Security of Cyber Physical Systems

Charlie Britton

Year 3, Semester 2, 2023

Contents

0.1 Administrivia . . . 2
0.1.1 Assessment . . . 2
0.2 Initial Definitions . . . 2
0.2.1 Hardware Issues . . . 3
0.2.2 Glitching Attacks . . . 3
0.3 Trusted Computing . . . 3
0.3.1 Protection Layers . . . 4
0.3.2 Trusting Things . . . 4
0.3.3 An Attacker's Trust Model . . . 4
0.3.4 Prevention Measures . . . 4
0.3.5 Definition of Trust . . . 5
0.3.6 Identification of Roles . . . 5
0.3.7 Trusted Computing Base . . . 6
0.3.8 UEFI . . . 6
0.3.9 Chain of Trust . . . 6
0.3.10 Root of Trust . . . 6
0.3.11 Evidence of Trust . . . 7
0.4 CPS: Cyber Physical Systems Security . . . 7
0.4.1 Why Security Matters . . . 7
0.4.2 CPS Layers . . . 7
0.4.3 Threats to a CPS . . . 8
0.4.4 Types of CPS Vulnerability . . . 9
0.5 Side Channel Attacks . . . 9
0.5.1 Cryptography Reminder . . . 9
0.5.2 Side Channel Information . . . 10
0.5.3 Simple Power Analysis (SPA) and Dynamic Power Analysis (DPA) . . . 10
0.5.4 Dynamic Power Analysis . . . 13
0.6 Lab 1 Clarification . . . 13
0.7 Trusted Execution Environment . . . 14
0.7.1 Confidential Computing Consortium . . . 14
0.7.2 Trusted Execution Technologies . . . 14
0.8 Healthcare . . . 15
0.8.1 Privacy & Security Goals . . . 16

0.8.2 Attack Types 16

Lecture 0: Introduction

2023-01-31T14:00

This module focuses on the types of attack and attacker on a range of systems and computing techniques to ensure security of the system. We look at side channel attacks and different types of wireless identification systems, such as Mifare, E-passports and other NFC systems. We also look at GSM and WEP and how these are weaker cryptosystems than their modern counterparts.

The other half of the module looks at physical security, wired and WiFi network security, infrastructure attacks and hardware trojans.

0.1 Administrivia

0.1.1 Assessment

There are 2 assessed labs, the first of which is looking at TPMs (trusted platform modules). This is due on the 6th March. We want to ensure that the underlying device is trusted such that we can cryptographically guarantee the device works. Details about the second lab have not yet been properly released by the module team. Each lab is worth 50% of the module.

0.2 Initial Definitions

Cyber physical systems are those that are concerned with the physical properties of a device and hardening it so that an attacker can't compromise something even when they have access to the system.

Difference betwwen trust, trustworthy and trusted:

- **Trust** –
- **Trustworthy** – The person or company has the potential to be trusted with the thing
- **Trusted** –

Definition 1. Security Certification – a standard of ensuring that the hardware and software a company provides complies with a set of standards such that the attack surface of the product is reduced.

We then discussed what recent attack there had been, such as stuxnet, heartbleed etc.

0.2.1 Hardware Issues

Meltdown is one of Intel's hardware vulnerabilities which allow attackers to use a side channel. It took 6 months for Intel to acknowledge the flaws within their architecture.

Although not a huge issue on dedicated hardware, these bugs allowed people on different VMs running on shared hardware to allow the attacker to jump between the isolation on different virtual machines.

Software is written a lot more than hardware and is often much more complicated than the underlying hardware. People will mostly exploit the bugs in software and the underlying hardware is much more immutable than the software.

0.2.2 Glitching Attacks

These are physical attacks where we have access to the raw hardware where we can perform an attack on the equipment. Glitching attacks include instruction skipping, malformed data read and writes and instruction decoding errors. You can either have non-invasive attacks which don't change the underlying hardware much but may include connections to test pads on the PCB.

Other techniques include modifying the IC packaging which can involve scraping layers off of the IC and will require a lot of knowledge.

Types of Attacks

Most of the clocks for the system have a rising and falling edge on the clock to synchronise the start and end of all instructions. Where we have an out of chip oscillator, then this clock must travel across the wire to get to the IC.

We can glitch the clock, power, thermals or apply some radiation to the board to trigger a glitch.

0.3 Trusted Computing

Lecture 1: Trusted Computing

2023-02-07T14:00

Trusted computing and security mean 2 very different things. The concept of trust can be applied to a system. We want to know how the hardware and firmware can interact to provide trusted execution.

