

Embedded Systems and Computer Security

ISAE

Rodolphe Ortalo
CARSAT Midi-Pyrénées
(rodolphe.ortalo@free.fr)
rodolphe.ortalo@carsat-mp.fr
<http://rodolphe.ortalo.free.fr/ssi.html>

ISAE – 2022/2023

Overall presentation (1/2)

- Fast paced computer security walkthrough
 - Security properties
 - Attacks categories
 - Elements of cryptography
 - Introduction to mandatory security policies
- Embedded systems and security
 - Specificities
 - Physical attacks (SPA, DPA)
 - TPM
- Software development and security
 - Security requirements and process
 - Static verification and software development tools
 - Common criteria / ISO 15408

Overall presentation (2/2)

- Case studies
 - Wireless networks
 - Next generation avionics systems
 - Network appliances
 - Mobile telephony
 - Gaming devices
- Wrap-up (if time permits)
 - IDS
 - Firewalls
 - Tripwire
 - Metasploit
 - Anti-virus

3

Overall presentation (1/2)

- **Fast paced computer security walkthrough**
 - Security properties
 - Attacks categories
 - Elements of cryptography
 - Introduction to mandatory security policies
- Embedded systems and security
 - Specificities
 - Physical attacks (SPA, DPA)
 - TPM
- Software development and security
 - Security requirements and process
 - Static verification and software development tools
 - Common criteria / ISO 15408

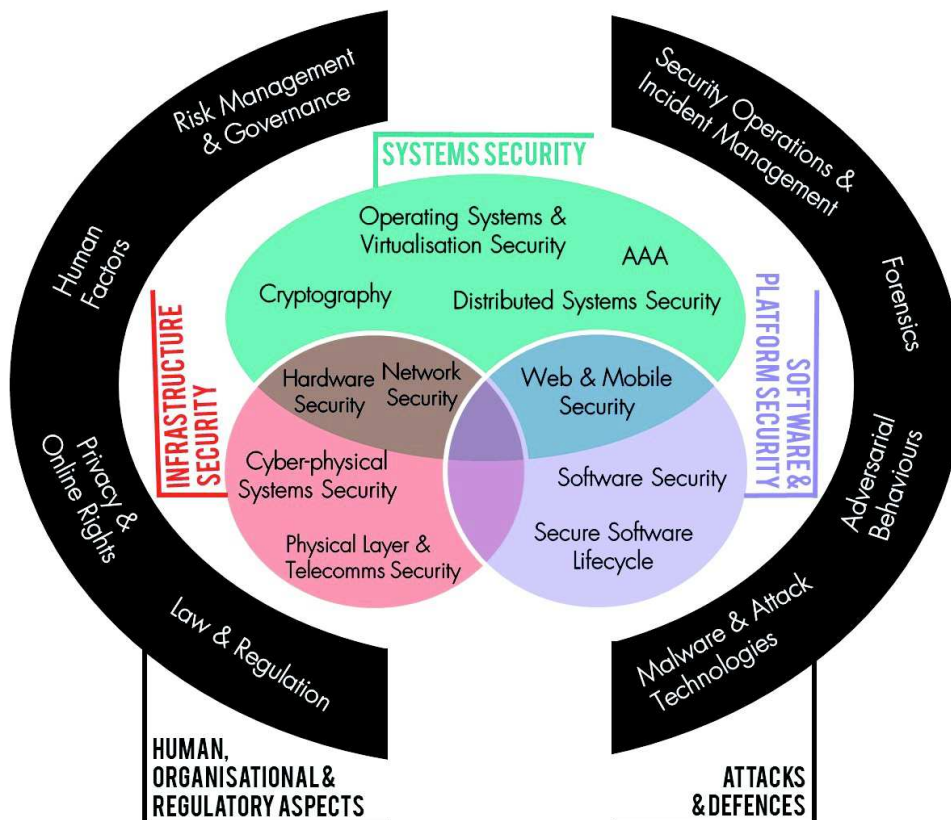
4

First, Computer Security

- Security
 - and privacy
- Computer science
 - and networks
- Also
 - IT security (& OT-)
 - Risk management (& control)
 - Defence (defined by its counterpart)

5

ISAE – 2022/2023



UK National Cyber Security Center
(NCSC) Body of Knowledge

7

Behind the acronyms, the codes

- NCSC
- is part of UK GCHQ
- once was government code & cypher school
 - GC&CS owned a mansion in Bletchley
 - where a few guys and devices became rather well known after WWII
- even earlier was British Signal Intelligence
 - beginning in August 1914
- You always find such things in computer security
 - that's the fun part
 - after-the-fact

8

Overall presentation (1/2)

- Fast paced computer security walkthrough
 - **Security properties**
 - Attacks categories
 - Elements of cryptography
 - Introduction to mandatory security policies
- Embedded systems and security
 - Specificities
 - Physical attacks (SPA, DPA)
 - TPM
- Software development and security
 - Security requirements and process
 - Static verification and software development tools
 - Common criteria / ISO 15408

9

Very specific fault hypothesis

- Malicious faults

- human
- intentional
- with ill will
 - direct
 - or not
- and malice
 - disinformation
 - disguise



by Travelling Runes (CC BY-SA 2.0)

10

Basic properties - **Confidentiality**

- Property of information not to be revealed to non-authorized users
 - prevent users from reading confidential data, unless they are authorized
 - prevent authorized users from communicating confidential data to non-authorized users

Basic properties - **Integrity**

- Property of information to be accurate
 - prevent inadequate alteration (creation or destruction) of data (either incorrect or performed by non-authorized users)
 - no user should be able to prevent a legitimate modification

12

Basic properties - **Availability**

- Property of information to be accessible when it is needed
 - allow access to authorized users for reading or writing
 - no user should be able to prevent authorized users from accessing information

13

What is information?

- Data
 - typed, generated, stored, transmitted, displayed, etc.
- «Meta-data » : associated to other data and accessed by computing processes
 - identities, names, addresses (user, computer, process, peripherals, etc.)
 - time (date of computation)
 - access rights
 - etc.

14

ISAE – 2022/2023

Other properties

- Anonymity = confidentiality of user identity
- Privacy = confidentiality of (personal data + user identity)
- Message authenticity = integrity of (content + sender identity + date + ...)
- Document authenticity = intégrité of (content + creator identity + date + ...)
- User authenticity = integrity of identity
- « Auditability » = availability of (who, what, when, where, ...) of an action
- Sender non-repudiation = availability of (sender identity + ...) + integrity of content
- Receiver non-repudiation = availability of (receiver identity + ...) + integrity of content
- Intellectual property protection = confidentiality of content (+ integrity of container)

15

Overall presentation (1/2)

- Fast paced computer security walkthrough
 - Security properties
 - **Attacks categories**
 - Elements of cryptography
 - Introduction to mandatory security policies
- Embedded systems and security
 - Specificities
 - Physical attacks (SPA, DPA)
 - TPM
- Software development and security
 - Security requirements and process
 - Static verification and software development tools
 - Common criteria / ISO 15408

16

ISAE – 2022/2023

Attackers and their motivations

- **Game** : exploration (to the limits), extend and apply knowledge, find new weaknesses, improve security : "**hackers**" ("pirates" = "**crackers**")
- **Emulation, sectarianism** : group of hackers : "**exploits**"
- **Vandalism** : strength demonstration, punish : "**web defacing**", **virus**, **worms**...
- **Political, ideological** : ex. CCC
- **Vengeance**
- **Profit : espionage, funds extortion** : unfair concurrency, organized crime
- **Cyber war, terrorism?**
- **Awareness raising, lobbying**
- **Abusive protection** : ex. SONY

17

Various attack classes

- Passive sniffing
- Interception
- Covert channels
- Cryptanalysis
- Repudiation
- Inference
- Masquerading
- Trapdoors
- Logical bomb
- Trojan
- Virus
- Worm
- Denial of service
- and complex attacks...

18

ISAE – 2022/2023

2015

- Innovations (?) in the automotive industry
 - VW
 - Jeep
- Reminder
 - Physical security > Org. security > Logical security



22

2016



23

2017

- Ransomware fever
 - WanaCry, NotPetya
 - extending into 2018, then 2019
- Opportunity for a bibliographical ref. too
 - Young A., Yung M., Cryptovirology : Extortion-Based Security Threats and Countermeasures, *17th IEEE Symposium on Security and Privacy, Oakland, 1996.*

~2017

- Actually very early 2018, but...
 - ... press coverage timeline is not always important
- Vulnerabilities involving CPU hardware design
 - Speculative execution, data/inst. Caches
 - Nicknames : Spectre, Meltdown
 - Academic names
 - Covert channels (circa. 1987)
 - Auxiliary channels (circa 1996)
- Computer apocalypse
 - Once again

25

ISAE – 2022/2023

WebAuthn vs. « March-2018! »

W3C Web Authentication: An API for accessing Public Key Credentials Level 1

W3C Candidate Recommendation, 20 March 2018

This version:
<https://www.w3.org/TR/2018/CR-webauthn-20180320/>

Latest published version:
<https://www.w3.org/TR/webauthn/>

Editor's Draft:
<https://w3c.github.io/webauthn/>

Previous Versions:
<https://www.w3.org/TR/2018/WD-webauthn-20180315/>
<https://www.w3.org/TR/2018/WD-webauthn-20180306/>
<https://www.w3.org/TR/2017/WD-webauthn-20171205/>
<https://www.w3.org/TR/2017/WD-webauthn-20170811/>
<https://www.w3.org/TR/2017/WD-webauthn-20170505/>
<https://www.w3.org/TR/2017/WD-webauthn-20170216/>
<https://www.w3.org/TR/2016/WD-webauthn-20161207/>
<https://www.w3.org/TR/2016/WD-webauthn-20160928/>
<https://www.w3.org/TR/2016/WD-webauthn-20160902/>
<https://www.w3.org/TR/2016/WD-webauthn-20160531/>

Issue Tracking:
[GitHub](#)

Editors:


26

2019

Local
recipe

Mais pourquoi le robot "Monsieur cuisine connect" de Lidl

est-il équipé d'un micro caché?



Deux passionnés en informatique ont réussi à pirater l'appareil pour

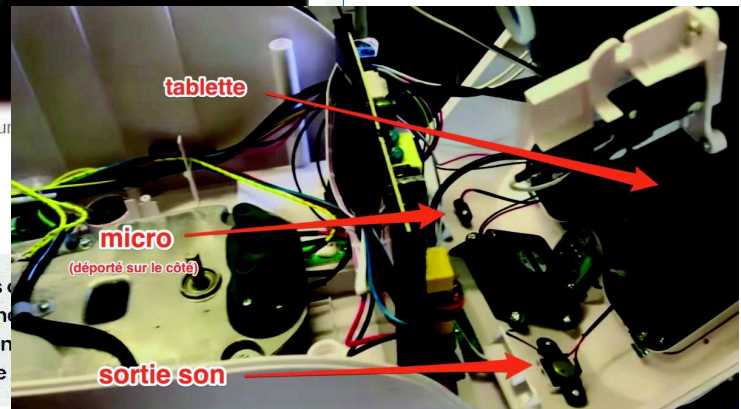
Capture Twitter @Siphonay

Publié le 13/06/2019 à 13:43, mis à jour à 14:47

f t p g in ✉

l'essentiel ▶ Dix jours après sa mise en vente, d'étranges découvertes... Cuisine Connect de Lidl, la réplique low cost du célèbre The... micro-inactif mais fonctionnel qui pourrait servir à espion... passionnés d'informatique ont pu s'appeler en utilisant le

La sortie du nouveau Monsieur Cuisine Connect de Lidl a fait beaucoup de bruit en France. Le 3 juin au petit matin, **les clients se sont précipités dans les magasins de l'enseigne pour acheter le robot** à un prix



28

2019

- Merck
 - 2017, NotPetya \$870m damage
 - Insurance claim
 - \$150m deductible, \$1.65bn cap
- Allianz, AIG
 - Denied coverage
 - « hostile or warlike » act or an act of terrorism (excluded)
- \$1.3bn claim in court in New Jersey atm for breach of contract, *featuring* :
 - a big pharmaceutical lab
 - a few big insurance companies
 - a few US/UK intelligence assessments
 - evil foreign hackers, presidents tweets, tons of bitcoins, ...

Winner!
(1/13/22)

30

2020

Proof that the current world can operate (more or less) without computer security ?

31

ISAE – 2022/2023

2021

- Trojan Source
 - Tricks with Unicode
 - <https://lwn.net/Articles/874951/>
- Arrests !?!?!
 - Ransomware groups (REvil) and RU-test
- MoonBounce
 - UEFI firmware tampering
- Log4j
 - Who did not hear about it ?
 - Mediatization impact ?
 - Interface bugs (see later) identification ?
 - Open source supply chain...

32

2022 – Cyber warfare effectiveness

News Latest Popular

2:42 pm Russia Preparing Cyberattacks on Ukraine's Critical Infrastructure – intelligence

2:10 pm **The New Russian Revolution Is at Hand**
EXCLUSIVE OP-ED

2:03 pm Kremlin Admits Call-Up "Errors" but no Decision to Close Border

1:54 pm Arson Attacks Sweep Across Russia

1:33 pm Putin's Decisions Show Russia "Struggling Badly" in Ukraine – Sullivan

1:32 pm WATCH: Video Shows Moment Russian Jet Shot Down Over Ukraine

12:46 pm Russia School Shooting Leaves 13 Dead, Including Children

12:42 pm **Time to Expose Putin as an Unreliable Ally**
EXCLUSIVE OP-ED


12:15 pm Ukrainian Heavy Industry Ground Down By Russia's War

11:56 am **What the Italian Election Means for Ukraine?**
EXCLUSIVE OP-ED

11:15 am Blinken: We Told Russians To Stop Talking About Nuclear Weapons

Russia Preparing Cyberattacks on Ukraine's Critical Infrastructure – intelligence


Sept. 26, 2:42 pm by UkrInform



Sept. 26, 2:10 pm **EXCLUSIVE**

The New Russian Revolution Is at Hand

Ilya Ponomarev



34

2022 (not last and not least)

- Family benefits, healthcare, hospital and... pensions.
- &french !

Campagne de tentative de phishing aux couleurs de l'Assurance Retraite - Message (HTML)

Fichier Message Aide Rechercher des outils adaptés

Supprimer Archiver Répondre Répondre à tous Répondre dans Teams Partager Actions rapides Déplacer Déplacer Indicateurs Modification Immersion Traduction Zoom Signaler le message FindTime

Campagne de tentative de phishing aux couleurs de l'Assurance Retraite

NationalisSupportBR
A BordeauxEnsembleDesUtilisateurs, ClermontAgents, CnavAdherentsSI, DijonToutLeMonde, LilleAgentsA-C, LilleAgentsD-G, LilleAgentsH-N, LilleAgentsO-T, LilleAgentsU-Z, LimogesToutLeMonde, LyonInfoAgentsTous, MarseilleDiffusionAgentsHORSIEGE, MarseilleDiffusionAgentsSIEGE, MontpellierAgents, NancyAgentsCarsatEnTotalite, 7 autres
Cc NationalisServicesBR, NationalisNiveauService

Bonjour,

Plusieurs signalements de tentatives de phishing usurpant l'Assurance Retraite reçues en interne ou en externe sont remontés (voir ci-dessous le type de message). Il s'agit de faux messages avec des liens douteux. **Nous vous invitons à être particulièrement vigilant vis-à-vis des mails reçus, notamment :**

- ne pas cliquer sur des liens que vous recevez dans des mails pointant vers des sites que vous ne connaissez pas et que vous n'avez pas l'habitude d'utiliser, comme des messages invitant à se connecter sur des espaces partagés online frauduleux,
- ne pas ouvrir de pièces jointes provenant d'expéditeurs inconnus ou de message dont la structure est inhabituelle voire vide,

Exemple de message reçu ce jour :

Votre demande de relevé de situation individuelle

l.assurance@retraite.cnaf.fr
A Gbaroni
dim. 27/11/2022 03:43

l'Assurance Retraite Tout savoir sur la retraite de la Sécurité Sociale
La retraite se base des données de l'Insee, du commerce et des services

Bonjour,

Votre relevé de situation individuelle est disponible sur notre site www.assurance-retraite.fr.

Pour le consulter ou le télécharger, [connectez-vous](#) à l'espace Services en ligne.

L'Assurance retraite vous remercie de votre confiance.

Cet e-mail vous est envoyé automatiquement. Merci de ne pas y répondre, car votre message ne sera pas traité en retour. Vous recevez cet e-mail car vous avez utilisé cet service en ligne sur le site www.assurance-retraite.fr. Si vous ne souhaitez plus recevoir d'e-mail relatif aux services en ligne de l'Assurance retraite à votre adresse, vous avez la possibilité de vous abonner dans le rubrique ["Gérer nos contacts"](#) de votre espace personnel sur www.assurance-retraite.fr. Conformément aux dispositions de la loi n° 78-17 du 06 janvier 1978 modifiée, vous disposez d'un droit d'accès, de rectification, de suppression et d'opposition aux données vous concernant. Appel 09 69 69 69 69

Cordialement,

Gilles Moya
Coordinateur / Référent ISI Support
BR
Direction des Systèmes d'Information
DOSI - Pôle Services
Tél. 9900 - 09 69 50 70 87

Retrouvez-nous sur www.assurance-retraite.fr et sur les réseaux sociaux

Cnav la sécurité sociale Agit avec vous

2022 & All time winner

- M\$!
- CVE-2022-37967

Timing of updates to address CVE-2022-37967

Updates will be released in phases: the initial phase for updates released on or after November 8, 2022 and the Enforcement phase for updates released on or after April 11, 2023.

