Embedded Systems Security: A perspective from the battleground

IR. ERIC VISEUR

DECEMBER 5TH, 2017

Bibliographic references

[1] ULg - Cryptographie et Sécurité informatique (INFO0045-2) - http://www.montefiore.ulg.ac.be/~dumont/pdf/crypto.pdf

[2] ISO/IEC 27000:2016 - Information technology — Security techniques — Information security management systems — Overview and vocabulary - http://standards.iso.org/ittf/PubliclyAvailableStandards/c066435_ISO_IEC_27000_2016(E).zip

[3] Reverse Engineering of Chiasmus – Hands on COM - https://prezi.com/bzyvzzdsxtkm/ubicrypt-chm/

[4] Ubuntu Forums hack exposes 2 million users - http://www.zdnet.com/article/ubuntu-forums-hack-exposes-two-million-users/

[5] Intel Data Protection Technology with AES-NI and Secure Key delivers fast, affordable data protection and security - https://www.intel.com/content/www/us/en/architecture-and-technology/advanced-encryption-standard--aes-/data-protection-aes-general-technology.html

[6] Intel AES-NI Performance Testing on Linux/Java Stack - https://software.intel.com/en-us/articles/intel-aes-ni-performance-testing-on-linuxjava-stack

[7] Raspberry Pi 3 Model B - https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[8] Cortex-A53 - https://developer.arm.com/products/processors/cortex-a/cortex-a53

[9] ARM Cortex-A53 MPCore Processor Cryptography Extension Technical Reference Manual - https://static.docs.arm.com/ddi0501/f/DDI0501.pdf

[10] Mobile Processor Exynos 8 Octa (8890) - http://www.samsung.com/semiconductor/minisite/Exynos/Solution/MobileProcessor/Exynos_8_Octa_8890.html

[11] USB Armory - https://inversepath.com/usbarmory

[12] i.MX537 Processors - https://www.nxp.com/products/microcontrollers-and-processors/arm-based-processors-and-mcus/i.mx-applications-processors/i.mx-mature-processors/multimedia-applications-processors-hd-video-high-end-advanced-applications-arm-cortex-a8-core:i.MX537

[13] Analysis of Lightweight Cryptographic Solutions for Internet of Things – http://www.indjst.org/index.php/indjst/article/download/98382/71755

[14] The Tiny Encryption Algorithm (TEA) - http://143.53.36.235:8080/tea.htm

[15] Botnets: Le retour des morts-vivants… connectés - https://blog.emsisoft.com/fr/2017/05/23/botnets/

[16] The Horus scenario - https://horusscenario.com/

[17] Securing the Internet of Things: Mapping Attack Surface Areas using the OWASP IoT Top 10 - https://www.owasp.org/images/5/51/RSAC2015-OWASP-IoT-Miessler.pdf

Who am I?

THALES

Seminar objectives

Gain knowledge over general concepts

- Back to basics: What is information security?
- Cryptographic algorithms families and evolutions
- Embedded systems specific challanges
- Modern threats
- A quick detour in the world of Industrial Control Systems

Understand the overall security analysis workflow: Risk analysis, EBIOS, Common Criteria

Practice over the theoretical concepts

Seminar organisation

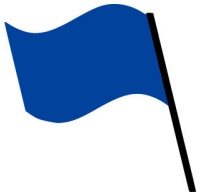
Theoretical sections, divided per concept and finished with a quick quiz

Most sections are followed by challenge sessions

- Done in groups of 2-3 students
- The answer to each challenge must be submitted to me directly. Play the game, don't yell your answers!

The whole course is a Capture the flag competition:

- Each correct answer earns you a certain amount of points
- The later you submit, the less points you get
- **The best team gets a prize!**



We have a **tight planning**, so let's not lose time !

Challenge: Let's get started!

I found a long lost treasure map in some flea market during a travel for work and uploaded it to:

https://treasure.ev1z.be/

On the back of the map, some strange markings were present:

> *Walk down the source, far below the headers of the island. At the map's tag, go back up your steps to find a cavern. Grab what you find and head back on your route towards the mountain. There you will realize you went way too far and the final clue was in a lighthouse you passed long ago.*
>
> *Use the tool you found in the cavern on the hidden message of the lighthouse to discover the flag.*

Open the map and try to make sense of these instructions… There's a treasure flag hidden somewhere!

(Clue: There's no crypto involved in this, it's warm up !)

Section 1
Information Security

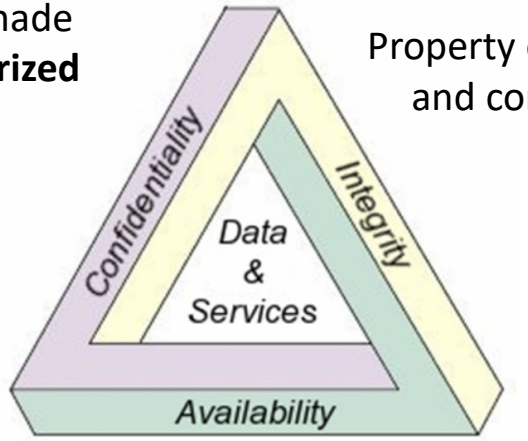
GENERAL DEFINITIONS

Note: This section is heavily based on [1]

Information Security

ISO 27000:2016 definitions [2]: *information security is the preservation of **confidentiality**, **integrity** and **availability** of information*

Property that information is not made **available** or disclosed **to unauthorized** individuals, entities, or processes



Property of **accuracy** and completeness

Property of being **accessible** and usable upon demand by **an authorized entity**

Confidentiality

Confidentiality is breached when information is made available to **unauthorized** parties

Breaking the confidentiality is a **disclosure**

Equifax data breach: What happened

Equifax, one of the nation's three main credit reporting agencies (the other two are Experian and TransUnion), announced on September 7 it was the victim of a major hack that exposed the personal information of 143 million U.S. consumers — or two-thirds of all Americans with credit reports.

According to Equifax, hackers exploited a security vulnerability in a U.S.-based application to gain access to consumers' personal files. The company has not yet said which application or which vulnerability was the source of the unauthorized breach.

Plus, the threat isn't necessarily immediate. Criminals have the information they need and they can decide to use it whenever they want — whether it be weeks, months or years from now. This is why it is so important that all Americans take the right precautions now!

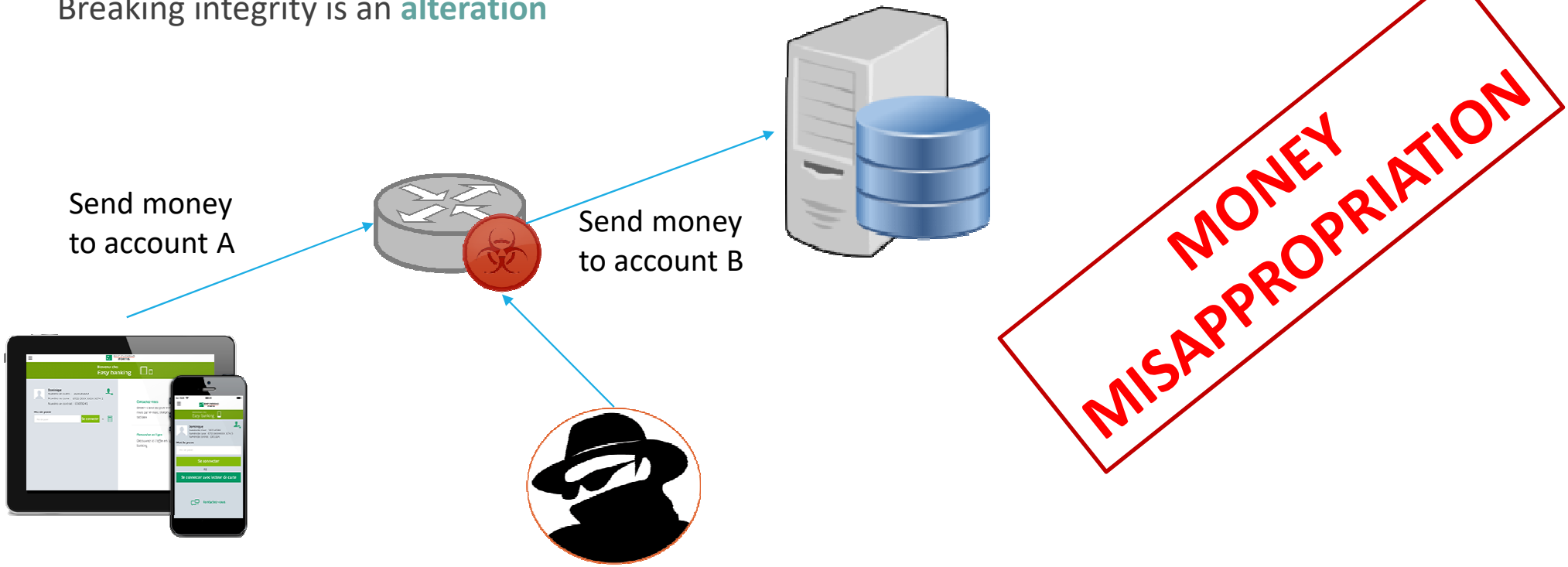
What personal information was stolen in the hack?

Hackers were able to gain access to consumers' names, Social Security numbers, birth dates, addresses and, in some cases, driver's license and credit card numbers.

Integrity

Integrity is breached when information is **transformed** by unauthorized parties

Breaking integrity is an **alteration**



Availability

Availability is breached when information **cannot be accessed** anymore by its **rightful** viewers

Breaking integrity is a **disruption**

LILY HAY NEWMAN SECURITY 10.21.16 01:04 PM

WHAT WE KNOW ABOUT FRIDAY'S MASSIVE EAST COAST INTERNET OUTAGE

FRIDAY MORNING IS prime time for some casual news reading, tweeting, and general Internet browsing, but you may have had some trouble accessing your usual sites and services this morning and throughout the day, from Spotify and Reddit to the New York Times and even good ol' WIRED.com. For that, you can thank a distributed denial of service attack (DDoS) that took down a big chunk of the Internet for most of the Eastern seaboard.

This morning's attack started around 7 am ET and was aimed at Dyn, an Internet infrastructure company headquartered in New Hampshire. That first bout was resolved after about two hours; a second attack began just before noon. Dyn reported a third wave of attacks a little after 4 pm ET. In all cases, traffic to Dyn's Internet directory servers throughout the US—primarily on the East Coast but later on the opposite end of the country as well—was stopped by a flood of malicious requests from tens of millions of IP addresses disrupting the system. Late in the day, Dyn described the events as a "very sophisticated and complex attack." Still ongoing, the situation is a definite reminder of the fragility of the web, and the power of the forces that aim to disrupt it.

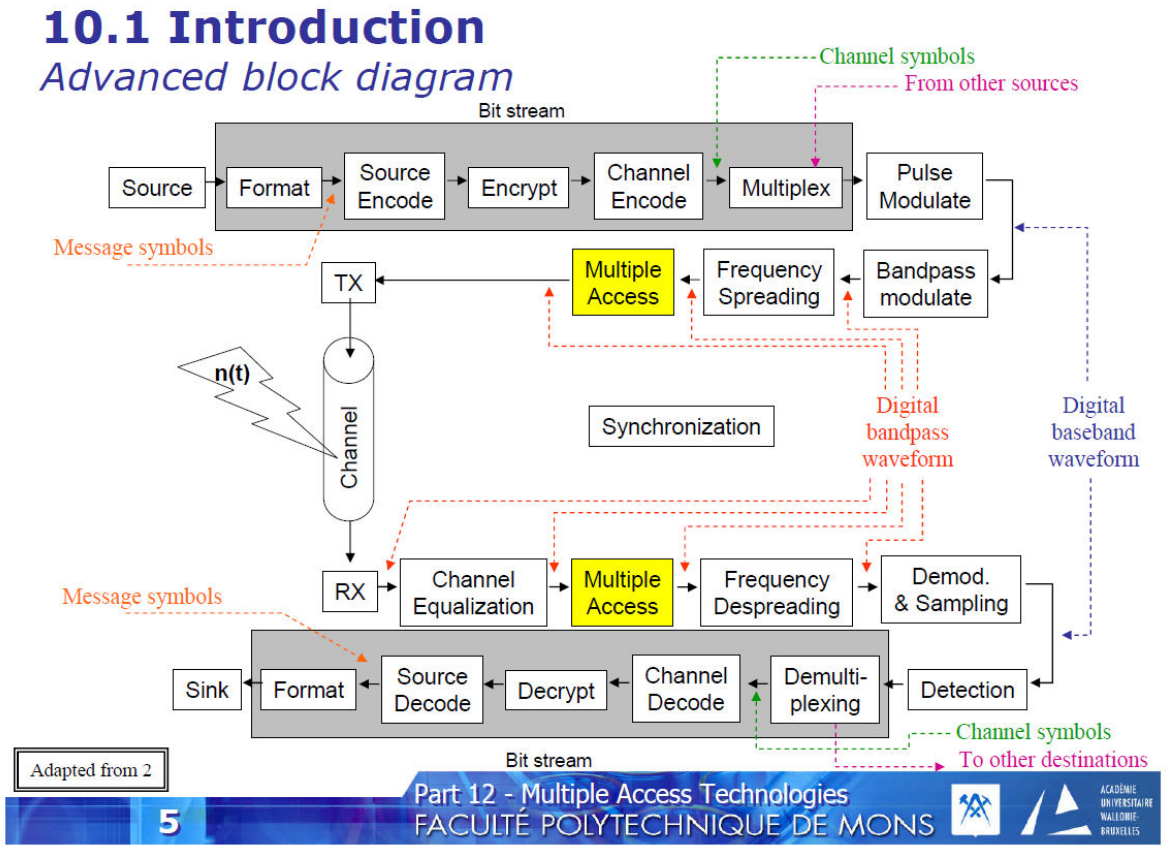
But wait, there's more !

ISO 27000:2016 definition of **information security** continues:

> *In addition, other properties, such as **authenticity**, **accountability**, **non-repudiation**, and **reliability** can also be involved.*

- **Authenticity** — property that an entity is what it claims to be
- **Accountability** — ability to prove who is the originator of an action
- **Non-repudiation** — ability to prove the occurrence of a claimed event (2.25) or action and its originating entities
- **Reliability** — property of consistent intended behaviour and results

Think this only applies to IT? Think harder!



Think this only applies to IT? Think harder! (2)

TCE-500 is a secure narrow-band telephone set for use on standard analogue networks (**PSTN**), developed by Thales Norway (previously: Thomson CSF, Alcatel and STK) in 1993.

The TCE-500 and the TCE-500B are approved by NATO for all level of classification and have a built-in modem.

Think this only applies to IT? Think harder! (3)

RT-3600 was an VHF wide-band FM military radio set, developed for the Dutch Army by Philips Telecommunications Industry (PTI) in Hilversum (Netherlands) during the late 1960s and early 1970s.

[…] The radio covers 26-70 MHz with 10 kHz deviation and a channel spacing of 50 kHz. Output power is 2W […]

The RT-3600 radio is often used in combination with […] voice encryption units.

Authentication

CIA definitions speak of “authorized parties”… **How does one gets authorized, exactly?**

AAA protocol

Typical access control mechanisms follow **3 steps**:

1. Identification/Authentication: Who are you? Prove it!
2. Authorisation: Are you allowed to do what you request?
3. Accounting/Audit : What did you do?

Accounting/Audit

- Accounting: List actions
- Audit: Log all actions, their outcomes, with very detailed tracks

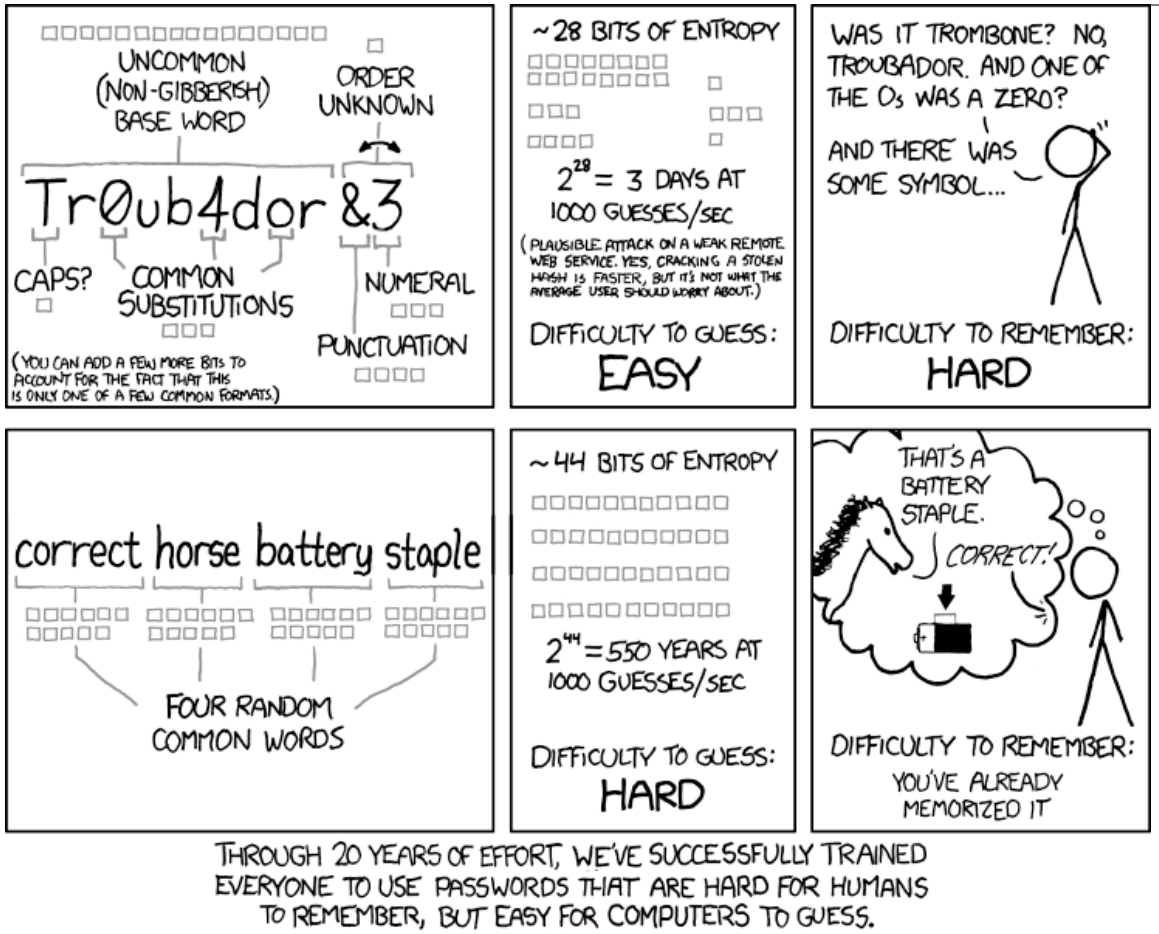
Weak & Strong Authentication

There are numerous ways to prove your identity:

- What you know (password, PIN code, …)
- What you own (smart card, access badge, …)
- What you are (fingerprints, retinal print, …)

Strong authentication uses **at least 2 factors**.

Passwords are a complex thing



We'll come back to this, but remember:

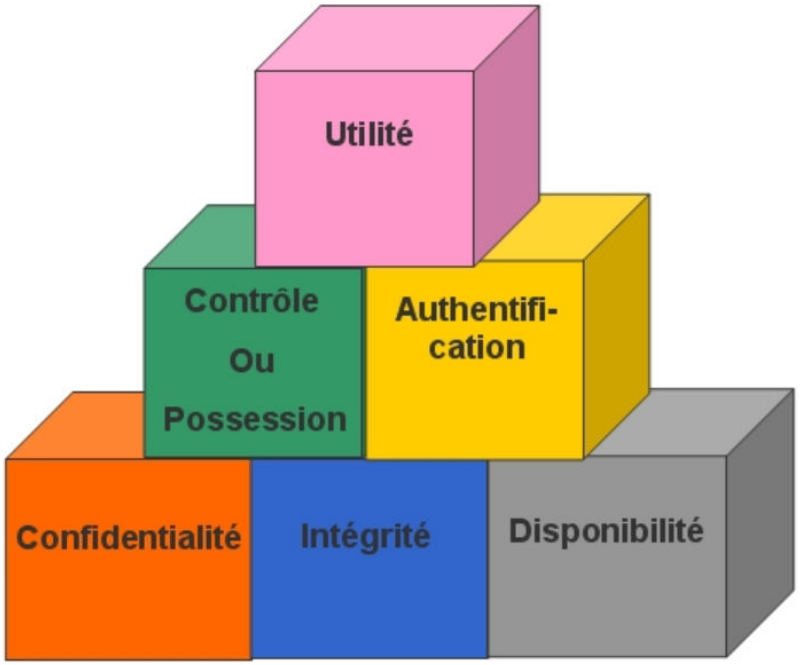
- You're a **human**: things that make sense are easier to handle
- The opponents often use **computers**: they don't care about sense

Put yourself in the proper perspective to gain a sense of real security!

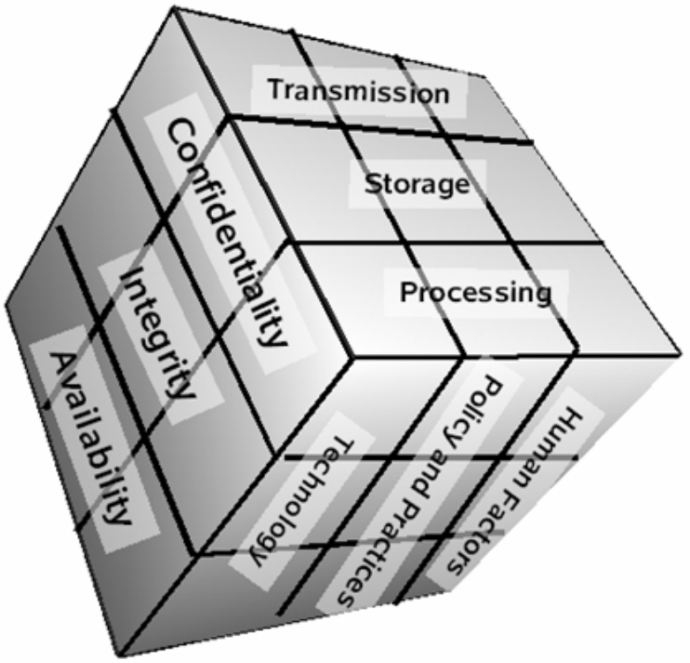
Entropy is the only thing standing between you and brute force

Information security is not a fixed thing

PARKERIAN HEXAD (2002)

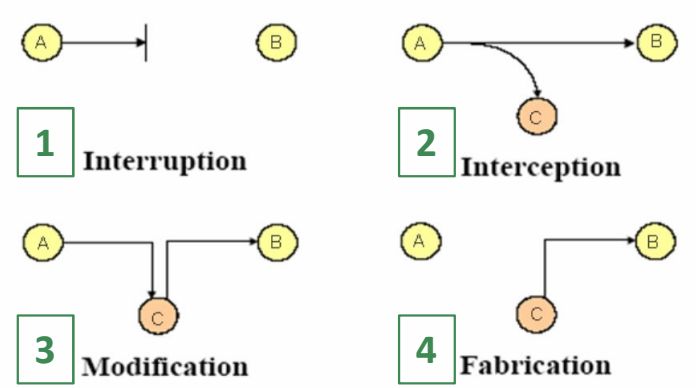


CUMBER CUBE (1991)



Challenge: Who's who?