On motherboards, there are lots of chips serving different types of functionality on the board. This lecture will look at the protection layers and how trusted we are with the computer.

0.3.1 Protection Layers

We have to make a decision of whether we trust the application. Is it possible to trust the application if we can't trust the OS? Can we trust the OS if we don't trust the firmware? Can we trust the firmware if we can't trust the hardware.

If there is an attack at a lower level, then is it possible for the layer above to detect this?

These layers are as follows:

- Trusted Application
- Middleware
- Operating System
- Firmware (e.g. BIOS)
- Hardware

0.3.2 Trusting Things

When connecting to a server from a phone, then we don't need to actually trust all of the infrastructure in the middle. Behind the scenes connecting to this server, we go through several different machines which manage our network and then we reach what we think is the server at the other end.

The software at the other end is trusted if we use something such as PKI with SSL/TLS. If we want to connect or use something insecurely, then we need to trust all of the parts in the chain.

0.3.3 An Attacker's Trust Model

The attacker has a goal of connecting to an asset. They will typically install some malware that will take over the OS. The OS then usually has access to all of the hardware. Whilst the OS usually has access to most things, some parts of the hardware are locked out to the OS. There is a special operation mode on most CPUs which allow the programmer to gain access to all of the hardware in special cases.

The lower level a malware attacks, the more access it will have. Bootkits and rootkits are the way that attackers can gain access to hardware at the firmware level and will be impossible to see from the operating system.

0.3.4 Prevention Measures

To prevent the attacker getting control of the device, we may provide digital signiatures on software that is distributed to ensure that applications are isolated. We can also provide access control on the asset to stop the attacker.

Additionally, providing a firewall can stop the attacker from connecting to the computer.

0.3.5 Definition of Trust

This is a hard thing to define. There are different definitions of trust. The TCG (Trusted Computing Group) provide the definition of trust as:

- An entity can be trusted if it always behaves in the expected manner for the intended purpose. (2004)
- 1. It can be unambiguously identified 2. It operates unhindered 3. The user has first-hand experience of consistent, good, behavior or the users trusts someone who vouches for consistent, good, behavior (2010)
- A platform that uses Root-of-Trust to provide reliable reporting of the characteristics that determine its trustworthiness (2017)

As we can see, the definition of trust is constantly growing and changing to adapt to modern times. Other definitions we need to know are:

- **Attestation** – The process of vouching for the accuracy of information. External entities can attest to shielded locations, protected capabilities, and Roots of Trust. A platform can attest to its description of platform characteristics that affect the integrity (trustworthiness) of a platform. Both forms of attestation require reliable evidence of the attesting entity.
- **Trust** – Trust is the expectation that a device will behave in a particular manner for a specific purpose
- **Measurement** – The process of obtaining the identity of an entity. Normally this is a cryptographic hash
- **Security** – A condition that results from the establishment and maintenance of protective measures that ensure a state of inviolability from hostile acts or influences

Trust is action-specific. We can confuse trust and security quite easily. People can be trusted to do one thing but not another. It is not a one-time decision and we can rethink the decisions we make about the trust of the machine.

0.3.6 Identification of Roles

Who

The user has roles and restrictions that need to be enforced. They power on the computer and provide their credentials.

The platform is the different machines in the stack that have roles and responsibilities.

The software is knowing the identity of it whilst executing and help to stop malware and enforce user access policies.

How

Users are authenticated in different ways (something you know, have and are). The platform is given with persistent identifiers that identify it. In terms of software, this is the measurements that create fingerprints of the binary/config and this stops malware hiding.

0.3.7 Trusted Computing Base

This is the smallest possible subset of hardware and channels that we use, including a verified OS kernel to try and ensure that we trust everything that is being used.

0.3.8 UEFI

UEFI is the first stage of the bootloader and can be about 2 MB. The OS is then on top of this and can be in the order of millions of lines of code, which we have to trust.

0.3.9 Chain of Trust

The chain of trust is based on the trust model that we introduced earlier to extend the trust from the hardware to the software.

The chain is created by verification of the previous stage. For example, when powering on a device, it will create 'Entity A'. This will then be verified by 'Entity B' which will then create a link in the chain. Once we get to the component that we want to launch (such as an application), then we can trust the component because we trust the previous component.

