VULNERABILITY OF REMOTE MONITORING AND CONTROL SYSTEMS IN THE OIL AND GAS INDUSTRIES

George Tyson and David Allen
OilTek Systems LLC

INTRODUCTION

The Modern oil and gas industry makes extensive use of systems to remotely monitor and control operations throughout the process. Supervisory Control and Data Acquisition (SCADA) systems have had wide acceptance and have been used for years.

Many SCADA systems use the Modbus protocol, developed in 1979, to communicate between the parts of the system. Most of these systems operate under MS Windows or DOS which creates an environment of ever expanding vulnerabilities. Hackers have used these vulnerabilities to wipe out revenue and destroy infrastructure.

These vulnerabilities are growing. Every time MS Windows is updated there is a chance of a new vulnerability to be introduced into the operating system or in third party software. Increased use of WiFi and Bluetooth for data transfer has increased exposure to system attacks and exploits

The new system that is entering service in the industry is based on the Internet of Things (IoT) or the Industrial Internet of Things (IIOT) system architecture. The very nature of the IoT architecture provides native security that exceeds SCADA and other legacy systems and can limit the exposure to attacks through WiFi, Bluetooth, cellular, LoRa and other wireless technologies/protocols along with traditional hacks similar to malware or phishing.

SCADA

SCADA systems are industrial control systems that monitor, report data, and can automate controls and responses. This technology has been applied in many Industries which have experienced how SCADA systems are the convergence of information technology (IT) and operational technologies (OT), resulting in substantial gains in efficiency and lower costs. Such benefits do not come without risks.

SCADA systems create interdependencies between the cyber environment and the operational world providing a path for cyber-attacks to affect real world operations. While distributed resources and automation of these decentralized systems tends to decrease physical risk, the required increase in communications and reliance on the Internet increase cyber and operational risks.

The first and second generation SCADA systems were limited to single site networks or a single building as a standalone or sealed system. In these systems, risk was limited to physical attacks or on-site hacking. The third generation SCADA system, which is connected to the internet, provides multiple routes for a cyber-attack increasing security risks. The risk is multiplied as parallel distributed SCADA systems are connected to a single supervisor, or master, in a network architecture.

The third generation SCADA systems (Figure 1) have three major attack points. The first would be through a corporate server or LAN which communicates directly with the SCADA master. Credentials could be acquired through phishing, Trojans or other malware giving hackers access. This type attack can corrupt the corporate computer exposing the SCADA master even when the attack occurred through a

non-SCADA user. Insuring good security practices,such as having all OS security patches installed and insuring anti-malware applications are up to date, will prevent most automated attacks. The area of the most vulnerability on these systems is the users.

The second would be a direct attack on the SCADA master. Since the SCADA master must be exposed to the internet,so that it can communicate with the Communications Servers which are connected to remote nodes, it leaves it vulnerable to direct brute force attacks. Again good security practices prevent most attacks. The user vulnerability becomes less due to less users, usually administrators, having access.

The third attack avenue is through the communications servers. These units are normally located in remote areas and directly connect to Remote Terminal Units (RTU) which interface the sensors to the SCADA system. The communications server connects to the SCADA master over the internet. It can also connect to other communications servers creating a miniature WAN system. The vulnerability of these systems are that they are located in remote areas and are difficult to update resulting in out of date security.

Any of the three attack points can provide access to other parts of the distributed SCADA system by exposing security credentials, system addresses and discovery of usernames/passwords, allowing the attacker almost unlimited access and control of the SCADA system and anything the SCADA system commands.

SCADA compromised

SCADA systems have been compromised in the past. Some examples include:

The New York Bowman Dam's SCADA system was compromised by Iranian hackers in 2013. The dam which controls storm surges had it SCADA system connected to the internet via a cellular modem. No control features were accessed because the SCADA system was in maintenance mode disabling the control features.

In 2010 one of the most famous SCADA attacks happened in Iran. Stuxnet, a complex malware worm specifically written to attack SCADA networks, destroyed as many as one-fifth of Iranian atomic centrifuges. This malware would rewrite software at the PLC level resulting in corrupted data being reported to the SCADA master and loss of operational control.

In 2014 alerts were published by ICS-CERT about a version of Malware called BlackEnergy. This malware infected SCADA systems through SCADA products such as HMI Master Stations.

Also in 2014 a German steel mill suffered a SCADA attack through the mills business network. The attack was able to compromise the SCADA production network preventing a blast furnace from shutting down correctly, causing extensive damage to the mill.