Map the different active attacks below with the impacted property



A. Integrity

B. Availability

C. Confidentiality

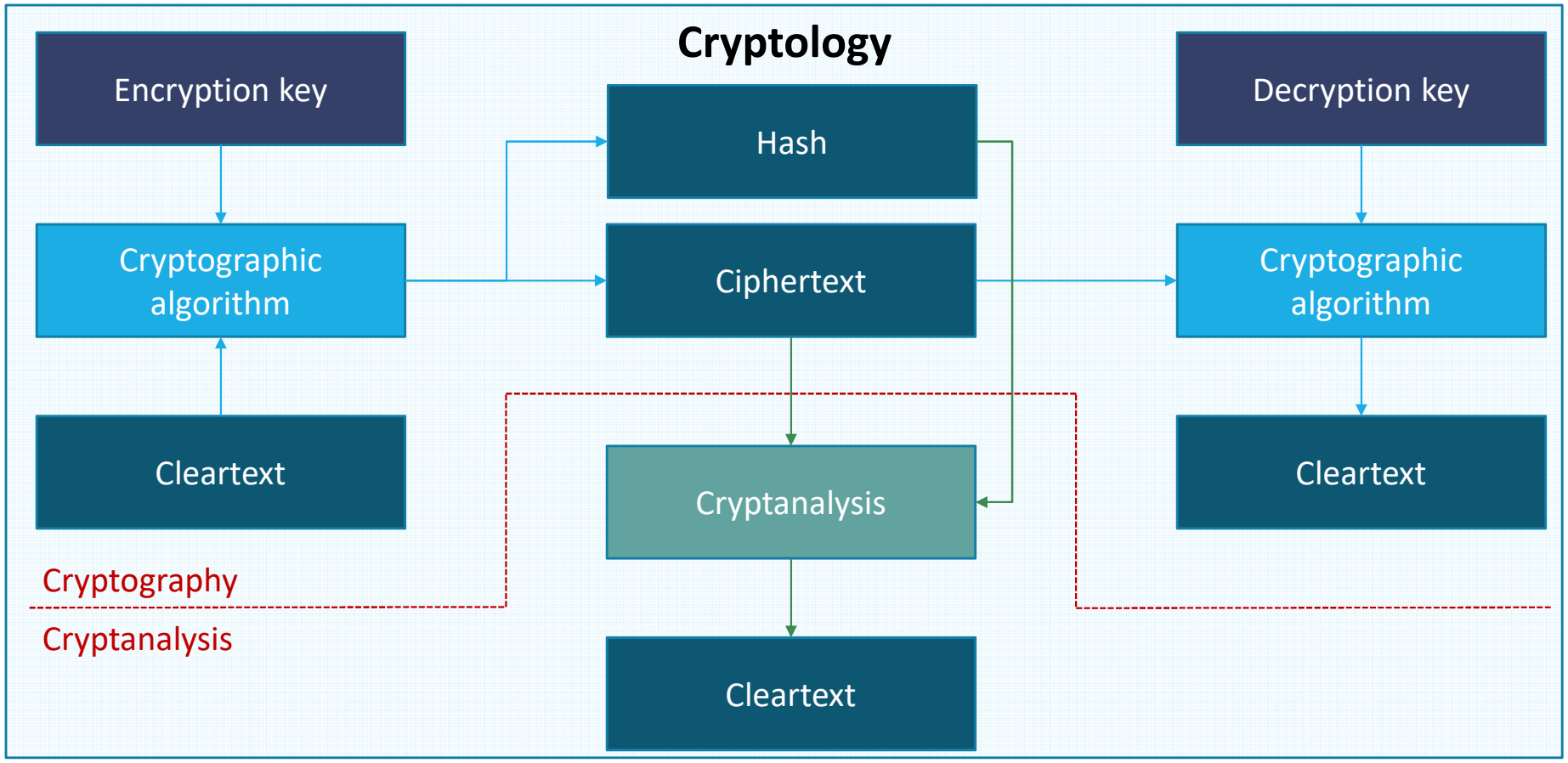
D. Authenticity

Section 2
Cryptographic algorithms

SYMMETRIC & ASSYMETRIC CRYPTOGRAPHY, HASHING, MAC AND OTHER ASSORTED ALGORITHMS

Note: This section is heavily based on [1]

Cryptography?



Some definitions

Cryptology: branchs of mathematics dedicated to studying all means to guarantee information security, including how to break them

Cryptography: « protect » branch of cryptology

Cryptanalysis: « break » branch of cryptology, e.g. how to break cryptographic algorithms without knowledge of the key and/or the algorithm

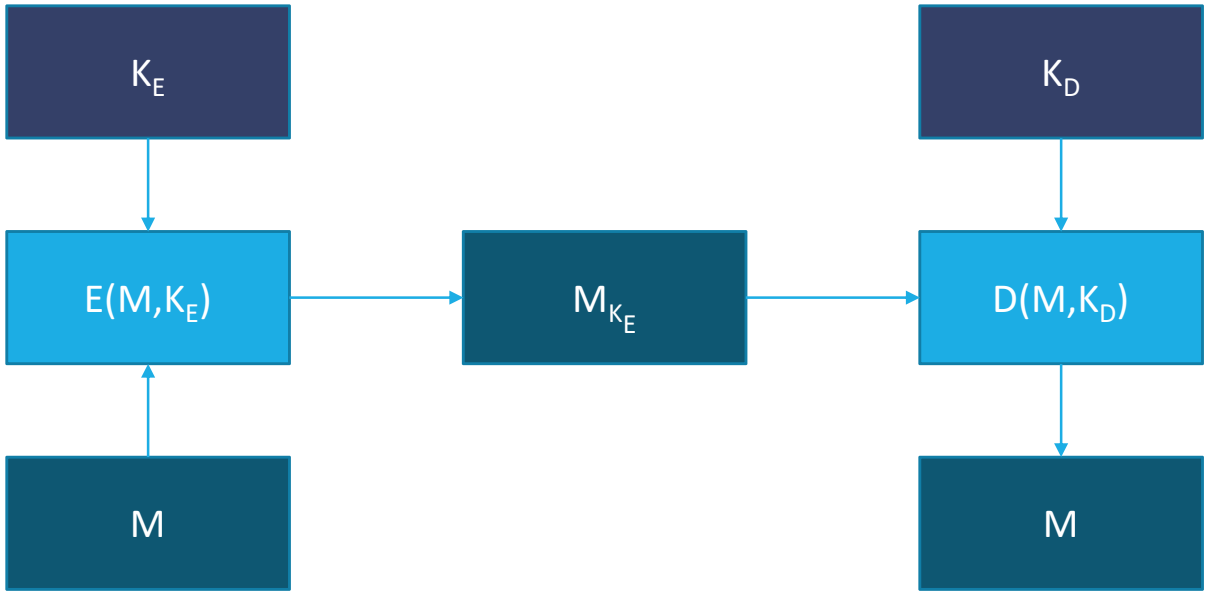
Encryption: Transformation of a cleartext into a new version, impossible to understand if you don't know about the encryption algorithm and key: the ciphertext

Hashing: One-way transformation of an arbitrary length cleartext into a fixed length string that strongly differs depending on the input

Encryption

Mostly used to guarantee **confidentiality** of the cleartext

Often seen as a synonym of cryptography, but **that's completely false**



Symmetric $K_E = K_D$

Assymmetric $K_E \neq K_D$

Challenge: Substitution ciphers

Each symbol is replaced by another one, following a more or less complex function

- Cryptanalysis is usually very easy: if you just substitute, letter frequency analysis will help you find the « key »
- Often used in enigmas, childs games, and so on..
- Many examples can be found throughout history, one of them being:

Caesar's cipher can be used as a symmetric, alphabetic-only encryption algorithm.

Without using Internet, find out which **k** value can decrypt the following message:

PBATENGHYNGVBAF LBH UNIR SBHAQ GUR GUVEQ SYNT

Bonus flag: What is the usual name of this cipher, and why?

Encryption algorithm families

SYMMETRIC ENCRYPTION

$K_E = K_D = K$ « the secret key »

Fully random keys

Based on transposition & substitution of bits in the cleartext, using K as a parameter

Key sizes < 512 bits

Usually very fast

Main issue: keys distribution

ASSYMMETRIC ENCRYPTION

$K_E = K_{PUB}$ « the public key »

$K_D = K_{PR}$ « the private key »

Keys based on a random seed, manipulated so that KPUB can be deduced from KPR, but not the opposite

Based on complex mathematical problems

Key sizes < 8192 bits

Slow

Easy key distribution

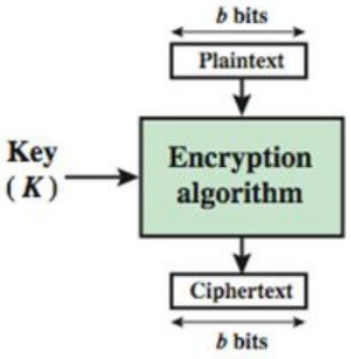
Symmetric algorithm modes

BLOCK CIPHER MODES

Plaintext is cut into blocks of b bits, with padding if needed

Block modes differ by the way successive blocks impact each other and additional data

Common ones: ECB, CBC

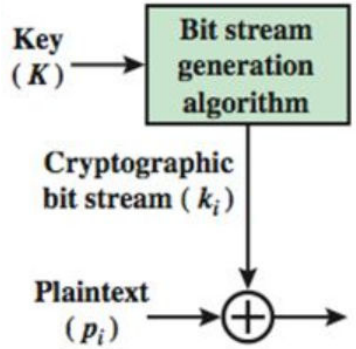


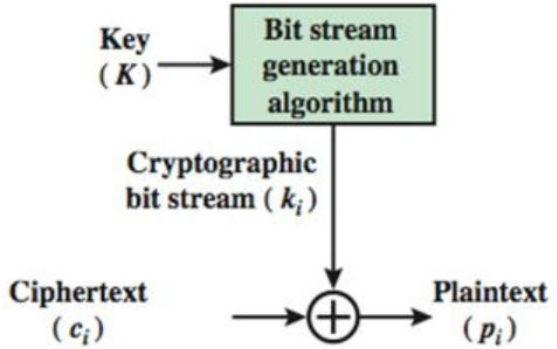
STREAM CIPHER MODES

Plaintext is modified bit by bit through a XOR with a value derived from the key

Stream modes differs from the generation algorithm

Common ones: OFB, CTR, GCM, CCM





Challenge: Block cipher modes

Let's try out the AES algorithm in some different modes.

On your virtual machine, under the `/opt/3` folder, your will find:

- `text`, `text2`: Two different text files
- `lock.jpg`: A lock picture in the JPEG format
- `encrypt-jpg.sh`: A Bash script that will encrypt the input JPEG file using OpenSSL
- `genkey-32b.sh`: Generate a 32 bit random hex string

Before we start looking for the flag - using these elements, you are asked to:

- Test the ECB & CBC modes with key sizes of 128 and 256 bits. Use several random keys of the appropriate lengths.
- Inspect your results with the `hexdump -vC` (text) and `GIMP` (image) applications.

OK, let's start looking for the flag!

The flag is hidden in the file `/opt/3/flag3-dec17`. It was encrypted using AES, with a mode and key size you have to determine based on the file characteristics.

This is all the info you need: `204A6176615363726970742E0D0A372E`

Hashing

Mainly used to guarantee **integrity**

Principle: Arbitrary length string `M` → Fixed-length string `H(M)` « the hash »

Main properties:

- Unidirectional functions: Knowing `H(M)`, you cannot find back `M`
  Note: Finding back `M` from `H(M)` is a **preimage attack**
- No collisions: There exists no pair `(M,M')` so that `H(M) = H(M')`
- Diffusion: Slight difference between `M` and `M'` result in vastly different `H(M)` and `H(M')`

Popular algorithms

- MD5, SHA1: Deprecated
- SHA2, SHA3: Currently safe and secure

Challenge: I've lost a password!

On a drunk day, my paranoia took over and I created this very weird cryptographic method:

- I write my key into a text file
- This file is hashed using the CRC32 algorithm
- The resulting hash is transformed then used as the encryption key for the **/opt/4/clever-crypto** application

The sad new is that the clever-crypto algorithm is lost for good. On another drunk night, I vomited on the design documentation and destroyed the only backup of its source code using a baseball bat.

So, nowadays, if you lose the keyfile, you've pretty much completely lost your data.

On your virtual machine, you will find the encrypted fourth flag in **`/opt/4/flag4`**. I've lost the key file, but I still have the CRC32 hash: **`19ac3c84`** (The offset has no importance here.)

First, find out why my drunk self made a huge mistake in the design of this method (Tip: *collision*). Use your findings to recover the fourth flag!

You would think they would fix this…

Ubuntu Forums hack exposes 2 million users

An unnamed hacker took usernames, email addresses, and salted and hashed passwords.

By Zack Whittaker for Zero Day | July 15, 2016 -- 20:06 GMT (21:06 BST) | Topic: Security

The company that builds Ubuntu, a popular Linux distribution, has said its forums were hacked Thursday.

Canonical, which develops the operating system, said in a statement on Friday that two million usernames, email addresses, and IP addresses associated with the Ubuntu Forums were taken by an unnamed attacker.

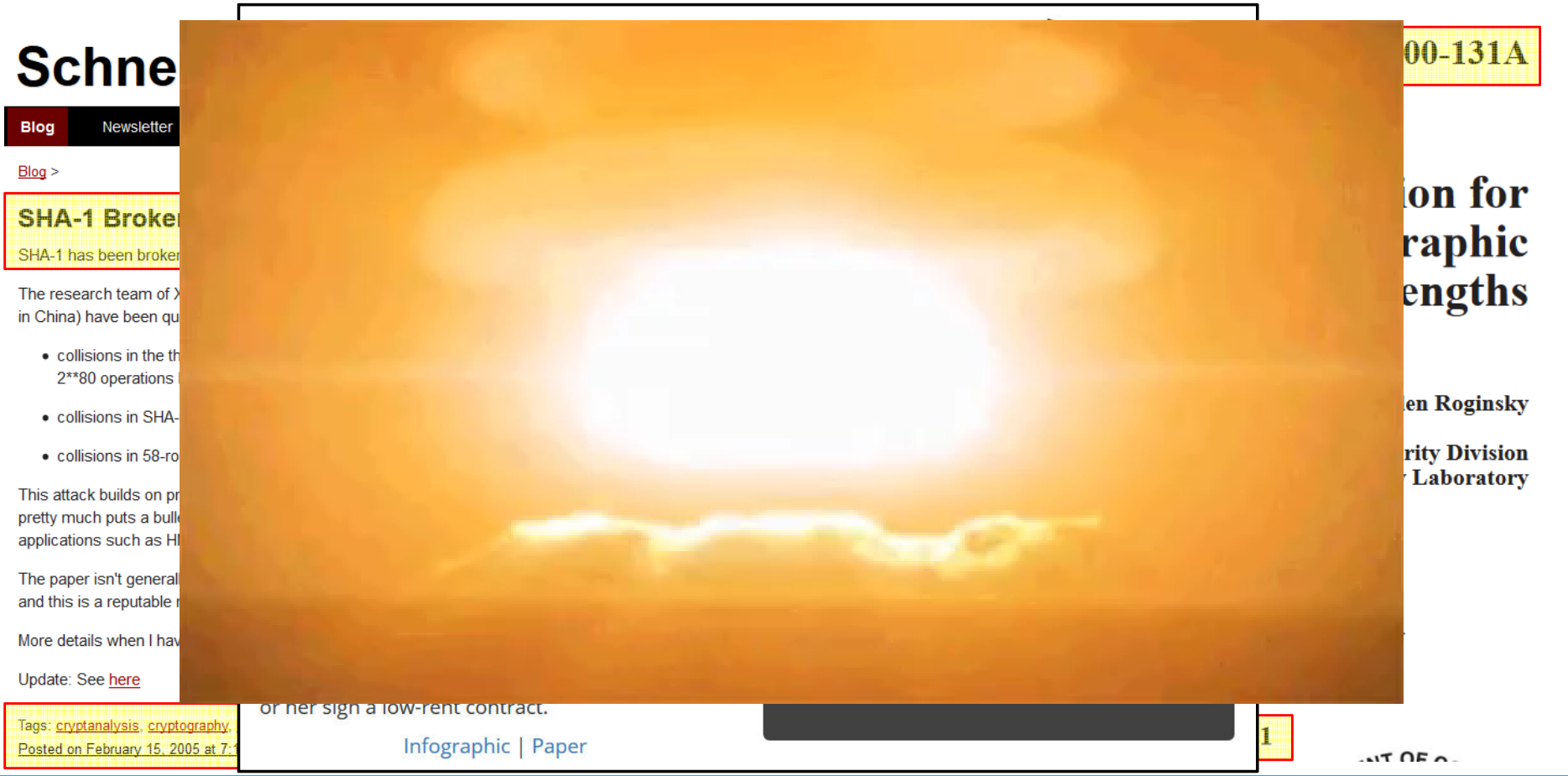
The statement added that although the forums relied on Ubuntu's single sign-on service, the passwords were hashed and salted, turning them into randomized strings of data. But the statement did not say which hashing algorithm was used -- some algorithms, like MD5, are still in use but are deprecated, as they can be easily cracked.

Extract of **[4]**

Cryptography is a constant hide and seek game that involves hackers, mathematicians, more or less talented programmers, and money.

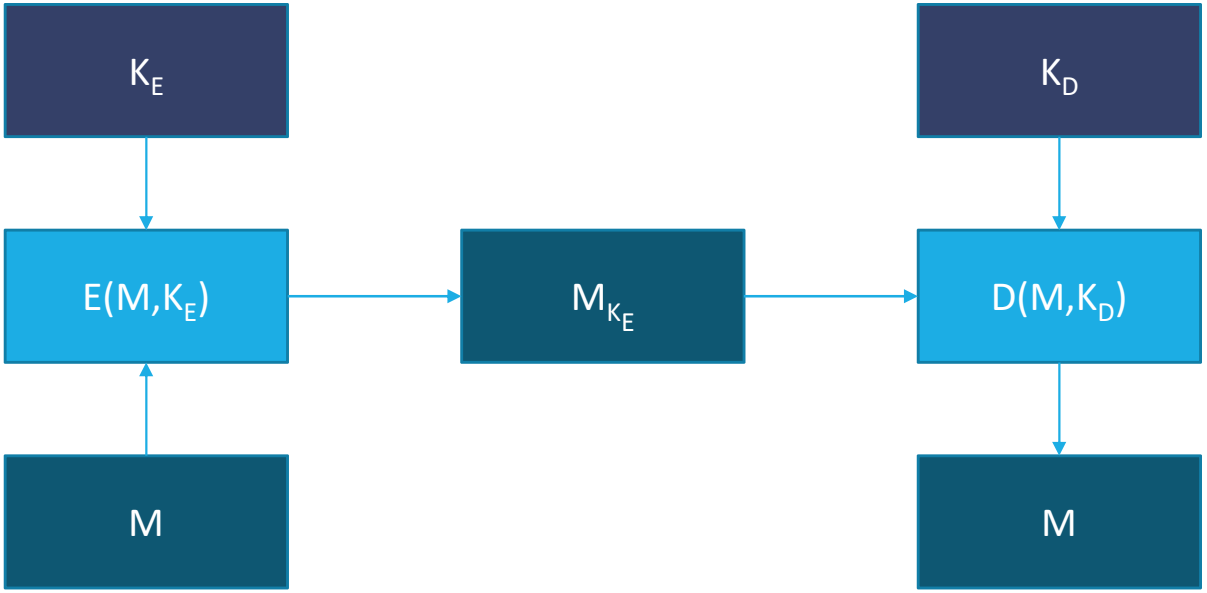
If you want to get involved, be ready to up your game !

