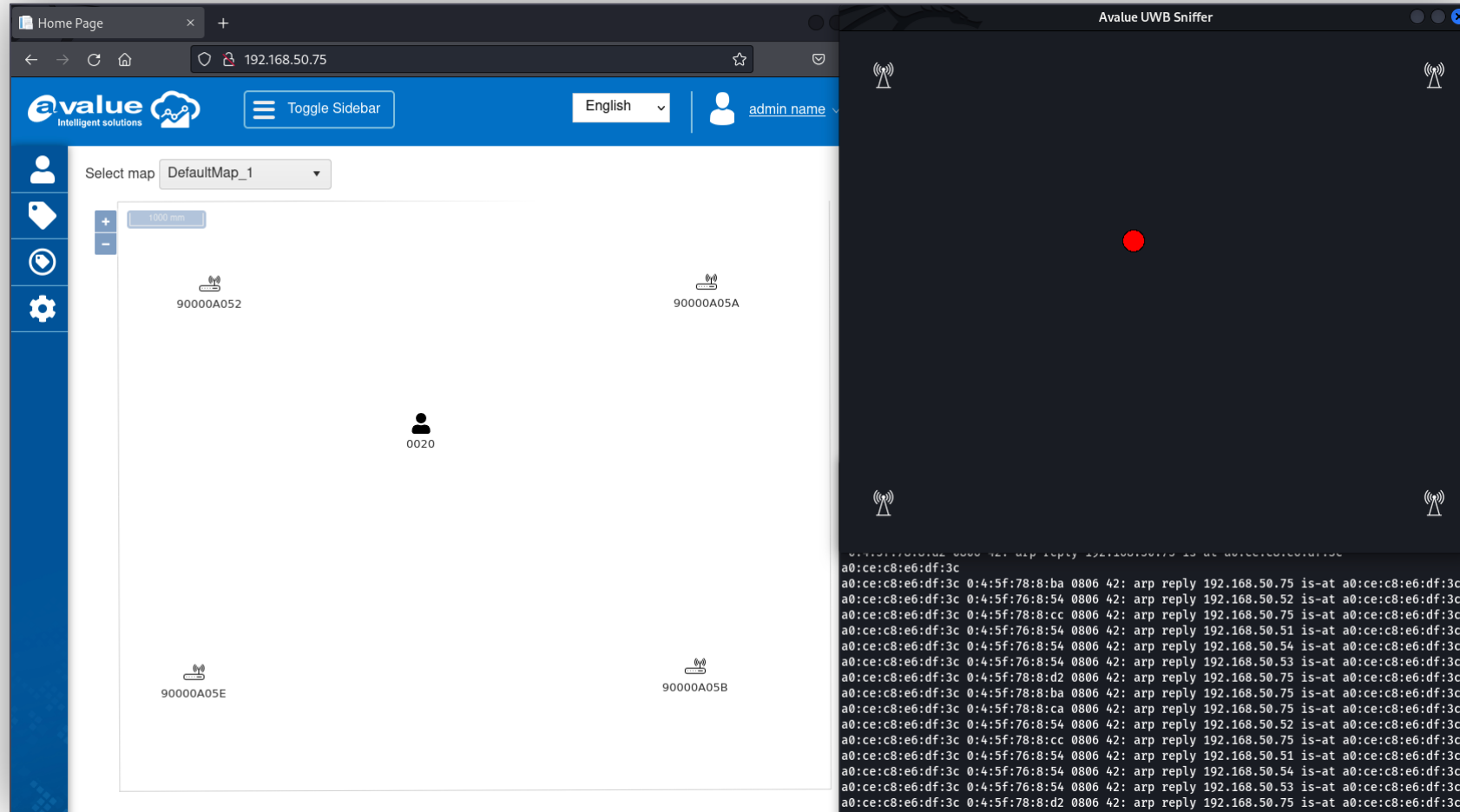
04

Position of target is altered in RTLS