Good cyber security and constant vigilance can and do stop SCADA cyber attacks however, the basic distributed SACDA architecture provides multiple opportunities for bad actors to take over the complete system. An alternative architecture is the Internet of Things (IoT) or the Industrial Internet of Things (IIoT). The basic architecture provides less attack routes, more opportunity to stop an attack and the built in ability to stop attacks from compromising the complete system when compared to SCADA.

INTERNET OF THINGS

IoT systems are designed to provide machine to machine communication, automation of processes and the use of Artificial Intelligence (A.I.). The architecture incorporates cloud computing and storage creating a natural block to hackers. Just as SCADA, IoT is able to monitor, report data, and automate controls and responses. It can even incorporate A.I. providing a more intelligent way of creating a hands-off process control function.

A basic IoT distributed system has a four layer architecture. The sensing layer contains the sensors and actuators this is connected to the network layer which contains the Edge Computers which perform data aggregation and conversion along with functioning as data gateways. The data gateways are connected to the data processing layer over the internet using WiFi, Cellular, LoRa or LoRaWAN, Ethernet, etc. where the data is analyzed and processed. The final layer is the application layer where the users can manage the data and control systems through the HMI. Both the data processing layer and the application layer exist within the cloud services.

A properly designed and implemented IoT system should be flexible, redundant, have multiple user logins and provide near real-time data. It should also be inexpensive and require minimum effort to deploy or modify. An IoT design must be flexible so that it can meet the specific needs of an installation without a completely new design supporting similar operations that have distinct needs. Just as systems to drill and develop wells are modified to meet the requirements of each well, a properly designed IoT system will not require a redesign but only a modification to meet the different system requirements.

Flexibility of IoT is achieved through the use of distributed isolated parts which, as explained later in the paper, also provides native protection against cyber threats. Any given function of an IoT system is separate from the other parts creating a natural firewall. Data collection is separate from data processing which is separate from data storage which is separate from the human/machine interface (HMI), As long as the inputs/outputs remain the same, each part can be changed or modified without affecting the other parts. This provides a flexible method to meet differing requirements without a full system redesign. Legacy data systems suffer from a centralized approach. A small change in one subsystem creates havoc in other portions of the program, resulting in a static system that requires operations to conform to it, not it conforming to the operation.

A redundant IoT system uses Edge Computing, cloud processing and storage which insures the retention of data and control during unforeseen circumstances, disasters or a cyber-attacks. Communication outages, power failure, hostile attacks or even negligence can disable all data collection, processing and control for a centralized system. A properly designed IoT system has the ability to switch to sections that are not affected and/or switch data storage and control to an Edge Computer. When the interruption ceases, the Edge Computer sends all data, results and actions its dedicated cloud module.

Quality real-time (or near real-time) data allows for accurate decisions with immediate observable results. Having real-time data and operational control at your fingertips allows the operation to be fine-tuned for efficiency and output and/or discover an issue before it results in loss of revenue. Figure 3 exemplifies a real-world example of how IoT can help discover issues.

IoT Parts

Sensing layer; Data collection and/or control – Sensors, actuators, valves, etc. are located near or in the operations that the IoT system is designed to measure and/or control. Examples would include collection of flow rates, volumes, pressure, temperature and the movement of valves, activation/deactivation of pumps and/or operation of a relay.

Network layer; Edge Computing – A processing unit located near the operation which collects data from sensors and does basic processing. The Edge Computer A.I. uses this information, and optional input from users, to make changes to the operation through control devices The Edge Computer could also be used to contact users directly if an action is required or to inform users of a change in the operation. All activity and results are uploaded to a dedicated cloud module by the Edge Computer. In many cases this Edge Computer is a smart PLC or single board computer.

Edge Computing within an IoT system can support real-time or near real-time data collection and operational control by reducing the bandwidth requirements. Legacy systems require all data to be sent to a central processing system and all control commands be issued by the same central system. This results in a bottleneck for the data and limits the amount of data that can be collected. It reduces the number of

devices that can be measured and how often measurements are done. Edge Computing provides basic data processing and packaging, allowing more operational data to flow through a smaller data pipe at a faster pace. Multiple data pipes to the cloud functions can be achieved by adding additional Edge Computers, providing a way for more data to be collected during an operational day.

Data processing layer; Cloud computing service – The cloud provides advanced data processing, powerful A.I. and access to the data and analytics for multiple users. Unlike legacy data systems, an IoT system can have multiple processing units each receiving data from different operations and doing different types of processing, yet having all results available through one Human Machine Interface (HMI) multi-user/type port and saved to a cloud database. The amount and type of services can also be easily adjusted in response to system load unlike static legacy systems. For maximum security against cyber threats each Edge computer would connect to a single dedicated cloud computing module.