…or take recommendations into account



It’s a cat & mouse game, all the time

Assymetric cryptography



K_{PUB} can be derived from K_{PR}, not the opposite

$K_E = K_{PUB}$ → Encryption

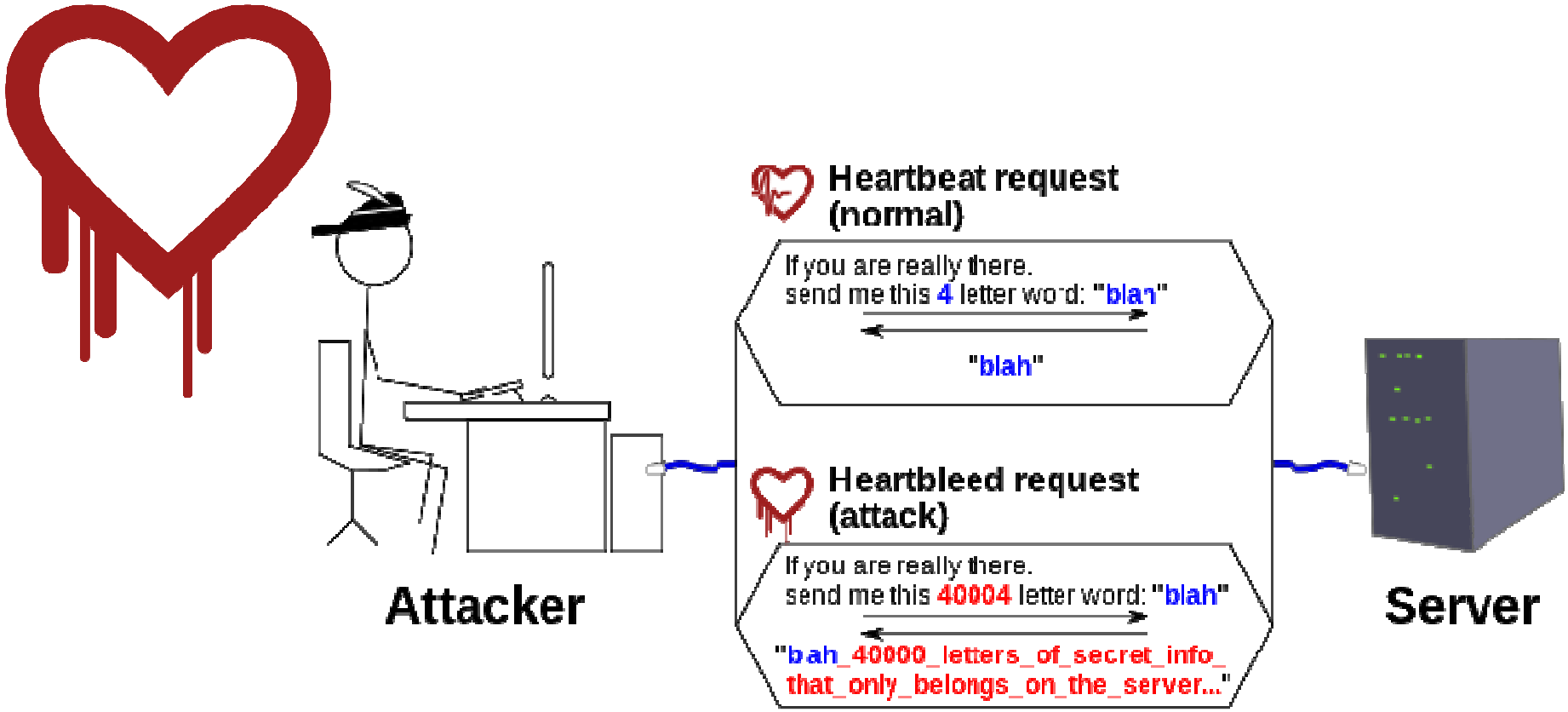
$K_E = K_{PR}$ → Signature

Signature is useful

- Authentication
- Non-repudiation

Mathematically less robust than symmetric encryption, but more varied use cases

Heartbleed


Heartbeat request
(normal)
If you are really there.
send me this 4 letter word: "blah"
"blah"
Heartbleed request
(attack)
If you are really there.
send me this 40004 letter word: "blah"
"bah_40000_letters_of_secret_info_
that_only_belongs_on_the_server..."
Attacker
Server

Kerckhoff's principle

The secret should only reside in the key and **never in the algorithm**

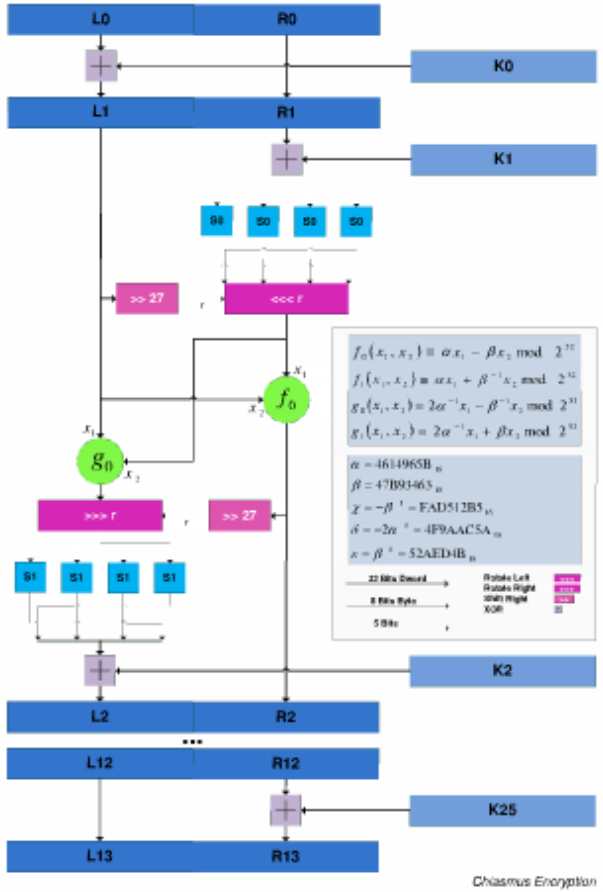
Why?

- The opponent will **always** find out about your systems. **ALWAYS.**
- Public algorithms can be challenged by a virtually infinite community – AES has yet to be broken.

Example: **Chiasmus**

Wikipedia: *Chiasmus is a **secret** German government **block cipher** that was leaked by **reverse engineering** […] GSTOOL […] **insecure Electronic Codebook (ECB) mode** […] key with a PRNG […] **current system time** […] can easily be broken by brute force.*

Picture: Block scheme of Chiasmus encryption, from **[3]**



Chiasmus Encryption

Don’t roll your own crypto

A cryptographic algorithm has many ways to fail

- Related-key
- Sandwich

Common hash functions [edit]

Collision resistance [edit]

Main article: Collision attack

Hash function	Security claim	Best attack	Publish date	Comment
MD5	2^{64}	2^{18} time	2013-03-25	This attack takes seconds on a regular PC. Two-block collisions in 2^{18}, single-block collisions in 2^{41}.[1]
SHA-1	2^{80}	$2^{63.1}$	2017-02-23	Paper.[2]
SHA256	2^{128}	31 of 64 rounds ($2^{65.5}$)	2013-05-28	Two-block collision.[3]
SHA512	2^{256}	24 of 80 rounds ($2^{32.5}$)	2008-11-25	Paper.[4]
BLAKE2s	2^{128}	2.5 of 10 rounds (2^{112})	2009-05-26	Paper.[5]
BLAKE2b	2^{256}	2.5 of 12 rounds (2^{224})	2009-05-26	Paper.[5]

- Mod-n cryptanalysis

Don’t roll your own crypto (2)

…and the

Side-channel

Main article: Si

How vu

16

soumis il y

Knowi
steps
md5 (
or not
I wonc
vulner

- Bla
- Mai
- Pr
- R
- R
- Timing analysis

th a pse

Most Android Apps are Crypto Fails

This study from Carnegie Mellon and UCSB analyzed 11,748 android apps that use crypto and found that 10,327 of them (88%) were flawed. They built a tool to check for *extremely obvious* crypto implementation errors like

- Using ECB mode.
- Using a non-random IV for CBC mode.
- Using constant encryption keys.
- Using constant salts for password hashing.
- Using fewer than 1000 iterations in password hashing.
- Seeding the random number generator with a static value.

Except for the "1000 iterations" one, these are all obvious flaws, and anyone who knows anything about cryptography should know that they are a bad idea. Especially "using constant encryption keys" - that's *insane*.

Anyway, here are their results summarized in a table.

# apps	violated rule
5,656	Uses ECB (BouncyCastle default) (R1)
3,644	Uses constant symmetric key (R3)
2,000	Uses ECB (Explicit use) (R1)
1,932	Uses constant IV (R2)
1,636	Used iteration count < 1,000 for PBE(R5)
1,629	Seeds SecureRandom with static (R6)
1,574	Uses static salt for PBE (R4)
1,421	No violation

Their results make two things clear:

- You shouldn't implement crypto yourself. *Even when* you have a high-level API.
- Just because an app "uses military-grade AES encryption", that does not mean it is secure. It probably isn't.

Very Bad Password Advice

This post on How-To-Geek about generating passwords from the command line advocates generating passwords from the current date and time.

```
[root@bender /]#
[root@bender /]# date | md5sum
6cf5f6cb1eb9f625c94d0c0ebc97fcc1  -
[root@bender /]#
[ 0.82 0.50 0.41 bender ][           0-$ shell1  (1*$sh
```

The first command they give is (note double fail: base64-encoding hex):

```
date +%s | sha256sum | base64 | head -c 32 ; echo
```

They do provide a command that gives a good alphanumeric password:

```
tr -cd '[:alnum:]' < /dev/urandom | fold -w30 | head -n1
```

However, they also mention this one:

```
date | md5sum
```

About the above command, they say, "I'm sure that some people will complain that it’s not as random as some of the other options, but honestly, it’s random enough if you’re going to be using the whole thing." That is *absolutely wrong* and demonstrates a complete lack of understanding what a hash function is.

Lessons Learned:

- Hashing does not add randomness. The output of a hash is as random as its input.
- Use a cryptographically-secure random number generator to generate passwords.

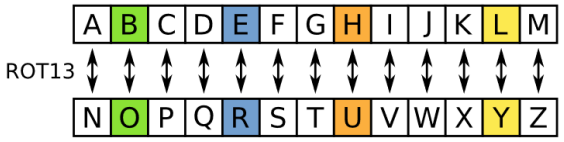
Most secure stream cipher ever!

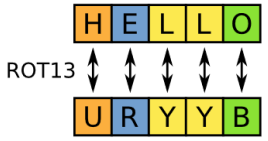
ROT13

ROT13 ("rotate by 13 places", sometimes hyphenated ROT-13) is a letter substitution cipher that replaces a letter with the letter 13 letters after it in the alphabet.

ROT13 is an example of the Caesar cipher, developed in ancient Rome.

For more information on ROT13 see http://en.wikipedia.org/wiki/ROT13



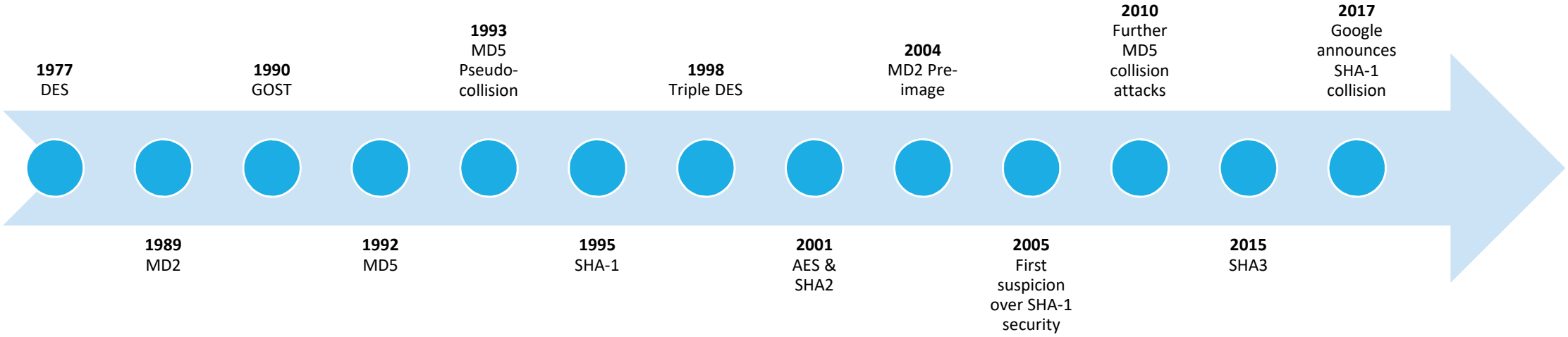


ROT26 is just like ROT13 but twice as secure!

From Hack.lu 2016: The « salsa meter »

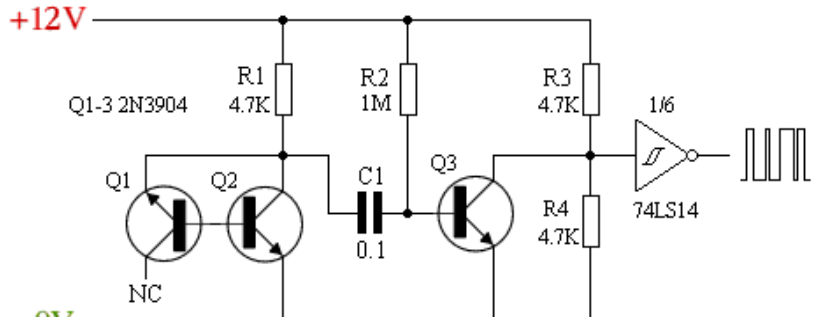
Remember WannaCry?

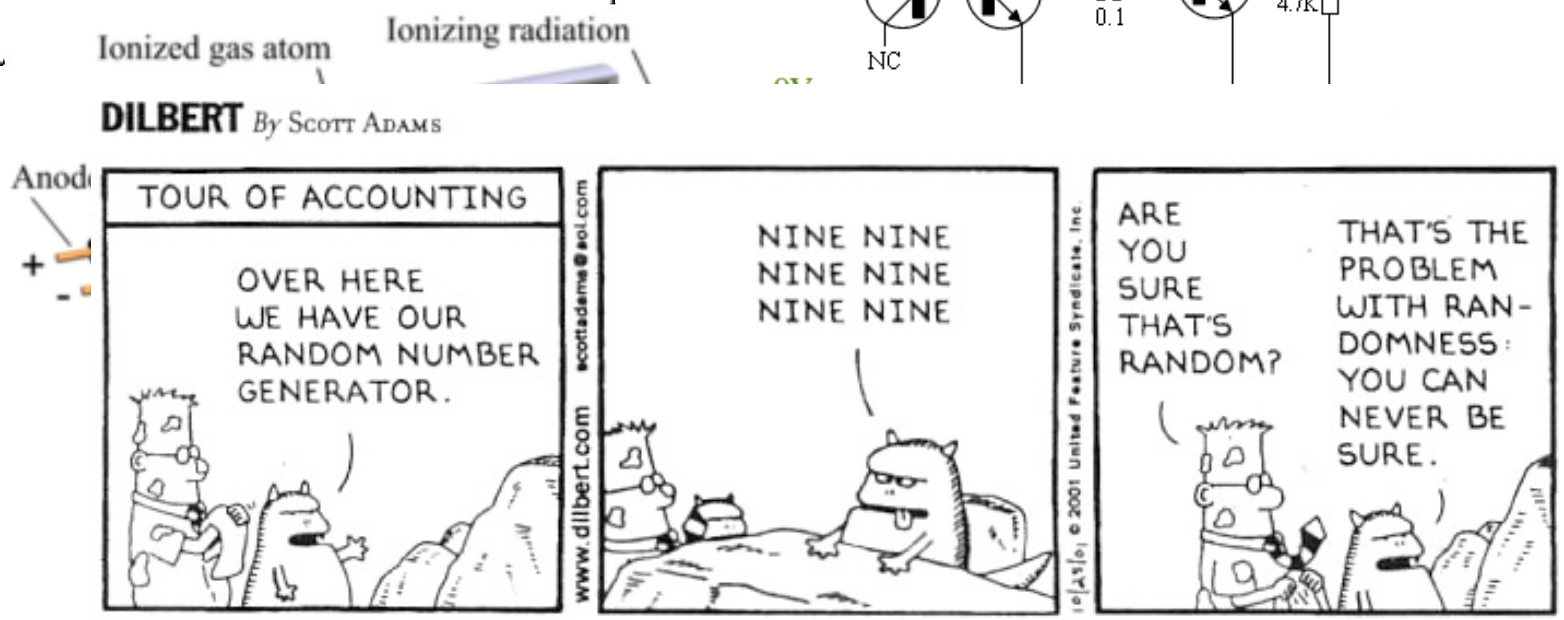
A short modern crypto timeline



Before we leave the crypto area

```
int getRandomNumber()
{
    return 4;  // chosen by fair dice roll.
               // guaranteed to be random.
}
```



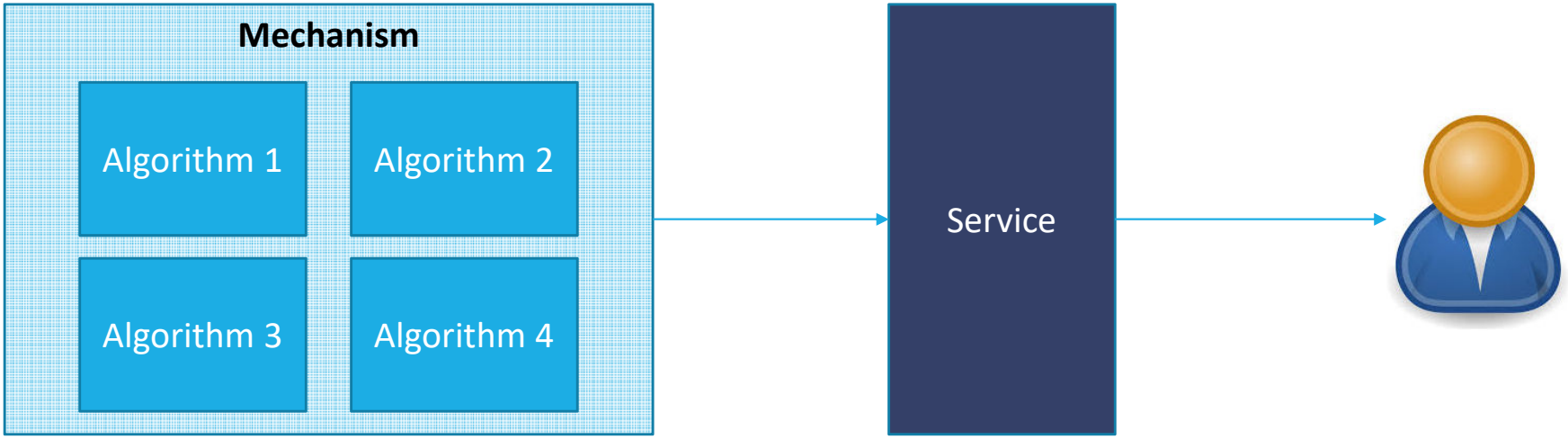


Section 3
Service, mechanism, algorithm

PROVIDING VALUE TO THE END USERS THROUGH PROPER USE OF CRYPTOGRAPHIC ALGORITHMS

Service – Mechanism - Algorithm

Service (ITILv3):	*A means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks.*
Mechanism:	*Technical mean used to achieve a given service (the* Cryptographic suite *term used by the NIST has a similar meaning)*
Algorithm:	*Building blocks of a mechanism*

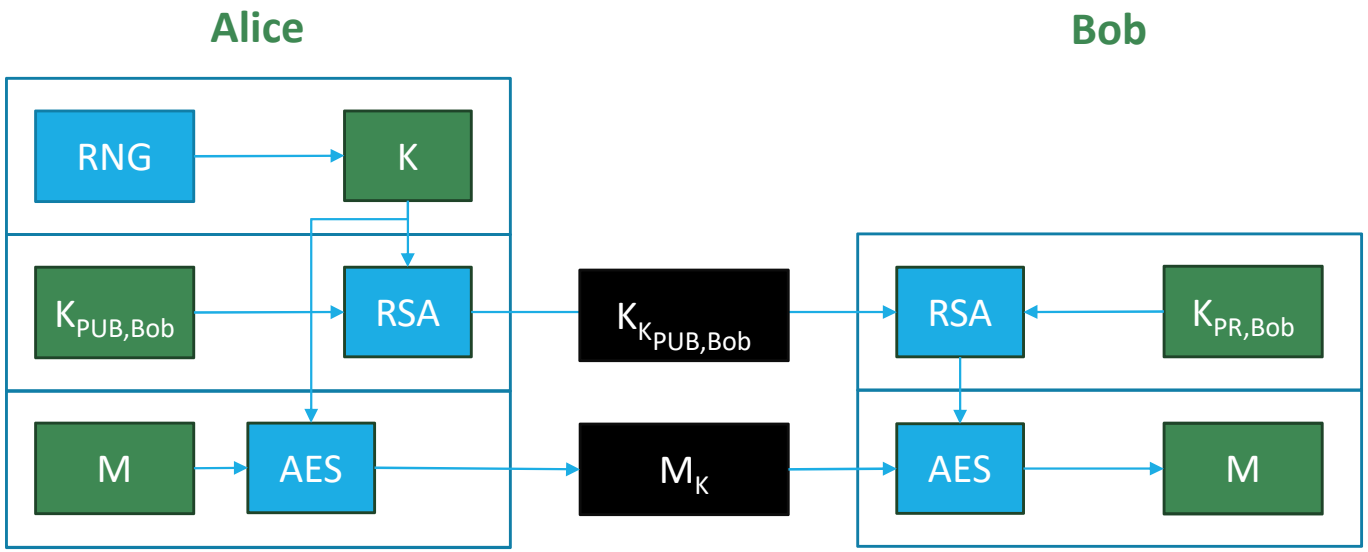


Confidentiality

Service: Guarantee the confidentiality of Alice and Bob's exchanges

Mechanism: Symmetric encryption using a one-time session key

Algorithms: RSA, AES, RNG



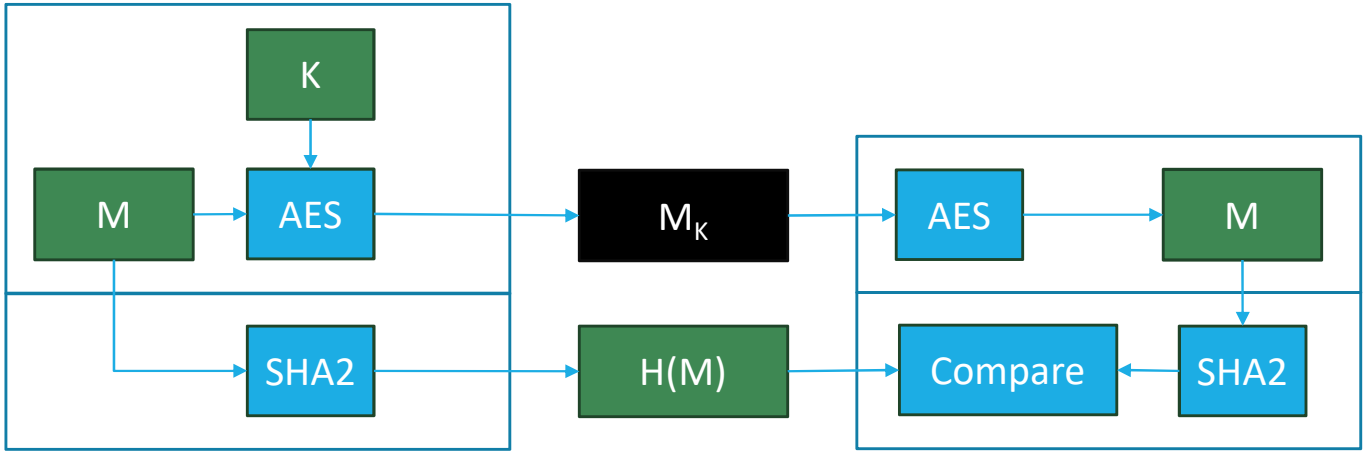
Integrity

Service: Prove to Bob that Alice's encrypted message wasn't tampered with

Mechanism: Hash-based message validation

Algorithms: SHA2

Some block mode ciphers provide both confidentiality and integrity, e.g. GCM. If they are not available (e.g.: you use a hardware accelerator that only supports AES-128-CBC and SHA-256). In that case, a separate hash must be computed and also sent to Bob.



Serial & Parallel

PARALLEL SECURITY

Let's imagine a computer that can be unlocked using several means:

- Password
- Fingerprint
- USB security dongle

How well is it protected?