Example. The chain of trust in a typical computer starts with the Intel ME (which is the management engine provided by Intel. This is then trusted by the BIOS which is usually UEFI. Then the UEFI launches grub which loads the operating system. Once the OS is loaded, we typically stop trust at the application level.

0.3.10 Root of Trust

In real systems, the industry adds a hardware root of trust which always behaves in an expected manner.

in the root of trust we store a security version. In 2017, the first intel me was released with a security version of 0. If a bug was found, we increase the security version by 1.

0.3.11 Evidence of Trust

We measure the firmware and put data in a hash to verify it. We collect evidence during the boot and then report on them at a later point. We may also have a verified boot which verifies that each measured component is appropriate for the boot policy.

If the verification fails boot stops immediately but in measured boot then the platform continues to run until the platform fails.

0.4 CPS: Cyber Physical Systems Security

Lecture 2: Cyber Physical Systems Security – An Intro

2023-02-14T14:00

CPS is a network of cyber physical devices that systematically control and interact with the network of physical, computational components and humans.

The key parts of a CPS is the collection of data from sensors, sharing of that data via some sort of communication network, analysis of that data, generation of the control commands for that system and then controlling or guiding the physical domain via actuators.

There are many different types of CPS. These can range from smart homes (industrial consumer IoT), to oil refineries, smart grids, water treatment, medical devices or smart cars. These are all controlled typically by a SCADA–a supervisory, control and data acquisition system.

0.4.1 Why Security Matters

There are so many domains and applications of IoT in today's world, with many billions of connected devices on a steady upwards trend. There is also a huge amount of data being generated from these devices.

0.4.2 CPS Layers

There is a taxonomy of the layers of a cyber-physical system. These can be split into the following:

- **Perception Layer** – The data collection and information, such as sensors, actuators, RFID tags and GPS. These are vulnerable to eavesdropping, port scanning and passive replay attacks. These would fall under the confidentiality, privacy and authentication targets.
- **Transmission Layer** – This is where the collected data is sent to the remote server for analysis. It is vulnerable to MITM, DoS, repudiation and replay attacks. Need to focus on the confidentiality, integrity, availability and authentication.

- **Application Layer** – This is where we analyse the data and make the necessary changes to the system. This layer is vulnerable to malicious code, botnets, trojans, worms and buffer overflows. The target of these is privacy, security, safety and authentication.

Components such as sensors will talk to an aggregator which collects all of the values. This will then make its way to a PLC (programmable logic controller), which can then adjust actuators. The management of the PLC is done from a distributed control system, which is on a management network often air gapped from the internet.

5 Principles of Security

These are the confidentiality of the data, integrity, availability, authenticity and non repudiation of the data.

As a reminder, non repudiation is the situation where the author cannot dispute the authorship or validity of a contract of the data.

0.4.3 Threats to a CPS

CPS can often have safety-critical functions that must always work or the system must go into a fail-safe state. When designing a system, the following threats need to be considered:

- **Wireless Exploitation** – it is very easy to eavesdrop and transmit interfering signals on the wireless spectrum. You can stop the system from operating correctly by jamming the frequency or listening in to the signals transmitted and transmitting your own at a later point in time.
- **Reconnaissance** – Performance of operations targeting a nation's computational intelligence and industrial control systems, typically through a malware that gets spread to the systems.
- **Remote Access** – a user gains access to the system remotely. Typically much easier than a physical attack and has less consequences if caught.
- **Information Gathering** – especially important in households, where individuals may be having private conversations or in a business where there may be IP which needs protecting.
- **Physical Damage** – communications over a wire are still susceptible to being tapped by a 3rd party.
- **Spoofing**
- **Tracking**

0.4.4 Types of CPS Vulnerability

There are 3 types of vulnerabilities within a CPS. These are network vulnerabilities, where an attacker can do something on the network; platform vulnerabilities such as the hardware, software, configuration and database; and management where there are a lack of security guidelines, procedures and policies in place.

Up to slide 12

0.5 Side Channel Attacks

Lecture 3: Side Channel Attacks

2023-02-28T14:00

Side channel attacks can take many forms and can be as a result of physical attacks or software that is being run on the same machine. In this lecture, we focus on the attacks that are relevant to physical attacks.

When we have access to the device, we can attach probes to wires on PCBs and components. Using a probe, we can connect to to a device such as a HackRF One, and by putting the probe in close proximity, we can tune the frequency we listen in on. This will give us some form of trace.

A device like this can work on something that runs at lower frequencies (such as a microcontroller, or a lower end CPU in the 100 MHz region. EM devices such as the HackRF are one type of side channel attacks.