Application layer: Data Storage – Cloud based databases are utilized so that the data and results being generated are stored properly. This allows the database to conform to the type of data and use instead of forcing data into an incompatible database. Legacy systems typically have one type of database which forces all data to conform to its format, resulting in lost information and a lack of access to multiple users.

Application layer; Human/Machine Interface – All data access and system control is done through an HMI access port using Smart Phones, Tablets and/or PCs. The HMI can reside within the cloud or on a separate server. This allows access to the IoT system from anywhere, anytime and by any authorized person.

IoT Cyber Vulnerabilities

Three attack points exist in an IoT system; the Edge Computers in the network layer, the cloud service and end users. The Edge Computers are primarily single board computers running a form of Linux with limited computing power. While they are connected to the internet, the attack paths within a properly secured (Unique Linux user id and password) Edge Computer is far more limited than a SCADA Windows based machine. Also, unlike SCADA, if an Edge Computer is compromised it cannot provide a path to the other Edge Computers nor can it take over any processes that are not directly connected to the compromised computer. For hackers to access the other Edge Computers, the attack path would have to pass through the cloud service.

The second attack point is the cloud service (cloud). The attack path could occur from a compromised Edge Computer, a direct attack against the cloud or from the user side. Companies who provide cloud services take security very serious (their business depends on security) and will compel users to implement security rules (example 1) which reduce the risk from both the Edge Computers and the users. For an Edge Computer to access the cloud services it must not only authenticate but it must also follow security rules which highly restrict access within the cloud. The same applies to the HMI or application layer.

The Edge Computer would connect to a dedicated cloud module which communicates with the other cloud modules which in-turn communicate with their independent Edge Computer. The security rules dictate how the Edge Computer interacts with the cloud and how the data is processed. Even if a hacker was able to compromise an Edge Computer and steal its credentials, the hacker could only interact the connected cloud service within the parameters of the security rules. They would be unable to create a direct connection to any other Edge Computer or the HMI.

Preventing an attacker from acquiring a copy of the data going through the data gateway is easily thwarted by encryption. Many IoT data gateways are wireless providing easy access to the data being transferred. To prevent the eavesdropper from utilizing the data simple to complex encryption schemes can be utilized.

The final attack point is the end users, the risks here are similar to SCADA with one important difference, the attack can only occur through the users of the IoT system. Poor passwords, phishing attacks, spoofing sites, etc. have all been shown to be effective in compromising user credentials. Good security practice will include multilevel access and security rules that prevent any one user from controlling the entire network.

WIRELESS COMMUNICATION

Both SCADA and IoT system are expanding their use of wireless data transmission. It is being used for communications between all parts of their architectures and poses a threat to data loss, data theft and data spoofing.

Data loss

Many operations on in the oilfield still use electric motors that are classified as Incidental Radiators by the FCC. From part 15(n); "*Incidental radiator. A device that generates radio frequency energy during the course of its operation although the device is not intentionally designed to generate or emit radio frequency energy. Examples of incidental radiators are dc motors, mechanical light switches, etc.*"

In some cases the radio frequency (RF) strength and frequency can unintentionally interrupt WiFi and Bluetooth signals resulting in data loss. The greater the concentration and size of motor the greater the risk of data loss. The way to overcome interference is to either move the system away from the motor or to use a hardwire connection. Figure 4 shows data collected from a fracking operation where WiFi was being blocked by RF interference.

RF interference can also be caused by intentional means. While jammers/blockers are illegal in the United States they can still be purchased from China. These devices are usually designed to block multiple frequencies that include WiFi and Bluetooth, are extremely portable and can be purchased for less than $200.

While that lost data cannot be recovered an IoT monitoring system can have an alert function in the Edge Computer and/or cloud service that notifies user(s) of the problem in real-time, allowing for a quick fix or the ability to discover the saboteur before large amounts of data is lost.

Cyber-attack

WiFi vulnerabilities not only affect homes and offices but can also affect equipment in the field, it is imperative for WiFi system to be set up according to good security practices (unique SSIDs and Passwords, up to date software/firmware). However successful attacks do still occur making the system architecture and security foremost in stopping the attack and limiting the damage.