If several cryptographic means can be used, the overall system is only **as strong as the weakest one**

This is referred to as **parallel security** and should be avoided as much as possible

SERIAL SECURITY

If several security mechanism work in serial, all the layers must be passed through to successfully access a resource.

One can think of a corporate network:

- Hardware edge firewall
- Link-level encryption on the network
- Software firewall on each endpoint
- Two-factor authentication

This is referred to as serial security or **defense in depth**. This is the privileged approach: never trust only one guy to protect you secrets!

Section 4
Embedded systems security

LET'S PUT SECURITY IN THIS DAMN SMALL MICROCONTROLLER

Embedded systems

Wikipedia: *An embedded system is a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts.*

My definition: *An embedded system is a computer system with a dedicated function whose primary goal is not end-user interaction, with limited HMI means and specific interfaces*

Where can you find embedded systems?

Embedded systems are often used in **low-power situations**: no, that Core i7 with a 145W TDP is not an alternative

Arduino Uno

```
Done compiling.
Sketch uses 10086 bytes (31%) of program storage space. Maximum is 32256 bytes.
Global variables use 1659 bytes (81%) of dynamic memory, leaving 389 bytes for local variables. Maximum is 2048 bytes.
Low memory available, stability problems may occur.
```

- 131 Powerful Instructions
- Most Single Clock Cycle Execution
- 32 x 8 General Purpose Working Registers
- Fully Static Operation
- Up to 20 MIPS Throughput at 20MHz
- On-chip 2-cycle Multiplier

- 32KBytes of In-System Self-Programmable Flash program Memory
- 1KBytes EEPROM
- 2KBytes Internal SRAM
- Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
- Data Retention: 20 years at 85°C/100 years at 25°C
- Optional Boot Code Section with Independent Lock Bits
  - In-System Programming by On-chip Boot Program
  - True Read-While-Write Operation
- Programming Lock for Software Security

➔ Good candidate for a simple embedded system

Dedicated cryptographic accelerators

What is It? [5]

Intel® AES New Instructions (Intel® AES NI) is a new encryption instruction set that improves on the Advanced Encryption Standard (AES) algorithm and accelerates the encryption of data in the Intel® Xeon® processor family and the Intel® Core™ processor family.

Comprised of seven new instructions, Intel® AES-NI gives your IT environment faster, more affordable data protection and greater security; making pervasive encryption feasible in areas where previously it was not.

5.2.2.AES256

		NON AES-NI		AES-NI		RESULTS	
Test Run	Size	Encryption Time (msecs)	Decryption Time (msecs)	Encryption Time (msecs)	Decryption Time (msecs)	Encryption Acceleration (%)	Decryption Acceleration (%)
1	50 MB	1026	1066	563	677	45.13	36.49
2	50 MB	947	1084	559	638	40.97	41.14
					AVG	**43.05**	**38.82**

[6]

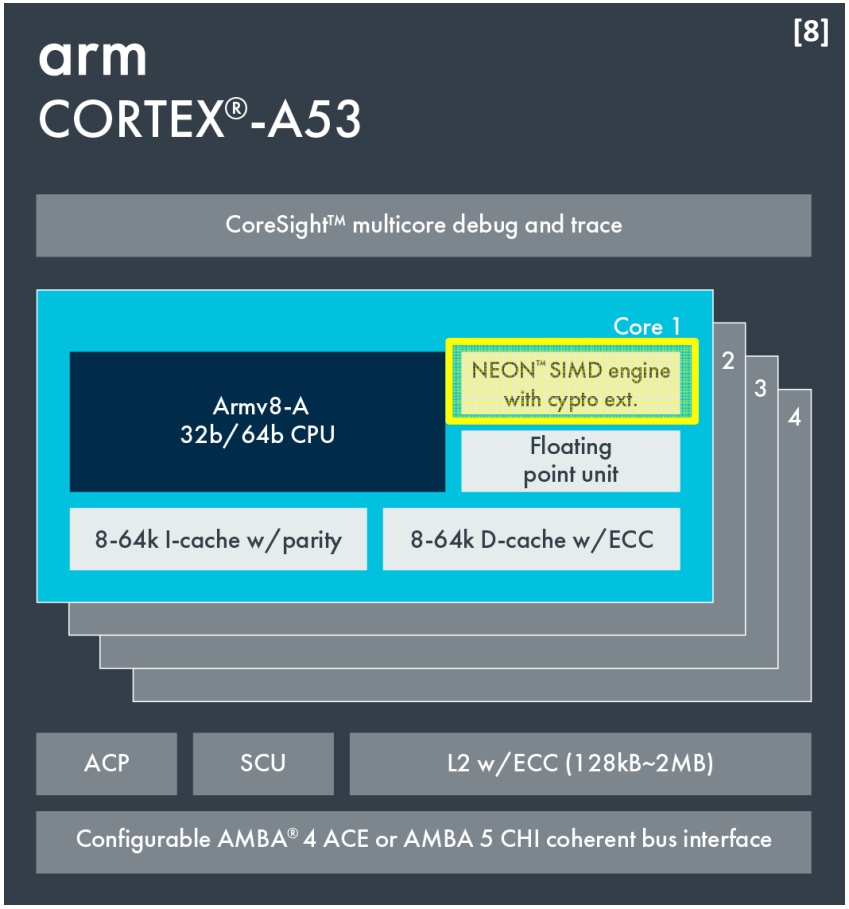
Software implementations don't cut it? Fine, we'll offload it to hardware !

Dedicated cryptographic accelerators (2)

[7] *The Raspberry Pi 3 is the third-generation Raspberry Pi. […] Broadcom **BCM2837** 64bit CPU […]*

BCM2837 is based on the **ARM Cortex A53 (ARMv8)** design

[9] *The Cortex-A53 processor Cryptography Extension supports the ARMv8 Cryptography Extensions. The Cryptography Extensions add new A64, A32, and T32 instructions to Advanced SIMD that accelerate Advanced Encryption Standard (**AES**) encryption and decryption, and the Secure Hash Algorithm (SHA) functions **SHA-1**, SHA-224, and **SHA-256**.*



Dedicated cryptographic accelerators (3)

Samsung Galaxy S7

CPU: Samsung Exynos 8890

There is a crypto accelerator mentioned in **[10]**, but no detailed information → Probably AES & SHA2, but **we cannot be sure**

Dedicated cryptographic accelerators (4)

Inverse Path USB Armory

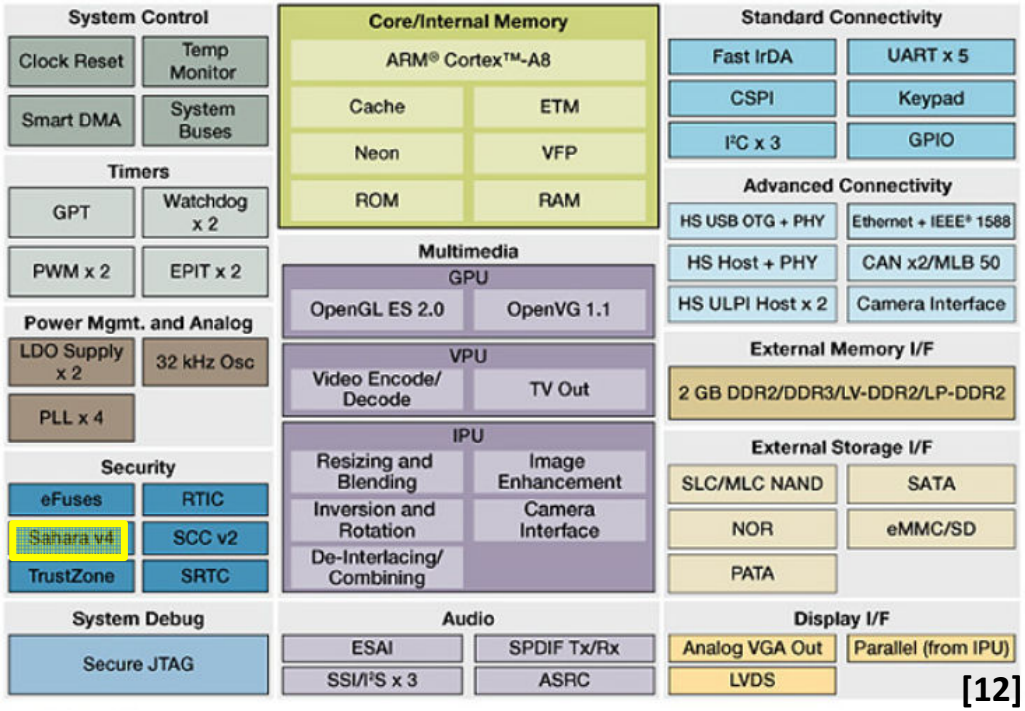
[11]

USB-stick sized SoC dedicated to information security

CPU: NXP i.MX53 ARM® Cortex™-A8 800Mhz

Contains a **SAHARAv4 Lite accelerator**:

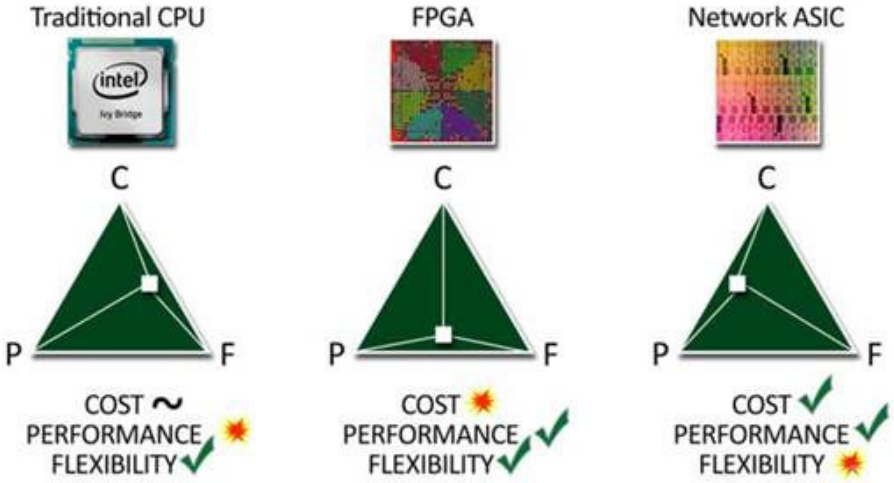
SAHARA (symmetric/asymmetric hashing and random accelerator), version 4, is a security coprocessor. It implements symmetric encryption algorithms, (AES, DES, 3DES, RC4 and C2), public key algorithms (RSA and ECC), hashing algorithms (MD5, SHA-1, SHA-224 and SHA-256), and a hardware true random number generator

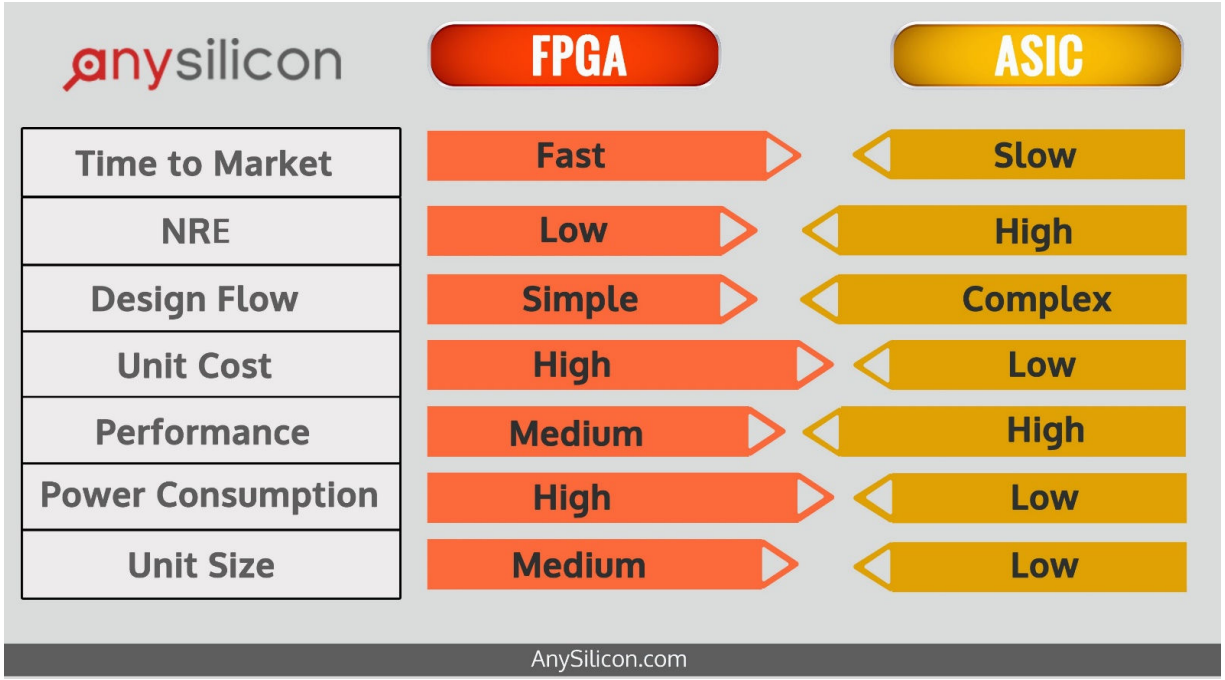


[12]

Why stop on COTS solutions?

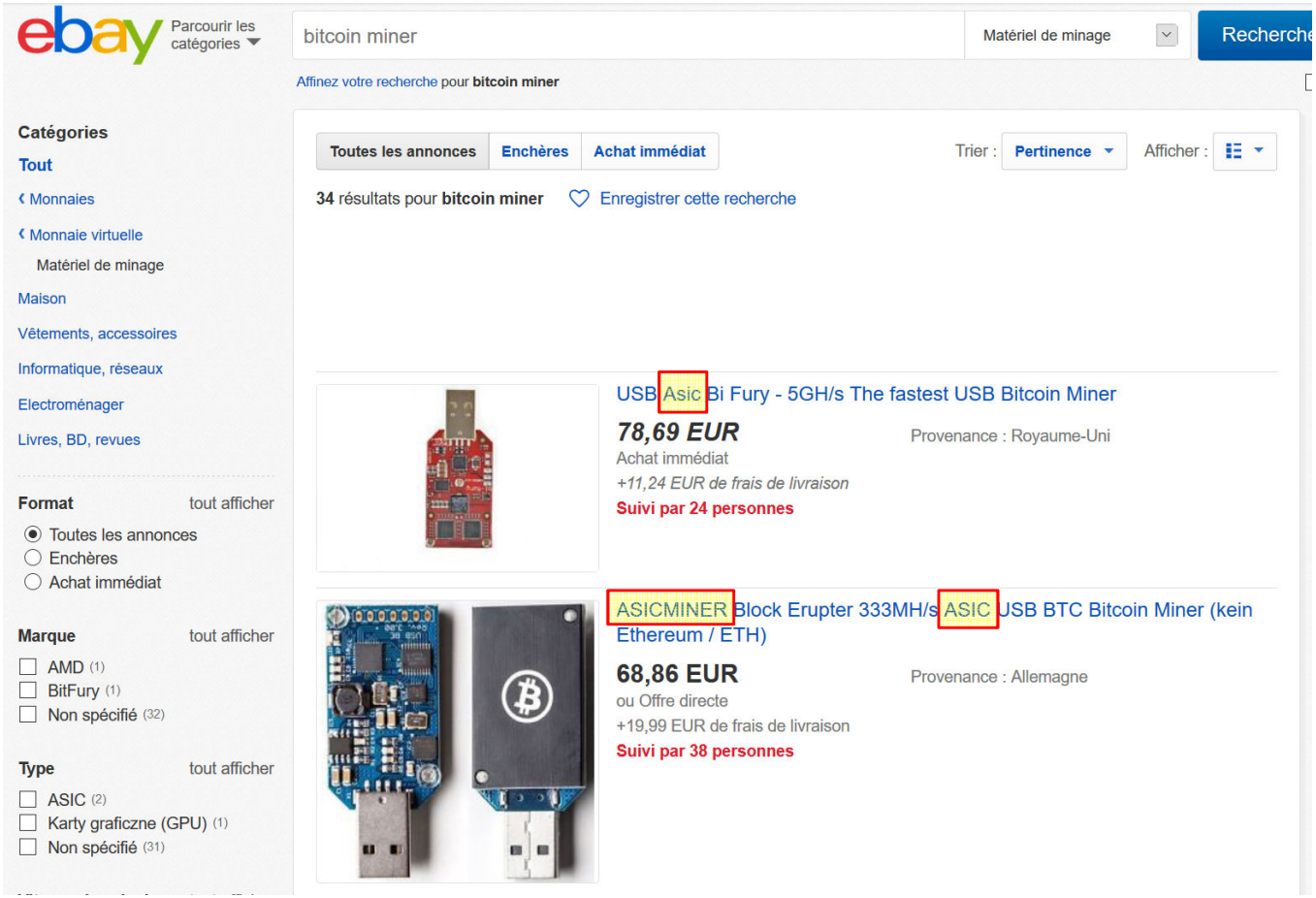
If you require a specific cryptographic mechanism not available on COTS accelerators, you can always **create your own cryptographic accelerator**





But, remember… **Unless you know what you do, never roll your own crypto!**

Anyone like bitcoin?



What if I can't use an accelerator?

We discussed common cryptographic algorithms currently in use: AES, SHA & RSA.

However, there are much more existing algorithms:

V·T·E	Block ciphers (security summary)
Common algorithms	AES · Blowfish · DES (Internal Mechanics, Triple DES) · Serpent · Twofish
Less common algorithms	Camellia · CAST-128 · IDEA · RC2 · RC5 · RC6 · SEED · ARIA · Skipjack · TEA · XTEA
Other algorithms	3-Way · Akelarre · Anubis · BaseKing · BassOmatic · BATON · BEAR and LION · CAST-256 · Chiasmus · CIKS-1 · CIPHERUNICORN-A · CIPHERUNICORN-E · CLEFIA · CMEA · Cobra · COCONUT98 · Crab · Cryptomeria/C2 · CRYPTON · CS-Cipher · DEAL · DES-X · DFC · E2 · FEAL · FEA-M · FROG · G-DES · GOST · Grand Cru · Hasty Pudding cipher · Hierocrypt · ICE · IDEA NXT · Intel Cascade Cipher · Iraqi · Kalyna · KASUMI · KeeLoq · KHAZAD · Khufu and Khafre · KN-Cipher · Kuznyechik · Ladder-DES · Libelle · LOKI (97, 89/91) · Lucifer · M6 · M8 · MacGuffin · Madryga · MAGENTA · MARS · Mercy · MESH · MISTY1 · MMB · MULTI2 · MultiSwap · New Data Seal · NewDES · Nimbus · NOEKEON · NUSH · PRESENT · Prince · Q · RC6 · REDOC · Red Pike · S-1 · SAFER · SAVILLE · SC2000 · SHACAL · SHARK · Simon · SM4 · Speck · Spectr-H64 · Square · SXAL/MBAL · Threefish · Treyfer · UES · Xenon · xmx · XXTEA · Zodiac

Maybe something fit for low-end embedded systems exists? Well… yes and no.

Table 3. Comparison of symmetric lightweight cryptography algorithms in IoT

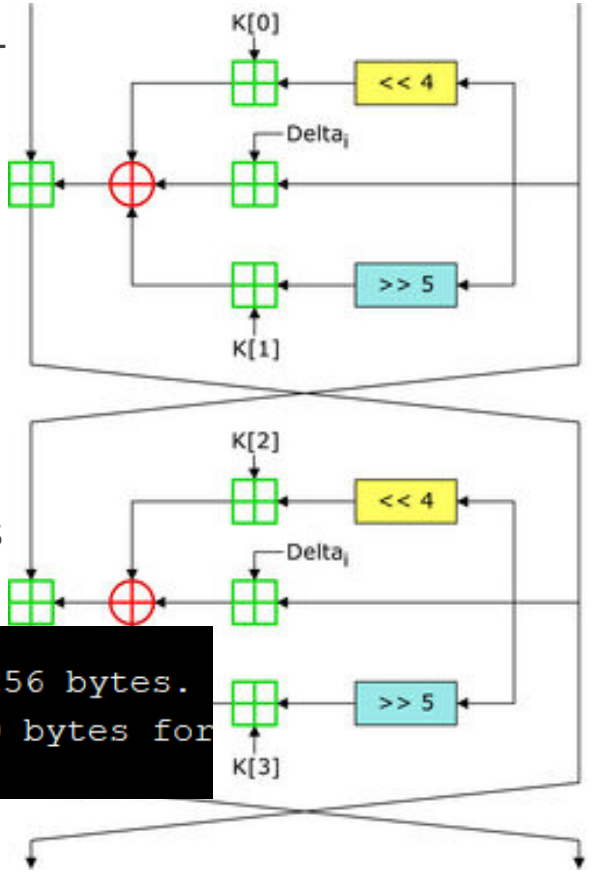
Symmetric Algorithm	Code length	Structure	Number of rounds	Key Size	Block Size	Possible Attacks
AES	2606	SPN	10	128	128	Man-in-middle attack
Hight	5672	GFS	32	128	64	Saturation attack
TEA	1140	Feistel	32	128	64	Related Key Attack
PRESENT	936	SPN	32	80	64	Differential attack
RC5	Not foxed	ARX	20	16	32	Differential attack

[13]

Let’s have a look at TEA

[14] […] operations from mixed (orthogonal) algebraic groups – XOR, ADD and SHIFT in this case. This is a very clever way of providing Shannon's twin properties of diffusion and confusion which are necessary for a secure block cipher, without the explicit need for P-boxes and S-boxes respectively.

It encrypts 64 data bits at a time using a 128-bit key. It seems highly resistant to differential cryptanalysis, and achieves complete diffusion (where a one bit difference in the plaintext will cause approximately 32 bit differences in the ciphertext) after only six rounds.




```
Sketch uses 2426 bytes (7%) of program storage space. Maximum is 32256 bytes.
Global variables use 268 bytes (13%) of dynamic memory, leaving 1780 bytes for
```

AES-256-GCM was **5x bigger** ! (but here we don’t any block mode)

Using crypto in embedded systems

What we learned about cryptography on embedded systems:

- Cryptographic algorithms are CPU-intensive
- Most mechanisms will create a data overhead (e.g. HMAC-SHA256)
- Not all embedded systems have a lot of memory available: all algorithms don't fit all use cases

What we learned from the previous sections: **information must be secured**

Possible solutions:

- Use a COTS hardware accelerator
  Drawbacks: Limited choice of algorithms, usually no upgrades possible
- Develop your own hardware accelerator
  Drawbacks: Can be very expensive, not always easy to upgrade, are you a skilled cryptographic developer?
- Use algorithms developed specifically for the embedded world
  Drawbacks: Often new, unchallenged algorithms, implementations are not always readily available for your platform, be careful of algorithms that trade security for « efficiency »

There is no silver bullet. You must make a choice that takes into account the sensitivity of the handled data, the available resources and the opponent potential.