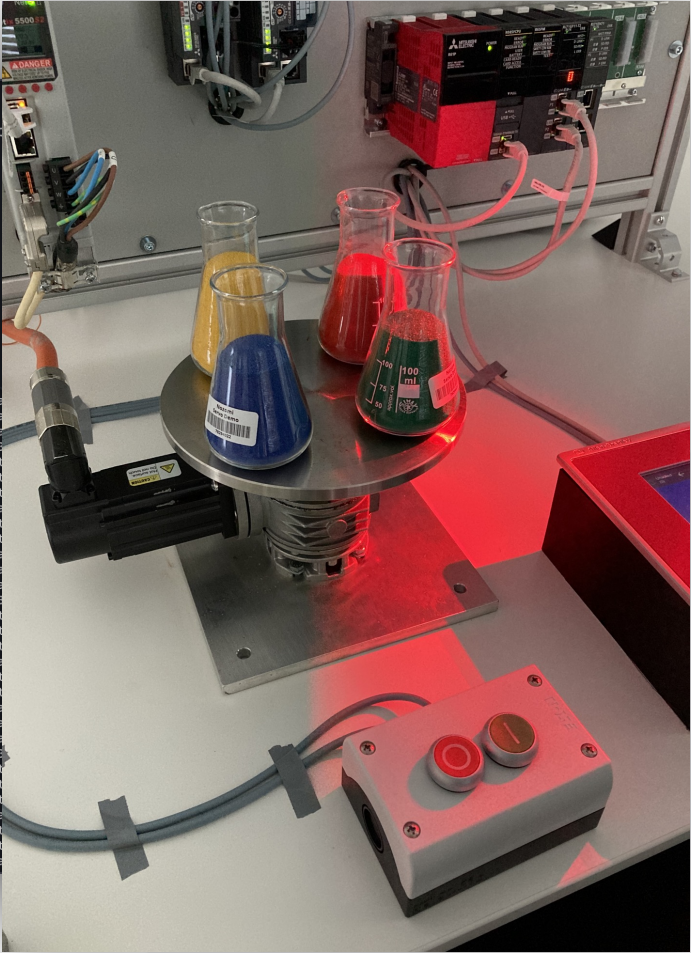
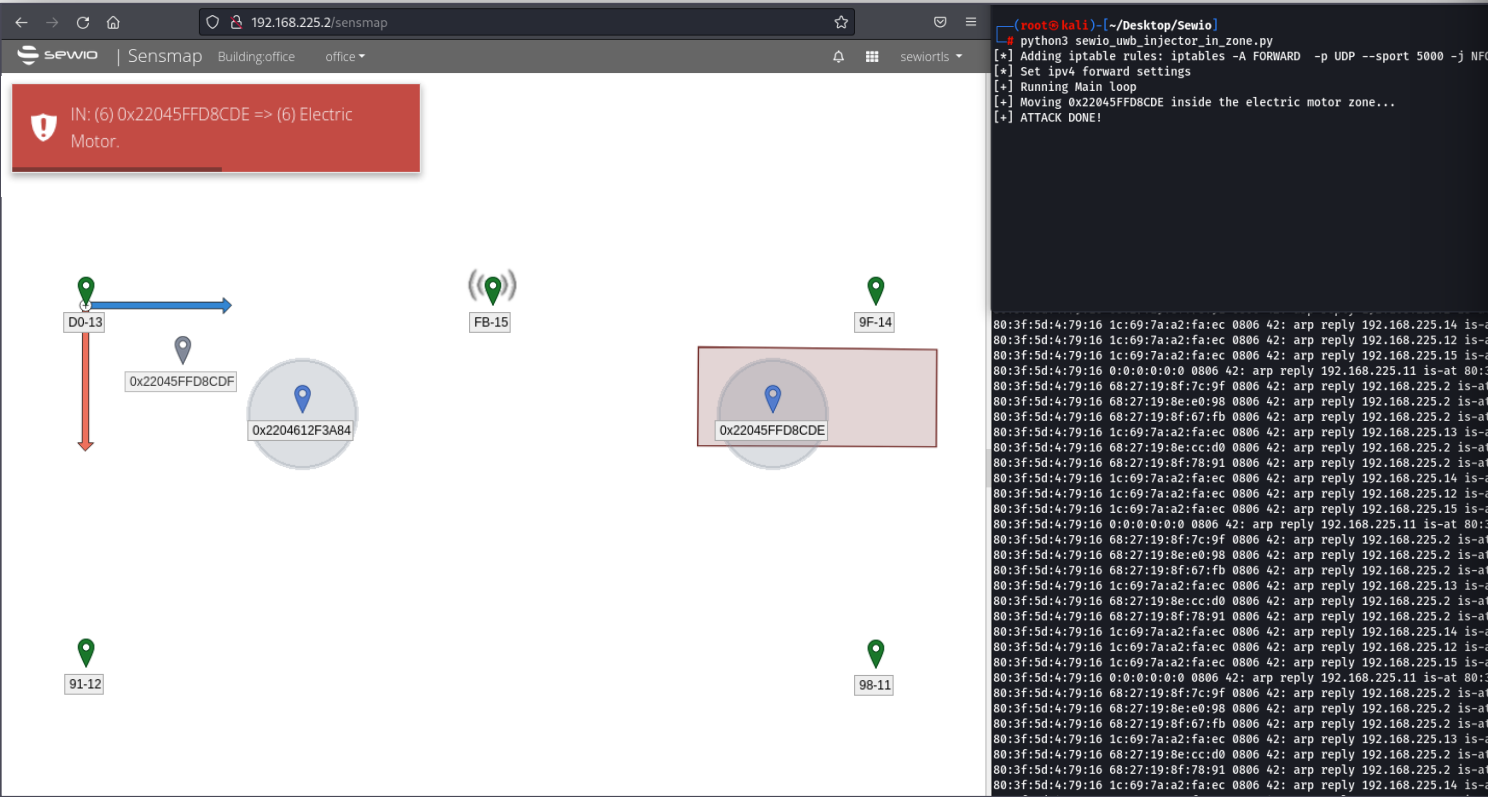
Attack Demos 