WiFi networks can be exploited in the following ways:

Location of access points in the field can be problematic. It needs to be in a location where it can be accessed by other systems, has power, and is secure from tampering. Getting physical access to an entry point is similar to finding, or stealing, the keys to a sports car. The attacker can easily gain full control of the network.

The use of WEP protocol which is based on the RC4 cypher. The cypher itself is not the issue but how it was implemented in WIFI protocols which allows for the reuse of cypher keys. Combined with several other vulnerabilities WEP becomes an immediate way for less than mediocre hackers to gain access to the network.

WPA2 protocol was created and released to address the issues of WEP however, it too has a vulnerability. Due to the way the four-way handshake occurs at first connection a man in the middle attack

can intercept and decode the security keys. WPA3 has been developed and released to address this issue.

Systems that do not encrypt their data could be vulnerable to packet sniffing. Attackers are able to intercept wireless packets and read the data and any unencrypted data would then be exposed. This is the most common exploit and is the majority of attacks on WiFi networks.

There are many other WiFi exploits that can be utilized against a network, some like Warshipping requires a small piece of hardware to be placed in an area where it can access the network while others, such as AirJack, are software based and can be carried out over the internet.

Bluetooth

Bluetooth has been adopted by almost every sector of business and industry. From smart phones to cars to home appliances it is everywhere and is now being used in the oilfield. The upside is that it can connect to almost anything, the down side it is even less secure then WiFi.

The five common Bluetooth attacks are:

BlueBourne – It can affect all operating systems including Windows, Linux, Android and iOS. It is an airborne attack (no physical connection) and allows attackers to install malware along with the ability to penetrate other attached devices. It does not require the attacking device to be paired with the device under attack and has multiple attack vectors.

Bluesnarfing – The target has to be set to “discoverable” which allows nearby devices to locate and pair with it. The attack exploits vulnerabilities with in the object exchange (OBEX) protocol that is built in to Bluetooth. Once the devices are paired, the attacker is able to download data form the attacked device. This type of attack is usually directed toward phones but has been used against remote devices.

Bluejacking – One of the least damaging Bluetooth attacks but it can still be used to corrupt data or send erroneous commands to a device. As with the Bluesnarfing, the device to be attacked must be discoverable. The attacker pairs there device with the target device and then sends unsolicited messages. These messages could contain commands the device recognizes, can cause excess use of batteries or the volume of messages could overwhelm the devices processor resulting in lost data.

Bluetooth Impersonation Attacks (BIAS) – By utilizing a vulnerability in the Bluetooth standard an attacker can impersonate a master or slave device that had been previously attached the victim’s device. In this way the attacker can establish a secure connection while impersonating a different device. This allows the attacker to intercept data that would be exchanged between the devices.

BlueBugging – This is another attack that requires the victim to be in discoverable mode. It takes a skilled attacker for successful execution. After discovery of the software version and hardware manufacture of the Bluetooth in use, the attacker will use a known vulnerability to place a “bug” in the attacked device. This bug can ease drop intercepting commands and security information along with using device built in commands to take control.

SUMMARY

Any system that is distributed and utilizes the internet is at risk. Bad actors of all types are always developing new ways to disrupt, control and blackmail users of distributed data systems. Wireless communication is always vulnerable to attackers and/or jamming with Bluetooth currently being the most susceptible to attack.

System security is always a changing battlefield. As new attack vectors are found, manufactures make adjustment to their protocols and software to prevent successful attacks. This in turn forces the bad actors to find new ways to disrupt systems and the cycle starts over.

The architecture of IoT lends itself to better security then SCADA and prevents entire networks from being corrupted and controlled by hostiles however,to insure the best protection against attackersgood security practices must still be observed and followed. Some of these include:

1. Recognize and understand the threats. Stay abreast of the new threats and how they can affect your system.
2. Use security rules enabled by Cloud Service providers. In an IoT architecture the cloud is your best firewall to attackers. Use the tools provided to keep unauthorized access out and from corrupting your system.
3. Keep systems updated. As new attack vectors are discovered, manufactures will send out updates for software and firmware. Be sure that all systems, including remote stations, are updated.
4. Use encryption. This is mandatory if you are using wireless communication in any part of a system. It is extremely easy to intercept wireless communications which, if unencrypted, gives away your data.
5. Monitor all devices and activity. Watch for unusual or excessive data traffic as this is an excellent indicator that an attack is occurring. In many cases this is the first indicator of an attack and if found early it may allow the damage to be contained.
6. Disable unused devices and user accounts. Abandoned accounts, old nodes that are still functional, unused but connected routers, etc. are all excellent gateways for an attacker because, in most cases, they are not monitored or updated.
7. Security training for end users and administrators. Many attacks start with the user, they need to spot possible attacks, understand the risks and have a way to report suspicious activity.