November 8, 2022 - Initial deployment phase	✓
December 13, 2022 - Second deployment phase	✓
April 11, 2023 - Third deployment phase	✓
July 11, 2023 - Initial Enforcement phase	✓
October 10, 2023 - Full Enforcement phase	✓

36

2023

- AI news (undetectable.pdf)

Planting Undetectable Backdoors in Machine Learning Models

Shafi Goldwasser
UC Berkeley

Michael P. Kim
UC Berkeley

Vinod Vaikuntanathan
MIT

Or Zamir
IAS

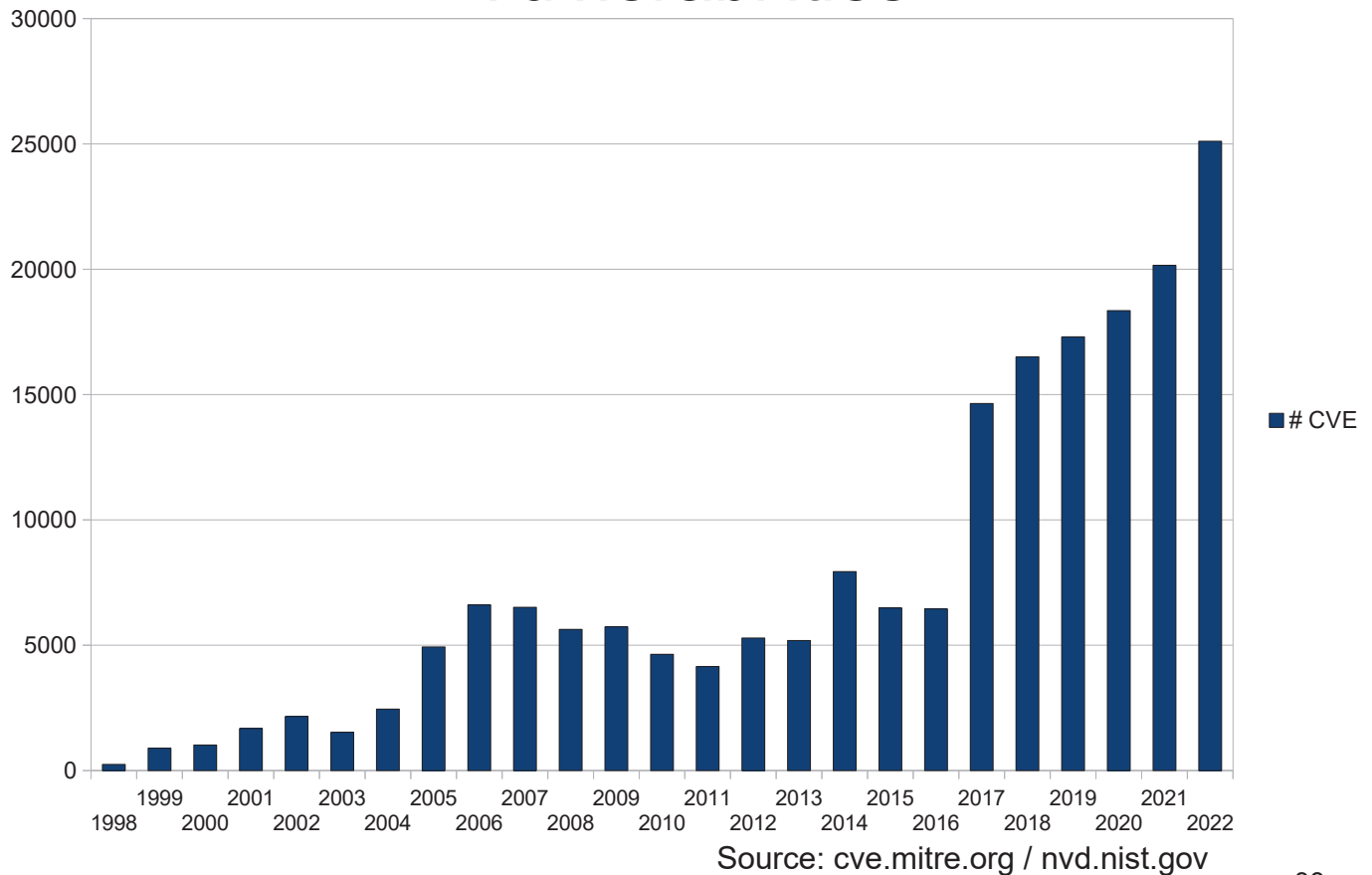
Abstract

Given the computational cost and technical expertise required to train machine learning models, users may delegate the task of learning to a service provider. Delegation of learning has clear benefits, and at the same time raises *serious concerns of trust*. This work studies possible abuses of power by untrusted learners.

We show how a malicious learner can plant an *undetectable backdoor* into a classifier. On the surface, such a backdoored classifier behaves normally, but in reality, the learner maintains a mechanism for changing the classification of any input, with only a slight perturbation. Importantly, without the appropriate “backdoor key,” the mechanism is hidden and cannot be detected by any computationally-bounded observer. We demonstrate two frameworks for planting undetectable backdoors, with incomparable guarantees.

- First, we show how to plant a backdoor in *any model*, using digital signature schemes. The

Vulnerabilities



38

Most famous vulns walkthrough

- Primarily because :
 - Not knowing them can cause vexation
 - Knowing them prevents over-estimation
 - Vulnerability lists are weak prevention tools
 - But there is need for demonstration
 - It illustrates pretty well software security problems

Buffer overflows

- Buffer overflows are a notorious problem
- Many exploits are based on them
- They are very easily introduced by simple programming mistakes
- BTW, very nice reference for *applied* secure programming
 - <http://www.openbsd.org/papers/>

*Most C examples taken or adapted from
“Puffy at Work”, Henning Brauer, Sven Dehmlow*

41

Buffer overflow

- What happens when a function is called (in C)?
 - General registers are saved on the stack
 - The CPU return address is computed and saved on the stack
 - Function arguments are stored too
 - The local variables of the function are also stored in the CPU stack
- Details are hardware dependent, but the overall idea is the same

42

Exemple

- A function

```
void function(char *str) {  
    char buffer[16];  
    strcpy(buffer, str);  
}
```

- A buffer overflow

```
int main(void) {  
    char *s = "Soy demasiado largo  
    para este espacio.";  
    function(s);  
}
```

43

ISAE – 2022/2023

Impact ?

- Program behavior is unpredictable
- Write to unexpected stack sections
- Can we overwrite the return address?
- With carefully chosen values, it is possible to enforce where the CPU execution returns at the end of the function
- This could be in code under our control, if we manage to inject it somewhere in memory (e.g. on the stack itself)



44

Not always that obvious

```
void function(int a, int b, int c) {
    char buffer1[8];
    char buffer2[16];
    int *ret;
    ret = buffer1 + GAP_TO_PC_ON_STACK;
    (*ret) += WIDTH_OF_1_CINSTRUCTION;
}

void main() {
    int x;
    x = 0;
    function(1, 2, 3);
    x = 1;
    printf("%d\n", x);
}
```

45

ISAE – 2022/2023

Not always that obvious

- `GAP_TO_PC_ON_STACK` and `WIDTH_OF_1_CINSTRUCTION` depend on the environment
 - e.g. : i386 linux 2.4 with gcc 2.95:12, 8
- This program prints 0 NOT 1
 - Possibly some kernel insult too
- Might be very interesting to overjump a line
 - Especially if there is a call to an authentication function or access control on that line

46

Prevent buffer overflows

- Be careful writing to buffers
 - Length check is mandatory
- Never do any tricks in C that you do not understand
 - *Never do any tricks in C*
 - Education first, then tools...
- strcpy and strcat are forbidden
 - use strncpy and strlcat (*if available*)

47

Format strings

```
int function(char *user) {
    fprintf(stdout, user);
}
```

- Problem: what if `user` is `"%s%s%s%s%s%s"`
- Most likely: program crash
- If not, program will print memory content

48

How does it work ?

- `printf` is called as a function
- functions get their arguments passed on the stack
- each format directive in a format string usually has a corresponding argument passed along
- for interpreting format directives, `printf` walks up the stack, expecting the right arguments to be there ; but, if they do not...

- Better :

```
int function(char *user) {
    fprintf(stdout, "%s", user);
}
```

49

Affected functions

- Any function using a format string
- Printing
 - `printf`, `fprintf`, *`sprintf`*, `snprintf`, `asprintf`
 - `vprintf`, `vfprintf`, *`vsprintf`*, `vsnprintf`, `vasprintf`
- Logging
 - `syslog`, `err`, `warn`

SQL Injection

- Building the query naively

```
statement = "SELECT * FROM users WHERE name = '" +
  userName+"'" AND pwd = '" +userPassword+"'" ;"
```

- What if

- userName is « ' OR '1'='1'; -- ' »
 - userPassword is not a problem anymore
- userName is « ' OR '1'='1'; DROP TABLES; -- ' »
 - The application is not a problem anymore either

- Mitigation

- Prepared statements (+ parse + execute)

```
SELECT * FROM users WHERE name = ? and pwd = ?;
```

- External libraries (for auth. or SGDB mapping)
- Parsing or escaping (**not** recommended)

51

SEL/**/ECT

- Obfuscation techniques are frequently used
- Sample ideas (for SQL injection)
 - Abuse of white space or comments
 - Fragmentation of the injected query
 - HTTP parameters
 - Comments (impl. specific ones, special comments)
 - Unprobed areas in packets
- Possible lessons
 - A full parser for parameter validation
 - Intrusion detection is not so easy
- NB: Numerous examples of code encryption or signature among attackers

Overall presentation (1/2)

- Fast paced computer security walkthrough
 - Security properties
 - Attacks categories
 - **Elements of cryptography**
 - Introduction to mandatory security policies
- Embedded systems and security
 - Specificities
 - Physical attacks (SPA, DPA)
 - TPM
- Software development and security
 - Security requirements and process
 - Static verification and software development tools
 - Common criteria / ISO 15408

53

ISAE – 2022/2023

Terminology

- Cryptology = cryptography + cryptanalysis
 - Cryptography (κρυπτος = hidden) :
messages non understandable by third parties
 - Cryptanalysis : discover secret(s), decypher
- Not to be confused with steganography
(στεγανος = covert) → invisible ink
watermark
- Cypher, encryption, decryption, clear (text),
cryptogram

54

Preamble (1/2)

- A domain of mathematics which exhibits some of the most significant advances of the end of 20th century, but
 - Mathematical proofs (of strength) are rare
 - Ciphers do break
 - Implementations do break too
 - Few experts (possibly few knowledgeable people)
- Difficult and counter-intuitive
 - example: encrypting twice can be dangerous

55

ISAE – 2022/2023

Preamble (2/2)

- Recent and unverifiable release of military control over cryptology
- Theroetical issues combine with many implementation difficulties
 - examples : random number generators, key generation, key protection, empty space padding, etc.
 - also at the level of hardware implementation
 - *covert channels (timing, energy, sound)*
 - *full module verification, e.g. :*
 - *out of order execution*
 - *TLB and caching interactions*
 - *memory (refresh, bugs, etc.)*

Spectre

Meltdown

Rowhammer

micro-code? (prediction and not crypto)

56

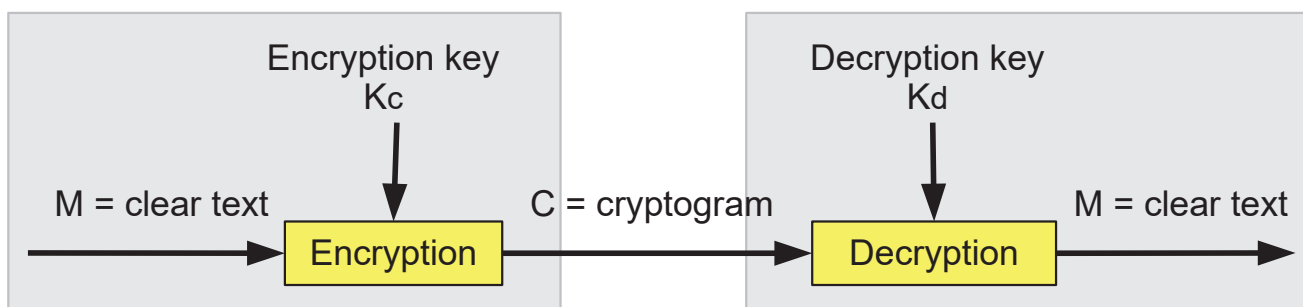
Crypto unifying theme

- The presence of an adversary
 - Eavesdropper, imitator, liar, forger, briber, stealer, extortionist, *whatever*
 - *Not* a god, but worst case
 - Big state-of-the-art super-computer and numerous best-in-world brains (*always*)
 - Controls the communications (aka Dolev-Yao hypothesis, *always*)
 - Could be an authorized (power) user
 - Quantum capable ?
- Always some hypothesis to check...

57

ISAE – 2022/2023

Encryption (confidentiality)



- Notation

encryption	$C = \{M\}_{K_c}$
decryption	$M = [C]_{K_d}$
- Confidentiality
 - Without knowing K_d , it must be « impossible » to find M
 - It must be « impossible » to find K_d , even knowing C and M (« (known) clear text » attack)
 - It must be « impossible » to find K_d , even knowing C while choosing M (« chosen clear text » attack)

58

Symmetric ciphers $K_c = K_d (= K)$

- All known ciphers until 1976 !
- Examples
 - DES (1976)
 - 56 bits key (+8 parity bits)
 - 64 bits blocks
 - AES (2002)
 - keys of 128, 192 or 256 bits
 - 128 bits blocks

59

DES : Data Encryption Standard (1975)

- Story
 - Base from IBM. With improvements from NSA.
 - The first algorithm scrutinized by NSA to become public... thanks to the standardization body.
- 64 bits blocks. Key of 56 bits + 8 bits (ex.: parity)
- Design oriented towards hardware implementation
- 3DES : common (generic) improvement
 - 112 bits key
- Huge public cryptology efforts associated to DES
- Feistel cipher family
- Lots of variants (ex.: *key-dependent S-boxes*)

60

AES : Advanced Encryption Standard (2001)

- Story
 - Selected by NIST from 15 proposals over a 5 year public selection process
 - Originally called Rijndael.
- 128 bits blocks. Keysize of 128, 192 or 256 bits
- Fast in both software and hardware
- Still resistant to open attacks (after a decade)
- Substitution-permutation network family
- Algebraic representation over $GF(2^8)$
- Now **very** wide adoption
 - AES-NI instruction set (Intel/AMD)
 - Common in most of encrypted flows nowadays

61

Symmetric ciphers modes of operation

- $$M = M_1 \cdot M_2 \cdot \dots \cdot M_n \quad C = C_1 \cdot C_2 \cdot \dots \cdot C_n$$
- ECB – *Electronic Codebook*
 - $C_i = \{M_i\}_K$
 - $M_i = [C_i]_K$
 - CBC – *Cipher Block Chaining*
 - $C_i = \{M_i \oplus C_{i-1}\}_K$
 - $M_i = C_{i-1} \oplus [C_i]_K$
 - IV sort of M_0
 - Stream ciphers
 - CFB – Cipher Feedback Mode
 - OFB – Output Feedback Mode

62

Public key ciphers $K_c \neq K_d$

- Knowing K_c , it must be «**impossible**» to find K_d
 - K_d is private (one must know K_d to decrypt)
 - K_c is public (everyone can encrypt): notion of public keys directory
- Ex.: RSA (1976)
 - (Probably) based on the (big) numbers prime factorization problem
 $e \cdot d \equiv 1 \pmod{(p-1)(q-1)}$ $K_c = \{pq, e\}$ $K_d = \{p, q, d\}$
- Ex.: El Gamal (1985)
 - Based on the discrete logarithm computation problem in finite fields
 - $y = g^x \pmod{p}$ $K_c = \{x\}$ $K_d = \{y, g, p\}$

63

One-time pad : perfect cipher

- The key is a serie of random bits as long as the message and the algorithm is exclusive-or
 - $C_i = \{M_i\}_{K_i} = M_i \oplus K_i$
 - $M_i = [C_i]_{K_i} = C_i \oplus K_i$
- According to information theory (Shannon), this is a perfect cipher (the key must **never** be reused)
 - Not very convenient
 - Possible

64

exclusive-or : brown paper bag cipher

- $C = M \oplus K$ et $M = C \oplus K$
 - No security
 - Compute $C \oplus C \gg k$ with $k = \{ 1, 2, \dots \}$ and count identical bytes. The coincidence indice indicates the key length n (in bytes).
 - $C \oplus C \gg n = M \oplus M \gg n$ removes the key.
 - Find the clear text using intrinsic redundancy of the original message (1,3 bit of information per byte in ASCII english for example).
 - Few minutes cryptanalysis.
- NB: Vigenère polyalphabetical cipher (1523-1596)

65

Strengths of symmetric ciphers

- Speed
 - ~ 1 Gb/s in hardware
 - ~ 100 Mb/s in software
- « Short » keys
 - 80 bits typically to withstand brute force attacks (today)
- Convenient to encrypt personal files (no need to share a key)

66

Weaknesses of symmetric ciphers

- To communicate, the secret key must be shared
 - sender and receiver have to trust each other, and both carefully protect the secret key
- How to distribute or renew the key?
 - Encrypt the new session key with the old one
 - Encrypt the session key with a device-specific key ⇒ trusted keys repository (directory)
 - Use a public key algorithm (Diffie-Hellmann)
 - Quantum cryptography
 - Avian carrier

67

Strengths of public key ciphers

- No trust needed between sender and receiver
- « **Easy** » key management
 - Public directory of public keys or peer to peer exchange
 - The private key must « **never** » be sent
- Allow for new kind of usage : symmetric keys distribution, electronic signature, certificates, etc.

68

Symmetric keys agreement

- Example : Alice generates a random (symmetric) session key K and encrypt it with the public key of Bob

- Exemple : Diffie-Hellmann

Alice randomly generates :

n : big prime number with $(n-1)/2$ prime
and chooses g = generator of a subgroup q de n
(typically, $g = 2$, $q = (n-1)/2$)

x (Alice's secret key) is such as $\log_g n < x < q$

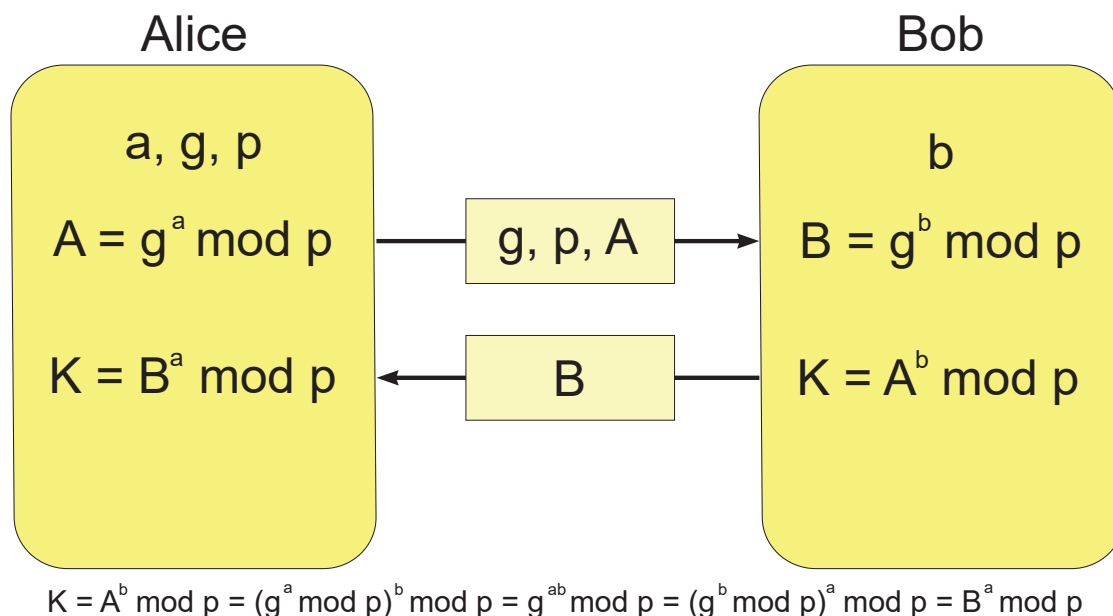
1. Alice computes $K_a = g^x \bmod n$ and sends (n, g, K_a) to Bob.
2. Bob randomly generates y (Bob's secret key), computes $K_b = g^y \bmod n$, and sends K_b to Alice.
3. Alice and Bob now each compute a session key separately
 $K = K_b^x \bmod n = K_a^y \bmod n = g^{xy} \bmod n$

69

ISAE – 2022/2023

Diffie-Hellman again

- But borrowed from Wikipedia page
 - Yep, it's nice and you'll see why twice later



70

Weaknesses of public key ciphers

- Complex computation
 - slow (~ 1 Mb/s)
 - long keys (1024 or 2048 bits),
except with elliptic curves (~ 160 bits)
- Specific problems
 - Integrity of public keys directory
 - Keys lifetime
 - Revocation
 - Private key sharing necessity?
 - Algorithms limitations : e.g. encrypt a small M with RSA

71

Hash functions \rightarrow fingerprint

- « One-way hash function » H
 - Fingerprint or hash $H(M)$ has a fixed width n (e.g.: 128 bits) whatever the length of M
 - The probability that 2 different messages M et M' have the same fingerprint $H(M)=H(M')$ is $\sim 1/2^n$
 - Knowing M , it is easy to compute $H(M)$
 - Knowing M , it must be impossible to find $M' \neq M$ with $H(M') = H(M)$
- Examples: MD5, SHA-1, SHA-256, DES in CBC mode
- Typically, one slices M in blocks m_1, m_2, \dots, m_k
 $h_1=F(\text{cte},m_1)$, $h_2 = F(h_1,m_2)$, ..., $h_k = F(h_{k-1},m_k) = H(M)$

72

Application : integrity

- Networking : against man-in-the-middle send message and fingerprint through distinct channels
- Files : modification detection
 - Examples : Tripwire, Samhain
 - On a trusted host, compute the fingerprints of stable files (OS, configuration, main programs, ...) and keep them in protected storage
 - Regularly or in case of doubt, recompute fingerprints to check them (with a trusted computer)

73

Crypto. up&down example

- 2004
 - Collision classes found in MD5
 - Extrapolation opportunities to SHA-1
- 2005
 - MD5 considered untrusted
 - Theoretical doubts with SHA-1 (numerous collisions)
- 2006, 2007, 2008
 - Rumors around SHA-1
- 2007 - 2012
 - NIST public competition for SHA-3
 - Five SHA-3 finalists since 2010-12-09
 - BLAKE, Grøstl, JH, Keccak and Skein
 - SHA-3 selected in 2012 (Keccak)

74

Julius. Caesar
Via Appia 1
Rome, The Roman Empire

Julius. Caesar
Via Appia 1
Rome, The Roman Empire

May, 22, 2005

To Whom it May Concern:

Alice Falbala fulfilled all the requirements of the Roman Empire intern position. She was excellent at translating roman into her gaul native language, learned very rapidly, and worked with considerable independence and confidence.