Locating and Targeting People/Assets



Geofencing



Contact Tracing

The screenshot displays a web browser window at `192.168.225.2/sensmap` showing a contact tracing map. The map features several location pins labeled with IDs: D0-13, FB-15, 9F-14, 91-12, and 98-11. A central Venn diagram shows the intersection of two MAC addresses: `0x22045FFD8CDE` and `4612F3A84`. A blue notification box in the top left corner states: "ENCOUNTER 0x22045FFD8CDE and 0x2204612F3A84".

To the right, a terminal window shows the execution of a Python script `python3 sewio_uwb_injector_gen_false_contact.py`. The script's output includes:

```
[*] Adding iptable rules: iptables -A FORWARD -p UDP --sport 5000 -j NFQUEUE --queue-num 1
[*] Set ipv4 forward settings
[+] Running Main loop
[+] Inducing false contact between 0x2204612F3A84 and 0x22045FFD8CDE
```

Below the terminal output, a series of network logs are visible, showing ARP requests and replies between various IP addresses and MAC addresses, such as `80:3f:5d:4:79:16` and `1c:69:7a:a2:fa:ec`.

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.

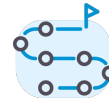
SIMATIC RTLS4030G
Operating Instructions, 04/2021, C79000-G8976-C515-06

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

The screenshot shows the Nozomi Networks Alerts dashboard. The top navigation bar includes 'Dashboard', 'Appliances', 'Alerts', 'Environment', 'Analysis', and 'Smart Polling'. The 'Alerts' section is active, displaying a table of alerts. The table has columns for 'RISK', 'TIME', 'NAME', and 'DESCRIPTION'. The first alert is a 'MITM attack' with a risk level of 10, occurring on 2022-06-01 at 17:51:03.601. The description states: 'Attacker identified by MAC address 00:24:32:16:95:fe is acting as a MITM, its victims are: 192.168.225.13, 192.168.225.2'. A detailed view of this alert is shown on the right, including the text: 'A potential MITM attack has been detected. The attacker is ARP-poisoning the victims. The attacker node could alter the communication between its victims.' and 'Attack analysis: This alert could be: A Execution tactic (technique MITM, T0830)'.