Challenge: Let's use this camera!

Here is the situation:

- The network camera (`10.10.10.101`) is connected directly to the Internet.
- The video server (`10.10.10.200`) continuously receives information from this camera, through the Internet.

We'll have a look at the camera GUI together. Once this tour is done, you have **10 minutes** to devise an attack scenario that:

- Enables to take down the video server, or any other one connected to the Internet, granted that you have multiple IP cameras available
- Doesn't disrupt the normal flow of operations of the system, e.g. the video server is still working correctly if you're not attacking it.

You are free to ask any question you want (I won't answer to all of them!) and to come back and look at the GUI during the exercice.

There is no flag to find here: the points will be allocated based on how feasible the scenario you submit is.

Bonus points for the first team to answer this: Which information security properties do you disrupt with your scenario?

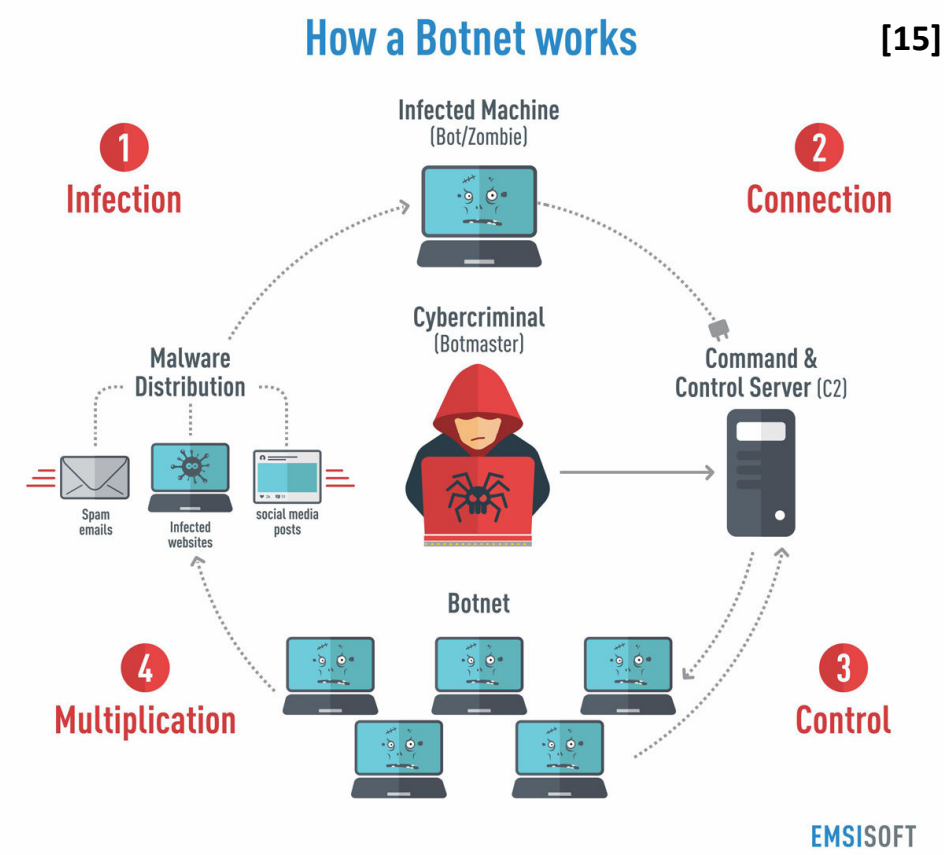
Mirai

Wikipedia:

*Mirai (Japanese for "the future", 未来) is malware that turns networked devices running Linux into remotely controlled "**bots**" that can be used as part of a botnet in large-scale network attacks.*

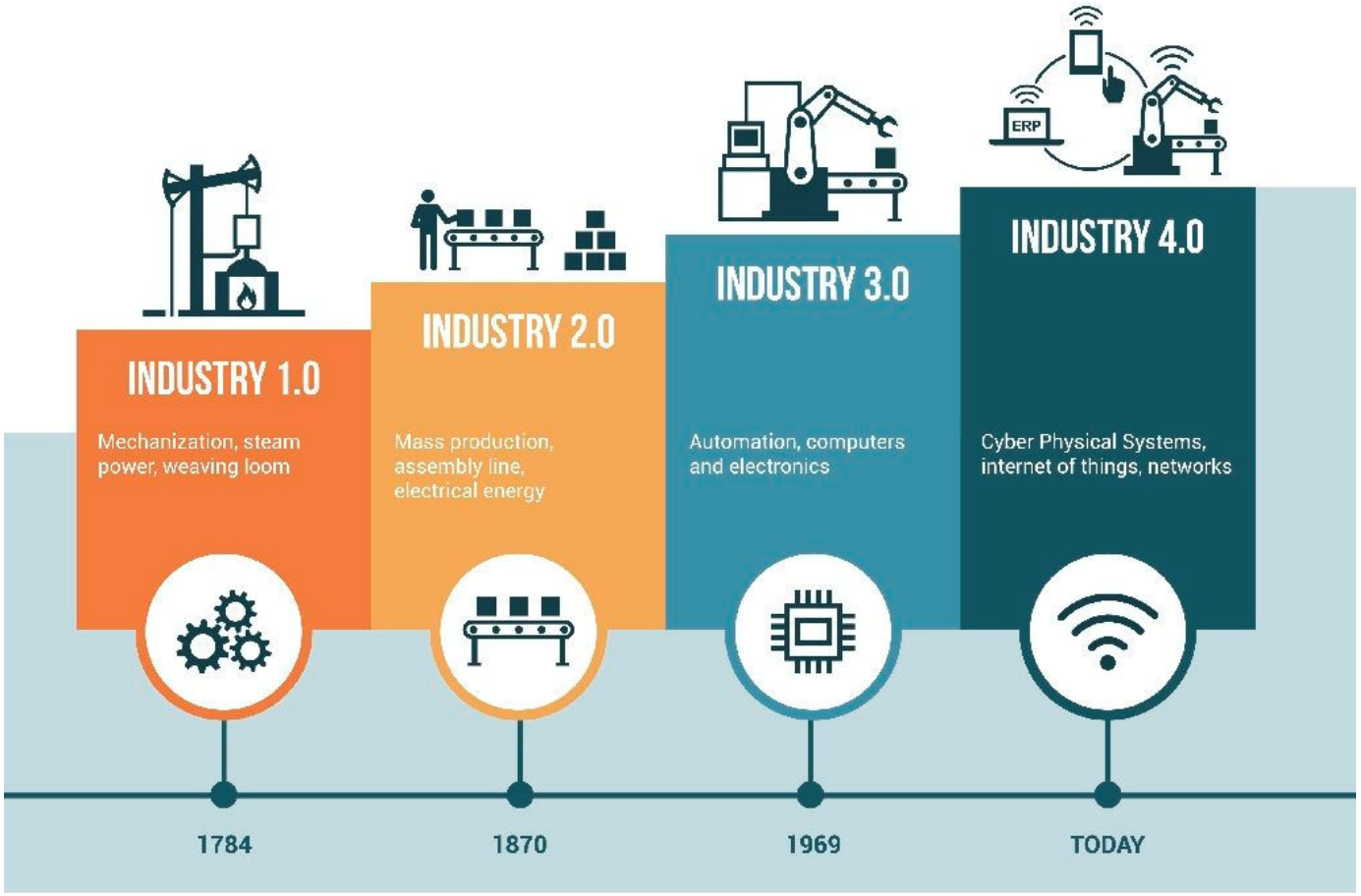
It primarily targets online consumer devices such as IP cameras and home routers.

Embedded systems, becoming more proeminent with IoT and Industry 4.0, are a privileged at target for botnets.



[15]

Industry 4.0


INDUSTRY 1.0
Mechanization, steam power, weaving loom
INDUSTRY 2.0
Mass production, assembly line, electrical energy
INDUSTRY 3.0
Automation, computers and electronics
INDUSTRY 4.0
Cyber Physical Systems, internet of things, networks
ERP
1784
1870
1969
TODAY

September 2016: OVH under attack

```
log /home/vac/logs/vac.log-last | egrep
bps" | awk '{print $1,$2,$3,$6}' | sed
1,2,3,7,8,10,11 -d '|' | sed "s/.......
"s/.....pps/Mpps/" | cut -f 2,3,4,5,6,
rep "gone" | sed "s/gone|//"
Sep|18|10:49:12|tcp_ack|20Mpps|232Gbps
Sep|18|10:58:32|tcp_ack|15Mpps|173Gbps
Sep|18|11:17:02|tcp_ack|19Mpps|224Gbps
Sep|18|11:44:17|tcp_ack|19Mpps|227Gbps
Sep|18|19:05:47|tcp_ack|66Mpps|735Gbps
Sep|18|20:49:27|tcp_ack|81Mpps|360Gbps
Sep|18|22:43:32|tcp_ack|11Mpps|136Gbps
Sep|18|22:44:17|tcp_ack|38Mpps|442Gbps
Sep|19|10:13:57|tcp_ack|10Mpps|117Gbps
Sep|19|11:53:57|tcp_ack|13Mpps|159Gbps
Sep|19|11:54:42|tcp_ack|52Mpps|607Gbps
Sep|19|22:51:57|tcp_ack|10Mpps|115Gbps
Sep|20|01:40:02|tcp_ack|22Mpps|191Gbps
Sep|20|01:40:47|tcp_ack|93Mpps|799Gbps
Sep|20|01:50:07|tcp_ack|14Mpps|124Gbps
Sep|20|01:50:32|tcp_ack|72Mpps|615Gbps
Sep|20|03:12:12|tcp_ack|49Mpps|419Gbps
Sep|20|11:57:07|tcp_ack|15Mpps|178Gbps
Sep|20|11:58:02|tcp_ack|60Mpps|698Gbps
Sep|20|12:31:12|tcp_ack|17Mpps|201Gbps
Sep|20|12:32:22|tcp_ack|50Mpps|587Gbps
Sep|20|12:47:02|tcp_ack|18Mpps|210Gbps
Sep|20|12:48:17|tcp_ack|49Mpps|572Gbps
Sep|21|05:09:42|tcp_ack|32Mpps|144Gbps
Sep|21|20:21:37|tcp_ack|22Mpps|122Gbps
Sep|22|00:50:57|tcp_ack|16Mpps|191Gbps
You have new mail in /var/mail/root
```

Octave Klaba ✓
@olesovhcom

Follow

Last days, we got lot of huge DDoS. Here, the list of "bigger that 100Gbps" only. You can see the simultaneous DDoS are close to 1Tbps !

7:37 AM - Sep 22, 2016

54 705 585

Octave Klaba ✓
@olesovhcom

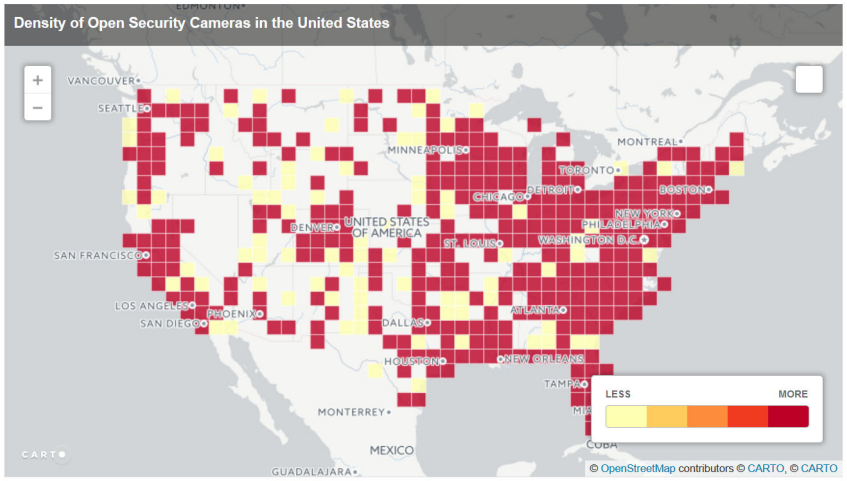
Follow

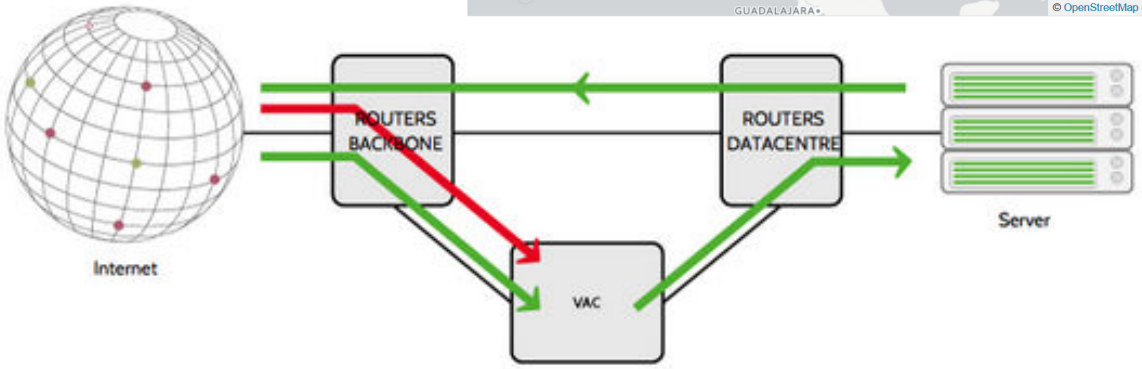
Replying to @olesovhcom

This botnet with 145607 cameras/dvr (1-30Mbps per IP) is able to send >1.5Tbps DDoS. Type: tcp/ack, tcp/ack+psh, tcp/syn.

2:31 PM - Sep 23, 2016

25 573 385





Could this have been avoided?

Wikipedia:

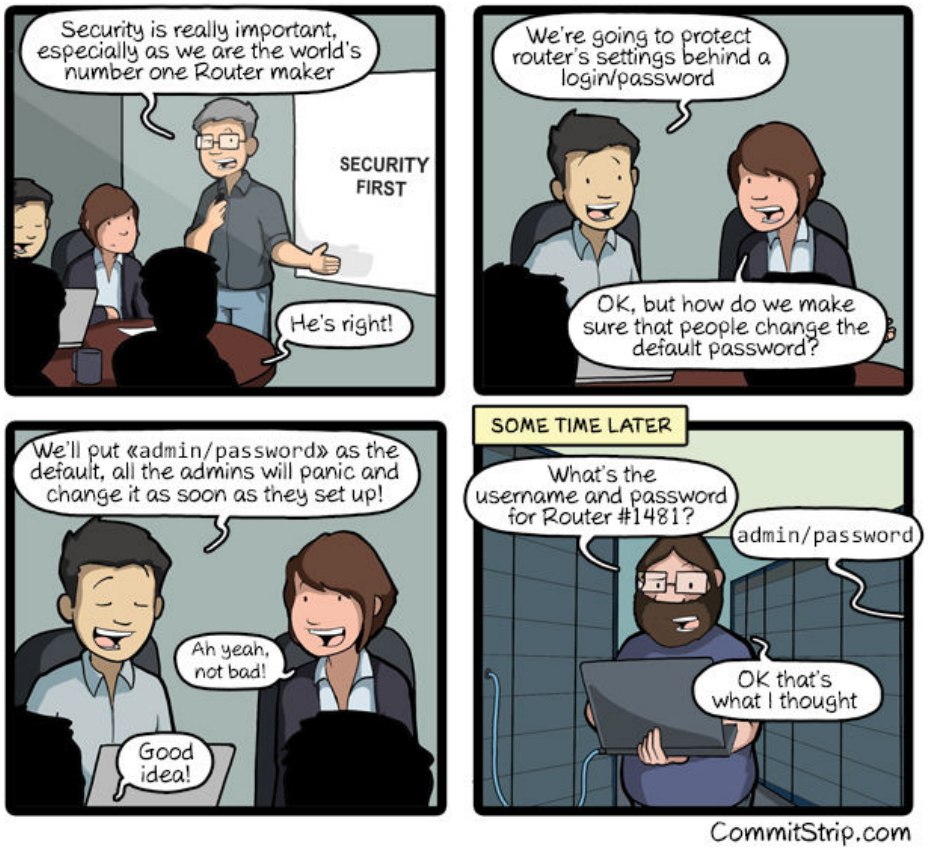
Devices infected by Mirai continuously scan the internet for the IP address of IoT devices. Mirai includes a table of IP Address ranges that it will not infect, including […] United States Postal Service and Department of Defense.

Mirai then identifies vulnerable IoT devices using a table of more than 60 common factory default usernames and passwords, and logs into them to infect them with the Mirai malware. **Infected devices will continue to function normally**, except for occasional sluggishness, and an increased use of bandwidth. A device remains infected until it is rebooted […] unless the login password is changed immediately, the device will be reinfected within minutes. […]

There are hundreds of thousands of IoT devices which use default settings, making them vulnerable to infection. Once infected, the device will monitor a command and control server which indicates the target of an attack.

[…] bypass some anti-DoS software which monitors the IP address of incoming requests and filters or sets up a block if it identifies an abnormal traffic pattern […] marshall more bandwidth than the perpetrator can assemble alone, and to avoid being traced.

Who is to blame, then?



Vendors could have updated, but..

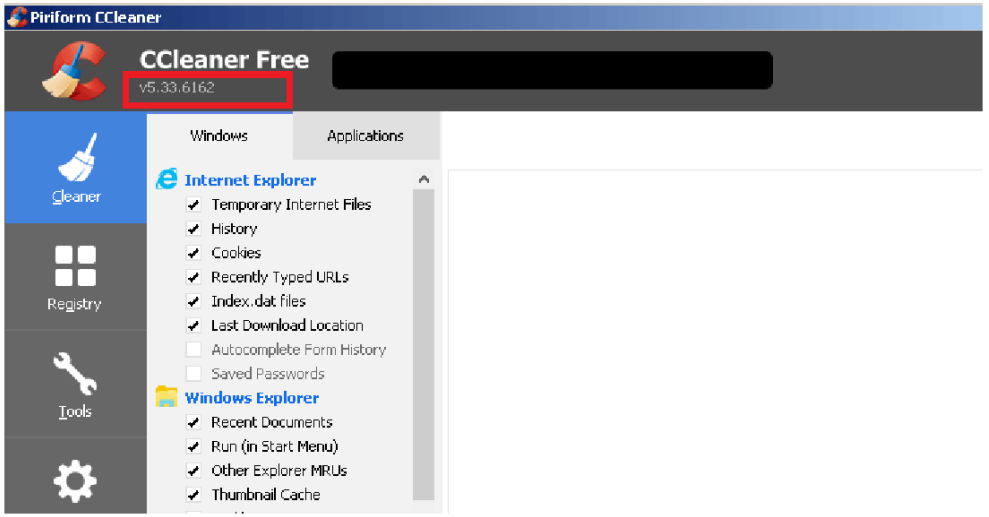
In the embedded world, **updates are rarely modular**. The most common situation is an update completely replaces a major part of the binaries, somethimes even deleting existing settings.

Let's imagine what could happen..

A new malware strain called BrickerBot is bricking Internet of Things (IoT) devices around the world by corrupting their storage capability and reconfiguring kernel parameters.

Detected via honeypot servers maintained by cyber-security firm Radware, the first attacks started on March 20 and continued ever since, targeting only Linux BusyBox-based IoT devices.

Detected on 13 September, the malicious version of CCleaner contains a multi-stage malware payload that steals data from infected computers and sends it to attacker's remote command-and-control servers.



Moreover, the unknown hackers signed the malicious installation executable (v5.33) using a valid digital signature issued to Piriform by Symantec and used Domain Generation Algorithm (DGA), so that if attackers' server went down, the DGA could generate new domains to receive and send stolen information.

Updates, yes, but the right way

Updates should be signed, with a properly protected private key and widely known associated public key

Updates should be modular, and only affect necessary modules

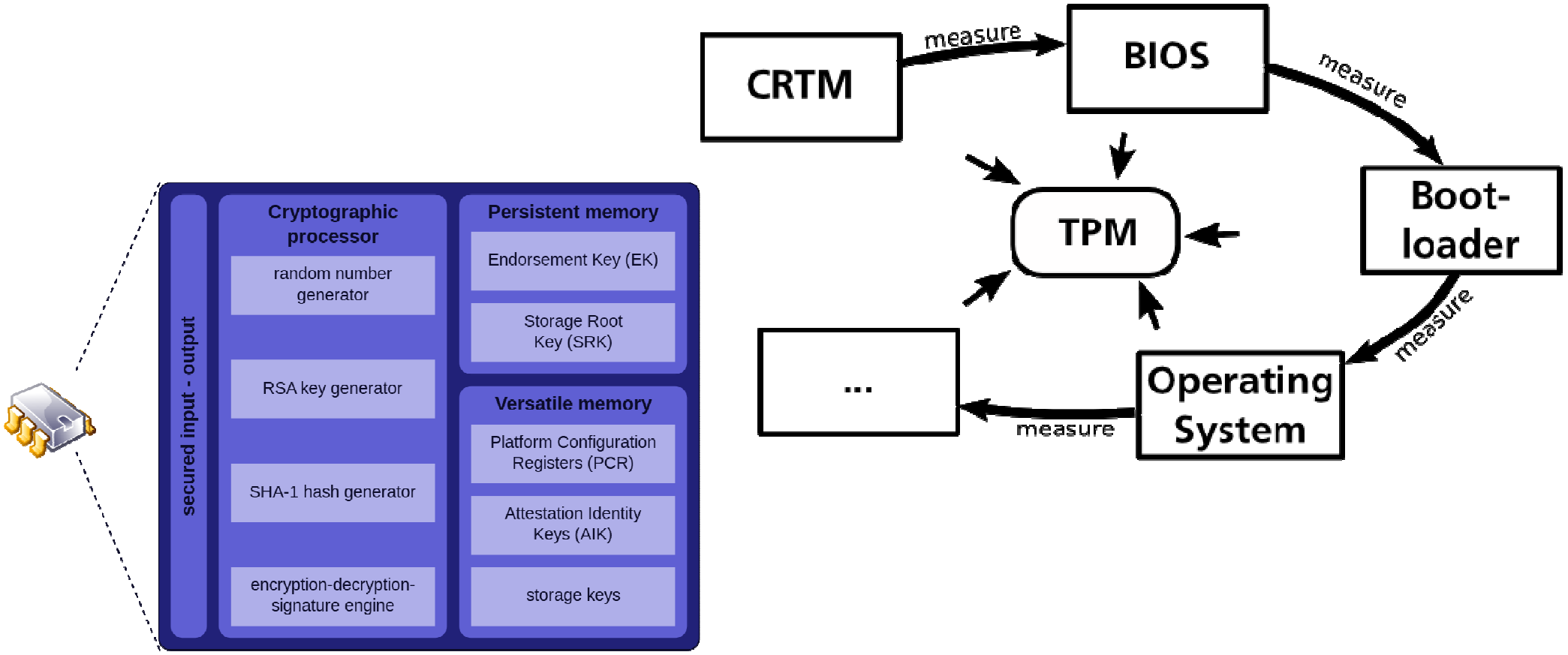
Updates shoud be thoroughly tested, **including versus security risks**: automated tests, fuzzing, CVE checking, …

Updates should be strongly protected against their **integrity**

Updates should be **enforced**: if you can live with the previous version, you'll be reluctant to upgrade. Create urgency, make it necessary.