Her basic work habits such as punctuality, interpersonal deportment, communication skills, and completing assigned and self-determined goals were all excellent.

I recommend Alice for challenging positions in which creativity, reliability, and language skills are required.

I highly recommend hiring her. If you'd like to discuss her attributes in more detail, please don't hesitate to contact me.

Sincerely,

Julius Caesar

May, 22, 2005

Order:

Alice Falbala is given full access to all confidential and secret information about GAUL.

Sincerely,

Julius Caesar

<http://www.cits.rub.de/MD5Collisions/>

```
ortalo@hurricane:~/ $ md5sum letter_of_rec.ps order.ps
a25f7f0b29ee0b3968c860738533a4b9  letter_of_rec.ps
a25f7f0b29ee0b3968c860738533a4b9  order.ps
ortalo@hurricane:~/ $
```

75

ISAE – 2022/2023

RSA+AES+SHA3

- The ideal combination or the minimum baseline for computer security ?
- Now come quantum, beware !
 - or not
 - and diversify...

Post-Quantum Cryptography

- NIST latest crypto competition
 - quantum computer resistant public-key cryptographic algorithm
 - Post-Quantum Cryptography Standardization
<https://csrc.nist.gov/projects/post-quantum-cryptography>
 - Selected Algorithms 2022
 - PK encryption and Key setup : CRYSTALS-Kyber
 - DSA : CRYSTALS-Dilithium, Falcon, Sphincs+
 - Round 4 (*call for more*) submissions opened
 - July 5, 2022 – June, 1 2023
- A thought experiment can be pretty fruitful
 - That's a tradition now in quantum mechanics

77

Use crypto. correctly

Use proven code instead of rewriting, do not reinvent the wheel (or the brakes)

- Nintendo Wii
 - Used strncmp() instead of memcmp() to compare the SHA hash
- Works well when one feeds it a signature that starts with null bytes
- Strings in C are null terminated
- A null byte is only 256/2 random attempts away on average

78

Really, *use* it correctly

- Shortcomings *are* shortcomings
- Engineering shortcuts are *exactly* the things an attacker would abuse
 - What do you think *smart attackers* do ?
- For example
 - Imperfect Forward Secrecy : How Diffie-Hellman Fails in Practice*, 2015 (WeakDH.org)
 - Pre-computation attacks for 512, 726 or 1024 bits groups (discrete log calculation)
 - Practical tests
 - Original idea, 1976 (DH), TLS implementation circa 90-2000, prioritization of improvement in 2015...
 - In the meantime, Snowden leaks show that at least one major agency did its homework... discrete logs tables

79

Other topics (undetailed)

- Steganography
- *Watermarking*
- Random generators
- Prime generation
- Key escrow
- Voting
- Timestamping
- Destruction
- Protocols
- Cryptanalysis

Overall presentation (1/2)

- Fast paced computer security walkthrough
 - Security properties
 - Attacks categories
 - Elements of cryptography
 - **Introduction to mandatory security policies**
- Embedded systems and security
 - Specificities
 - Physical attacks (SPA, DPA)
 - TPM
- Software development and security
 - Security requirements and process
 - Static verification and software development tools
 - Common criteria / ISO 15408

81

Security policy and security model

- The security policy
 - « *specifies the set of laws, rules and practices that regulate how sensitive information and other resources are managed, protected and distributed within a specific system.* » [ITSEC, 1991]
 - physical, personnel or procedural, logical
- A security model
 - Formal description or mathematical abstraction
- Classical partition between model entities
 - active: subjects s
 - passive: objects o

82

Discretionary and mandatory policies

- Discretionary policy
 - each object o is associated to a specific subject s , its owner who manipulates access rights at his discretion
 - the owner can freely define and grant such access rights to himself or another user
- Mandatory policy
 - discretionary rules (access rights)
 - *and* : mandatory rules (habilitation level)

83

ISAE – 2022/2023

Access control matrix model

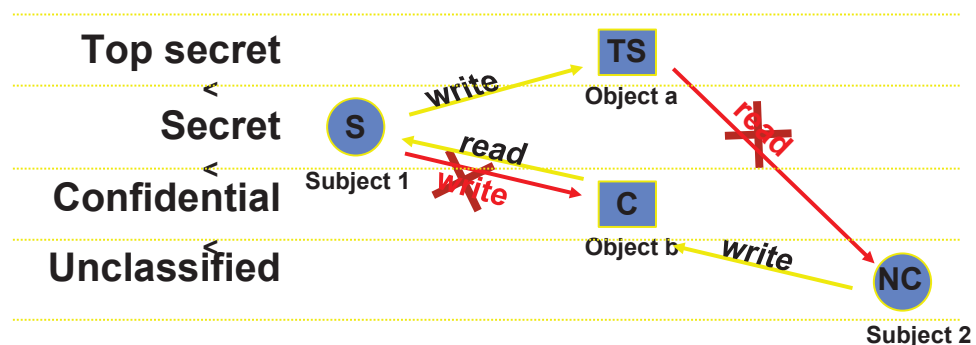
[Lampson 1971]

- State machine : state = (S, O, M)
 - O set of objects
 - S set of subjects ($S \subseteq O$)
 - $M(s, o)$ is the set of rights that subject s holds over object o
 - rights belong to a finite set A

84

Multilevel mandatory policy of Bell-LaPadula (1975)

- (habilitation) level of subjects $h(s)$
- (classification) level of objects $c(o)$
- prevents information flow from an object to a lower level object
- prevent any subject from gaining information from an object which level is higher than their habilitation



85

Bell-LaPadula Model

- classification cl : totally ordered set
- compartment C : set of categories
- $n=(cl, C)$, $n'=(cl', C')$: $n \leq n' \Leftrightarrow cl \leq cl'$ et $C \subseteq C'$ (treillis)
- simple property

$$\forall s \in S, \forall o \in O, \text{read} \in M(s, o) \Rightarrow c(o) \leq h(s)$$
- \star -property

$$\forall s \in S, \forall (o, o') \in O^2, \text{read} \in M(s, o) \wedge \text{write} \in M(s, o') \Rightarrow c(o) \leq c(o')$$

86

Other policies and models

- Non-interference
- Non inference
- HRU
- Clark-Wilson
- Chinese wall
- RBAC
- etc.

... for further studies.

87

Weaknesses of BLP and Biba

- Weaknesses
 - Overclassification degrades (security) information continuously (or out-of-model declassification procedures are introduced)
 - The model does not represent all information flows and does not take into account covert channels
- Biba (integrity) policy
 - dual of BLP for integrity assurance
 - rights = { write, read, invoke }
 - similar weakness : information integrity level degrades continuously

88

Policy, protection and access control

- Security rules are enforced via security mechanisms (hardware or software)
- Easy to imagine for rules like « it is permitted to... » or « it is forbidden that... » – protection mechanisms – privileged instructions, memory access control, file access control, etc.
→ authorization
- Harder for rules like « it is mandatory that... » or « it is recommended that... »
→ action triggers, ressource management

89

ISAE – 2022/2023

Overall presentation (1/2)

- Fast paced computer security walkthrough
 - Security properties
 - Attacks categories
 - Elements of cryptography
 - Introduction to mandatory security policies
- **Embedded systems and security**
 - **Specificities**
 - Physical attacks (SPA, DPA)
 - TPM
- Software development and security
 - Security requirements and process
 - Static verification and software development tools
 - Common criteria / ISO 15408

90

Embedded systems characterization

- Various designation (different real cases)
 - real-time
 - critical
 - embedded (in a vehicle)
 - autonomous / distant from the power plug
 - hidden / distant from any user
 - distributed (communicating?)
 - integrated (in a hardware platform)
 - other? : lost, stolen, fallen from the shelf (re-purposed...), numerous&similar?
- Up to now, not so different from a regular computer (esp. from the security point of view)

91

Inventory attempt

- | | |
|---------------------------|---------------------------|
| • smart cards | • RAID cards |
| • switches/routers | • coprocessors |
| • game consoles | • chronotachygraph (?) |
| • GPS receivers | • (artificial) satellites |
| • ADSL+TV boxes | • rockets |
| • mobile phones | • automatic pilots |
| • digital video recorders | VAL, train? |
| • home automation? | • switch/firewall AFDX |
| • (industrial) robots | (avionics) |
| • energy meters | • cars? |

Domains of application

- Industry
 - Industry automation and robotics
 - Energy (smart grid)
- Vehicles
 - Avionic domain
 - Space domain
 - Ground-transport domain
- Consumer electronics
 - mobile telephony
 - video games
 - Internet acces (high speed)
 - media broadcasting

93

Multiple security requirements

- Supplier/content protection
 - GSM phone
 - Media distribution
- ES environment protection
 - The vehicle itself, its passengers
 - Vehicle ressources (e.g. satellite)
- ES owner protection
- ES self protection
 - smart card, cryptographic chipset
- And the protection of an embedded *information system*, *i.e.* several networked ES

Evolution

from

- Some security functionalities

to

- Security management at the system design and architectural level
(both hardware and software)

Security of industrial systems is getting a lot of attention recently (and then?)

95

ISAE – 2022/2023

Motivations for evolution

- Widening attack range
 - logical, physical, auxiliary channel
- Limited computing resources
 - especially wrt. computational needs (crypto.)
- Limited ressources in general
 - especially energy (storage also)
- A need of modularity/flexibility
 - fast moving components and standards
- Multiple different security functionalities expected by users

96

Challenges

- complexity
 - embedded software gets more and more complex
 - efficient languages (C, C++) are not specifically secure
- extensibility
 - Java, .NET: designed for extension
 - J2ME, JavaCard too
 - dynamic updates (with code execution)
 - mise à jour (exécution) dynamique
- networking
 - WiFi, bluetooth
 - Internet

Note : Nothing really specific to embedded systems...

98

Overall presentation (1/2)

- Fast paced computer security walkthrough
 - Security properties
 - Attacks categories
 - Elements of cryptography
 - Introduction to mandatory security policies
- Embedded systems and security
 - Specificities
 - **Physical attacks (SPA, DPA)**
 - TPM
- Software development and security
 - Security requirements and process
 - Static verification and software development tools
 - Common criteria / ISO 15408

Physical attacks

- Direct hardware attack
 - micro-probing
 - substrate reconstruction
 - debugging interface access (JTAG, etc.)
- Difficulties
 - Costly (with respect to other attacks)
 - Destructive
 - Alternative attack precursor
- Primary target: cryptographic chipsets

100

ISAE – 2022/2023

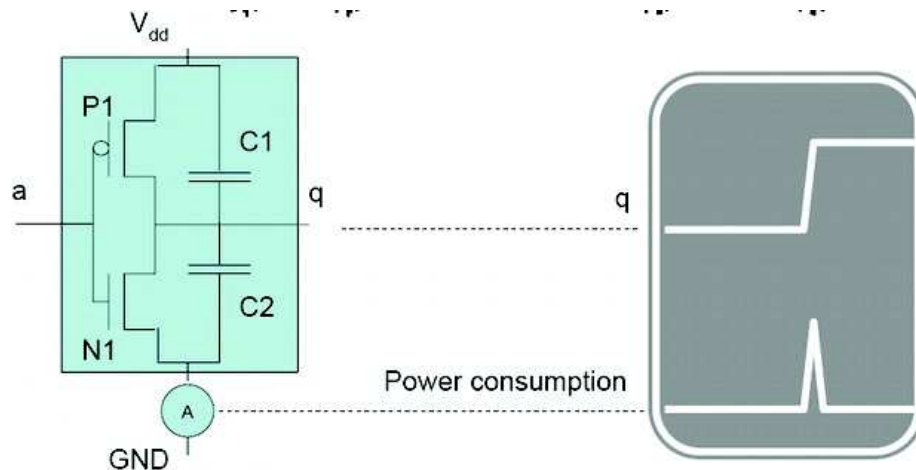
Auxiliary channels

- Timing (temporal) analysis
- Power analysis
 - SPA: simple power analysis
 - DPA: differential power analysis
- Impact
 - Find correlation between measurements and secret keys
 - Very efficient (DXX)
 - costly counter-measures
 - rigorous, counter-intuitive, patented

101

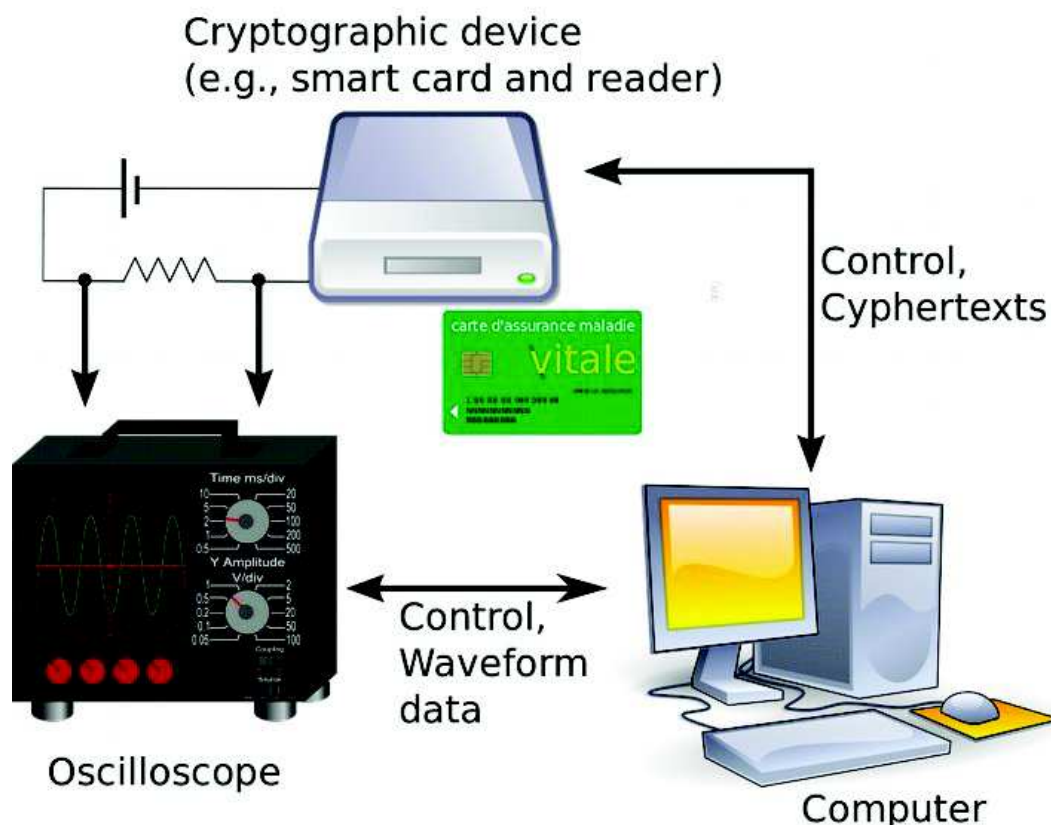
Power analysis

- References (most pictures reused)
 Elisabeth Oswald (Univ. Bristol) - dpabook.org
 Josh Jaffe & P.Kocher (*timing analysis*)
 (Cryptography Research, Inc.)