REFERENCES

Understanding Modbus Protocol - RTU vs TCP vs ASCII https://www.dpstele.com/modbus/index.php. Accessed 10 February 2022

SCADA hack on Florida water plant a reminder of risk to critical infrastructure posed by cyberattacks https://www.verdict.co.uk/water-cybersecurity-scada-hack/#:~:text=It%20is%20just%20over%20ten,with%2014%20major%20control%20systems. Accessed 10 February 2022

Inside the Cunning, Unprecedented Hack of Ukraine's Power Grid https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/ Accessed 10 February 2022

The 4 Stages of IoT Architecture https://www.digi.com/blog/post/the-4-stages-of-iot-architecture#:~:text=IoT%20system%20architecture%20is%20often,building%20or%20even%20a%20person. Accessed 10 February 2022

Architecture of Internet of Things (IoT) https://www.geeksforgeeks.org/architecture-of-internet-of-things-iot/Accessed 10February 2022

What is Cloud Security? https://www.forcepoint.com/cyber-edu/cloud-security Accessed 10 February2022

Cloud Security/Cloud Computing Security https://www.beyondtrust.com/resources/glossary/cloud-security-cloud-computing-security Accessed 10 February 2022

10 Firebase Realtime Database Rule Templates https://medium.com/@juliomacr/10-firebase-realtime-database-rule-templates-d4894a118a98 Accessed 27 January 2022

14 Major SCADA Attacks and What You Can Learn From Them https://www.dpstele.com/blog/major-scada-hacks.php Accessed 10 February 2022

Most Common Wireless Network Attacks https://www.webtitan.com/blog/most-common-wireless-network-attacks/ Accessed 25 February 2022

How to Protect Yourself From Bluetooth Hacking https://www.vectorsecurity.com/blog/how-to-protect-yourself-from-bluetooth-hacking Accessed 28 February 2022

How to Hack Bluetooth Devices: 5 Common Vulnerabilities https://hackernoon.com/how-to-hack-bluetooth-devices-5-common-vulnerabilities-ng2537af Accessed 28 February 2022

ARMIS Research BlueBourne https://www.armis.com/research/blueborne/ Accessed 28 February 2022

What Is Bluesnarfing? https://www.easytechjunkie.com/what-is-bluesnarfing.htm Accessed 28 February 2022

BIAS: Bluetooth Impersonation AttackS https://francozappa.github.io/publication/bias/paper.pdf Accessed 28 February 2022