Updates are just a particular case of tampering with your software base, only here it's with the user consent. But how can you protect against software tampering?

Trusted Computing

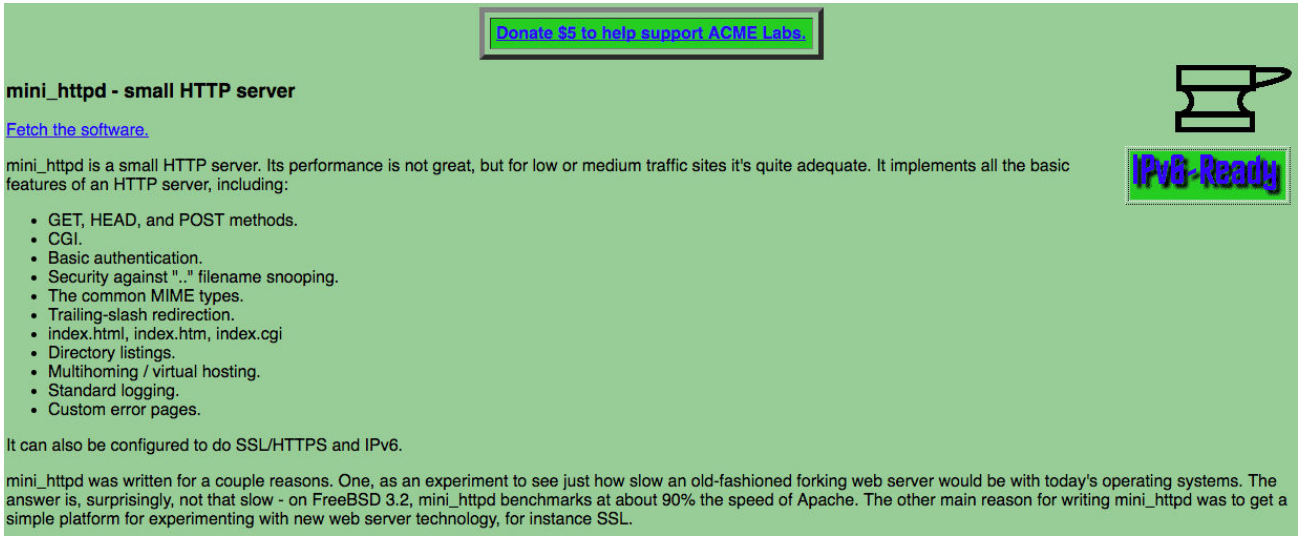


Be careful what you embed

This is **not** supposed to be a production ready web server

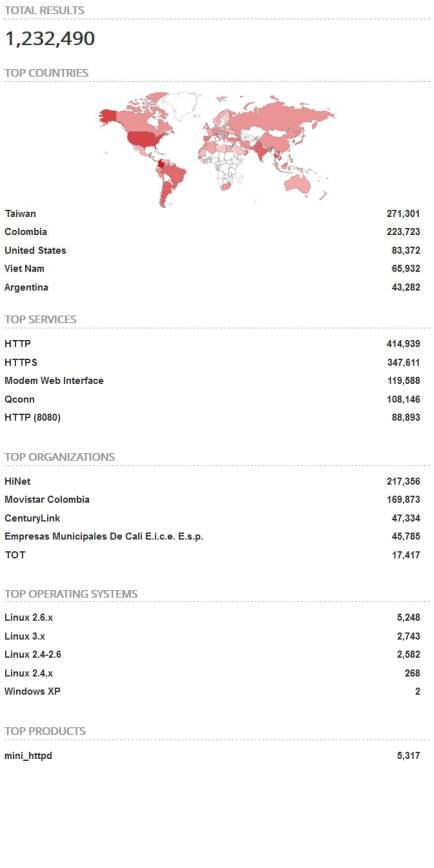
But, hey..

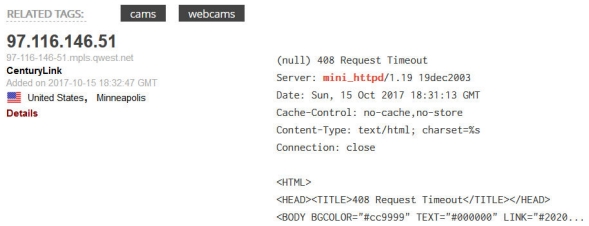
- It's free
- It's feature-rich
- It's easier to configure than Apache or Nginx
- It even has SSL!



OWASP found out in 2017 that there are many embedded systems running it. Let's check that.

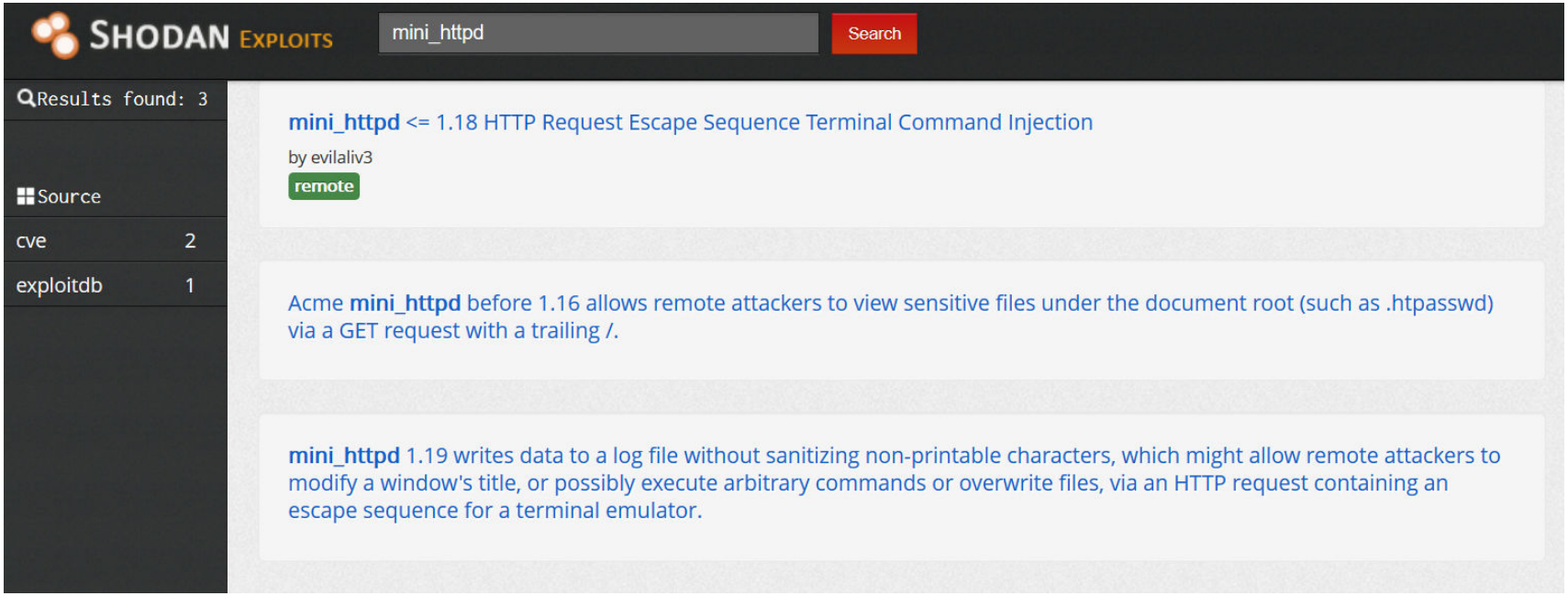
Introducing… SHODAN!





Come on, you've got to be kidding me!

At least there are no available exploits, RIGHT?



Speaking of bad things…

OWASP did a research in 2016-2017, using tools such as Shodan, to find out what was available on the web.

The results are not good. NOT. GOOD.

- Old versions of Busybox dating back to 2005
- Versions of Dropbear dating back to 2008
- Out of date versions of OpenSSL back to 2002
- Blank admin passwords
- Password stored locally in the clear
- Lack of updates

The success of Mirai such makes so much more sense.

The silver lining of Open Source

Depending on the license, Open Source allows you integrate more or less tightly and more or less openly

Consequences are multiple

- More and more products integrate Open Source components (that's excellent news)
- These components are often not updated
- Not even followed up by their integrators

End of the day, software with known vulnerabilities can still be found out in the fields with tools like Shodan; they're just awaiting to be hacked.

Capabilities (Without Application Licensing Restriction)	GPL (Linux)	Dual-GPL (MySQL)	LGPL/MPL (OpenOffice, Firefox)	Apache/BSD (Apache, FreeBST)
1) Download	✓	✓	✓	✓
2) Evaluate	✓	✓	✓	✓
3) Deploy	✓	✓	✓	✓
4) Redistribute	⊘ [1]	✓ [3]	✓	✓
5) Modify	⊘ [2]	⊘ [2]	⊘ [2]	✓ [4]

1) Application needs to be licensed under GPL if redistributed with the GPL asset.
2) Library code modifications need to be licensed under the same license as the originating asset.
3) Usually requires a commercial license from the copyright holder.
4) Although much more permissive than an OSI license, some BSD based licenses, such as Apache V2, still have some copyleft materials.

OWASP Principles of IoT Security

Assume a Hostile Edge - Edge components are likely to fall into adversarial hands. Assume attackers will have physical access to edge components and can manipulate them, move them to hostile networks, and control resources such as DNS, DHCP, and internet routing.

Test for Scale - The volume of IoT means that every design and security consideration must also take into account scale. Simple bootstrapping into an ecosystem can create a self denial of service condition at IoT scale. Security countermeasures must perform at volume.

Internet of Lies - Automated systems are extremely capable of presenting misinformation in convincing formats. IoT systems should always verify data from the edge in order to prevent autonomous misinformation from tainting a system.

Exploit Autonomy - Automated systems are capable of complex, monotonous, and tedious operations that human users would never tolerate. IoT systems should seek to exploit this advantage for security.

Expect Isolation - The advantage of autonomy should also extend to situations where a component is isolated. Security countermeasures must never degrade in the absence of connectivity.

Protect Uniformly - Data encryption only protects encrypted pathways. Data that is transmitted over an encrypted link is still exposed at any point it is unencrypted, such as prior to encryption, after decryption, and along any communications pathways that do not enforce encryption. Careful consideration must be given to full data lifecycle to ensure that encryption is applied uniformly and appropriately to guarantee protections. Encryption is not total - be aware that metadata about encrypted data might also provide valuable information to attackers.

Encryption is Tricky - It is very easy for developers to make mistakes when applying encryption. Using encryption but failing to validate certificates, failing to validate intermediate certificates, failing to encrypt traffic with a strong key, using a uniform seed, or exposing private key material are all common pitfalls when deploying encryption. Ensure a thorough review of any encryption capability to avoid these mistakes.

System Hardening - Be sure that IoT components are stripped down to the minimum viable feature set to reduce attack surface. Unused ports and protocols should be disabled, and unnecessary supporting software should be uninstalled or turned off. Be sure to track third party components and update them where possible.

Limit what you can - To the extent possible limit access based on acceptable use criteria. There's no advantage in exposing a sensor interface to the entire internet if there's no good case for a remote user in a hostile country. Limit access to white lists of rules that make sense.

Lifecycle Support - IoT systems should be able to quickly onboard new components, but should also be capable of re-credentialing existing components, and deprovisioning components for a full device lifecycle. This capability should include all components in the ecosystem from devices to users.

Data in Aggregate is Unpredictable - IoT systems are capable of collecting vast quantities of data that my seem innocuous at first, but complex data analysis may reveal very sensitive patterns or information hidden in data. IoT systems must prepare for the data stewardship responsibilities of unexpected information sensitivity that may only be revealed after an ecosystem is deployed.

Plan for the Worst - IoT systems should have capabilities to respond to compromises, hostile participants, malware, or other adverse events. There should be features in place to re-issue credentials, exclude participants, distribute security patches and updates, and so on, before they are ever necessary.

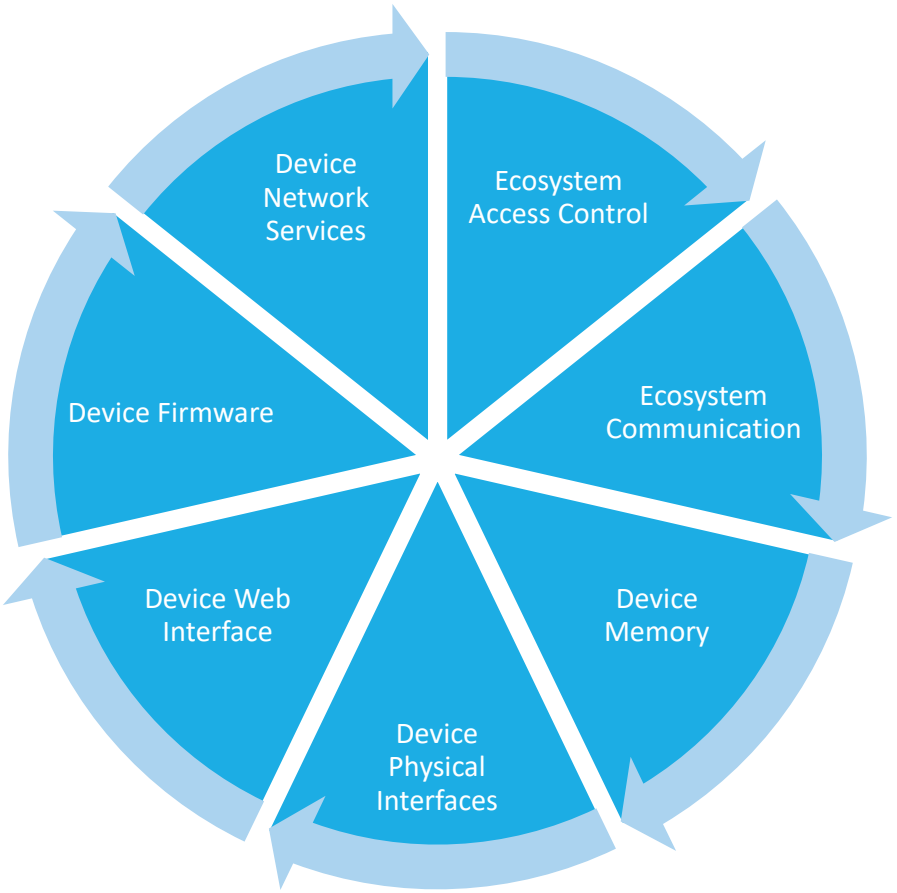
The Long Haul - IoT system designers must recognize the extended lifespan of devices will require forward compatible security features. IoT ecosystems must be capable of aging in place and still addressing evolving security concerns. New encryption, advances in protocols, new attack methods and techniques, and changing topology all necessitate that IoT systems be capable of addressing emerging security concerns for years after they are deployed.

Attackers Target Weakness - Ensure that security controls are equivalent across interfaces in an ecosystem. Attackers will identify the weakest component and attempt to exploit it. Mobile interfaces, hidden API's, or resource constrained environments must enforce security in the same way as more robust or feature rich interfaces. Using multi-factor authentication for a web interface is useless if a mobile application allows access to the same API's with a four digit PIN.

Transitive Ownership - IoT components are often sold or transferred during their lifespan. Plan for this eventuality and be sure IoT systems can protect and isolate data to enable safe transfer of ownership, even if a component is sold or transferred to a competitor or attacker.

N:N Authentication - Realize that IoT does not follow a traditional 1:1 model of users to applications. Each component may have more than one user and a user may interact with multiple components. Several users might access different data or capabilities on a single device, and one user might have varying rights to multiple devices. Multiple devices may need to broker permissions on behalf of a single user account, and so on. Be sure the IoT system can handle these complex trust and authentication schemes.

OWASP IoT Attack Surface Areas



OWASP IoT Security Framework

Edge

- Communications encryption
- Storage encryption
- Strong logging
- Automatic updates and/or version reporting
- Update verification
- Cryptographic identification capabilities
- No default passwords
- Strong local authentication
- Offline security features
- Configurable root trust store
- Device and owner authentication
- Transitive ownership considerations
- Defensive capabilities
- Plugin or extension verification, reporting and updating
- Secure M2M capabilities
- Secure web interface
- Utilize established, tested networking stacks and protocols
- Use latest, up to date third party components
- Capability to utilize hardware devices
- Support multi-factor authentication
- Support temporal and spacial authentication and functionality
- Tracks and contains data from potentially tainted (insecure) sources
- Features (interfaces) are disabled by default
- Written in a type safe programming language or subject to scrutiny
- Does not employ secrets in code
- Device monitoring and management capabilities

Gateway

- Multi-directional encrypted communications
- Strong authentication of components (edge, platform, user)
- Storage
- Denial of service and replay attack mitigation
- Logging and alerting
- Anomaly detection and reporting capabilities
- Use latest, up to date third party components
- Automatic updates and/or version reporting

Cloud

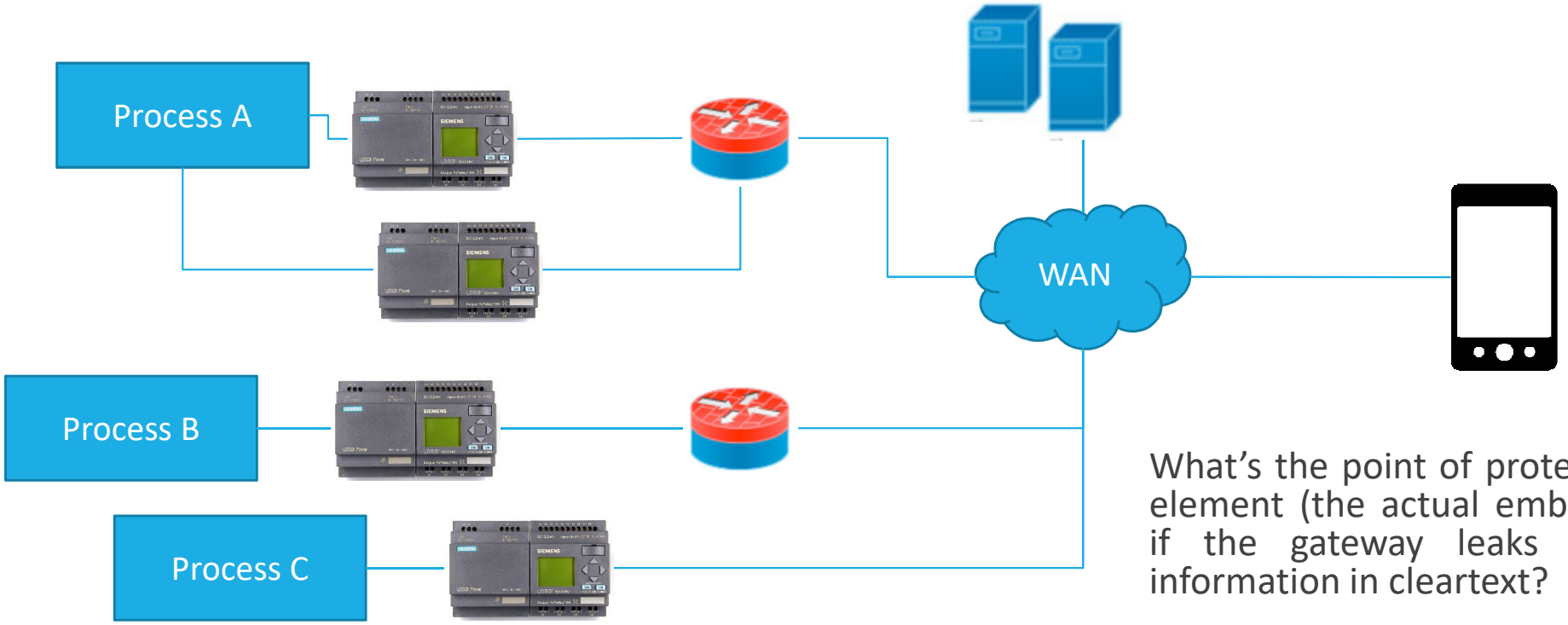
- Encrypted communications
- Secure web interface
- Authentication
- Secure Authentication Credentials
- Encrypted storage
- Capability to utilize encrypted communications to storage layer
- Data classification capabilities and segregation
- Security event reporting and alerting
- Automatic updates and update verification
- Use latest, up to date third party components
- Plugin or extension verification, reporting and updating
- Interface segregation and isolation based on utility (device, management interface, user interface, etc.)
- Application level firewall and defensive capabilities (IP blocking, throttling, account management, etc.)
- Ensure ecosystem segregation in the case of multi-tenant solutions
- Stack security considerations (no web UI to execute arbitrary code)
- Audit capability

Mobile

- Ensure mobile component enforces authentication requirements equal or greater to other components
- Local storage security considerations
- Capabilities to disable or revoke mobile components in the case of theft or loss
- Strong audit trail of mobile interactions
- Mobile application should perform cryptographic verification and validation of other components
- Encrypted communications channels
- Multi-factor authentication
- Capability to utilize mobile component to enhance authentication and alerting for other components

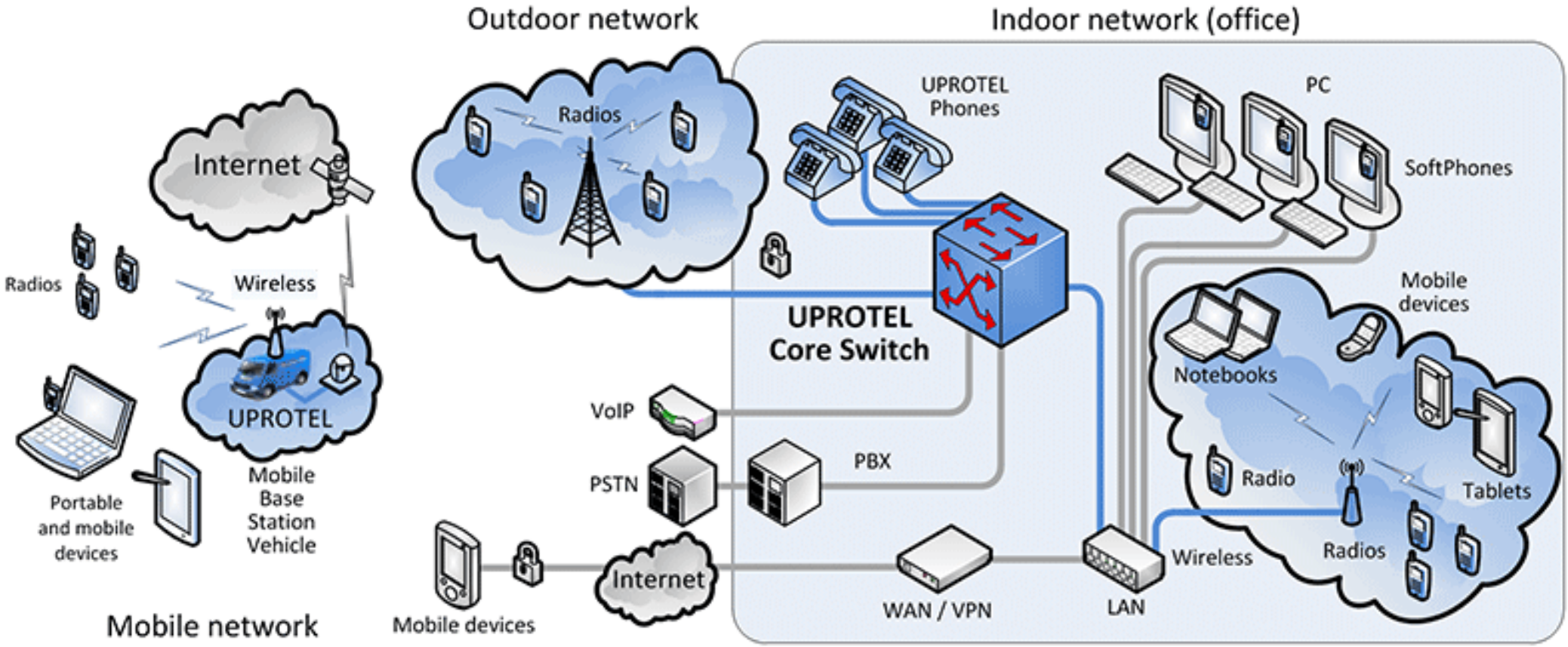
A system approach is necessary

OWASP defines an IoT security framework around four core elements: **edge, gateway, cloud, mobile**: that's the classic architecture of an IoT deployment, e.g. **industrial environment monitoring**.