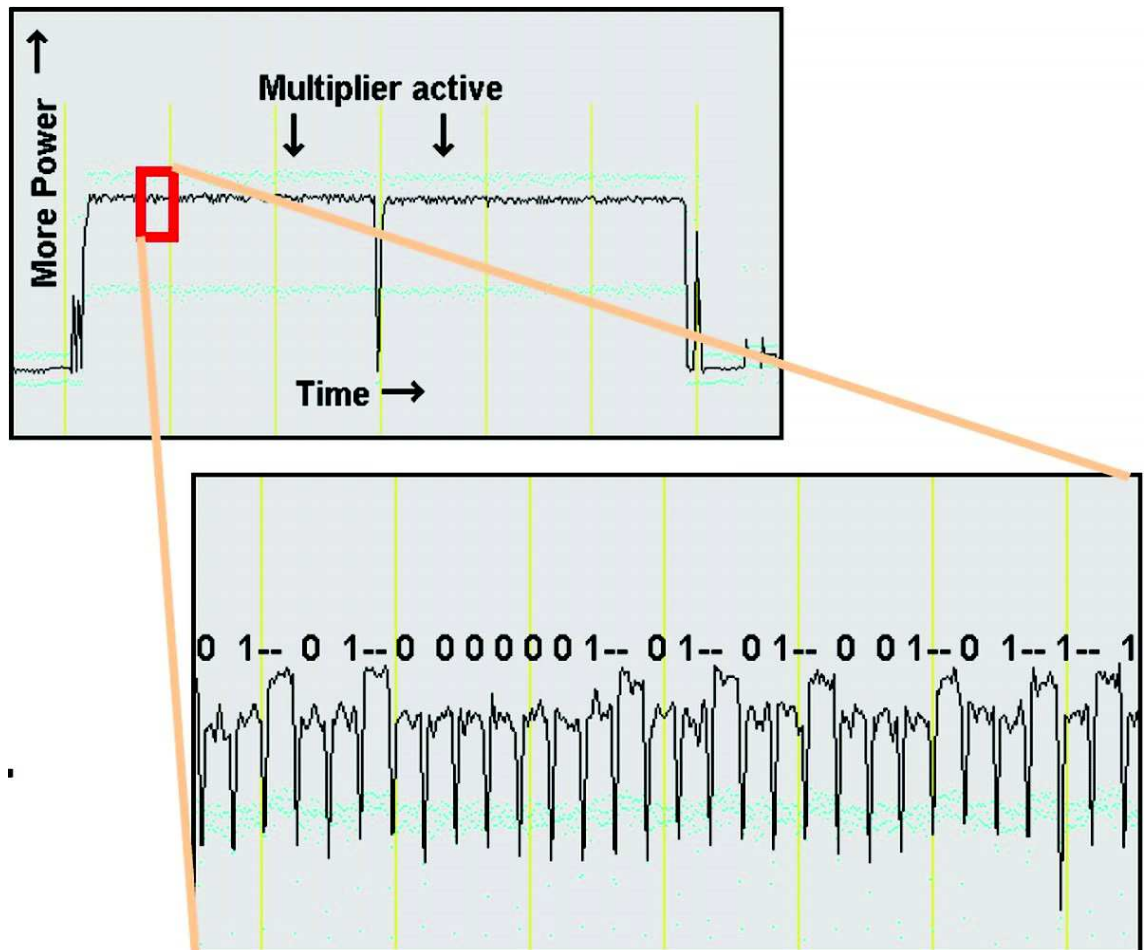
102

Power analysis typical setup



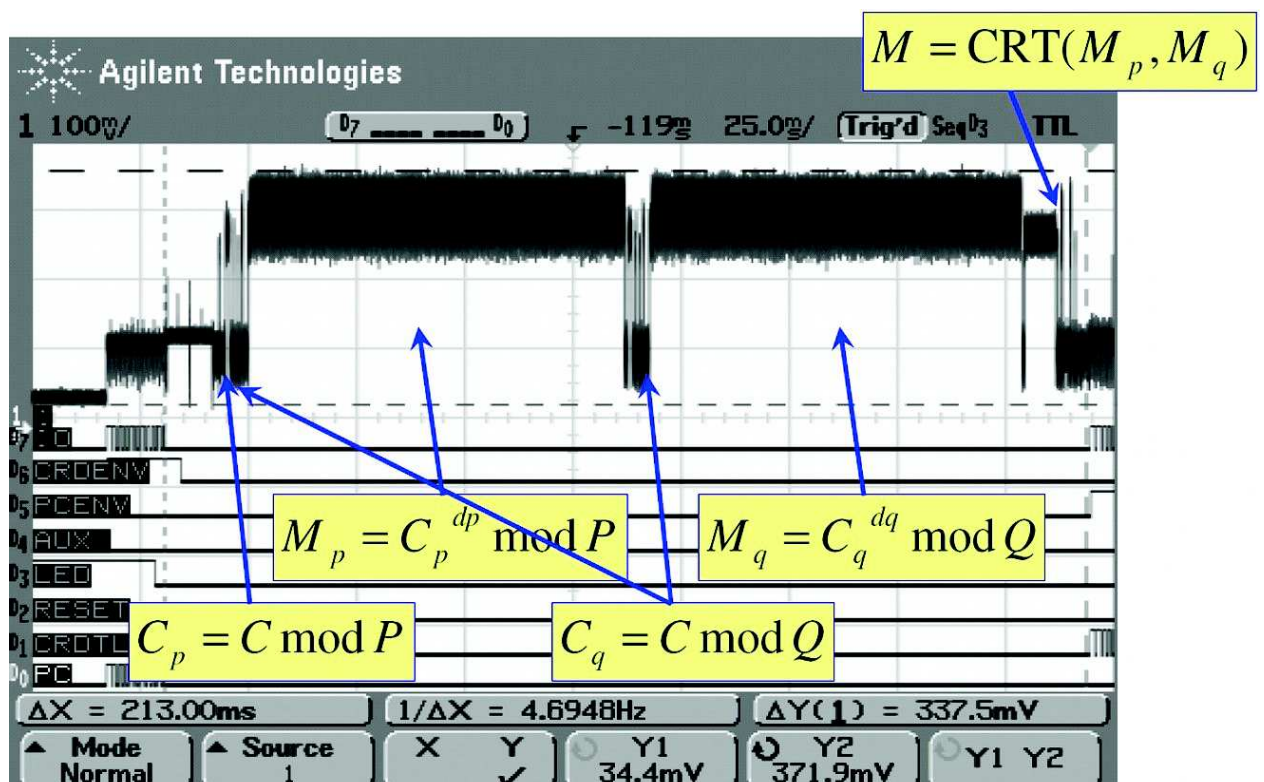
103

SPA Example



104

SPA Example



105

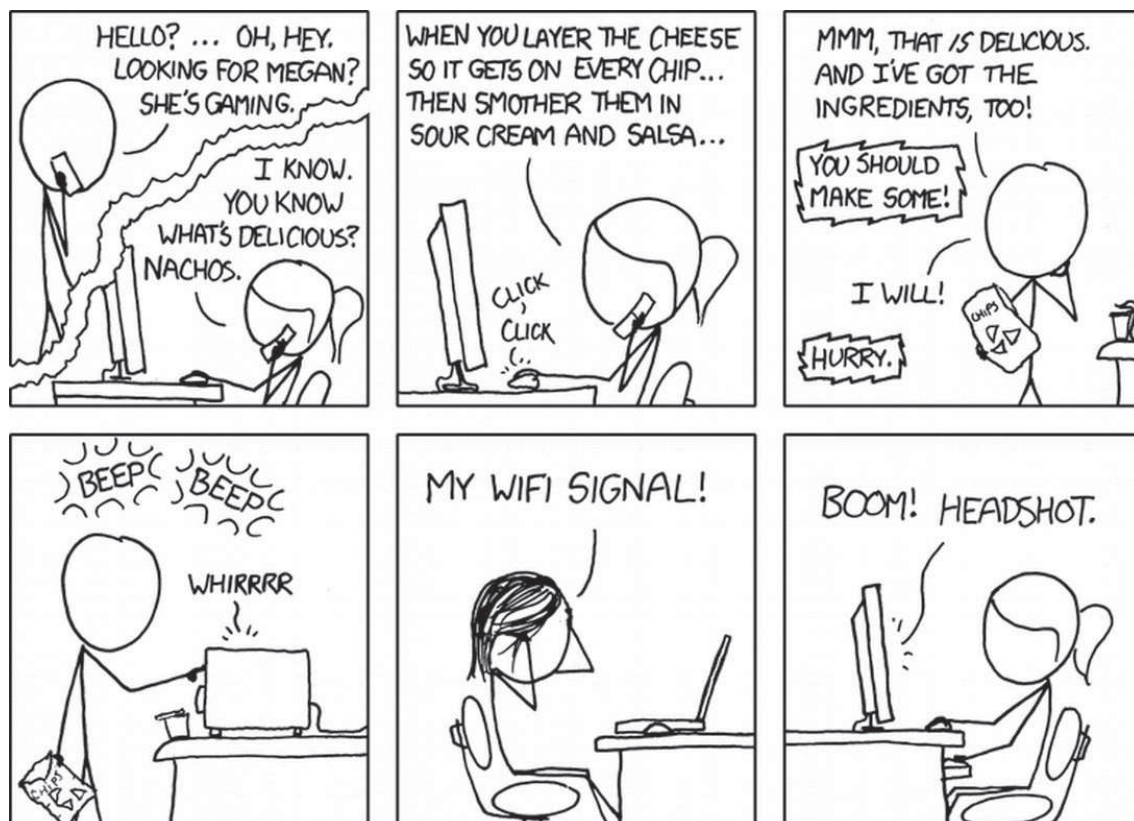
Alternative attacks

- DPA
 - Differential power analysis
 - Much less sensitive to noise disruptions
- Fault injection
 - induce normal behavior perturbation
 - may allow deduction of secret information
- EMA
 - electromagnetic analysis
 - best of *eighties* (TEMPEST protection)
 - remastered

106

ISAE – 2022/2023

NACHOS - <https://xkcd.com/654/>



107

Physical protection

- Secure computing
 - hardware
 - software
 - hybrid
- Attack resistance
 - Trusted Computing
 - e.g.: TPM (TCPA)

108

ISAE – 2022/2023

Overall presentation (1/2)

- Fast paced computer security walkthrough
 - Security properties
 - Attacks categories
 - Elements of cryptography
 - Introduction to mandatory security policies
- Embedded systems and security
 - Specificities
 - Physical attacks (SPA, DPA)
 - **TPM**
- Software development and security
 - Security requirements and process
 - Static verification and software development tools
 - Common criteria / ISO 15408

109

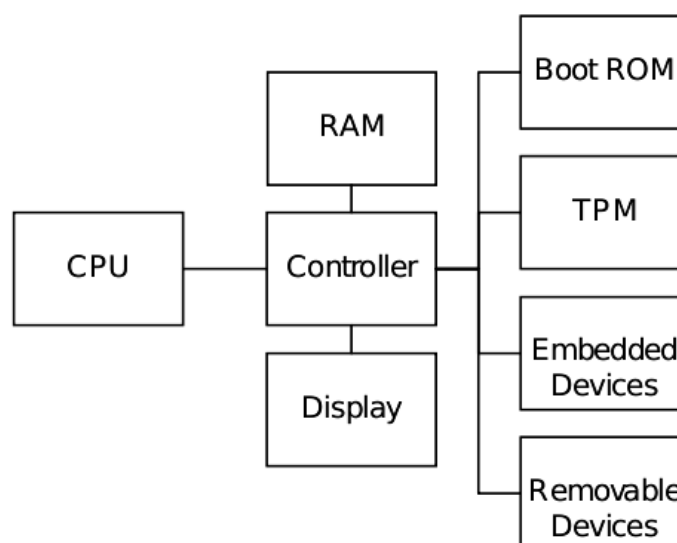
TPM

- *Trusted Platform Module*
- from the Trusted Computing Group (TCG)
 - <http://www.trustedcomputinggroup.com/>
 - « ... open, vendor neutral, industry standards for hardware-enabled trusted computing and security... »
 - Promoters (2008)
 - AMD, Fujitsu, HP, IBM, Infineon, Intel, Lenovo, Microsoft, Seagate, Sun, Wave
 - Contributors, Adopters... (140 members)
 - successor of TCPA (and competing Palladium?)
 - established in 2003
 - Slow but steady diffusion in common platforms through the 2010s
- Now in V2 !

110

ISAE – 2022/2023

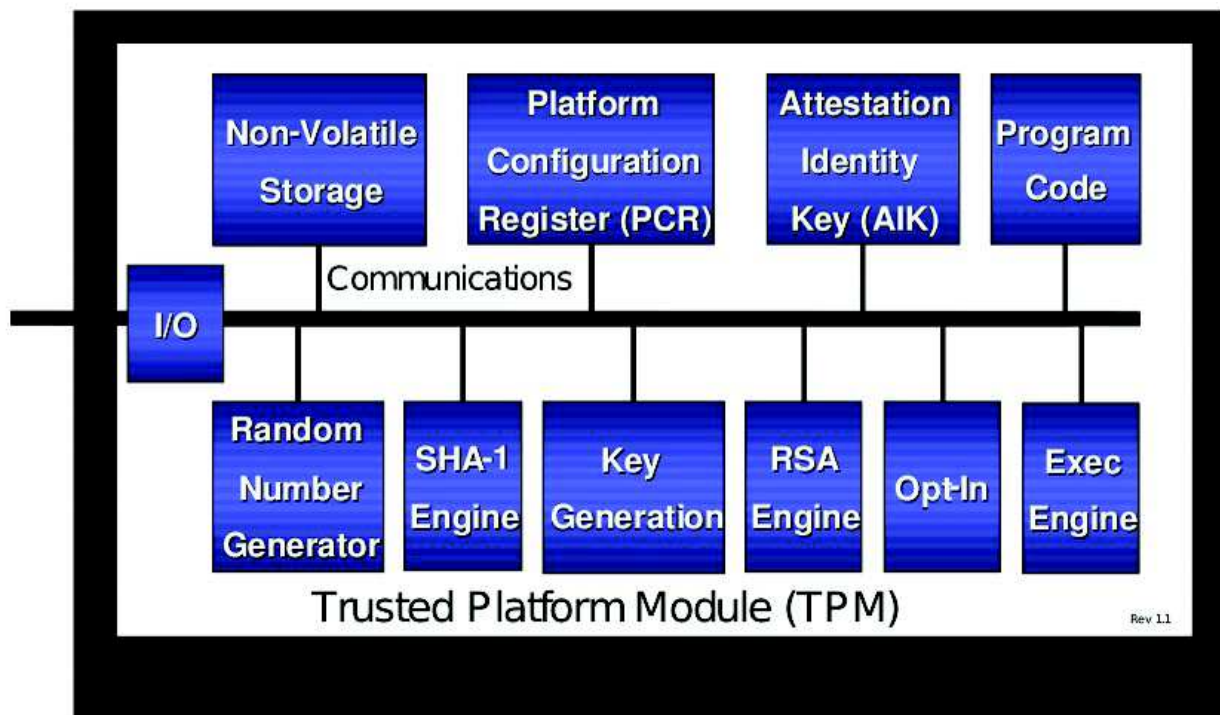
TCG Architecture



Very generic reference architecture

111

A TPM (V1) with RTR+RTS



112

Fundamental Trusted Platform Features

- Protected capabilities
 - shielded locations (register, memory, etc.)
 - and: key management, RNG, sealing, etc.
- Attestation
 - by the TPM, to the platform, of the platform, authentication of the platform
- Integrity Measurement, Logging and Reporting
 - metrics of integrity and digests (PCR)
 - recommended logging (optional)
 - attesting measurements
 - ➔ independent process to evaluate integrity (platform cannot lie)

113

Use on Linux

<http://www.grounation.org/index.php?post/2008/07/04/8-how-to-use-a-tpm-with-linux>

- Enable TPM in BIOS / Load drivers
- Install tpm-tools and TrouSers
- Take Ownership (once and for all)
- (Compile) Install and setup TrustedGRUB
 - Restart successfully
 - Contemplate PCRs
 - PCR 0 to 3 for the BIOS, ROMS...
 - PCR 4 contains MBR information and stage1
 - PCR 8,9 contains bootloader information stage2 part1,2
 - PCR 12 contains all commandline arguments from menu.lst and those entered in the shell
 - PCR 13 contains all files checked via the checkfile-routine
 - PCR 14 contains all files which are actually loaded (e.g., Linux kernel, initrd, modules...)
 - PCR 15 to 23 are not used

114

ISAE – 2022/2023

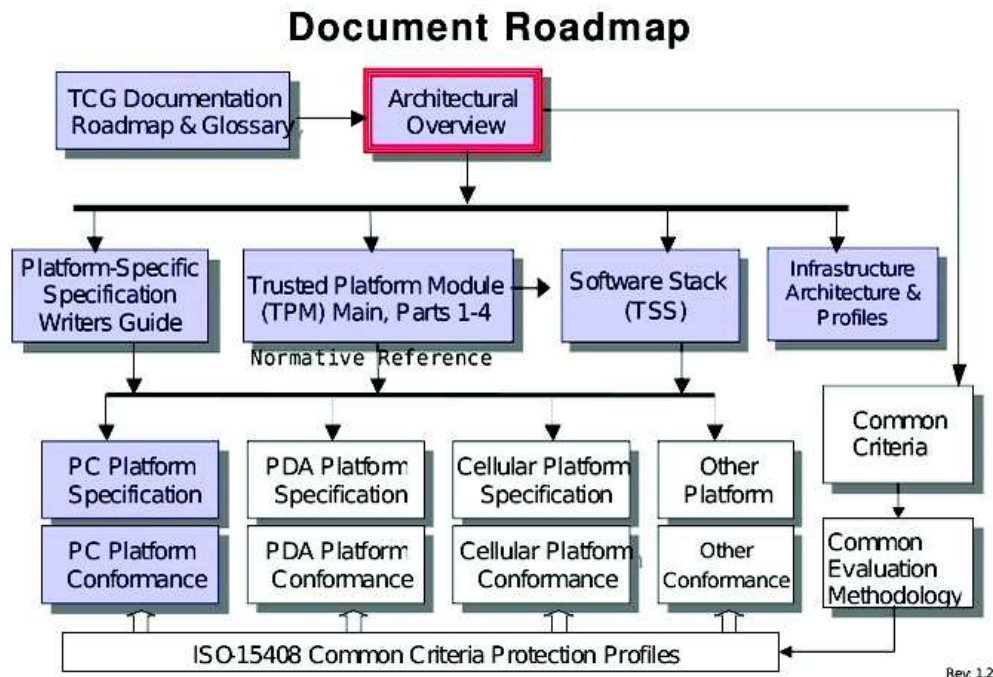
Use on Linux

<http://www.grounation.org/index.php?post/2008/07/04/8-how-to-use-a-tpm-with-linux>

- Use some TPM features
 - Add some « checkfile » or « pcr_verify » to grub.lst
- ```
/somehwere/check.file
fedb1cfff009e115f7f5f7b4533667a787798832d (hd0,1)/test1.file
485214eab2de87284de9d4e323e428bf60e0aa77 (hd0,1)/grub-0.92.tar.bz2
a6e171e989849dd44735a513c4270a0837c09174 (hd0,1)/test2.file
```
- Restart successfully
  - NB : Do **not** forget the
    - Owner password
    - Storage Root Key (SRK) password
    - or stick to the precise software installed at setup time
      - with security verifications still enforced
      - and if you disabled boot on CD/DVD/USB of course...
  - Extend to :
    - TPM KeyRing
    - Ecryptfs PKI...

115

# Trusted Computing Group (TCG) and Trusted Platform Module (TPM)



<http://www.trustedcomputinggroup.com/>

116

## Other issues with TPM

- « (...) *The TPM has the EK generated before the end customer receives the platform. (...)*
  1. *The EK MUST be a 2048-bit RSA key (...)*
    - c. *The PRIVKEY SHALL exist only in a TPM-shielded location (...)* »
      - *TPM Main Part 1 Design Principles, Specification, Version 1.2, Level 2 Revision 103, 9 July 2007. Section 5 (lines 1004-1040).*
- « ... *If it's good enough for the NSA, it should be good enough for you.* »
  - Roger L. Kay, *Trusted Computing is Real and it's Here*, 2007.
- Trusted Computing or « *Treacherous Computing* » ?
  - (several) *anonymous*

117

## Contrast with UEFI

- Microsoft Secure boot
- The initial master key is controlled by Seattle
- And it delegates...
- Side note
  - Fortunately, there is JTAG...

118

ISAE – 2022/2023

## Contrast with...

- ... keys in an attached smartcard.

119

# Overall presentation (1/2)

- Fast paced computer security walkthrough
  - Security properties
  - Attacks categories
  - Elements of cryptography
  - Introduction to mandatory security policies
- Embedded systems and security
  - Specificities
  - Physical attacks (SPA, DPA)
  - TPM
- **Software development and security**
  - **Security requirements and process**
  - Static verification and software development tools
  - Common criteria / ISO 15408

120

ISAE – 2022/2023

## Introductory programmer comment

### World-writable memory on Samsung Android phones

Posted Dec 17, 2012 20:13 UTC (Mon) by **mikov** (subscriber, #33179) [\[Link\]](#)

My experience from most places: nobody cares, nobody reviews. If a problem is discovered later, we will fix it later - why worry now and delay the release? What "/dev/mem"? Enough with this mumbo-jumbo we have a release to make and management bonuses to earn.

In fact people who do care and worry about esoteric things like "security", or "good design" or "code quality" are universally viewed as trouble-makers or ivory tower idiots both by management and most of the engineers. It is an uphill battle even to do what used to be the baseline 10-15 years ago.

Commercial software engineering now is no different from accounting. The glory days are gone. It is all downhill from now on.

<http://lwn.net/Articles/529496/>

BTW, Cyanogen fix: <http://review.cyanogenmod.org/#/c/28568/>

121

# Problem to address (with respect to security requirements definition)

- Best ROI when done at application design phase
- When considered at all, they tend to be
  - general lists of security features
    - password, firewalls, antivirus, etc.
  - implementation mechanisms  $\neq$  security requirements
  - intended to satisfy *unstated* requirements
    - authenticated access, etc.
- Exist in a section by themselves (copied from a generic set)
  - no elicitation or analysis process, no adaptation to the target
- Significant attention is given to what the system should do
  - little is given to what it should not do (*in req. eng.*)
- Priority is not given to security (wrt ease of use for example)

122

## Note on security updates

- How can we manage software vulnerabilities?
  - Wait until they are exploited by an attacker
  - Quickly provide a patch that should correct the problem (without introducing another one)
  - Whine because system administrators do not install patches fast enough
- Astonishingly it is very popular
  - All serious editors do that
  - Users feel more secure (still?)