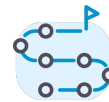
RISK	TIME	NAME	DESCRIPTION
10	2022-06-01 17:51:03.601	MITM attack	Attacker identified by MAC address 00:24:32:16:95:fe is acting as a MITM, its victims are: 192.168.225.13, 192.168.225.2
9	2022-06-01 17:52:03.080	New Node	New node 192.168.225.35 appeared on the network
7.5	2022-06-01 17:51:00.017	Duplicated IP	IP 192.168.225.14 is duplicated by MACs: 00:24:32:16:95:fe, 00:24:32:16:95:fe
7.5	2022-06-01 17:50:59.761	Duplicated IP	IP 192.168.225.15 is duplicated by MACs: 00:24:32:16:95:fe, 00:24:32:16:95:fe
7.5	2022-06-01 17:50:58.737	Duplicated IP	IP 192.168.225.12 is duplicated by MACs: 00:24:32:16:95:fe, 00:24:32:16:95:fe
7.5	2022-06-01 17:51:00.273	Duplicated IP	IP 192.168.225.13 is duplicated by MACs: 00:24:32:16:95:fe, 00:24:32:16:95:fe
7.5	2022-06-01 17:50:57.713	Duplicated IP	IP 192.168.225.11 is duplicated by MACs: 00:24:32:16:95:fe, 00:24:32:16:95:fe

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)

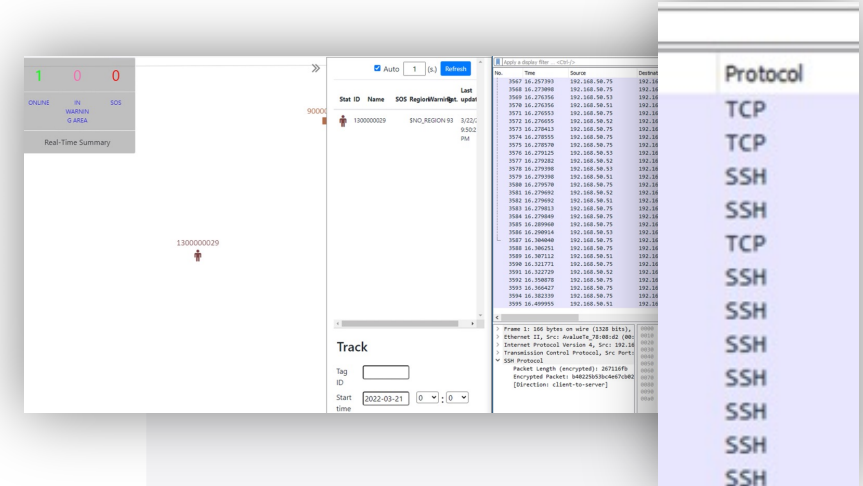
Traffic Encryption



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)