What's the point of protecting the edge element (the actual embedded system) if the gateway leaks the complete information in cleartext?

A system approach is necessary (2)



Challenge: A flag roaming free!

On your **ESS1 – ICS Simulator** VM (**`student:student`**), you have three containers running:

- **ics-sensor** (**`10.2.0.1`**) simulates an industrial **temperature** sensor. It sends values **ranging from 0 to 255 to ics-gw using a proprietary protocol**. There is no SSH access towards the sensor and the vendor won't disclose how this protocol works.
- **ics-gw** (**`10.2.0.2/10.3.0.2`**) receives the data from the sensor, thanks to an API provided by the vendor. We somehow managed to reverse engineer the thing thanks to that document. Some b64 magic was necessary. The GW sends the data towards the historian using JSON-formatted messages:

  ```
  { "source": "ics-sensor@10.2.0.1", "type": "temperature", "unit": "celsius", "value": 75, "control": "E85693A9" }
  ```

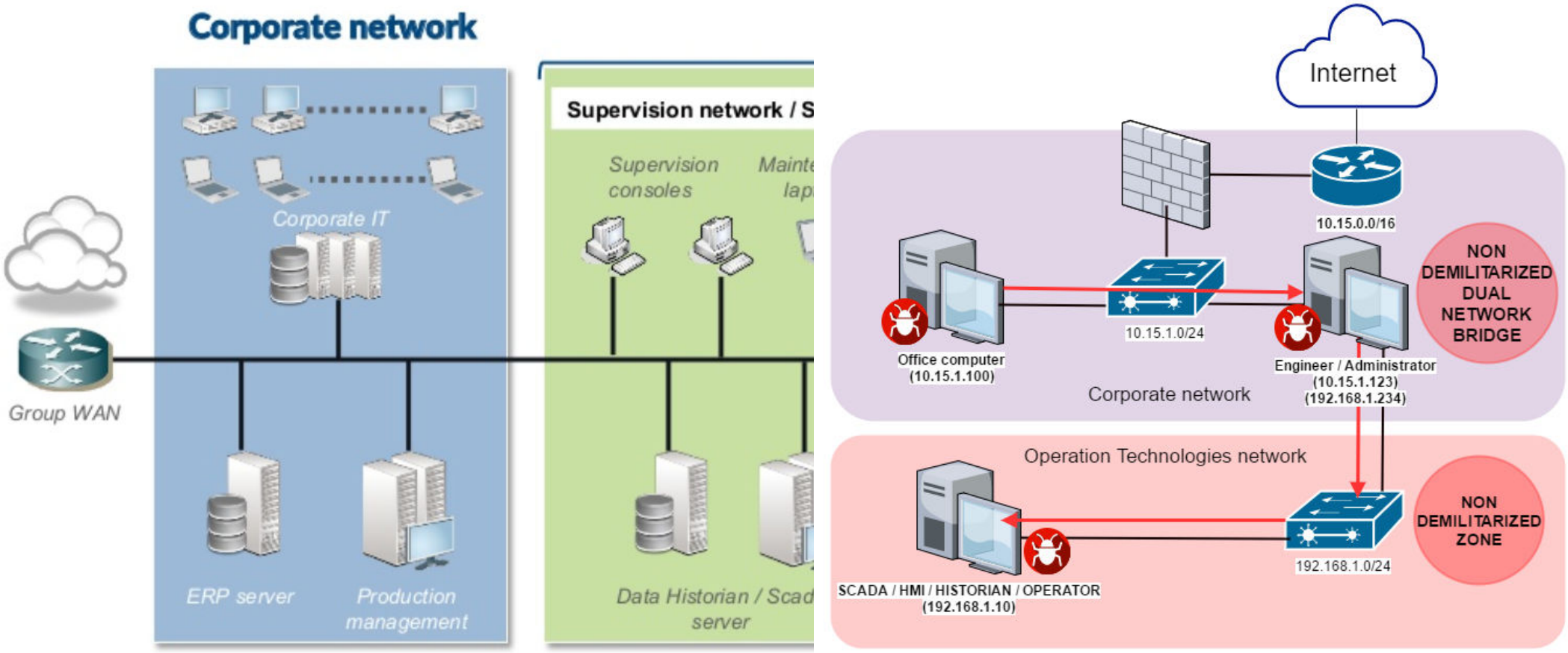
  You have a complete root access to **ics-gw (`5eh9kYaloGEPJEIYkn7G`)**.
- **ics-historian** (**`10.3.0.200`**) receives the data and stores them in flat-file databases. You have a low-privilege user access (**`historian:c48fb19a26`**) to this container, that is allowed to run `tcpdump` using `sudo`. The vendor considers your issue is out of support scope and won't help.

The flag is used as an authentication token for the exchanges between each component. Yesterday, the only sysadmin that knew it was fired. **Your mission is to recover the flag!**

There are several ways to solve this challenge. All the clues are there, you just have to use them.

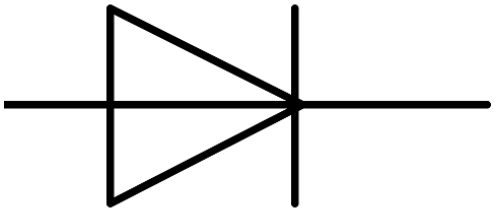
Note: You have no access to the running containers using the lxc commands, so you cannot directly open a console on the containers. Don't think it would have been that easy ☺

Systems interconnection



Data diodes

Challenge: Let's build a diode!



A data diode is the IP networks equivalent of the semiconductor: it enables **one-way data** transfer along a network link

Its main difference with a firewall is the implementation of unidirectionality by **physical means**: a higher level of assurance is delivered for this security function.

You are asked to propose, based on your existing knowledge, a solution to implement a data diode. **Don't cheat, close your laptops and set aside your smartphones!**

There is no flag to find here: the points will be allocated based on how trustable your solution is.

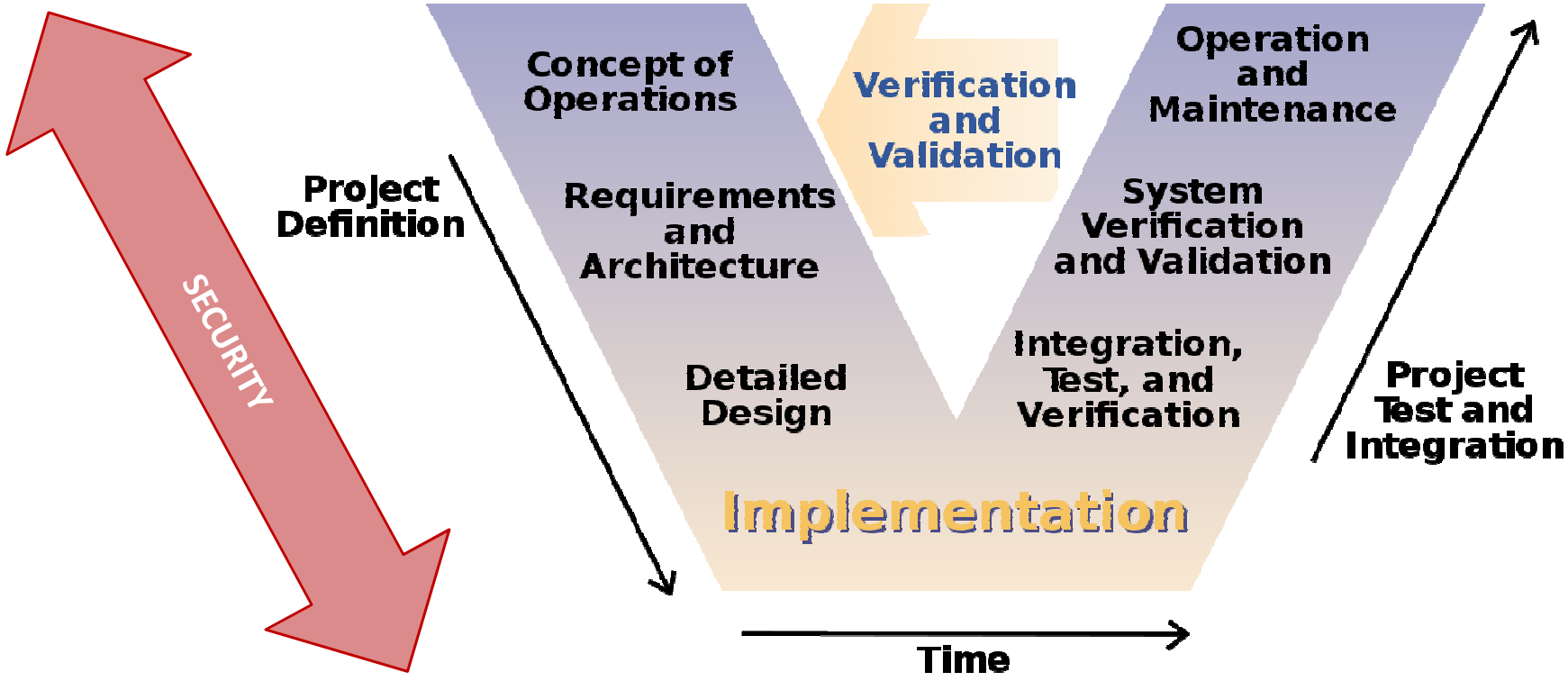
The simplest data diode: an optical fibre cable

There are many ways to build a data diode

- Optical fibre cable
- Software/FPGA implementation on copper-based networks
- Infrared links
- Optocoupler
- Serial link

Nothing comes close to the simplicity, performance and reliability of the optical fibre solution (you don't even need a power supply !)

The V-Model


SECURITY
Project Definition
Concept of Operations
Requirements and Architecture
Detailed Design
Implementation
Verification and Validation
Integration, Test, and Verification
System Verification and Validation
Operation and Maintenance
Project Test and Integration
Time

You'll experience this. That's a given.

Challenge: Killing the power grid

Nowadays, power grids become an integral part of the Industry 4.0 world:

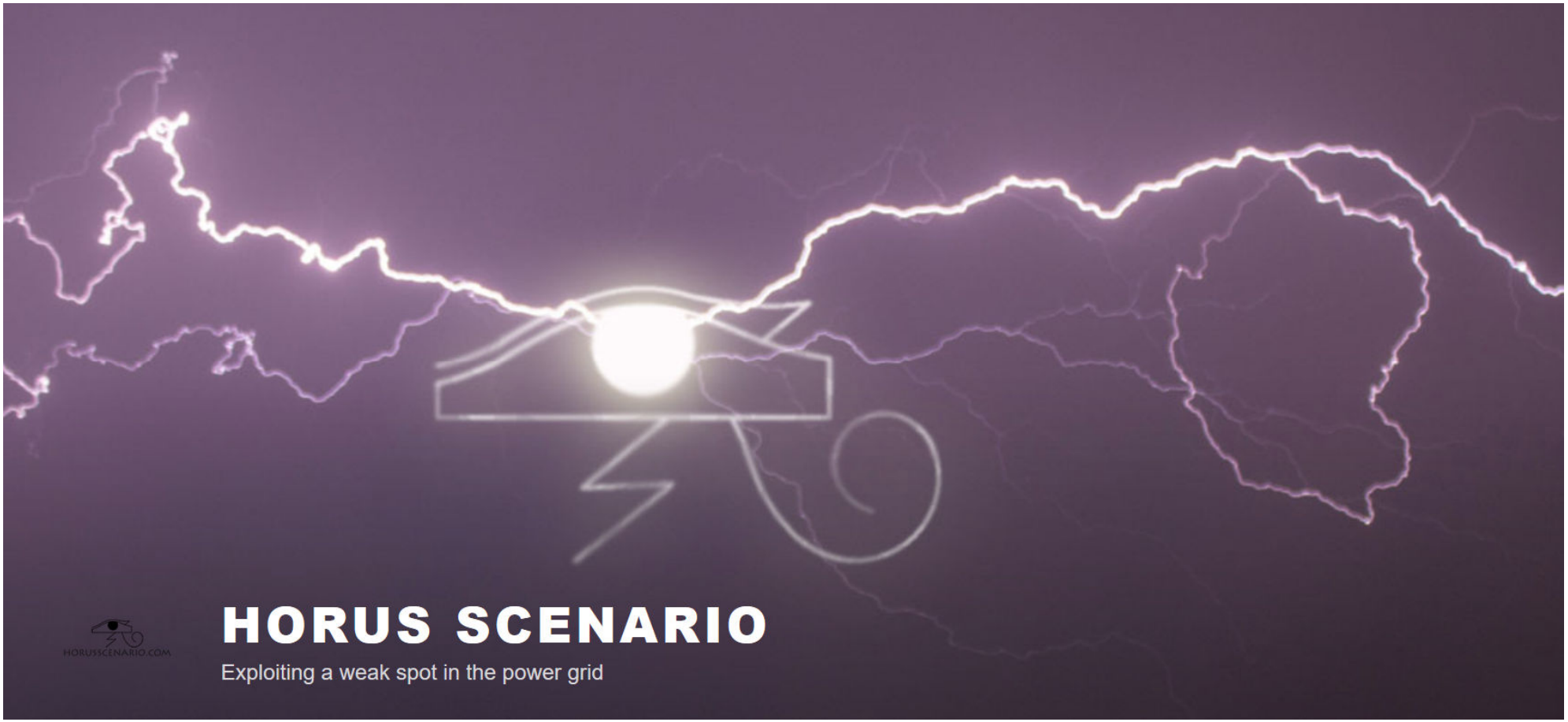
- Power plants have an increasing connection towards the ICT world
- The advent of personal solar panels transformed the power grid into a bidirectional system
- With technologies like smart grids and highly interconnected transport network, ICT is everwhere

While this presents excellent opportunities in terms of power saving and fossil energies phase out, it brings cybersecurity challenges into a power distribution world.

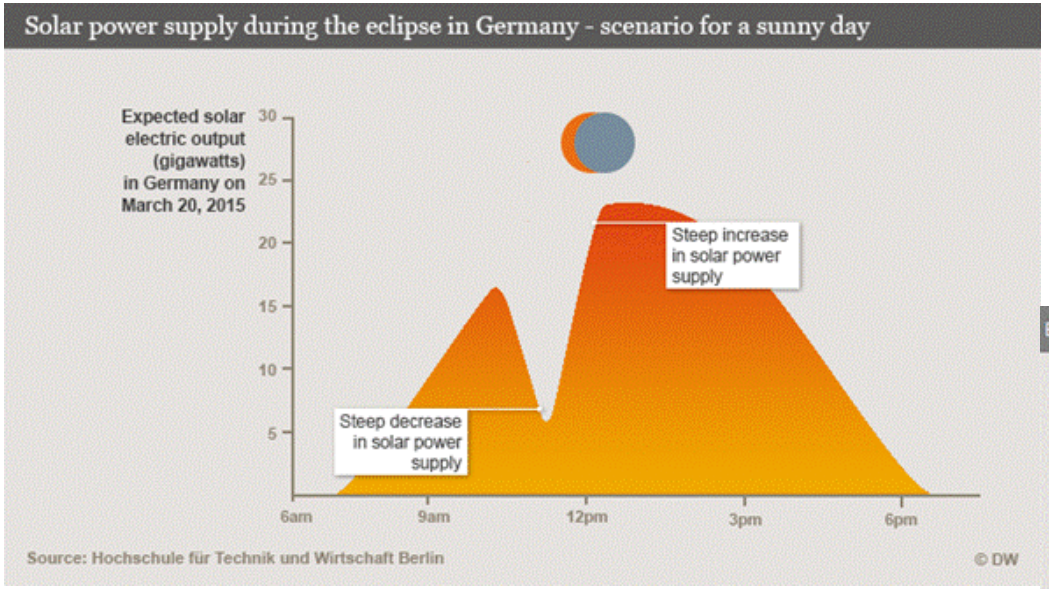
You are asked, based on your knowledge of power grids, to imagine a scenario where you take advantage of a weakness in the « cyber » power grid to create an damaging impact to end users.

There is no flag to find here: the points will be allocated based on how feasible the scenario you submit is based on existing knowledge in that field.

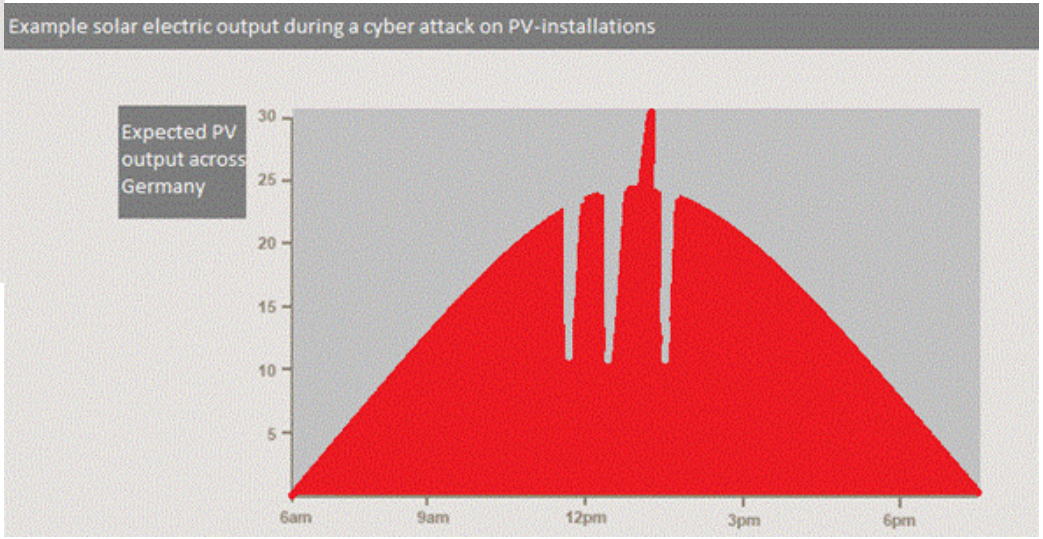
Just in case you need anymore convincing



The theoretical proof



[16]



Practical experiments

[16] […] *A live test setup was used to discover vulnerabilities in the market leading ,and likely the most secure, brand: SMA. Devices of this brand are generally considered to be the Mercedes among PV inverters* […]

Several laws and guidelines exists for power supply equipment and its cyber security […] *the PV installation businesses and PV inverter suppliers are in no way obliged to actually follow these laws and guidelines* […] *it can be expected that very little cyber security measures are in fact in place.* […]

These findings resulted in an attacker being able to remotely compromise the device completely. Not only was it possible to hack the device and control its flow of power, there were actually several different ways of doing this. […]

In total seventeen (later spliced to twenty-one findings as requested by the vendor) vulnerabilities were discovered. […]

Section 5
Certification framework & New legal challenges

PUTTING THE RIGHT EFFORT ON THE CRITICAL ASSETS

A word of introduction

EBIOS

Wikipedia: ***EBIOS (Expression des Besoins et Identification des Objectifs de Sécurité)*** *is a method for analysis, evaluation and action on risks relating to information systems. It generates a security policy adapted to the needs of an organization.*

Organized in five steps:

1. Circumstantial study - determining the context
2. Security requirements
3. Risk study
4. Identification of security goals
5. Determination of security requirements

Whenever you're designing an Information System (nowadays, an Embedded System is an IT System), **you start with analyzing how your system will be used, what it should protect, and versus with properties** ➔ Definition of a **FEROS** (Fiche d'Expression Rationelle des Objectifs de Sécurité).