123

## In other words

- It is not enough to apply patches to secure a system
- Also, you cannot rely only on firewalls or antivirus (or IT security tools)
- Security objectives of a piece of software should be identified
- Security implies a change in point of view
  - e.g.: it must *not* work
  - unavailable is better than destroyed
  - which (computer) is saved first ?
- i.e. : What do you *really* want *exactly* ?

124

ISAE – 2022/2023

## Drones firmware security update



- DJI firmware update
  - february 2015
  - Phantom 2
  - Phantom 2 Vision (+)
- integrates
  - a no-fly zone
  - 15.5 miles radius
  - around the...
    - White House
- guess why ?
- et l'Elysée au fait ?

125

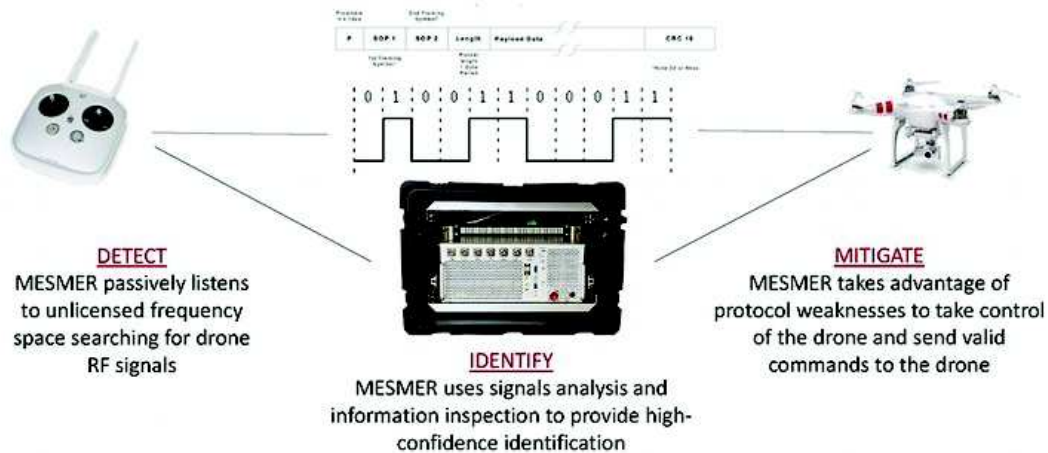


# Speaking of point of view...

“Countering the Threat of Unauthorized Drones with ...”

## Protocol Manipulation

department **13**



1

126

ISAE – 2022/2023

# Law 1

## Code des postes et des communications électroniques

### Partie législative (Articles L1 à L141)

#### LIVRE II : Les communications électroniques (Articles L32 à L97-4)

#### TITRE Ier : Dispositions générales (Articles L32 à L40-1)

#### Chapitre II : Régime juridique. (Articles L33 à L34-10)

#### Section 1 : Réseaux et services. (Articles L33 à L33-10)

Naviguer dans le sommaire du code

### Article L33-3-1

Version en vigueur du 01 février 2012 au 31 juillet 2021

Modifié par Ordonnance n°2011-1012 du 24 août 2011 - art. 40

I. - Sont prohibées l'une quelconque des activités suivantes : l'importation, la publicité, la cession à titre gratuit ou onéreux, la mise en circulation, l'installation, la détention et l'utilisation de tout dispositif destiné à rendre inopérants des appareils de communications électroniques de tous types, tant pour l'émission que pour la réception.

II. - Par dérogation au premier alinéa, ces activités sont autorisées pour les besoins de l'ordre public, de la défense et de la sécurité nationale, ou du service public de la justice.

NOTA :

Ordonnance n° 2011-1012, article 57: ces dispositions entrent en vigueur le 1er février 2012, toutefois, les installations radioélectriques permettant de rendre inopérants les téléphones mobiles de tous types tant pour l'émission que pour la réception, établies dans l'enceinte des salles de spectacles à la date de publication de la présente ordonnance, restent autorisées pendant un délai de cinq ans à compter de cette date. Pendant ce délai, l'utilisation de ces installations reste soumise aux conditions définies par application de l'article L. 36-6 du même code conformément au 2° de l'article L. 33-3 dans sa rédaction antérieure à celle issue de la présente ordonnance.

Versions ▾

Liens relatifs ▾

127

# Law 2

## [Code des postes et des communications électroniques](#)

### ■ **Partie législative (Articles L1 à L144)**

#### ■ [LIVRE II : Les communications électroniques \(Articles L32 à L97-4\)](#)

#### ■ [TITRE Ier : Dispositions générales \(Articles L32 à L40-1\)](#)

#### ■ [Chapitre II : Régime juridique. \(Articles L33 à L34-15\)](#)

#### [Section 1 : Réseaux et services. \(Articles L33 à L33-15\)](#)

Naviguer dans le sommaire du code

### > [Article L33-3-1](#)

**Version en vigueur depuis le 31 juillet 2021**

[Modifié par LOI n°2021-998 du 30 juillet 2021 - art. 24](#)

I.-Sont prohibées l'une quelconque des activités suivantes : l'importation, la publicité, la cession à titre gratuit ou onéreux, la mise en circulation, l'installation, la détention et l'utilisation de tout dispositif destiné à rendre inopérants des équipements radioélectriques ou des appareils intégrant des équipements radioélectriques de tous types, tant pour l'émission que pour la réception.

II.-Par dérogation au premier alinéa, ces activités sont autorisées pour les besoins de l'ordre public, de la défense et de la sécurité nationale, ou du service public de la justice.

L'utilisation par les services de l'Etat de dispositifs destinés à rendre inopérant l'équipement radioélectrique d'un aéronef circulant sans personne à bord est autorisée, en cas de menace imminente, pour les besoins de l'ordre public, de la défense et de la sécurité nationales ou du service public de la justice ou afin de prévenir le survol d'une zone en violation d'une interdiction prononcée dans les conditions prévues au premier alinéa de l'[article L. 6211-4 du code des transports](#). Un décret en Conseil d'Etat détermine les modalités de mise en œuvre de ces dispositifs, afin de garantir leur nécessité et leur proportionnalité au regard des finalités poursuivies, ainsi que les autorités compétentes pour y procéder.

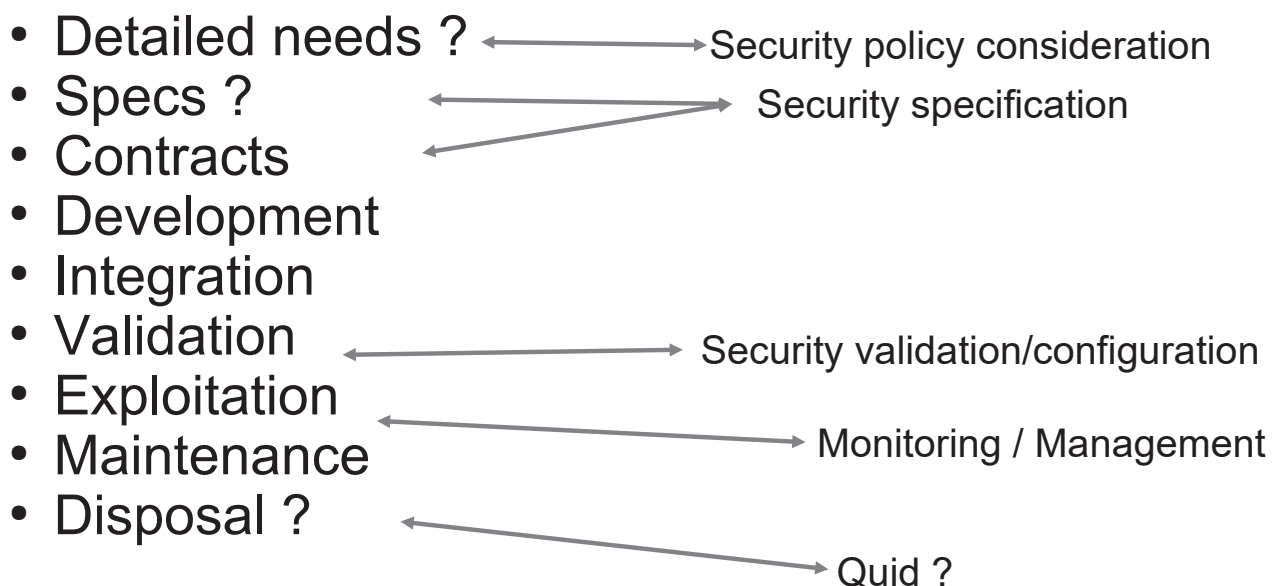
Versions ▾

Liens relatifs ▾

128

ISAE – 2022/2023

## Another view on project lifecycle



129



# Risk analysis

1. Identify assets and their value (\$\$)
  2. Define assets priority
  3. Identify vulnerabilities, threats and potential damages
  4. Define threats priority
  5. Optimize counter-measures selection
- Inherently qualitative (human/expert opinion)
  - Applicable to organization, system, project
  - Several methods available
    - MARION, MEHARI, EBIOS, etc.
    - HAZOP, FMEA, ISO31000, etc.

131

## Pros (my view)

- *Identification* of assets and their relative values
- Assets value offers an opportunity to budget realistically (for protection)
- Is understandable by end users
  - Quite easier than assembly language exploits or cryptographic hash functions
- Risk management alternatives
  - Transfer (insurance, state, etc.)
  - Acceptance (life is deadly after all)
  - Reduction (work, work, work, work, ...)
  - Avoidance (just do it the other way)
- Management could express clear priorities

132

## Cons (my view)

- Threat determination is an oracle problem
- May be used to demonstrate that (any) risk is (already) managed
  - Some forgotten successes of risk management
    - Lehman-Brothers financial risk exposure
    - Greek debt control
  - Qualitative also means manipulable
- Relies a lot on best practices or risks lists
  - Fuels paranoia and ready-made useless tools
  - Does not help target real assets
- Management rarely wants to decide
- Sometimes does not end well morally speaking
  - For example : product lifetime optimization  
(NB : Inherently viewpoint-based)

133

## Threats and use-case examples

- Trusted Computing Group
  - Mobile phone TPM use-case scenarios
  - *(Name,) Goal*
    - *Threats*
- Platform integrity
  - Ensure that device possess and run only authorized operating system(s) and hardware
    - Logic of device firmware modified
    - Device hardware modified
    - Device functions in a manner other than intended by the manufacturer
    - Device modified to broadcast false identification (IMEI)

134

# Threats and goals examples

- Device authentication
  - Assist user authentication
  - Prove identity of device itself
    - Identity spoofing to get unauthorized access to services
    - Identity no longer bound to the device
    - Theft of device
    - Device tracking
- Robust DRM implementation
  - Service and content providers need assurance that the device DRM is robust
- SIMLock / Device personalisation
  - Ensure that a mobile device remains locked on a particular network

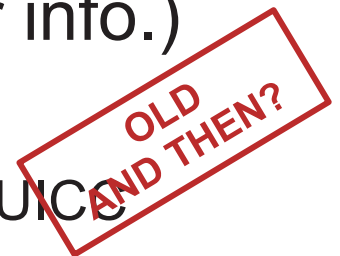


135

ISAE – 2022/2023

## Last use-case examples (for info.)

- Secure software download
- Secure channel between device and UICC (UMTS Integrated Circuit Card)
- Mobile Ticketing
- Mobile Payment
- Software use
  - User-available predefined software use policies
- Proving platform and/or application integrity to end user
- User data protection and privacy



136

# References

- DHS « Build Security In »
  - <https://buildsecurityin.us-cert.gov/>
- The Addison-Wesley Software Security Series
  - <http://www.softwaresecurityengineering.com/series/>
- CERT/CC
  - <http://www.cert.org/>
- « *Smashing the Stack for Fun and Profit.* »
  - Aleph One, Phrack Magazine 7, 49 (1996)  
File 14 of 16.
- OpenBSD
  - <http://www.openbsd.org/papers/>

138

## Some real programming

- Presentation based on work from real programmers in the neighbourhood
- First, sources :
  - Matthieu Herrb & lots of OpenBSD « good programming » examples
  - Vincent Nicomette and Eric Alata for some « details »

## Now real programming (*prereq.*)

```
#include <stdio.h>
void copie(char * s) {
 char ch[8] = "BBBBBBBB" ;
 strcpy(ch,s) ;
}
int main(int argc, char * argv[]) {
 copie(argv[1]) ;
 return(0);
}
AAAAAAAAAAAAAA[system_adr][exit_adr][shlibc_adr]
Bash$./a.out 'perl -e 'print "A"x12 . 0xb7ee1990 . 0xb7ed72e0 .
0xb7fcc0af' '
sh-3.1$
```

140

ISAE – 2022/2023

## Now real programming

- Number One : buffer overflow with string functions

```
strcpy(path, getenv("$HOME")) ;
strcat(path, "/");
strcat(path, ".foorc");
len = strlen(path);
```



- `strcat()`, `strcpy()`
  - no verification on buffer size, dangerous : do not use
- `strncat()`, `strncpy()`
  - leave strings non terminated, very difficult to use correctly
- `strlcat()`, `strlcpy()`
  - May truncate strings, but probably easier to use

# str{n,l}{cpy,cat} practical usage

STRCAT(3)

Linux Programmer's Manual

STRCAT(3)

NAME

strcat, strncat - concatenate two strings

SYNOPSIS

```
#include <string.h>
```

```
char *strcat(char *dest, const char *src);
```

```
char *strncat(char *dest, const char *src, size_t n);
```

No strlcat() on Linux ; so, from the BSDs (more precisely OpenBSD) :

```
size_t strlcpy(char *dst, const char *src, size_t dstsize);
size_t strlcat(char *dst, const char *src, size_t dstsize);
```

No strlcat on Linux... 142

ISAE – 2022/2023

## strncat() is difficult to use

```
strncpy(path, homedir, sizeof(path) - 1) ;
path[sizeof(path) - 1] = '\0' ;
strncat(path, "/", sizeof(path) - strlen(path) - 1) ;
strncat(path, ".foorc", sizeof(path) - strlen(path)
- 1) ;
len = strlen(path) ;
```

Note (on Linux) : g\_strlcpy() and g\_strlcat() exist in glib-2.0

Note (on BSD) : see next slide (*Yeah !!!*)

Additional note: C11 has removed gets() (was deprecated in C99) replaced by **gets\_s()**

## strl\*() look better

```
strncpy(path, homedir, sizeof(path)) ;
strcat(path, "/", sizeof(path)) ;
strcat(path, ".foorc", sizeof(path)) ;
len = strlen(path) ;
```

- May truncate, but no overflow
- Add checks for non testing code :
 

```
strncpy(path, homedir, sizeof(path)) ;
if (len >= sizeof(path)) return (ENAMETOOLONG) ;
strcat(path, "/", sizeof(path)) ;
if (len >= sizeof(path)) return (ENAMETOOLONG) ;
strcat(path, ".foorc", sizeof(path)) ;
if (len >= sizeof(path)) return (ENAMETOOLONG) ;
len = strlen(path) ;
```

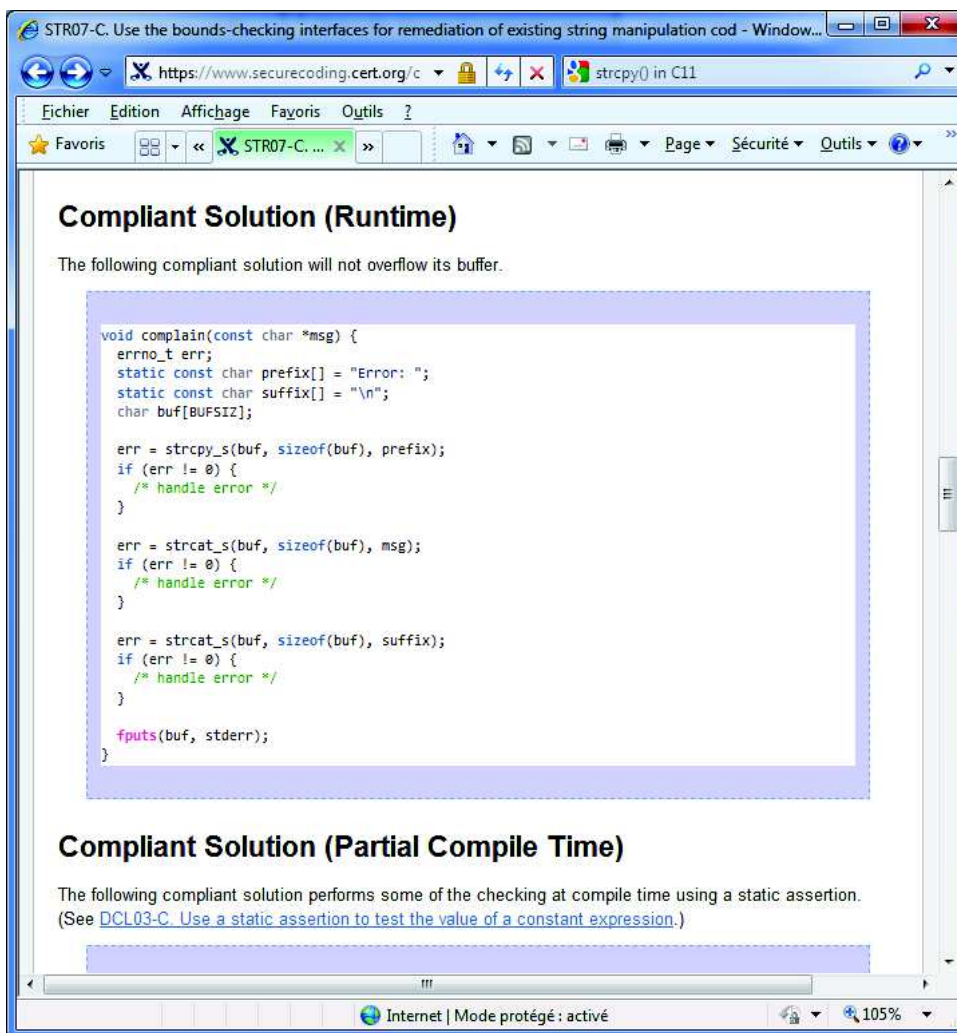
144

## C11 Annex K (ISO/IEC 9899:2011)

- C11 Ann.K « Bounds-checking interfaces » defines alternative versions of standard string-handling functions (from Microsoft)
- `strcpy_s()`, `strcat_s()`, `strncpy_s()` and `strncat_s()`
- *ie* :
 

```
errno_t strcpy_s(
 char * restrict s1,
 rsize_t s1max,
 const char * restrict s2
) ;
```
- See also : ISO/IEC TR24731-1:1999 and ISO/IEC:TR24731-2:2010 ...
- Note : `wchar_t`

145



146

## Time of check, time of use

- How to create a temp. file in /tmp without overwriting an existing file ?

```
/* Generate random file name */
name = mktemp("/tmp/tmp.XXXXXXXXXX");
/* verify file does not exist */
if (stat(name, &statbuf) == 0) {
 return EEXISTS;
}
/* ok, open it */
fd = open(name, O_RDWR);
```



- Opens a possible race condition with a concurrent process
- mktemp() deprecated in POSIX.1 (2011)

147



## Options

- Use `mkstemp()` to replace both system calls  
`fd = mkstemp("/tmp/tmp.XXXXXXXXXX") ;`
- Use `O_CREAT | O_EXCL`, `open()` flags that trigger an error if the file already exists  
`fd = open(name, O_CREAT | O_EXCL) ;`
- Note the difference between `fopen()` and `open()` return types (`FILE*` vs. `int` or streams vs. file descriptors)