We can also do voltage or power analysis, looking at simple power usage or differences in different parts of the circuit. Sometimes these attacks can be active, e.g., by manipulating the frequency or voltage.

Most cryptography is done on the TPM if one exists, otherwise the device makes use of the processor for operations. Neither the TPM nor the processors we use are resistant to most types of side-channel attacks.

HSMs are the more advanced version of a TPM and are used in things such as CAs for generation and storage of the root certificates.

0.5.1 Cryptography Reminder

AES: Advanced Encryption Standard

This is the main cipher which is used for encryption of content. This is a form of symmetric encryption where both parties have the key. The initial key is obtained through TLS and asymmetric keys initially.

We separate the data blocks into 128 bit lengths. We have a key length of 128, 192 and 256 bits, with a corresponding number of encryption rounds. The key length is the part of the algorithm that varies in addition to the plaintext we want to encrypt.

Each round of AES is identical but uses different constants in operations. Each round substitutes bytes, shifts rows, mixes columns and adds a round key. There are several rounds applied successively and the output is the ciphertext.

The cipher is completely transparent in that the algorithm and constants are open.

RSA

RSA is an important asymmetric encryption scheme, where you are free to distribute a public key and retain a private key for verification and decryption of messages encrypted with the public key.

Setup of RSA involves 2 primes p and q. We compute $n = pq$, then select d and e such that d is relatively prime to $(p-1)(q-1)$ and $ed \mod ((p-1)(q-1)) = 1$.

We then discard p and q, and distribute the public key (e, n) and retain the private key as (d, n). To encrypt and decrypt, we use the following:

$$C = M^c \mod n \tag{1}$$

to encrypt and

$$M = C^d \mod n \tag{2}$$

to decrypt.

0.5.2 Side Channel Information

We can capture each round of encryption from a voltage trace of the AES encryption. If we see a trace with 10 spikes, then there are 10 rounds of encryption and we know that the encryption is AES-128.

For an RSA trace, we will have a similar set of peaks, with different traces dependent on the multiply operations and the square operations. Dependent on the number of peaks we see, we can find out how many bits are being used.

Unlike other parts of the device, when we are running cryptographic operations, we will use a lot more power. For example, when starting up the EEPROM as seen in figure , we have the power trace of when the EEPROM is started and when it starts to send data. We can find the encryption timing based on the power usage.

add figure

0.5.3 Simple Power Analysis (SPA) and Dynamic Power Analysis (DPA)

These are more sophisticated and powerful analysis tools. These are done by a cryptanalysis expert to extract the keys from cryptographic devices.

The attacks for SPA can be mounted quickly and SPA attacks can take as little as a few seconds. A DPA attack can take several hours, where the device needs