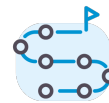


SSH tunnel PoC on Avalue RTLS



Advantages

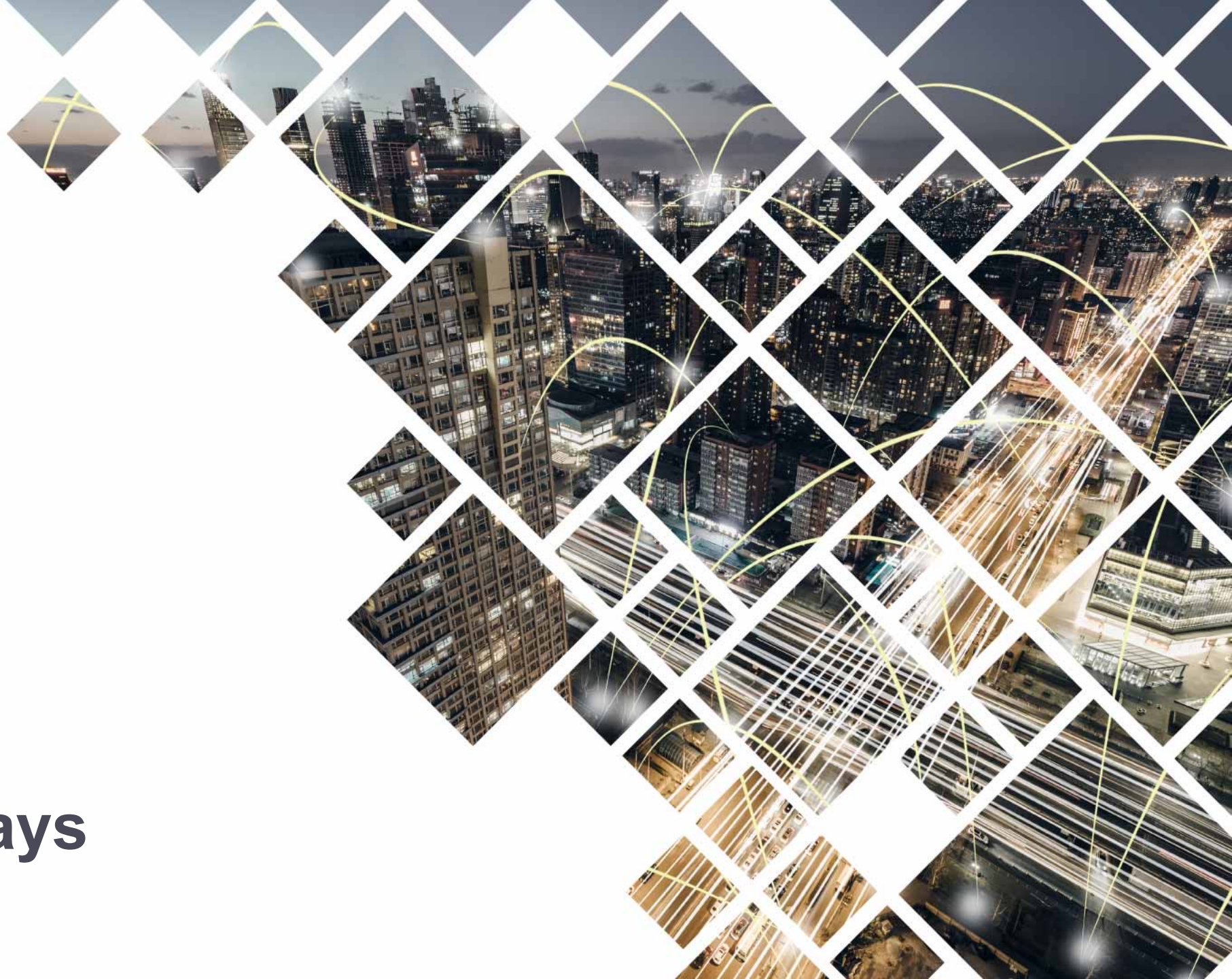
- The closest mitigation to completely solving the problem
- Allows basic RTLS functionalities to remain unaltered



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

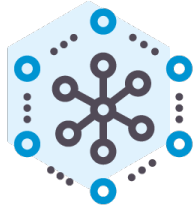
- IEEE 802.15.4z has out of scope areas, creating security loopholes
- Nozomi Networks Labs discovered zero-days in two popular UWB RTLS
- UWB RTLS is used for personnel tracking, geofencing, and contact tracing
- Threat actor TTPs are MitM and eavesdropping or manipulation tactics
- Mitigations include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