148

ISAE – 2022/2023

## Arithmetic overflows

```
n = getIntFromUser();
if (n<=0 || n*sizeof(struct item) > BUFMAX){
 return EINVAL;
}
```

- If  $n$  is big enough, the condition will not be true
- Use :

```
n = getIntFromUser();
if (n<=0 || n > BUFMAX/sizeof(struct item)){
 return EINVAL;
}
```

149

# Arithmetic overflows

```
n = getIntFromUser();
if (n<=0) {
 return EINVAL;
}
data = (struct item *)
 malloc(n * sizeof(struct item));
if (data == NULL) {
 return ENOMEM;
}
```

- If  $n$  is big enough, overflow occurs and a small memory allocation is done
  - opening the path to a memory overflow
- Use `calloc()` !

```
data = (struct item *)
 calloc(n, sizeof(struct item));
```

150

# Format strings issues

- Many standard display functions use a format for printing : `printf()`, `sprintf()`, `fprintf()`, ...
- Two variants exist, with and without format strings : `printf("%s", ch)` or `printf(ch)`
- What happens when you give « `%x` » to `printf` ?
  - `printf()` gets its next argument from the stack
- When user input is passed to such functions, it can generate this kind of situations
- This kind of situation may allow to access areas of memory for reading (sometimes for writing)

# Example

```
#include <stdio.h>
int main()
{
 char * secret = "polichinelle";
 static char input[100] = {0};
 Printf("Enter your name: ");
 scanf("%s", input);
 printf("Hello ");printf(input);printf("\n");
 printf("Enter your password: ");
 scanf("%s",input);
 if (strcmp(entree,secret)==0) {
 printf("OK\n");
 } else {
 printf("Error\n");
 }
 return 0;
}
```



152

# Example

- Normal use of the program

```
bash$./a.out
Enter your name: Jack
Hello Jack
Enter your password: ripper
Error
```

- « Abuse » of the program

```
bash$./a.out
Enter your name: %p%s
Hello 0x08049760polichinelle
```

- Allows to walk the stack and access internal program data

153

## Practical recommendations

- Design first
  - Often broken and insecure by design
- Obscurity does not help
  - Exploits against closed source may be just as easy as against open source
  - Obfuscation primarily works for people writing code, not crackers
- Quality is security
  - Most security problems are simple bugs
  - There is no security plugin
  - No ROI for security
    - But shorter test cycles
    - Less bugs, so less time spent fixing them
    - And usually better efficiency

154

ISAE – 2022/2023

## Practical recommendations

- Most code should be simple and boring
  - Easier to audit
  - Already formatted
  - Clever code is almost always wrong
- Fix a bug everywhere
  - Even automate for checking it
- Check return codes
- Design your APIs right...
- Understand semantics
  - File descriptors
    - Inheritance over fork
    - Access rights only checked on open()
  - Signal handlers *are* complex
    - Simple rule : only set volatile atomic flags in them

155

# Practical recommendations

- Most security issues come from abstraction layers violation (audit these cases)
  - Hidden variables
  - Concurrency
  - Overflows
  - Flow control on error
- All user input must be checked
  - Positive checks
  - Everything not static is like user input
- Be careful with optimizations
- There is no secure language or environment
  - Java does not suffer from simple buffer overflows but has integer overflows, logic errors, etc.

156

ISAE – 2022/2023

## Overall presentation (1/2)

- Fast paced computer security walkthrough
  - Security properties
  - Attacks categories
  - Elements of cryptography
  - Introduction to mandatory security policies
- Embedded systems and security
  - Specificities
  - Physical attacks (SPA, DPA)
  - TPM
- Software development and security
  - Security requirements and process
  - **Static verification and software development tools**
  - Common criteria / ISO 15408

157

# Capabilities of Security Analyzers

- Examining Calls to Potentially Insecure Library Functions
- Detecting Bounds-Checking Errors and Scalar Type Confusion
- Detecting Type Confusion Among References or Pointers
- Detecting Memory Allocation Errors
- Detecting Vulnerabilities that Involve Sequences of Operations (Control-Flow Analysis)
- Data-Flow Analysis (reducing false alarms)
- Pointer-Aliasing Analysis (primarily useful for the former)
- Customizable Detection Capabilities

158

ISAE – 2022/2023

## Classes of Tools

- Source code analysis tools
  - see below
- Penetration testing tools
  - Ports scanners
    - e.g. nmap
  - Vulnerability scanners
    - e.g. Nessus, ISS's Internet Scanner
  - Application scanners
  - Web application assessment proxy

159

# Analyzer mechanics

- « simple » searching (grep-like)
- lexical analysis
- abstract syntax tree (AST) construction (parsing)
- advanced work (may) start here
  - global / local analysis
  - type calculus, logical reasoning, range calculus
  - false alarms reduction techniques
  - IDE integration, specification-based verification
  - etc.
- Wikipedia has a pretty good reference and *tools* collection list
  - Under « static program analysis » (en)
  - And « list of tools for static code analysis » (en)
  - Apparently up to date and with the old things too...

160

## Examples (*somehow outdated*)

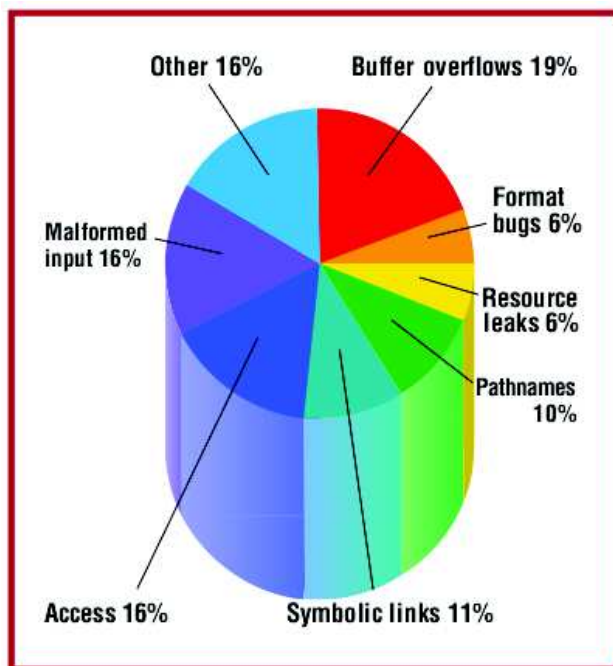
- Splint - <http://www.splint.org/>
  - evolution of good-old lint
  - lightweight static analysis
- Linux kernel hackers doing static analysis
  - smatch - <http://smatch.sourceforge.net/>  
links with sparse, C source inspection tool library
  - Died, and resurrected : and again

Coccinelle does/did that too, with OCaml

  - Herodotos (subtool)
- ASTREE - <http://www.astree.ens.fr/>
  - LIENS, started Nov. 2001
  - C programs
  - real-time embedded software static analyzer
  - based on abstract interpretation

161

# Splint – Quotes from the authors



**Figure 1. Common Vulnerabilities and Exposures list for the first nine months of 2001. Most of the entries are common flaws detectable by static analysis, including 37 buffer overflow vulnerabilities.**

- A tool knowing common vulnerabilities
- Exploiting annotations in programs
- Automated checking

*Improving security Using Extensible Lightweight Static Analysis*, David Evans and David Larochelle, *IEEE Software*, January/February 2002.

162

ISAE – 2022/2023

## False alarms

**Table I**

**False warnings checking wu-ftpd**

| Cause                  | Number | Percent |
|------------------------|--------|---------|
| External assumptions   | 6      | 7.9     |
| Arithmetic limitations | 13     | 17.1    |
| Alias analysis         | 3      | 3.9     |
| Flow control           | 20     | 26.3    |
| Loop heuristics        | 10     | 13.2    |
| Other                  | 24     | 31.6    |

163



# Annotation examples

## Library functions :

```
char *strcat (/*@returned@*/ char *s1, char *s2)
 /*@ensures s1:taintedness =
 s1:taintedness | s2:taintedness@*/;

char *strcpy (char *s1, const char *s2)
 /*@requires maxSet(s1) >= maxRead(s2)@*/
 /*@ensures maxRead(s1) == maxRead(s2)
 /\ result == s1@*/;
```

165

ISAE – 2022/2023

## Smatch

- <http://repo.or.cz/w/smatch.git>
- Smatch uses Sparse as a C parser
- (Linux kernel source code check)-oriented tool
  - Hacker (vs. Academic) approach
  -
- Smatch <https://lwn.net/Articles/691882/>
  - Static check over program model built from source
- Sparse <https://lwn.net/Articles/689907/>
  - C AST building and checking

166

Tue, 11 Jan 2022 11:40:05 +0100 (11 13:40 +0300)

tree 6ca845a72ae7eed79040060214cd0f2e41934888 [tree](#) | [snapshot \(tar.gz zip\)](#)

parent 3a0e5e2a339db6731b82f299f9997de7b7f8523e [commit](#) | [diff](#)

uninitialized: silence false positives with enum switch statements

One common cause of false positives is when:

```
switch (some_enum) {
 <All values handled. No default case>
}
```

Ideally if you had cross function analysis and whatever then Smatch would know the possible values and handle this correctly. Unfortunately a lot of code is too complicated.

The next most ideal situation would be if Smatch could determine the values of some\_enum based on the enum\_declaration. But right now it doesn't and that's hard to implement.

So this patch just silences the warnings in the check itself. If the switch is based on an enum, and there is no default case then mark everything which is sometimes initialized as always initialized. On my system this seems to silence about 3% of my warnings.

Reported-by: Jesse Brandeburg <jesse.brandeburg@intel.com>

check\_uninitialized.c [diff](#) | [blob](#) | [blame](#) | [history](#)

smatch.h [diff](#) | [blob](#) | [blame](#) | [history](#)

smatch\_states.c [diff](#) | [blob](#) | [blame](#) | [history](#)

167

ISAE – 2022/2023

## ASTREE

- Example of abstract interpretation application to software analysis
- Properties / objectives
  - sound (all possible errors)
  - automatic (no invariants required)
  - efficient
  - domain-aware, parametric, modular, extensible
  - hence, very precise
- Application / achievements
  - A340 fly-by-wire control software (C, 132kloc, 2003)
  - A380 electric flight control codes (2004)
  - C version of ATV automatic docking software (2008)

# Abstract interpretation

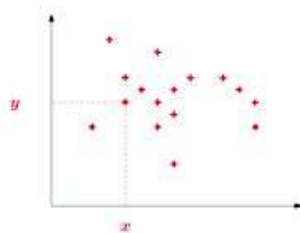
- Formalize the idea of approximation
  - to bring the correctness problem at range
- Application of abstraction to
  - the semantics of programming languages
  - static program analysis
- competes with
  - deductive methods
  - model-checking
  - type inference

171

ISAE – 2022/2023

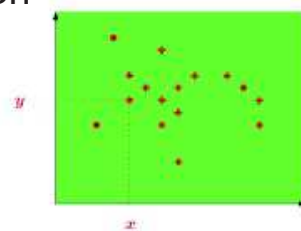
## A glance at the theory (1/3)

### Simple abstraction



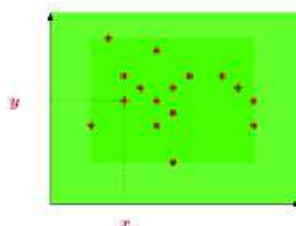
(a) [In]finite Set of Points

$$\{\dots, \langle 5, 7 \rangle, \dots, \langle 13, 21 \rangle, \dots\}$$



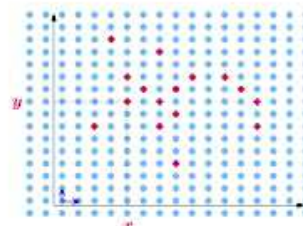
(b) Sign Abstraction

$$\begin{cases} x \geq 0 \\ y \geq 0 \end{cases}$$



(c) Interval Abstraction

$$\begin{cases} x \in [3, 27] \\ y \in [4, 32] \end{cases}$$



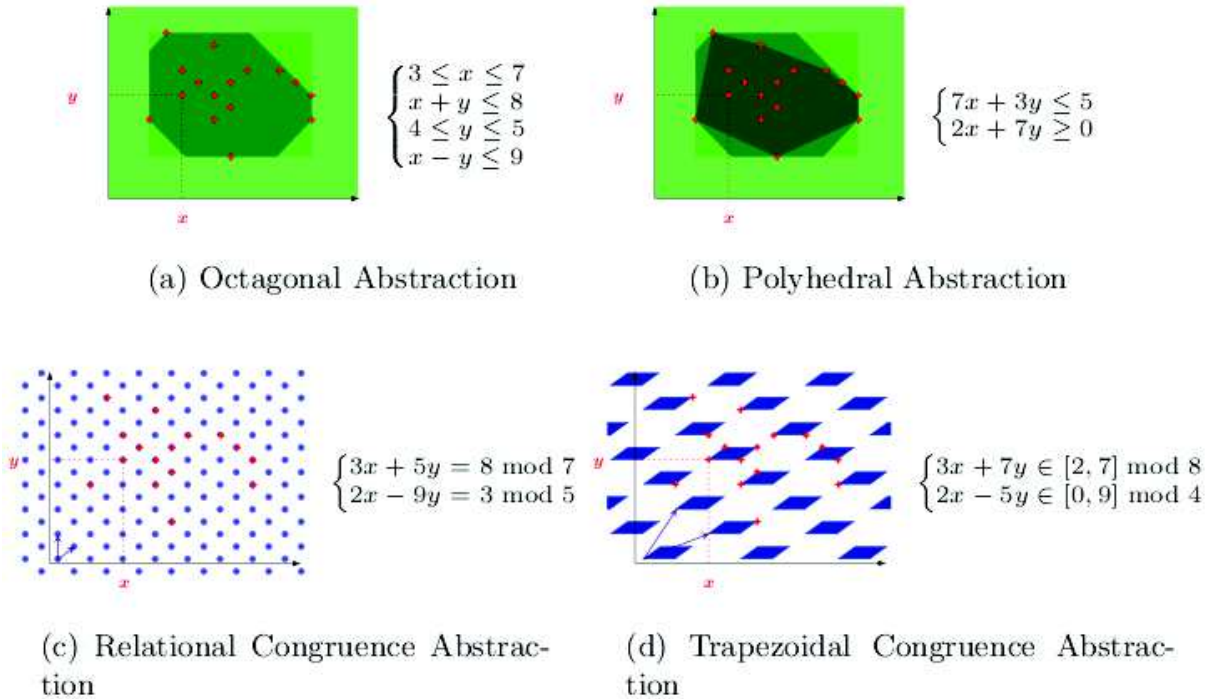
(d) Simple Congruence Abstraction

$$\begin{cases} x = 5 \bmod 8 \\ y = 7 \bmod 9 \end{cases}$$

172

## A glance at the theory (2/3)

### Effective abstraction

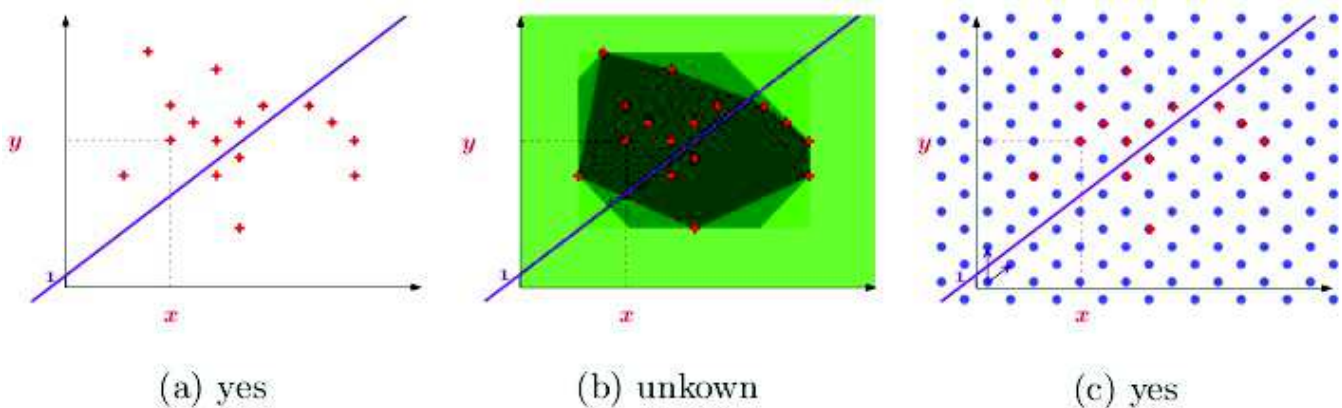


*Abstract Interpretation Based Formal Methods and Future Challenges*, Patrick Cousot, in *Informatics, 10 Years Back - 10 Years Ahead*, R. Wilhelm (Ed.), LNCS 2000, 2001.

173

## A glance at the theory (3/3)

### Information loss and checking



**Fig. 10.** Is  $1/(X+1-Y)$  well-defined?

*Abstract Interpretation Based Formal Methods and Future Challenges*, Patrick Cousot, in *Informatics, 10 Years Back - 10 Years Ahead*, R. Wilhelm (Ed.), LNCS 2000, 2001.

174

## Some concluding remarks

- Complete verification by formal methods
  - model checking / deductive methods
  - very costly in human resources
  - not likely to scale up
- Partial verification by static analysis
  - cost effective
- Program debugging
  - remains the prominent industrial « verification » method
  - well know deficiencies: uncompleteness, cost
- NB: Fault removal, but also fault prevention, fault tolerance and fault forecasting

176

## Security protocols analyzers

- Initial work with NRL PVS protocol analyser or Isabelle/HOL
  - IKE, Kerberos V4/V5, protocol specification
- Now
  - ProVerif
  - Maude-NPA
  - Athena
  - CPSA
  - Scyther
  - *Tamarin*
- Research example : Verified Secure Routing
  - Down to the implementation (SCION)

177

# Dolev-Yao attacker model

- Most common formal model used to prove properties of interactive cryptographic protocol
  - Attacker only limited by the constraints of cryptography
  - $E_k(M)$  reveals nothing about M
  - Adversary can hear, intercept and synthesize any message
  - « The attacker carries the message »

178

ISAE – 2022/2023

## Overall presentation (1/2)

- Fast paced computer security walkthrough
  - Security properties
  - Attacks categories
  - Elements of cryptography
  - Introduction to mandatory security policies
- Embedded systems and security
  - Specificities
  - Physical attacks (SPA, DPA)
  - TPM
- Software development and security
  - Security requirements and process
  - Static verification and software development tools
  - **Common criteria / ISO 15408**

179



## « Criteria »

- Genealogy
  - TCSEC – Trusted Computer System Evaluation Criteria – DoD 1985 (Orange book) and TNI – Trusted Network Interpretation of the TCSEC (Red book)
  - ITSEC – Information Technology Security Evaluation Criteria (EEC 1991)
  - JCSEC, CTCPEC, etc.
  - CC – Common Criteria also known as ISO15408 (ISO standard since ~2000)

180

## Orange book : criteria (1/2)

- |                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• Security policy           <ul style="list-style-type: none"> <li>• discretionary access control</li> <li>• Object reuse control</li> <li>• Labels</li> <li>• Mandatory access control</li> </ul> </li> <li>• Imputability (?)           <ul style="list-style-type: none"> <li>• Identification and authentication</li> <li>• Trusted path</li> <li>• Audit</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>• Operational assurance           <ul style="list-style-type: none"> <li>• System architecture</li> <li>• System integrity</li> <li>• Covert channels analysis</li> <li>• Installation management</li> <li>• Secure recovery</li> </ul> </li> </ul> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

182

## Orange book : criteria (2/2)

- Life cycle assurance
  - Security tests
  - Specification and verification
  - Configuration management
  - Secure distribution
- Documentation
  - User guide
  - Secure installation manual
  - Tests documentation
  - Security management documentation

183

ISAE – 2022/2023

## ITSEC - Criteria

- Functionality classes
- Assurance – Correctness : E1 to E6
- Assurance – Effectiveness
  - Construction
    - Suitability of functionality
    - Binding of functionality
    - Strength of mechanisms
    - Construction vulnerability assessment
  - Operation
    - Ease of use
    - Operational vulnerability assessment

184



## Nice quote on criteria

- CC – ISO 15408
  - Common Criteria
- « For the most part, the protection profiles define away nearly all of the interesting threats that most systems face today. » *in* Fedora and CAPP, lwn.net, 10 dec. 2008.