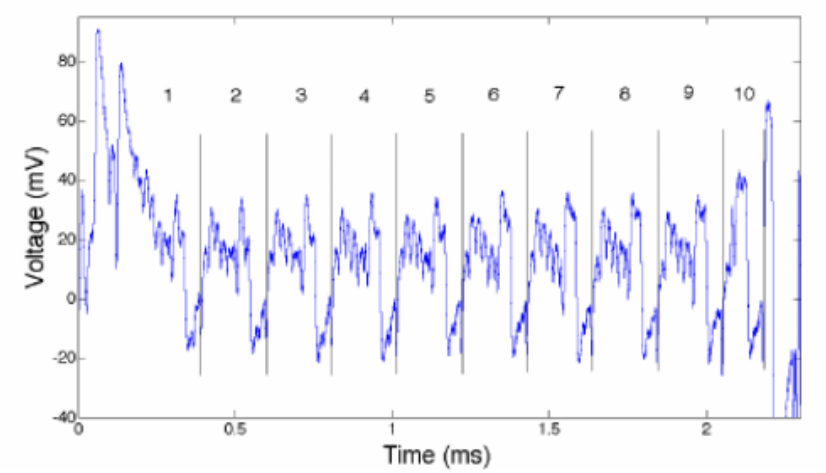

Figure 1: AES-128 Trace

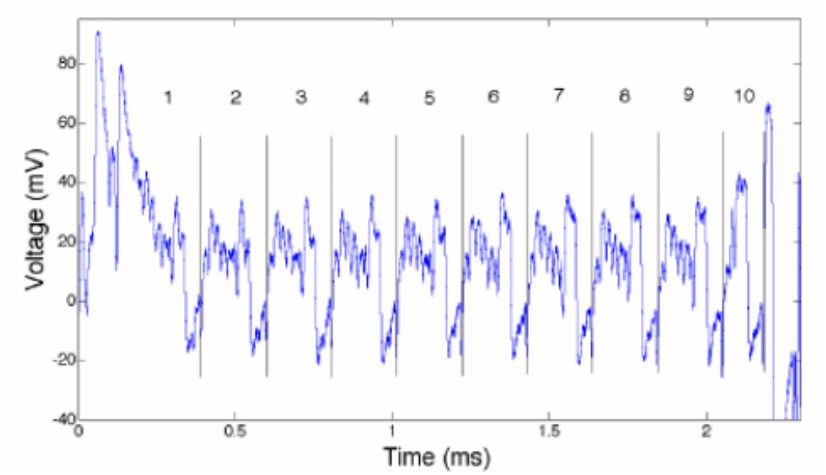

Figure 2: RSA Trace

to have a large number of traces in order for the device to be broken. For example, a device that takes 1M traces to break, it is much more secure than 1 trace.

The implementation of DPA is different company to company and the information is proprietary and not based on open research.

The main idea is that the detection and leakage from some part of cryptographic module can be done by attacking the module by taking inputs and outputs from the device in addition to various probes such as EMI and power usage to figure

out the keys by looking at common patterns and how the input affects the output.

Capturing EMI from Devices

We target specific frequencies e.g., the CPU, ALU, CPU extensions, serial or EEPROM which operate at different frequencies and then by tuning to the same frequency, we get different operations.

At a lower level, we can see instruction differences, timing differences and differences in the switching of different paths in the circuit.

To perform an attack, we pick the probe to used based on the type of attack, analyse the spectrum to pick frequencies and good probe positions. We then do some analogue processing to filter the signal, which we then capture and transfer to a larger machine. Finally, we do offline processing of the traces to try and figure out what we are looking at.

Single Trace Analysis

When doing an SPA, we start by looking at a single trace and try to find the start of the trace. We try and identify the start and the number of spikes. If we have 10 spikes, then it is likely to be 128 bit AES.

If there are a lot of XORs happening, then we look at what ciphers use a lot of XOR. For example, we can see how many rounds are done across a time and the distance between each peak.

Prevention of SPA

We avoid hard wiring specific algorithms and add a lot of dummy noise, e.g., add multiple dummy XOR operations, move registers around or do some shifting of various bits.

The idea is to not change the operation of the underlying cipher but add some useless parts to what is done to try and mask the actual algorithm. If we try and do the analysis in software, this is much more deterministic and it might be quite hard to hide what is happening.

Establishing the Algorithm from the Trace

very quickly went through in the lecture

Chosen Cipher Analysis

We use a specific input to search for the leaks. Where the behaviour differs in different parts of the input, we can establish that a change in a bit of the input causes a specific spike in the output trace.

Trace Pairs

By looking at the outputs from different traces and how a message difference leads to a different trace, we can reduce the number of searches that we need to bruteforce, then every bit we can determine from a power trace allows us to reduce the attack timeframe, halving with each bit of the key we know.

It is imfeasible and very expensive to do AES attacks by bruteforce but if we reduce the number of bits then it becomes feasible.

0.5.4 Dynamic Power Analysis

Whilst SPA provides that data for different ciphers, DPA requires us to capture enough traces to create a differential model that guarantees that a change in 1 bit in the output will cause a spike in the output of the device.

We have to measure many times and the number of required traces to get the information from the device.

Example. We feed aj input into a cryptographic model. The device has a 500 DPA rating and we change the input at bit 13 and generate a lot of inputs. If we generate 1000 traces where bit 13 is set to 0 and 500 where the bit is 1.

We can extract the power traces for just setting the bit to 0/1 without changing anything else. Based on these 1000 traces, we can see what parts of the power trace may leak this data. We can then take the averages for each of the sets of 0/1 respectively and then we can take the differences on the graph. Where there are spikes, this reveals the power leaks where the bit changes as confirmed by statistical analysis.

If the trace changes in the same way for lots of different bits, then this isn't very useful to us and we can't deduce much information from the model. If the power trace only changes for 1 bit, then we can be much more sure that changing a bit affects the output.

Lecture 4: Trusted Execution Environment 2023-03-07T14:00

0.6 Lab 1 Clarification

After initialisation of the TPM, when we create the primary, the `e` flag means that this is an endorsement hierarchy. This primary context is for endorsement only and is the main key that is the root of trust.