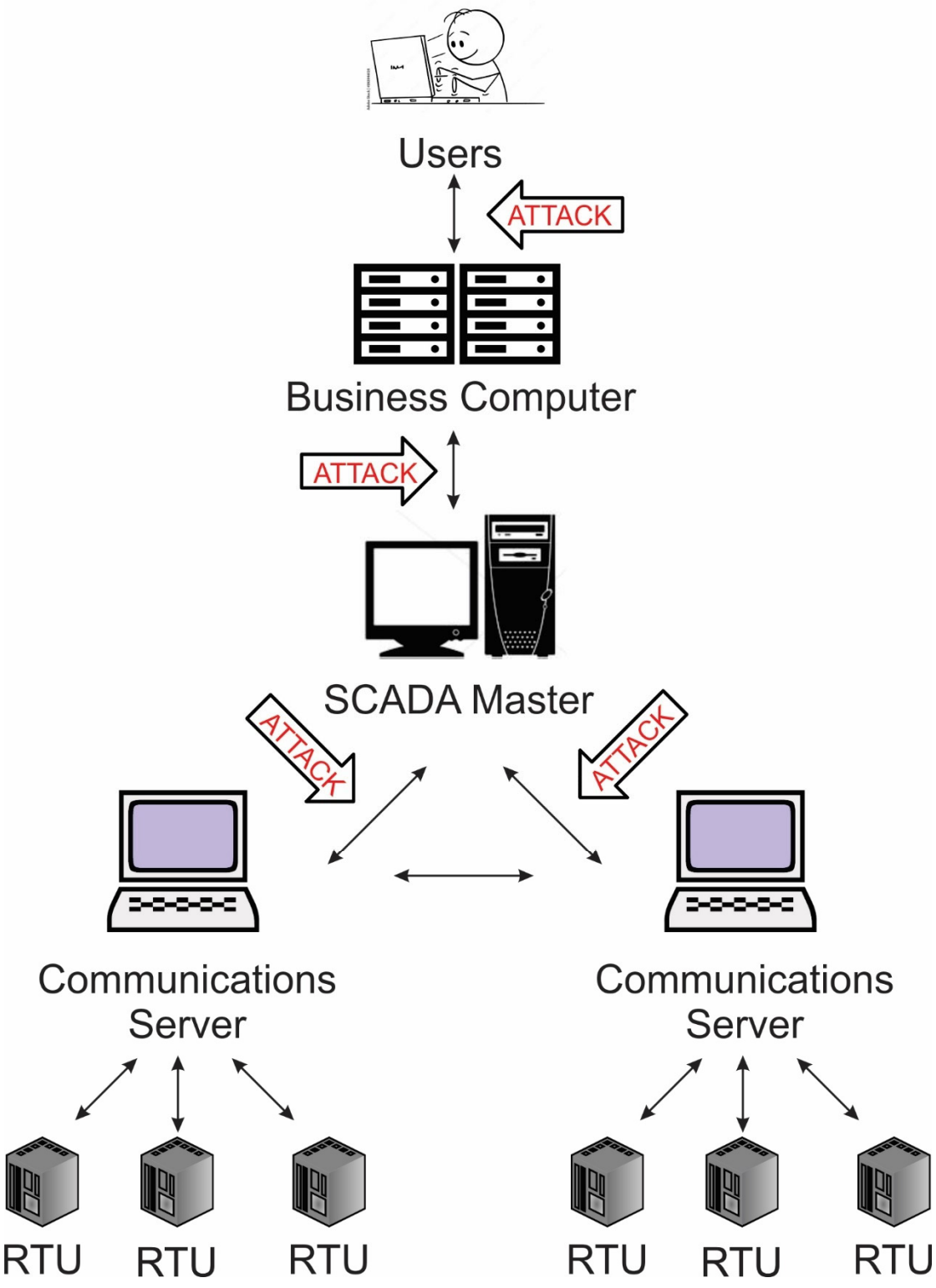

Figure 1 – Distrusted SCADA system and the attack points

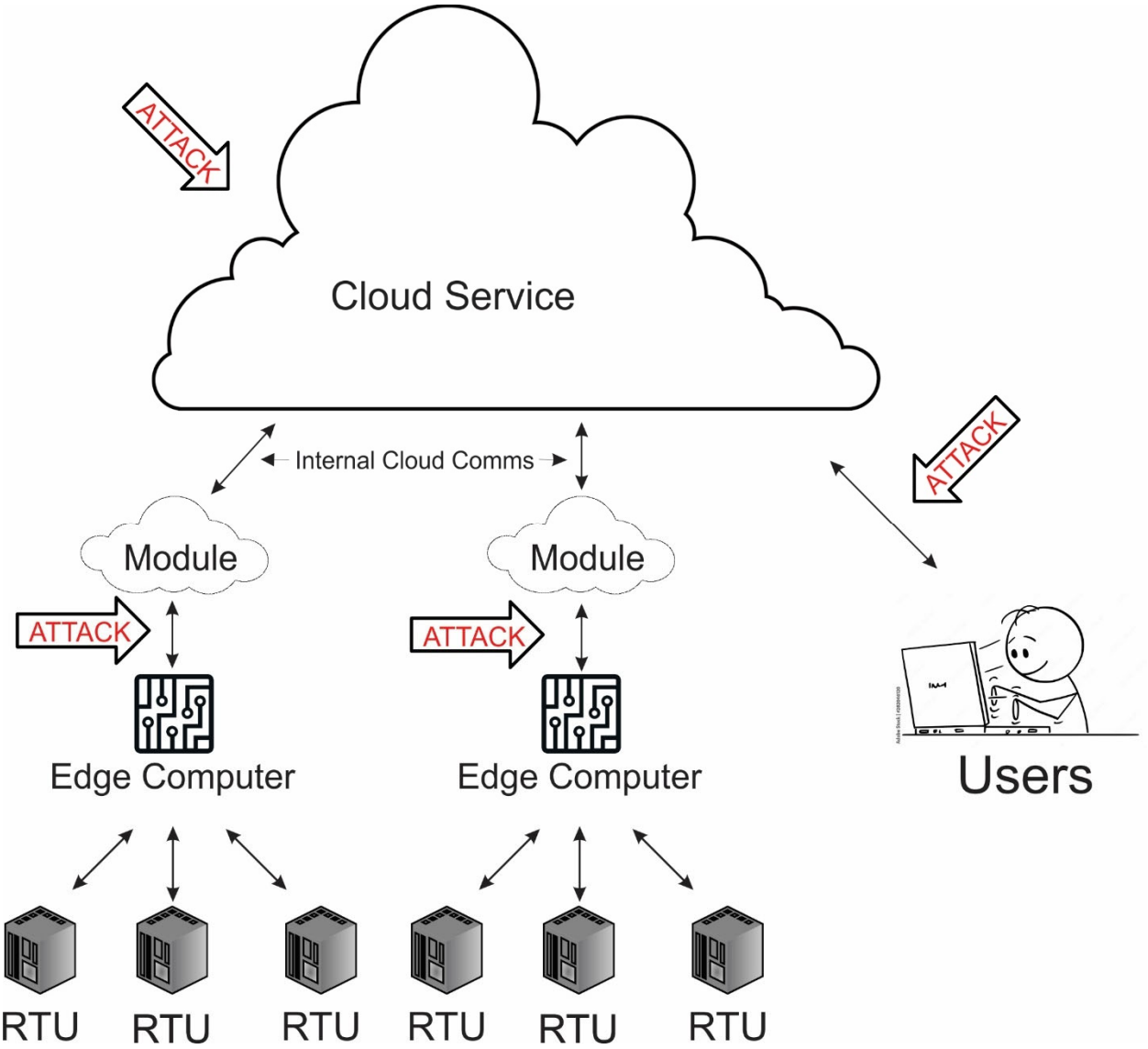

Figure 2 – Typical IoTsystem with attack points

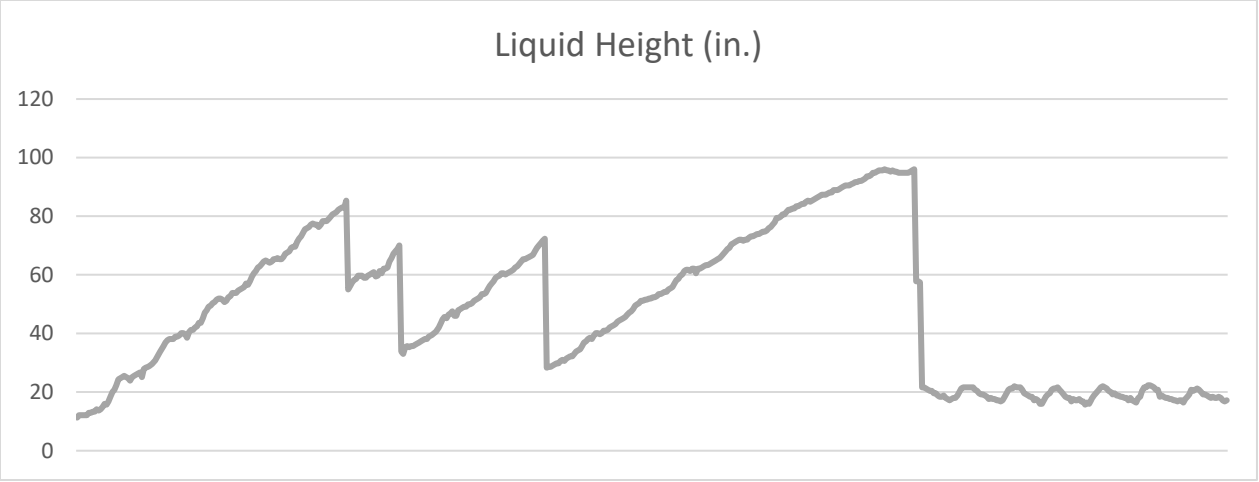

Figure 3 - The roll-off at the end of that last peak combined with no production after the final pick-up alerted the well owner to a potential down-hole issue within hours of the problem occurring.