Not the end of story however (oldest standard).

185

ISAE – 2022/2023

## The Blowfish

- « *Compared to many of the options found in Linux, unveil() is an exercise in simplicity.* », J. Corbet, [767137](#).
- `privsep +pledge()` :
  - `stdio`, `rpath`, `wpath`, `inet`, `dns`, `getpw`, `proc`, `exec`, ...
- Reducing ROP gadgets (RETGUARD) as (yet) another mitigation
- *Only two remote holes in the default install, in a heck of a long time!*
- <https://man.openbsd.org/>
  - « `man man`, *man* », D. Clar, circa 1991...
- Not even a word about [pf\(4\)](#) in the « security » page.

186

# Overall presentation (2/2)

- Case studies
  - Wireless networks
  - New generation avionics systems
  - Network appliances
  - Mobile telephony
  - Gaming devices
- Wrap-up (on-demand)
  - IDS
  - Firewalls
  - Tripwire
  - Metasploit
  - Anti-virus

187

ISAE – 2022/2023

## Now



Photo: resp.

188

# Still now

Automatic Taxi

vs.

Jeep Cherokee: Owned!



Photo: Mark Harris



Photo: Zoox



Photo: Whitney Curtis for Wired



Photo: Andy Greenberg for Wired

189

ISAE – 2022/2023

## Toaster hacking

**Fridge**

pwned?



IoT Village  
@IoTville

Follow

Can you own our #IoT #Samsung -  
RF28HMEBBSR fridge ::] @\_defcon\_



Une suggestion pour sauver  
l'électroménager français :  
la **balance espion**

**Already done!**

190

## Check too

- Abusive protection is the latest fashion...

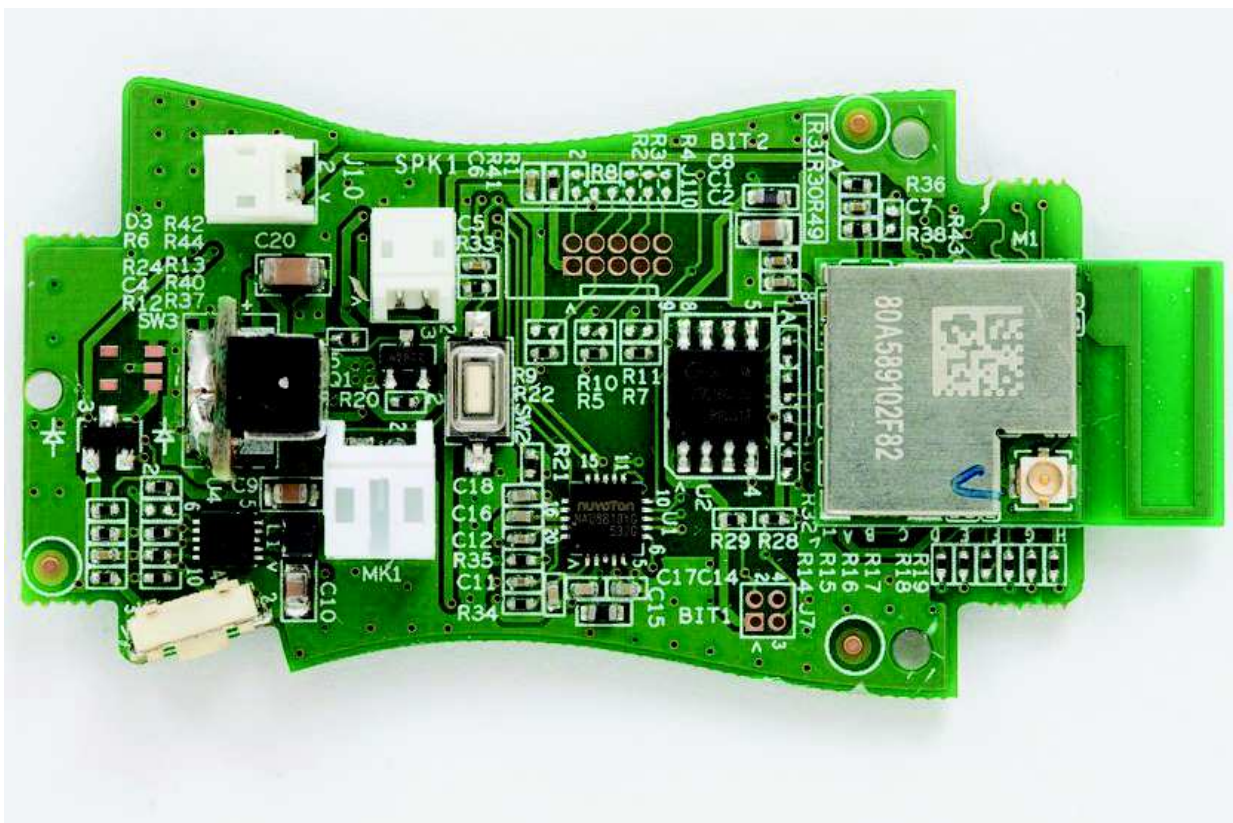


Photo: Corbis

191

ISAE – 2022/2023

## Nearly forgot to remember that



192



# Next ?



vs.



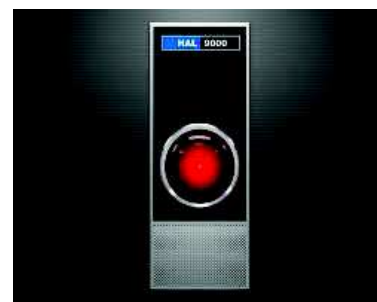
The only sure thing is that it will be the user's fault.

193

## NB : Past



23



HAL 9000

*2001 Space odyssey*, Stanley Kubrick & Arthur Clarke, 1968.

Note (*2010 Odyssey 2*): Contrary to duty imperative, R. Chisholm, 1963.

## Overall presentation (2/2)

- Case studies
  - **IoT Security**
  - **Wireless networks**
  - New generation avionics systems
  - Network appliances
  - Mobile telephony
  - Gaming devices
- Wrap-up (on-demand)
  - IDS
  - Firewalls
  - Tripwire
  - Metasploit
  - Anti-virus

195

ISAE – 2022/2023

## IoT Security :

Let's Forget All The Lessons From Traditional  
Network Security And Hope For The Best

*Keynote Speech*, James Mickens,  
27th Usenix Security Symposium (august 2018)

196

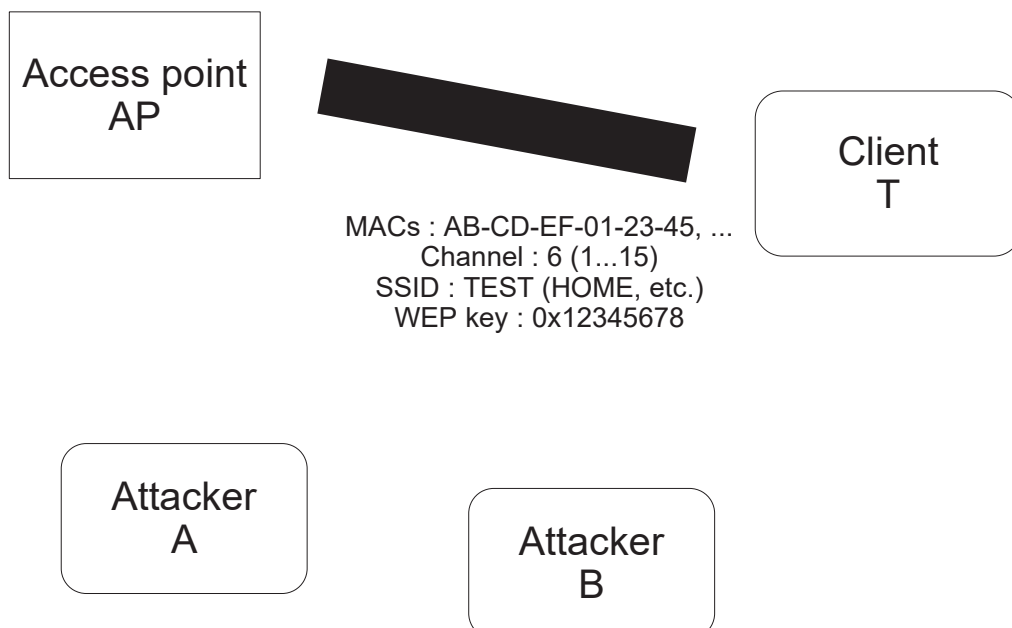
# A wireless network

- WiFi
  - IEEE 802.11a/b/g
  - radio waves
- secured by WEP
  - design fault : uses RC4
  - deprecated : WPA(TKIP), WPA2(CCMP), EAP
- attack example
  - source: Tom's Hardware Guide, 10&18/05/2005
  - tools: kismet, airodump, void11, aireplay, aircrack

197

ISAE – 2022/2023

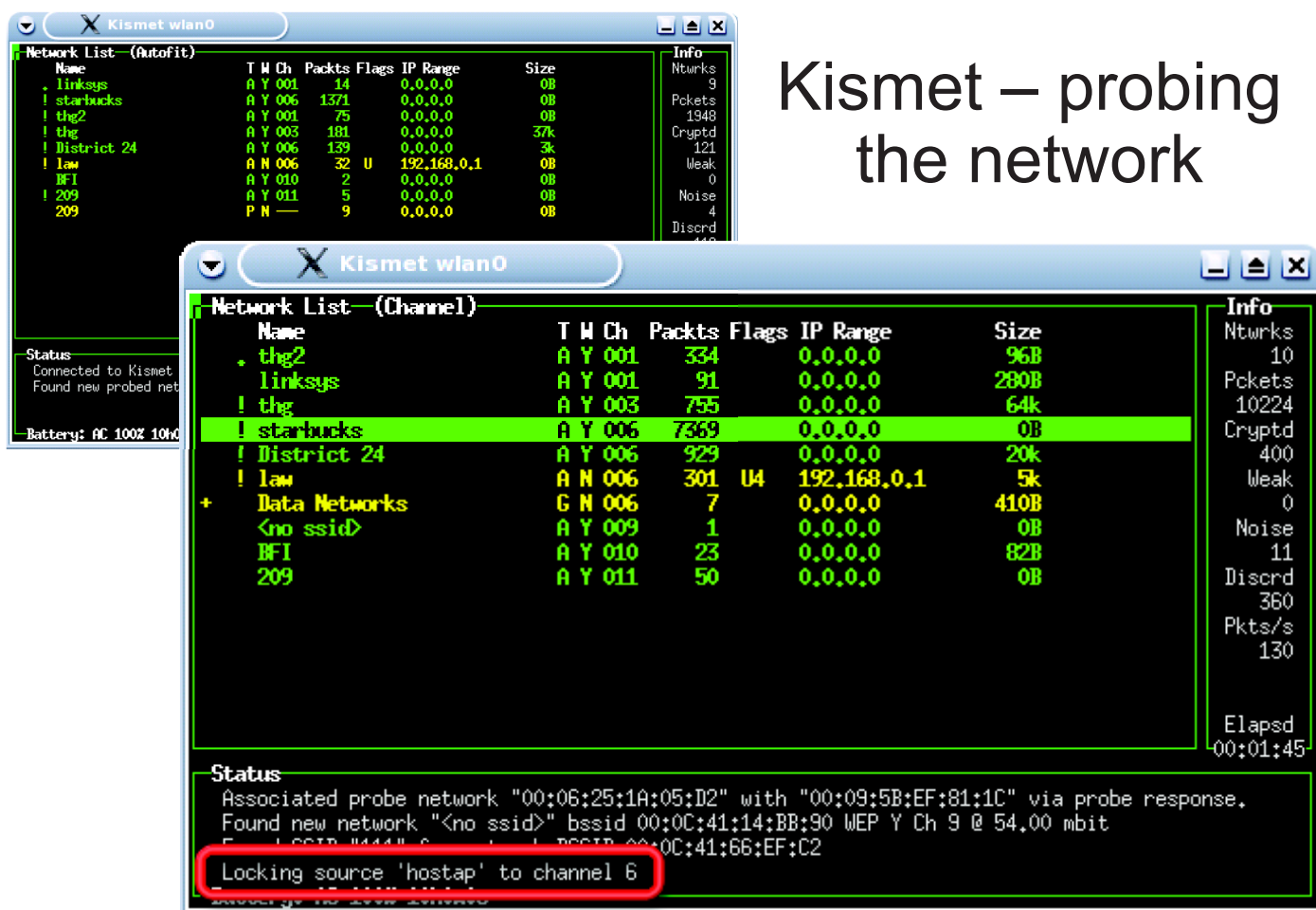
## Test network



198

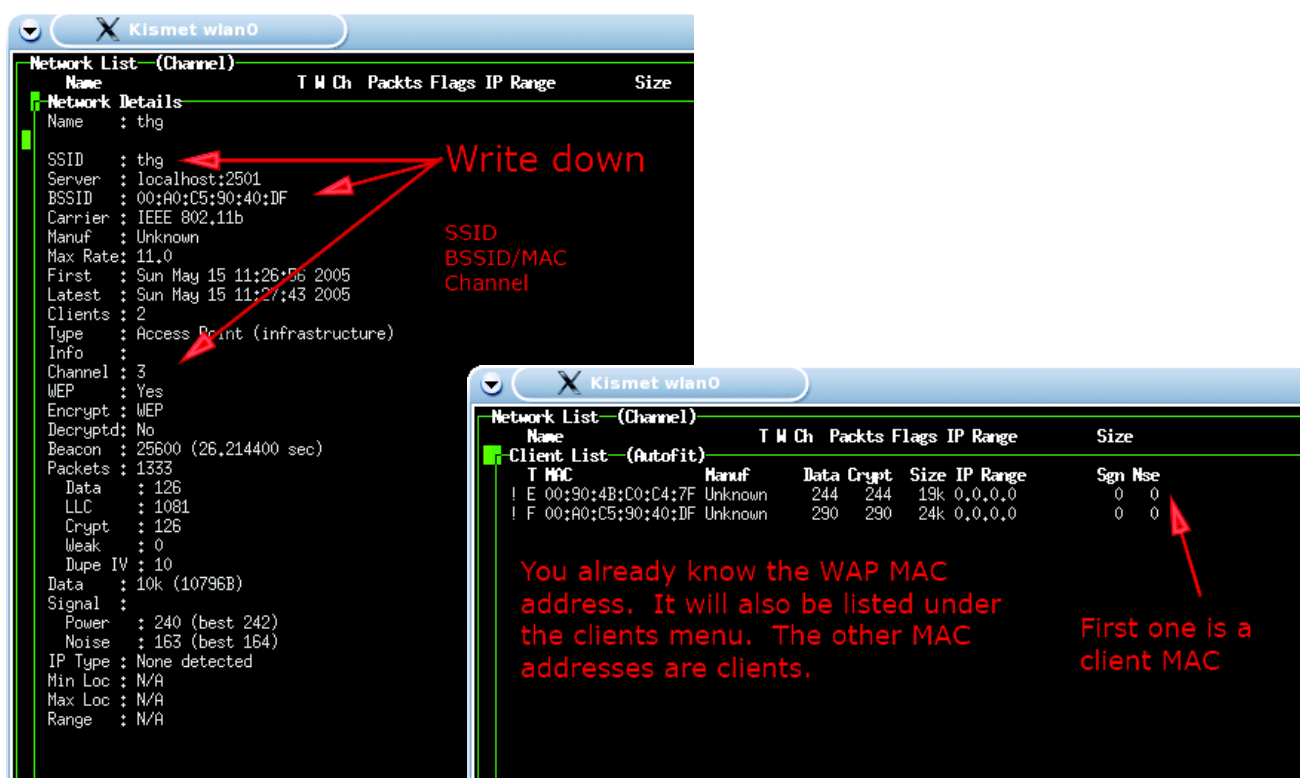


# Kismet – probing the network



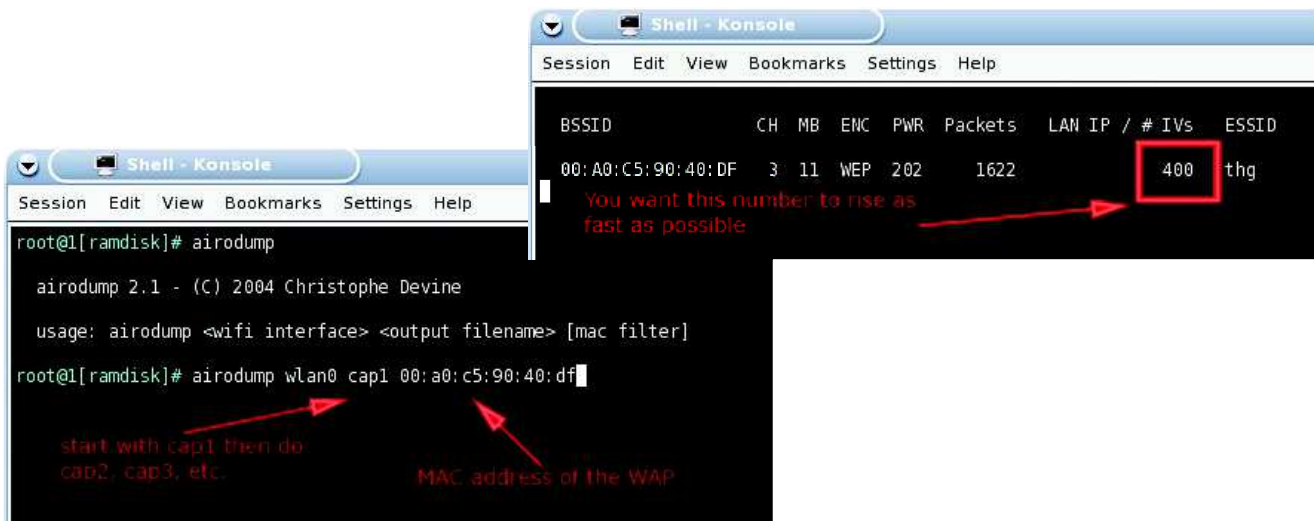
199

# Kismet – targetting



200

# Dumping packets (IVs) - airodump

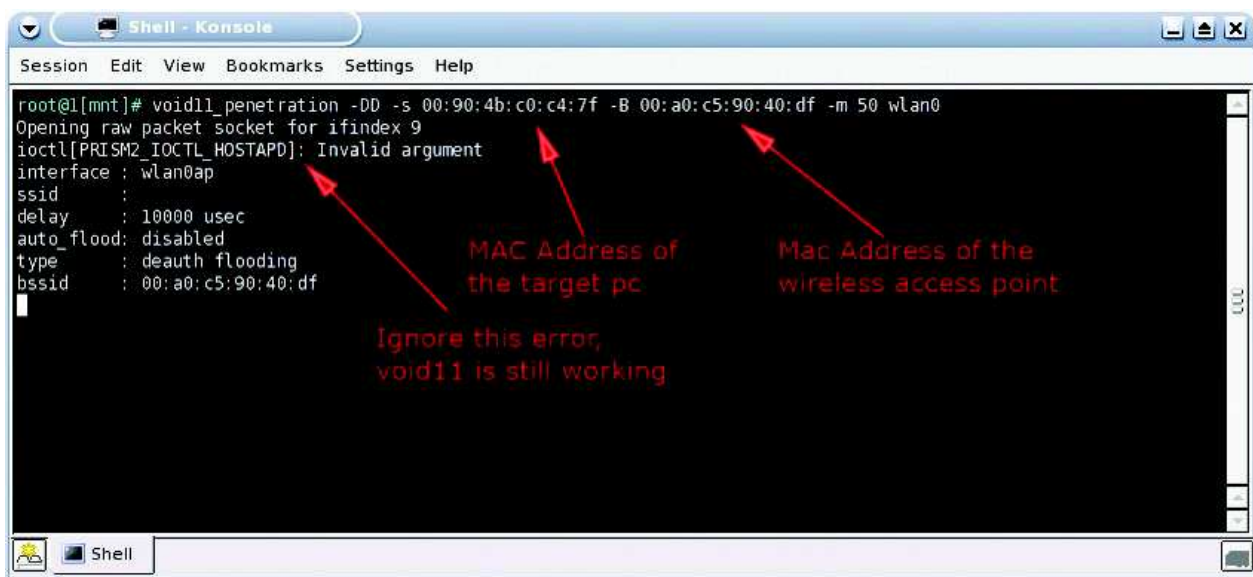


Number of needed packets

- 64bits WEP key : ~ 50 000 – 200 000 IVs
- 128bits WEP key : ~ 200 000 – 700 000 IVs