Common Criteria – ISO 15408

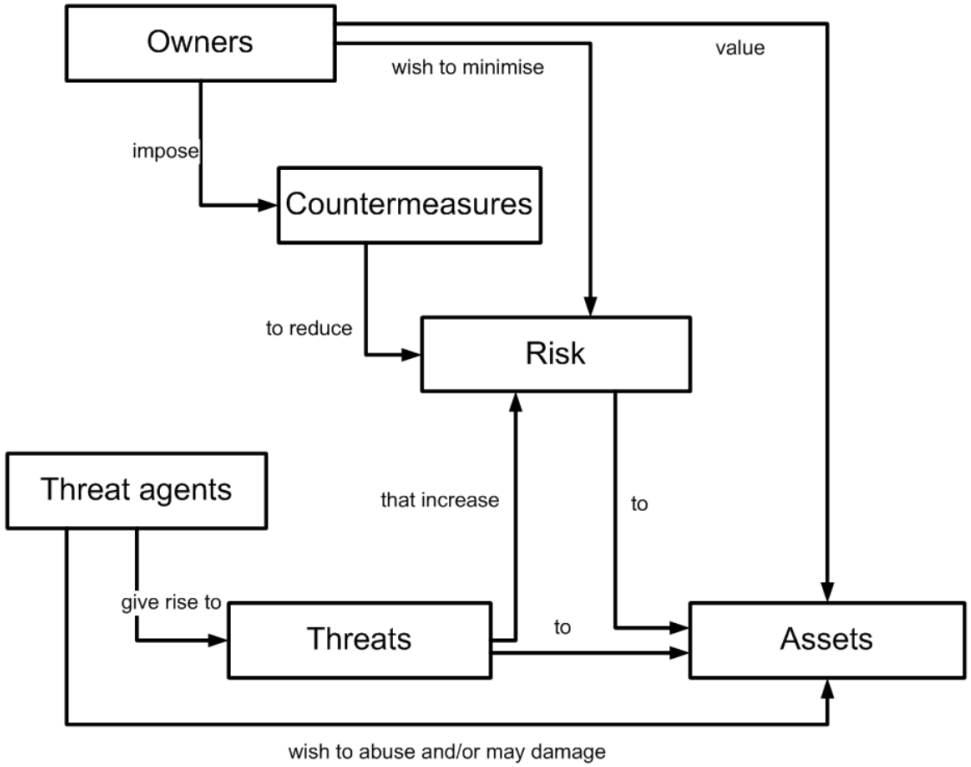
Wikipedia:

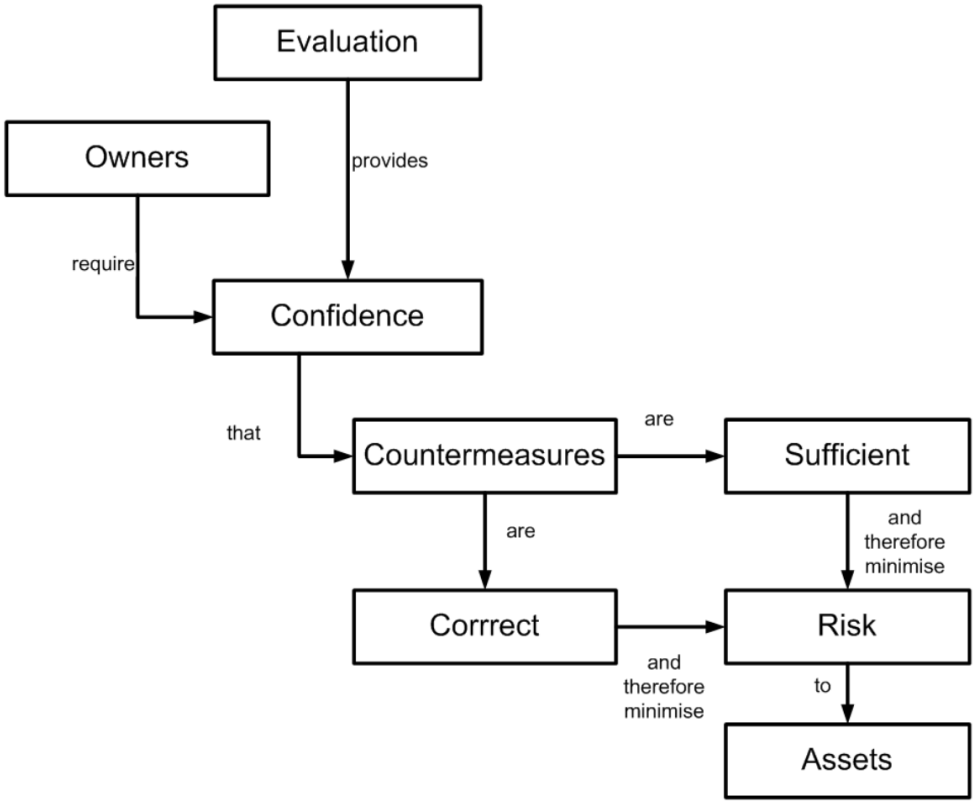
*The **Common Criteria for Information Technology Security Evaluation** (abbreviated as **Common Criteria** or **CC**) is an international standard (ISO15408) for computer security certification.* […]

specify their security functional and assurance requirements (SFRs and SARs respectively) through the use of Protection Profiles (PPs), vendors can then implement and/or make claims about the security attributes of their products, and testing laboratories can evaluate the products to determine if they actually meet the claims. […]

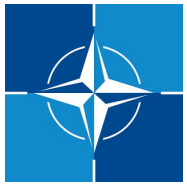
Common Criteria is used as the basis for a Government driven certification scheme […]

What Common Criteria does



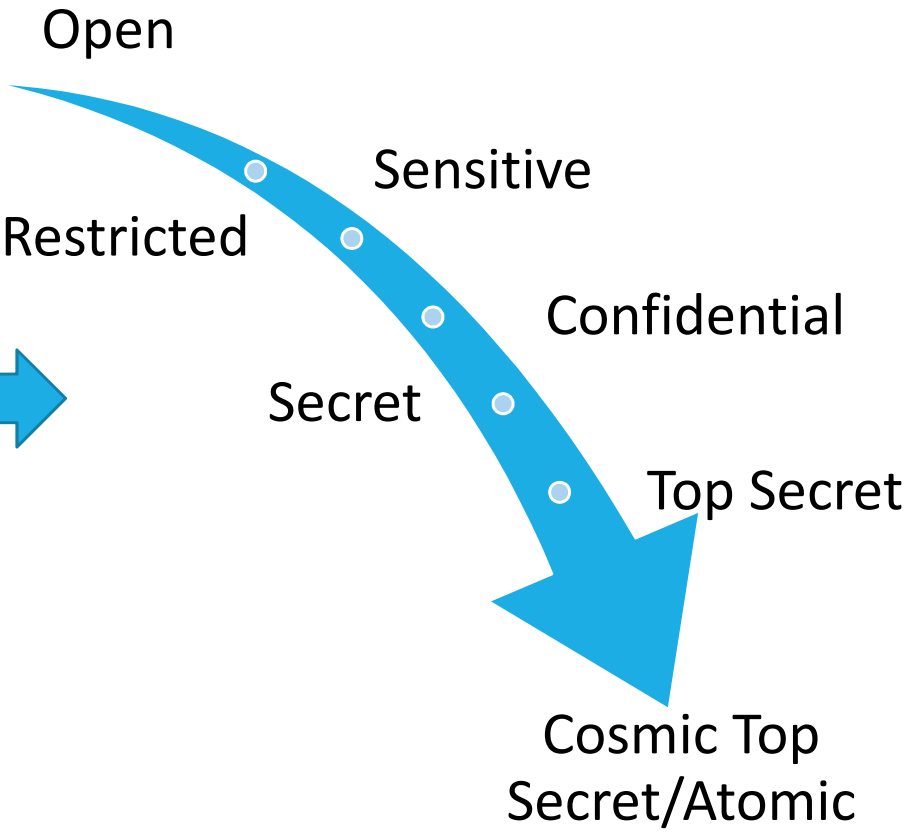


A link with data classification

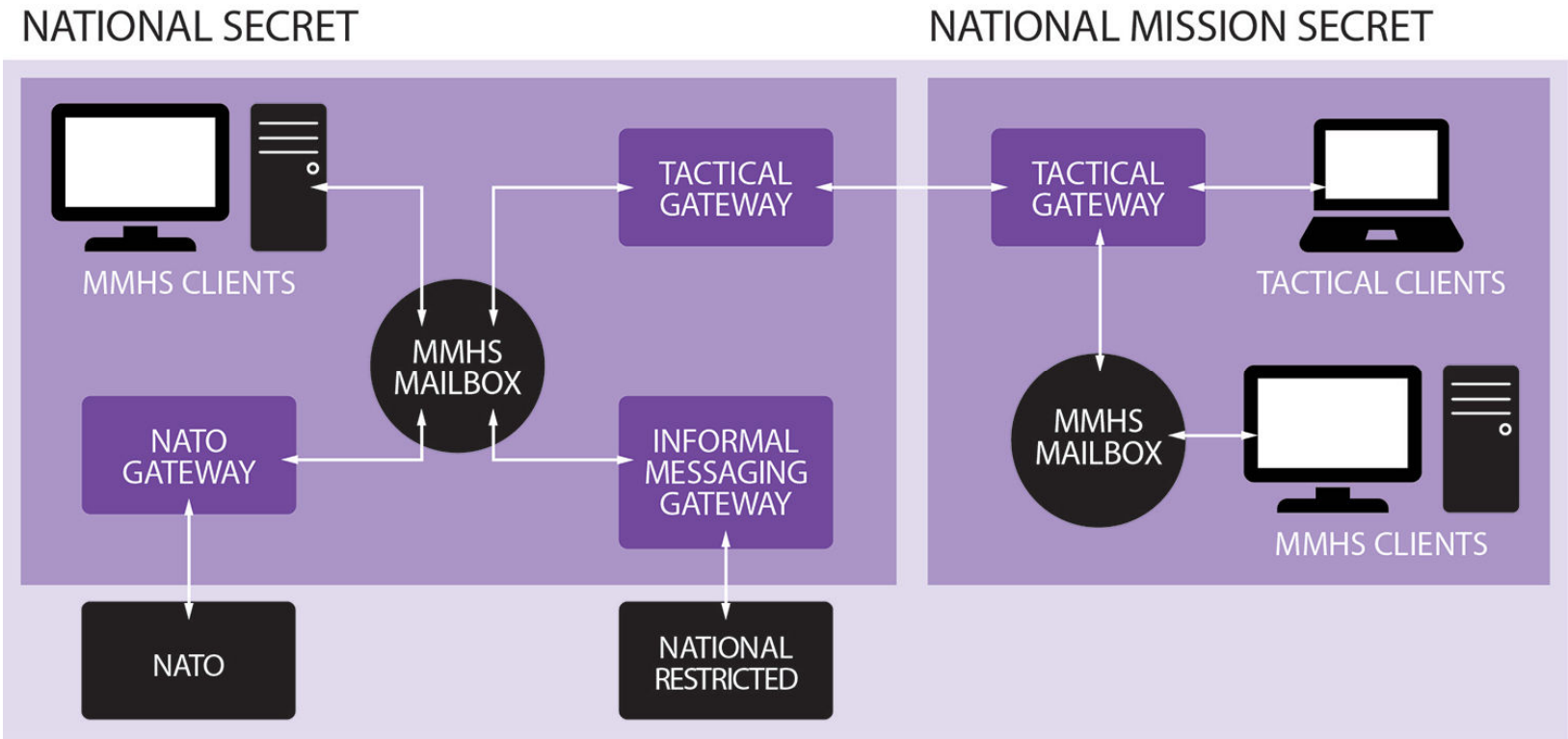


THALES

Specific regulations
TEMPEST
FIPS
Common Criteria



Not just a military matter anymore!



Cross domain solutions previously applied only to military situations are highly relevant to Industry 4.0, banking industry, and so on !

Belgium: December 11th, 1998 law

CHAPITRE II. - De la classification.

- **Art. 2.** Par classification, on entend l'attribution d'un degré de protection par ou en vertu de la loi ou par ou en vertu des traités ou conventions liant la Belgique.
- **Art. 3.** Peuvent faire l'objet d'une classification : les informations, documents ou données, le matériel, les matériaux ou matières, sous quelque forme que ce soit, dont l'utilisation inappropriée peut porter atteinte à l'un des intérêts suivants :
  - la défense de l'intégrité du territoire national et des plans de défense militaire;
  - l'accomplissement des missions des forces armées;
  - la sûreté intérieure de l'Etat, y compris dans le domaine de l'énergie nucléaire, et la pérennité de l'ordre démocratique et constitutionnel;
  - la sûreté extérieure de l'Etat et les relations internationales de la Belgique;
  - le potentiel scientifique et économique du pays;
  - tout autre intérêt fondamental de l'Etat;
  - la sécurité des ressortissants belges à l'étranger;
  - le fonctionnement des organes décisionnels de l'Etat;
  - la sécurité des personnes auxquelles […] des mesures de protection spéciales sont octroyées.

  Les matières nucléaires à usage pacifique […] ne sont pas classifiés au sens de la présente loi, sans préjudice des règles établies par ou en vertu des traités ou conventions qui lient la Belgique.
- **Art. 4.** La classification visée à l'article 3 comprend trois degrés : TRES SECRET, SECRET, CONFIDENTIEL.
  - Le degré TRES SECRET est attribué lorsque l'utilisation inappropriée peut porter très gravement atteinte à un des intérêts visés à l'article 3.
  - Le degré SECRET est attribué lorsque l'utilisation inappropriée peut porter gravement atteinte à un des intérêts visés à l'article 3.
  - Le degré CONFIDENTIEL est attribué lorsque l'utilisation inappropriée peut porter atteinte à un des intérêts visés à l'article 3.

  L'utilisation susvisée comprend notamment la prise de connaissance, la détention, la conservation, l'utilisation, le traitement, la communication, la diffusion, la reproduction, la transmission ou le transport.

NIS

Related to Operators of Essential Services (OES):

- ICT-controlled / ICT-based services: energy (power, gas, oil), transport (air, rail, water, road), banking, financial market infrastructures, health sector, drinking water supply & distribution, and Digital Infrastructure (IXPs, DNS service providers, TLD name registries).
- Criteria for OES identification
  - An entity provides a service which is essential for the maintenance of critical societal and/or economic activities
  - The provision of that service depends on network and information systems
  - An incident would have significant disruptive effects on the provision of that service.

Directive focused around **security incident prevention**:

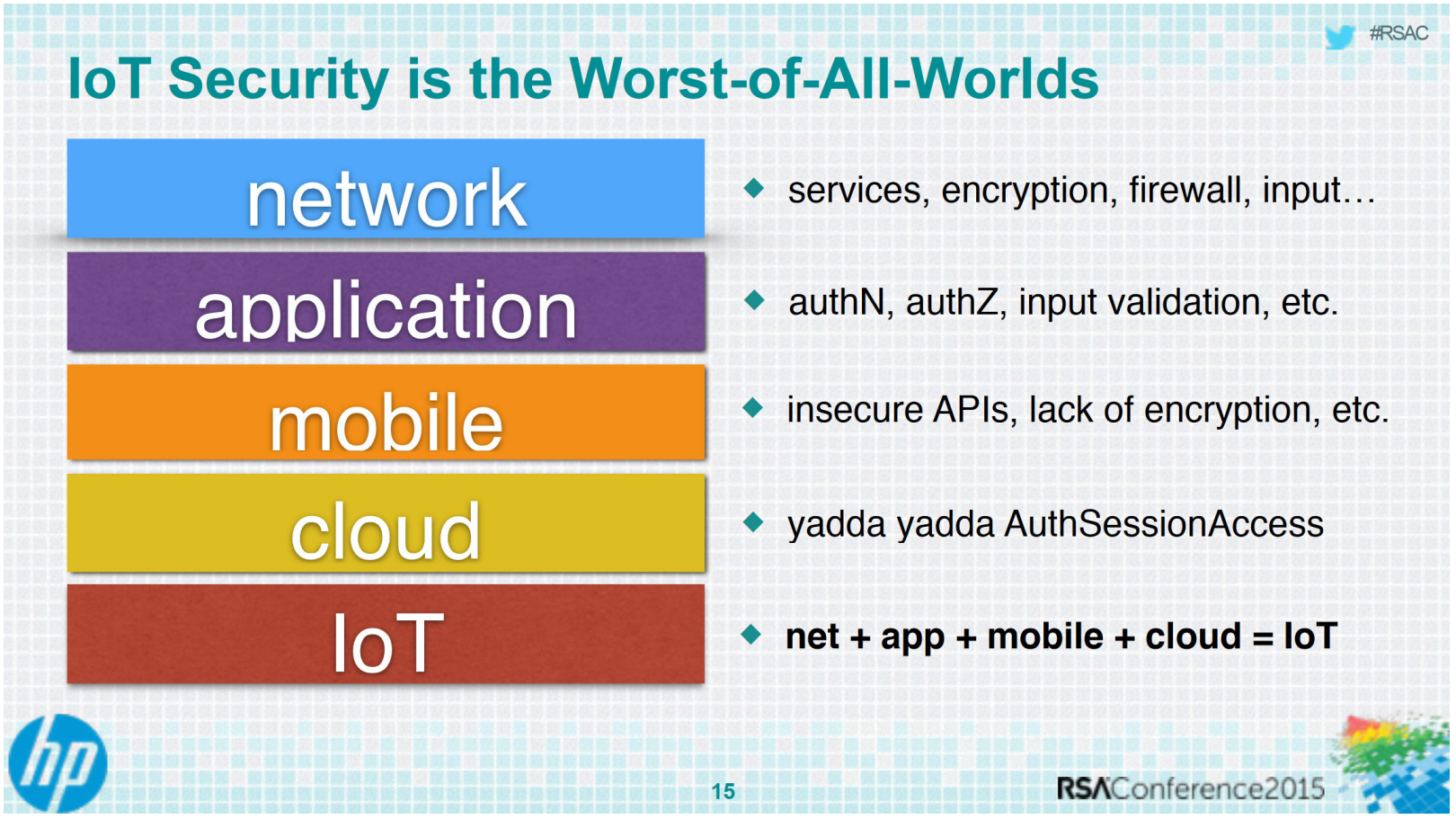
- Identification of critical assets onto which the directive applies (**rings a bell?**)
- Better coordination of European environment (CERT, NSA, ..)
- Mandatory disclosure of security incidents
- Fines in case the incident could have been avoided (e.g. negligence)


TERRITORIAL SCOPE
EU Establishments
Non-EU Established Organizations
Offer goods or services or engaging in monitoring within the EU.
THE PLAYERS
Data Subjects
Data Controllers
Data Processors
Supervisory Authorities
PERSONAL DATA
Identified
Identifiable
SENSITIVE DATA
Religious or Philosophical Beliefs
Racial or Ethnic Origin
Political Opinions
Trade Union Membership
Genetic Data
Biometric Data
Sex Life
Health
RESPONSIBILITIES OF DATA CONTROLLERS AND PROCESSORS
Security
Data Protection Officer (DPO)
Designate DPO if core activity involves regular monitoring or processing large quantities of personal data.
Record of Data Processing Activities
Maintain a documented register of all activities involving processing of EU personal data.
Data Protection by Design
built in starting at the beginning of the design process
Data Impact Assessment
For high risk situations
LAWFUL PROCESSING
Collection and processing of personal data must be for "specified, explicit and legitimate purposes" – with consent of data subject or necessary for
• performance of a contract
• compliance with a legal obligation
• to protect a person's vital interests
• task in the public interest
• legitimate interests
CONSENT
Consent must be freely given, specific, informed, and unambiguous.
GDPR
DATA BREACH NOTIFICATION
A personal data breach is "a breach of security leading to the accidental or unlawful destruction, loss, alteration, unauthorized disclosure of, or access to, personal data transmitted, stored or otherwise processed."
If likely to result in a high privacy risk → notify data subjects
Notify supervisory authorities no later than 72 hours after discovery.
RIGHTS OF DATA SUBJECTS
Transparency
Automated Decision Making
"Right not to be subject to a decision based solely on automated processing, including profiling."
Right to Erasure
Access and Rectification
Purpose Specification and Minimization
Right to Data Portability
ENFORCEMENT
Fines
Up to 20 million euros or 4% of total annual worldwide turnover. Less serious violations: Up to 10 million euros or 2% of total annual worldwide turnover.
Effective Judicial Remedies:
compensation for material and non-material harm.
INTERNATIONAL DATA TRANSFER
Adequate Level of Data Protection
Binding Corporate Rules (BCRs)
Privacy Shield
Model Contractual Clauses
TEACHPRIVACY
www.teachprivacy.com
Workforce awareness training by Prof. Daniel J. Solove
Please ask permission to reuse or distribute

Section 6
Final words

IT'S BEEN A HARD DAY'S NIGHT

Hey, let's be optimistic



[17]

One last OWASP extract

https://www.owasp.org/images/8/8e/Infographic-v1.jpg

Go and read stuff. Really. It's good for your brain and I'll help you being a kickass engineer.

Want to take this further?

Security is a very hot topic in the Engineering world. It's fresh, it's new, it's changing every day. The key to being relevant to **never stop learning**.

Moreover, **the solutions exist**. There are dedicated algorithms, research projects, standardisation efforts, working groups. **The information is just there, but too few people use it outside the cybersecurity community**.

It's costly, it takes time, it doesn't provide a immediate return on investiment, **it requires awareness and vision**.

Want to break the circle? You're the next ones that will be in charge. You'll design systems, you'll work on embedded devices, you'll participate in the decision process. **So be this clever person that will make a change**.

Date : 10/10/2017 - 20/05/2018

Du mercredi au vendredi de 13h00 à 17h00, le week-end et les jours fériés de 11h00 à 18h00

Fermeture du musée le lundi, le mardi, le 25 décembre et le 1er janvier

Lieu : Au Mundaneum - 76 rue de Nimy - 7000 Mons - Belgique

Prix : 7 euros - 5 euros (tarif réduit)

Plongez dans l'univers des écritures secrètes ! De Jules César à Edward Snowden, une nouvelle expo décode la cryptographie, pratique vieille comme le monde et plus que jamais d'actualité.

Impossible d'imaginer une société où toute l'information serait transparente et connue de tous. Depuis la nuit des temps, l'homme chiffre ses communications et tente de décrypter celles de ses ennemis : au VIe siècle av. J-C, Nabuchodonosor cachait des informations sous les cheveux de ses esclaves et, bien plus récemment, le génie de l'informatique Alan Turing contribua à écourter la Seconde guerre mondiale en cassant les codes de l'Enigma utilisée par les Nazis.

Le goût du secret se situe entre science et art, entre amour de la transgression et culture du « hacking ». Depuis les révélations de Wikileaks, la cryptographie s'impose comme un enjeu démocratique pour la confiance en notre société digitalisée. De l'Egypte des Pharaons aux mouvements sociaux actuels, d'innocents messages amoureux aux communications de guerre, en passant par le vote électronique et les trafics les plus divers, l'art des codes secrets se dévoile au Mundaneum, le centre d'archives montois connu comme le « Google de papier ».

Venez découvrir les machines utilisées pour le chiffrement, mais aussi les hommes et les femmes (dont de nombreux précurseurs belges) qui ont développé la discipline au fil des siècles.

Une exposition inédite … à ne pas garder secrète !

Embedded Systems Security

<eric.viseur@be.thalesgroup.com>

+32 497 92 36 80

Twitter: **EViseur**

Web: **https://ev1z.be/**