We then create subkeys for other operations and use the main endorsement key to create an RSA keypair which is restricted in its use.

The `fixedtpm` and `fixedparent` flags prevent us from exporting the key to another place. The `decrypt` flag sets the key such that it can only be used to decrypt.

When we sign the file, we only sign the hash of the program instead of signing the whole program. We guarantee that the binary is not altered through the hash but save signing the whole file.

The PCR extend function ensures that we have a proper chain of trust and that the data written is in the correct order, such that anything that comes after cannot overwrite what has happened previously. By appending and hashing values onto the TPM, we can cryptographically ensure that the chain of trust is intact.

0.7 Trusted Execution Environment

This is hardware that guarantees that the environment is safe. It is especially important for VMs that run in the cloud. When using a computer, we are trusting the hardware, firmware, operating system (or hypervisor), the operating system, a middleware and the apps.

We need to ensure that all of these aspects can be trusted and there is motivation to ensure that we limit the trust in an environment to as few people as possible.

The TEE is aimed at removing trust from the BIOS, hypervisors and virtual machines by ensuring that all of the software vendors cannot intercept nor tamper with the data that we are using for execution.

We create a trusted computing base, which takes into account the number of lines of code that we are trusting and then limit the amount we trust as much as possible. From when we start using this base, we want to ensure that the software is specifically isolated from the rest of the system. We can encrypt the memory to ensure that it cannot be read by anyone else in the system.

Similar to how the secure boot and TPM provide quotes, we can make use of quotes to provide a guarantee that the code we are running is trusted.

0.7.1 Confidential Computing Consortium

This is a buzzword and now a technology/consortium that was setup by mainly Microsoft to allow new public cloud where we have extremely sensitive data and also multi-party sharing in a secure way.

Confidential computing is ensuring that data in use by a program is secure and they assign data in transit and data at rest as being managed by something else and is not a part of the confidential computing consortium.

0.7.2 Trusted Execution Technologies

There are lots of technologies for trusted computing on the market, with Intel SGX being the first technology to provide. AMD followed with SEV, then

RISC-V, ARM and IBM also have secure environments which aren't as widely used.

SGX provides a separated execution for software partitions that prevent the OS and other parts of the system from accessing the base of the TCB. SGX provides hardware features to create enclaves which the software can use for transit of data to and from the CPU in an encrypted way such that the hypervisor can't get anything useful.

We divide the application into a trusted and an untrusted part, where the application creates the enclave that is protected. Only the part of the application that wants access to the enclave is allowed to access the enclave.

Intel SGX makes use of AES GCM 128 encryption, which provides the encryption for the memory in addition to a message authentication code which allows the encrypted message to be authenticated. The data authentication is stored in a Merkel tree which allows us to ensure that all of the data is secure.

The equivalent of the quote method of the TPM is the report function. In SGX, we have hardware that enforces the access to the data.

Finish slides possibly

0.8 Healthcare

Lecture 5: Cyber Physical Systems in Healthcare

2022-03-14T14:00

Healthcare is a huge area of cyber physical systems with lots of real-world attacks that can be done with potentially life threatening consequences. Healthcare is a multiple-trillion dollar industry, with lots of spending on both actual costs and waste costs. Making systems smarter can reduce the total overall cost and staff needed.

Healthcare systems continuously monitor and control vital life systems for patients and are embedded systems with sensing and communication built in. There are also WBANs (wireless body area network) which are a fundamental type of healthcare CPS.

We have 3 classes of CPS, with class 1 being things such as bandages or floss, class 2 being things such as electric wheelchairs and class 3 being things such as pacemakers.

Sensors can be on or in patients and form a part of the device. The device then communicates to some processing server through a network component, which looks through the data and then alerts and advises the healthcare provider.

By integrating more sensors into the body, we can build a better picture of the patient for the healthcare provider to use when diagnosing and treating the patient.

Goal	Tier 1	Tier 2	Tier 3
Authentication	✓	✓	✓
Non-repudiation	✓	✓	✓
Availability	✓	✓	✓
Device Anonymity	✓	✓	
Unlinkability	✓	✓	✓
Integrity		✓	✓
Data Anonymity		✓	✓
Communication Anonymity		✓	✓
Confidentiality		✓	✓

Table 1: Comparison of Privacy and Security Goals

0.8.1 Privacy & Security Goals

Within healthcare, we have three tiers of security and privacy goals, which are compared in Table 1.