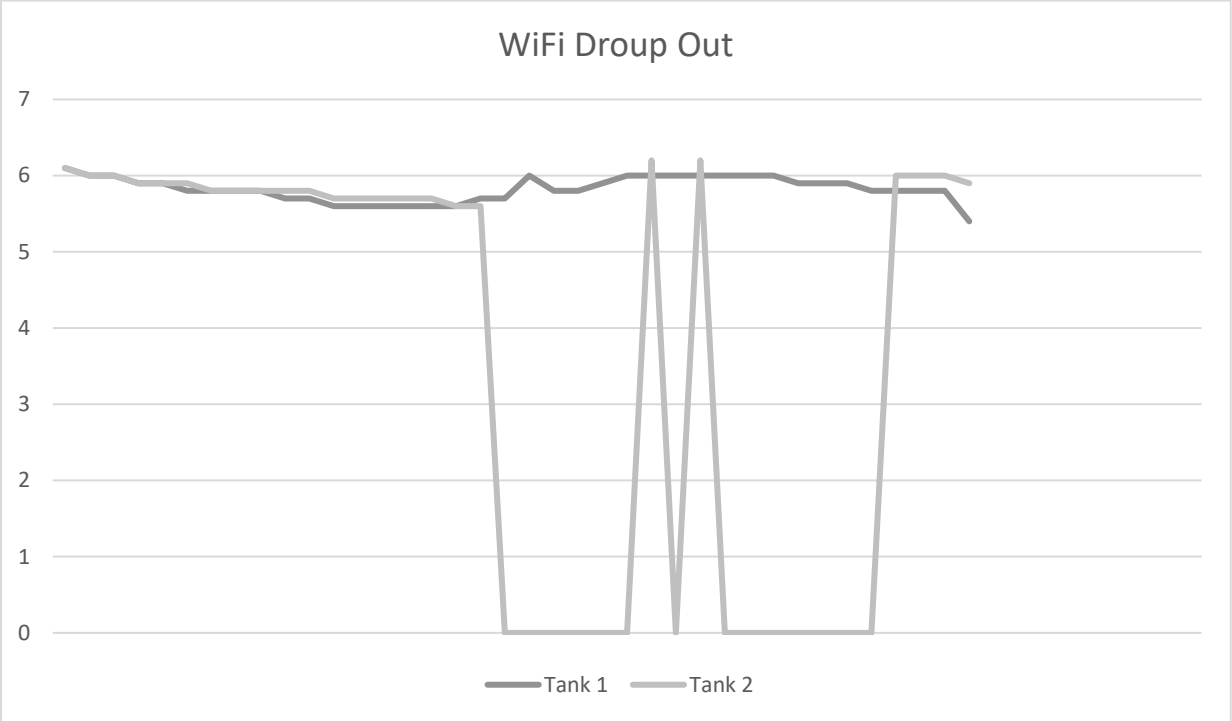

Figure 4 - Data analysis can show critical communications errors. Frac tanks 1 & 2 fluid levels were being monitored real time. Tank 1 was connected by Ethernet to the router while Tank 2 was connected by WiFi over an 18 inch gap. The RF interference was sufficient to block the WiFi signal, resulting in loss of data. The same issue has been observed with Bluetooth communications.

Example 1 – Cloud Services security rules for a database

The Google Firebase Realtime Data base has the ability to incorporate different security rules based on the user needs. The following examples are from Julio Marin at https://medium.com/@juliomacr/10-firebase-realtime-database-rule-templates-d4894a118a98

Rule Types
The rules have a JavaScript-like syntax that make it easy to understand and those comes in four types:

.read - Describes if and when data is allowed to be read by users.
.write - Describes if and when data is allowed to be written.
.validate - Defines what a correctly formatted value will look like, whether it has child attributes, and the data type.
.indexOn - Specifies a child to index to support ordering and querying.

Sample rules:

No Security
```
{
 “rules”: {
 “.read”: true,
 “.write”: true
 }
}
```

Full Security
```
{
 “rules”: {
 “.read”: false,
 “.write”: false
 }
}
```

Only authenticated users can access/write data
```
{
 “rules”: {
 “.read”: “auth != null”,
 “.write”: “auth != null”
 }
}
```

User Authentication from a particular domain
```
{
 “rules”: {
 “.read”: “auth.token.email.endsWith(‘@example.com’)”,
 “.write”: “auth.token.email.endsWith(‘@example.com’)”
 }
}
```
Example 1 continued

User Data Only
```
{
```

```
  "rules": {
    "users": {
      "$uid": {
        ".read": "$uid === auth.uid",
        ".write": "$uid === auth.uid"
      }
    }
  }
}
```

Validates user is moderator from different database location

```
{
 “rules”: {
 “posts”: {
 “$uid”: {
 “.write”: “root.child(‘users’).child(‘moderator’).val() === true”
 }
 }
 }
}
```

Validates timestamp

```
{
 “rules”: {
 “posts”: {
 “$uid”: {
 “timestamp”: {
 “.validate”: “newData.val() <= now”
 }
 }
 }
 }
}
```

Prevents Delete or Update

```
{
 “rules”: {
 “posts”: {
 “$uid”: {
 “.write”: “!data.exists()”
 }
 }
 }
}
```