A Tier 1 device may be invasive or non-invasive and is the device that the patient has on their person. A tier 2 device is the smartphone or computer that the data gets sent to to then be sent to the tier 3 device which is the health server or doctor and hospital.

0.8.2 Attack Types

There are many different ways to attack a device. We can modify the hardware of the device, work to make the device unavailable, sniff data from the device, modify the data that is sent to/from the device or leak information.

For hardware attacks, we can insert Trojan chips at the time of manufacture, which can then allow us to modify the logic on the controller or disable and enable certain bits of hardware.

We can also have software attacks, where we attack the software on the machines. There was an attack called Conficker, which affected X-ray machines, mammography and a gamma camera for nuclear medicine. It used a flaw in Windows and created a botnet.

We also had Wannacry, which was a ransomware which affected the NHS in a big way.

Up to slide 12

0.9 Smart Meters and Energy Theft

Lecture 7: Smart Meters and Energy Theft

2023-03-21T14:00

The power system is a large system which starts at some power plant, then is transmitted and distributed to end users. These users may be factories, facilities

such as swimming pools, offices, homes or end user appliances.

Traditional grids are being transformed into smart grids, with smart appliances that can shut off if needed, demand management–allowing some appliances to be used at off-peak times for the grid.

We also have processors, which can execute protection mechanisms for the grid quickly, storage providers to allow for storage of energy at off peak times to then be used when the grid is in high demand. The smart grid also has a network of sensors, which can detect for disturbances and fluctuations to isolate areas of the network.

0.9.1 Motivation for Smart Meters

90% of the power outages are caused by issues in the distribution networks. If we improve the distribution network, then we can have fewer interruptions to the network. Additionally, we now have smart meters, which allow for us to collect and aggregate the meter readings from all smart meters without the need for a meter reader to come and collect the readings.

With the advent of these smart meters, we do increase the attack surface for electricity theft as the meters themselves are not checked as often for bypass devices. Attackers may also spoof the meter readings from the device to send false readings instead of allowing the meter to send a true value.

We can have AMR (automated meter reading) and AMI (automated meter infrastructure), both of which are capable of being hacked. The infrastructure for AMI includes the smart meters which communicate their readings and the account to a base station aggregator, which then sends the readings to the utility provider.

In 2017, more than $96B was lost to the grid from non paying customers, with all consumers having to pay higher tariffs to make up for the defecit from the stolen electricity. This worked out as about 30 Euros per customer for electricity theft.

If metering reports lower usage, this can cause the utility companies to generate less electricity than is needed, resulting in worse power quality. Due to these losses in revenue, businesses are also encouraged to spend more on metering for their infrastructure, which costs the consumer more.

0.9.2 Smart Meter Functions

Smart meters are divided into primary electronic meters, AMR enabled smart meters and AMI enables smart meters.

All meters have a circuit to measure voltage and current and will be connected to a power supply on the grid operators side. There is then a microcontroller that communicates the current reading over an LCD/LED and stores the current

value to an EEPROM. AMR-enabled meters can then communicate the current value of the meter to the grid at given intervals, allowing the utility provider to see the usage over a given period.

On top of this, AMI-enabled meters can do net metering, where they calculate both imports and exports to accommodate those who may have solar panels, for example. AMI-enabled meters also have remote disconnect and can charge users different tariffs based on their grid usage at a specific time.

0.9.3 Attacks on Smart Meters

Measurement Level

Meters use transformers to measure the currents. You can alter transformers by adding magnets outside the meters to reduce the bill by 50-75%/ Wou can also lower the meter readings by decreasing the voltage/current/power factor of the circuit outside the meter.

Storage/Calculation Level

Adversaries can create a connection between a meter and an optical converter device. They can then tamper with consumption recording settings using software which is downloaded from the internet.

At all levels, we have possibilities for measurement to be interrupted by bypassing the physical tamper protections on the meter. We can also extract the passwords over optical communication. Meter storage can be tampered with and firmware can be changed. Communications can be intercepted and false readings can be reported to the backhaul nodes.

Consumption Attacks

These can be either major difference attacks, where the reported readings are much smaller or minor difference attacks, where they alter there readings to slightly underreport.

Another type of consumption attack can include load profile shifting, such that the attacker changes their recorded usage so there is no net change but such that they report their usage at a time when the pricing is lower.

Another way the attacker can evade detection is if they tamper with their neighbour's metering such that the neighbour picks up their deficit and over reports their usage. This would make the attacker much harder to detect as there is no loss at the provider level. Attackers may also do a combination of the above and at intermittent timings such that they are much harder to detect.

Pricing Attacks

The attacker may also compromise the pricing structure of the provider, usually at the server that does the accounting and billing, which can either be external to the company or may be done by bribing an employee with access to the tariffs for their job.

0.9.4 Countering Attacks

Machine Learning

We can make use of all of the data provided to us from various meters and process that data to build a model of what consumption should look like. We then apply the model, which is trained on large amounts of historical data to the new data and see if there is any anomalies.

We can use a support vector machine (SVM) to learn the features as an ML model for the customer. We can use CNNs too, or LSTM-based CNNs.

ML models can have issues with the data quality, can take a long time to detect and have issues with the thresholds at which to possibly trigger an investigation.

Statistical Methods

We can also make usre of Bollinger bands, which provide a moving average based on the historical data. The upper and lower bands are a significant number of standard deviations from the historical data. If the consumption is below or higher than is usual, then there is a potential issue.

Measurement Mismatch Based Detection

This is where we use lots of different sensors in the grid and take the user reported consumption. We measure the energ distributed to several users on the grid and compare the measurements at different points in the network to see if they add up. We have to take into account the losses in transmission but any large differences can indicate possible theft.

We can make use of dynamic programming to calculate the attack probability for a user for the year. If the user is likely to be an attacker, then we can install an FRTU (fully remote terminal unit), which provides the metering at a remote location.

Polynomial Approximation

This is a behaviour based method that uses a centralised observer's meter. We denote user's meters as $1, 2, \ldots, n$, with meter i belonging to a user. We take also E_j, which is the central observer meter's measurement at a time period j. We know that $E_j = \sum_{i=1}^{n} e_{i,j} = \sum_{i=1}^{n} \sum_{k=q}^{0} a_{i,k} x_{i,j}^{k}$. $a_{i,k}$ are unknown and $e_{i,j}, x_{i,j}$ denote user's i actual and reported consumptions at period j.

We can get the equations for different periods to get an approximation of the client's behaviour.

All of these methods have issues in that they are expensive to deploy and have issues with threshold selection.

0.10 UEFI and Secure Boot

T

his section has been added because it was all still very much a mystery to me how it worked and made it quite hard for me to understand what my coursework was asking me to do.

First, a couple of quick definitions and clarifications:

Definition 2. The BIOS is the legacy system that was established by IBM for getting the firmware of the system to boot into a bootloader.

Definition 3. MBR is the master boot record and is a partitioning scheme used for hard drives. It is older and thus more limited than GPT.

Definition 4. UEFI is the unified extensible firmware interface and is designed as the successor firmware specification to the original BIOS which uses the MBR.

Definition 5. The GPT partition table is a partitioning scheme for disks that is more flexible than the MBR and is used with UEFI.

Definition 6. CSM/legacy boot is a way for the UEFI firmware to boot using a BIOS/MBR style boot so that disks formatted for BIOS style booting can still be booted in a more modern system.

Definition 7. Secure boot is the process of using a TPM and keys to sign a boot process and ensure that the integrity of the bootloader remains intact.

When a computer starts, it performs a POST which ensures that the devices are installed correctly and that they work. Following the POST, the UEFI-compliant firmware is loaded from the motherboard somewhere and then we can possibly check the integrity or authorisation status of the bootloader.

Some systems choose to lock down the bootloaders that can be started from the

UEFI-compliant firmware interface by ensuring that they are signed, as Apple does with the iPhone to make sure that only their bootloader can run.

Another interesting gripe with the systems is that many motherboard vendors still refer to the UEFI-compliant firmware on the board as the BIOS, despite it not being the BIOS at all and just the motherboard's firmware.

When booting a device from a UEFI-compliant board, we set the boot order, which is stored to NVRAM. This boot order can be changed from within the OS that is booted provided that the OS is booted UEFI-native or in a UEFI-compliant way. Operating systems and bootloaders that are started in legacy BIOS/CSM mode may not access the EFI variables banks of NVRAM on the motherboard.

These variables can be managed by the underlying OS during an install automatically or can be manipulated (at least in Linux) by running the `efibootmgr` command.

Some implementations of the UEFI-compliant firmware may allow a user to enable or disable secure boot from within the UI but an important thing is to not allow it to be changed from within the OS or any way that doesn't require physical user interaction, otherwise an exploit that needed to alter the firmware could potentially just disable secure boot.