201

## Active attack – void11



Very noisy !  
~ 100 IVs generated per second

202

# Stealth attitude – aireplay

Packet capture (ARP)  
Re-send while masquerading  
as the target  
around 200 IVs per second

```

aireplay 2.2 - (C) 2004,2005 Christophe Devine

usage: aireplay [options] <interface #0> [interface #1]

interface #0 is for sending packets; it is also used to
capture packets unless interface #1 is specified.

source options:
 -i file : capture packet on-the-fly (default)
 -r file : extract packet from this pcap file

filter options:
 -b bssid : MAC address, Access Point
 -d dmac : MAC address, Destination
 -s smac : MAC address, Source
 -m len : minimum packet length, default: 40
 -n len : maximum packet length, default: 512
 -u type : fc, type - default: 2 = data
 -v subit : fc, subtype - default: 0 = normal
 -t tods : fc, To DS bit - default: any
 -f fromds : fc, From DS bit - default: any
 -w iswep : fc, WEP bit - default: 1
 -y : don't ask questions, assume yes

replay options:
 -x nbpps : number of packets per second
 -a bssid : set Access Point MAC address
 -c dmac : set Destination MAC address
 -h smac : set Source MAC address
 -o fc0 : set frame control[0] (hex)
 -p fc1 : set frame control[1] (hex)
 -k : turn chopchop attack on

root@l[ramdisk]# aireplay -i wlan0 -b 00:a0:c5:90:40:df -m 68 -n 68 -d ff:ff:ff:ff:ff:ff
Option -x not specified, assuming 256.
Seen 923 packets...

FromDS = 0, ToDS = 1, WEP = 1
BSSID = 00:A0:C5:90:40:DF
Src. MAC = 00:09:20:48:D1:26
Dst. MAC = FF:FF:FF:FF:FF:FF

0x0000: 0841 d500 00a0 c590 40df 0009 2d48 d126 .A.....H.&
0x0010: ffff ffff ffff c010 0c01 0000 f6b7 f698
0x0020: 28c8 658e d09c 8a89 2d1b 1757 a135 23b5 (.e.....W.5#
0x0030: a136 290a 9ca6 c0bd 7ac1 9189 b206 651f .6)....Z....e
0x0040: ef25 aa3e%>

Use this packet ? y

Saving chosen packet in replay_src-050515-230051.pcap
Sent 16074 packets...

```

Shell - Konsole

Session Edit View Bookmarks Settings Help

| BSSID             | CH | MB | ENC | PWR | Packets | LAN IP / # IVs | ESSID |
|-------------------|----|----|-----|-----|---------|----------------|-------|
| 00:A0:C5:90:40:DF | 3  | 11 | WEP | -1  | 135840  | 113735         | thg   |

Much faster with aireplay  
Took about ten minutes

203

## Last touch – aircrack

```

Shell - Konsole <2>
Session Edit View Bookmarks Settings Help

root@2[ramdisk]# aircrack

aircrack 2.1 - (C) 2004 Christophe Devine

usage: aircrack [options] <pcap file> <pcap file> ...

-d <start> : debug - specify beginning of the key
-f <fudge> : bruteforce fudge factor (default: 2)
-m <maddr> : MAC address to filter usable packets
-n <nbits> : WEP key length: 64 / 128 / 256 / 512
-p <nfork> : SMP support: # of processes to start
-q <quiet> : Quiet mode (Less print more speed)

root@2[ramdisk]# aircrack -f 2 -m 00:a0:c5:90:40:df -n 64 -q 3 cap*.cap

```

```

aircrack 2.1

* Got 50335! unique IVs | fudge factor = 4
* Elapsed time [00:00:12] | tried 4589 keys at 22945 k/m

KB depth votes
0 1/ 6 0B(12) F9(10) 69(5) F0(5) FC(3) 00(0)
1 2/ 9 69(12) 83(5) 7E(4) 80(3) EA(3) ED(3)
2 0/ 3 A2(32) 82(12) 8E(12) 22(5) 52(5) 7C(5)
3 5/ 10 6D(3) D7(3) D9(3) DA(3) FE(3) 00(0)
4 8/ 14 5A(3) 66(3) 67(3) 69(3) 6C(3) 81(3)

KEY FOUND! [0869A26D5A]

root@2[ramdisk]#

```

Crypto. attack against RC4  
(Fluhrer, Mantin, Shamir)  
aircrack-ptw (better?)  
WEP : K.O. (1min 3s?)

204

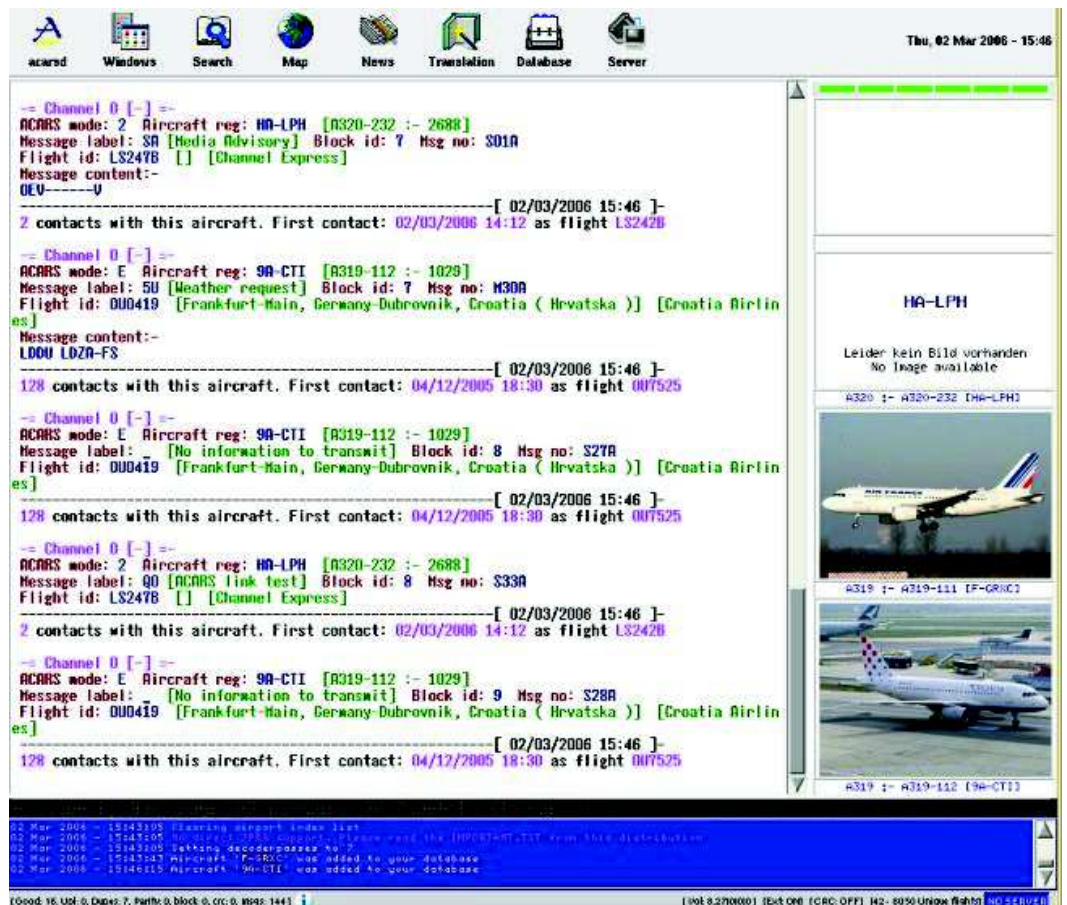
# Overall presentation (2/2)

- Case studies
  - Wireless networks
  - **New generation avionics systems - Industrial systems**
  - Network appliances
  - Mobile telephony
  - Gaming devices
- Wrap-up (on-demand)
  - IDS
  - Firewalls
  - Tripwire
  - Metasploit
  - Anti-virus

205

- [www.acarsd.org](http://www.acarsd.org) ISAE – 2022/2023

«Highly specific» technology  
1978-2018 ?



206

# ATA Spec 42

- « Aviation » certificates
  - Aviation Industry Standards for Digital Information Security
- X.509
  - RFC 5280

207

ISAE – 2022/2023

# ACARS Message Security

- ARINC 823P1
- Note ARINC 823P2
  - Key management
    - with ATA Spec 42

208



# EUROCAE / RTCA documents

- EUROCAE ED-202A / RTCA DO-326A, *Airworthiness Security Process Specification*, June 2014
- EUROCAE ED-203A / RTCA DO-356A, *Airworthiness Security Methods and Considerations*, June 2018
- EUROCAE ED-204A / RTCA DO-355A, *Information Security Guidance for Continuing Airworthiness*, September 2020

209

ISAE – 2022/2023

## ED-202A

- A risk management process
  - Classical risk management over a V process
- Aircraft and system levels
- Security environment and security perimeter
  - NB : Security « *measures* » (rules)
- Still significant effort on threat scenarios identification
- Security effectiveness
  - How well the aircraft is protected against unauthorized interactions
    - Similarity with CC evaluation assurance level or not ?

210

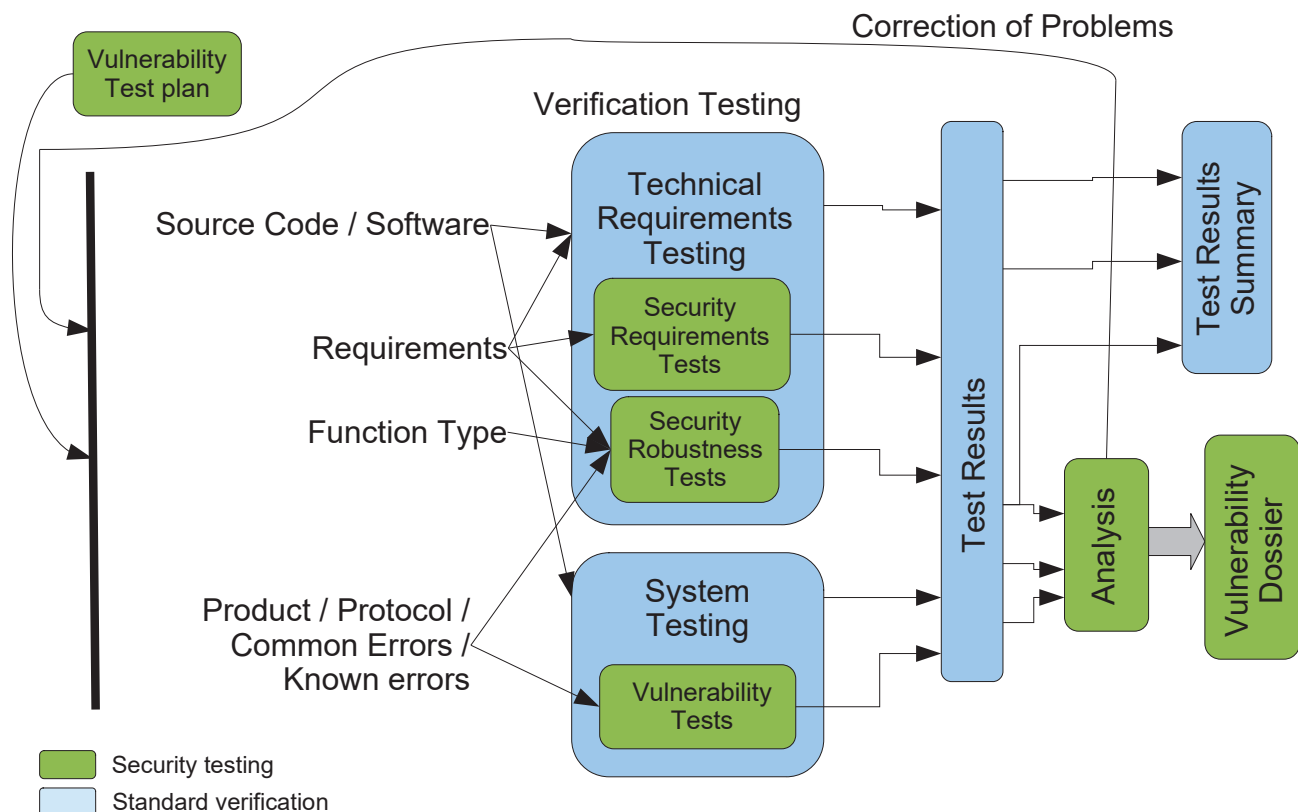
# Process mania

- Airworthiness security process activities
  - Aircraft Security Scope Definition
  - Preliminary Aircraft Security Risk Assessment
  - Aircraft Security Architecture and Measures
  - System Security Scope Definition
  - Preliminary System Security Risk Assessment
  - System Security Architecture and Measures
  - System Security Integrator's Guidance
  - System Security Verification
  - System Security Risk Assessment
  - Aircraft Security Operator's Guidance
  - Aircraft Security Verification
  - Aircraft Security Risk Assessment

211

ISAE – 2022/2023

## Zoom on Security Testing Activities



ED-202A, Fig.3-7

212



# ED-203A

- *Airworthiness Security Methods and Considerations*
  - Regulatory considerations
  - Risk assessment
  - Security assurance
  - Development of security architecture and ‘measures’
  - Security event logging
  - Appendixes
    - Security assurance objectives, and guidance
    - Process table
    - Examples : method, effectiveness, threat evaluation
    - STPA-SEC
    - Cybersecurity risk assessment methodology
    - Security architecture and measures

213

ISAE – 2022/2023

## Risk assessment and security assurance

- Risk (and threat) level determination and analysis
  - Lots of fault tree methodology reuse (condition identification and effect evaluation)
- Security security assurance
- Security development assurance
  - Can share some safety assurance considerations
- Security effectiveness
  - Introduction of refutation
  - Continued security effectiveness
- Security assurance levels (SAL)

214

# Principles of security architectures

## *Aircraft level*

- Defense in depth
  - Integrity of connected equipment
  - Continued airworthiness
  - Prevent bypass of security barriers
  - Keep security architecture as simple as possible
  - Detection and restoration
- ## *(Multi-)System level*
- Attack path refinement at system level
  - Consider security process specifics

- Minimize external interfaces
  - Disable all unused interfaces
  - Independence and isolation (esp. between SAL)
- ## *Security architecture at item level*

- Ensure proper error handling
- Least privilege
- Control access to connections

215

ISAE – 2022/2023

## Appendices notes

- Refutation (B)
- Risk scoring (F)
- STPA-SEC (G)
- Examples (of methodologies and measures)

217

## ED-204A (*Inf. Sec. Guid. for Continuing Airworthiness*)

- Generalities over 11 domains
  - Airborne software
  - Aircraft components
  - Aircraft network access points
  - Ground Support Equipment
  - Ground Support Information Systems
  - Digital Certificates
  - Aircraft Information Security Incident Management
  - Operator Aircraft Information Security Program
  - Operator Organization Risk Assessment
  - Operator Personnel Roles and Responsibilities
  - Operator Personnel Training

218

## ED-204A (*Inf. Sec. Guid. for Continuing Airworthiness*)

- For example
  - Hardening
  - Access control
  - Incident management
  - Transport, storage, lifecycle where appropriate
  - Decommissioning sometimes
  - A little more details on ground support equipment or « loadable software »
- Critique
  - Skills and training lack actual examples to assess level and types of knowledge involved
    - De facto, basic levels will apply

219

## « New » secure radio networks

- V2X aka connected vehicle
- IoT networks
- Link 16 aka MIL-STD-6016
- AeroMACS
- LDACS

220

ISAE – 2022/2023

## LDACS

- L-band digital aeronautical communication system
- [www.ldacs.com](http://www.ldacs.com)
- Core contributor German DLR
- Upcoming communications standard
  - Eurocontrol, ICAO
- VHF communication channel
- 550kbit/s to 2.6Mbit/s
- Cellular system
- Strong demands
  - Communication + navigation
  - Spectrum efficient
  - Secure (but bandwidth)

221

# LDACS security

- First : performance overhead checked
  - Note this is an extremely bad signal sent to security professionals...
- Private (encrypted) channels
- Aero X.509 PKI authentication
- Cell-attachement procedure
  - Formally verified !
  - Tamarin
  - Dynamic network management ?

222

ISAE – 2022/2023

# LDACS cell association

- Design goals (requirements)
  - Mutual authentication, perfect forward secrecy and secure key establishment
    - In the event of long-term keys compromise
    - That's 3 reqs ?
  - Robustness
    - Reuse of existing known protocols
  - Security levels
    - Different cipher suites
  - Certificate revocation
    - Via OCSP for reasons... 1... 2... 3...
      - Once again not a requirement but mechanism...
- Note then LDACS GS/AS certificates content
  - Is then defined !
    - With a post-quantum encryption algo option... and (mandatory?) 1 year (GS) / 3 years (AS) validity

## LDACS cell association (2)

- Main steps (CDS, GS, AS)
  - GS checks local certificates validity (GS,  $AS_i$ )
  - Identity advertisements with (public) keys and nonce plus address  $SAC_{AS}$
  - AS proves to GS and then GS answers to AS
  - Secret session key  $K_{AS,GS}$  derivation and encryption scheme proposal via MAKE algo (ephemeral secret selected on AS)
  - GS does its part ( $?!?_{step16}$ ) and answers
  - AS final check

N. Maurer, et al., "A Secure Cell-Attachment Procedure of LDACS," in 2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Vienna, Austria, 2021 pp.113-122.

224

## AeroMACS

- 5G in airports
- Lots of efforts towards EM compatibility
- Useful for many usages
- Security relying on cellular protocols and...
  - Aero PKIs of course...

# V2X : véhicule to everything

## V2V - Vehicle-to-Vehicle.

Alerts one vehicle to the presence of another. Cars "talk" using DSRC technology.

## V2D - Vehicle-to-Device.

Vehicles communicate with cyclists' V2D device and vice versa.

## V2P - Vehicle-to-Pedestrian.

Car communication with pedestrian with approaching alerts and vice versa.

## V2H - Vehicle-to-Home.

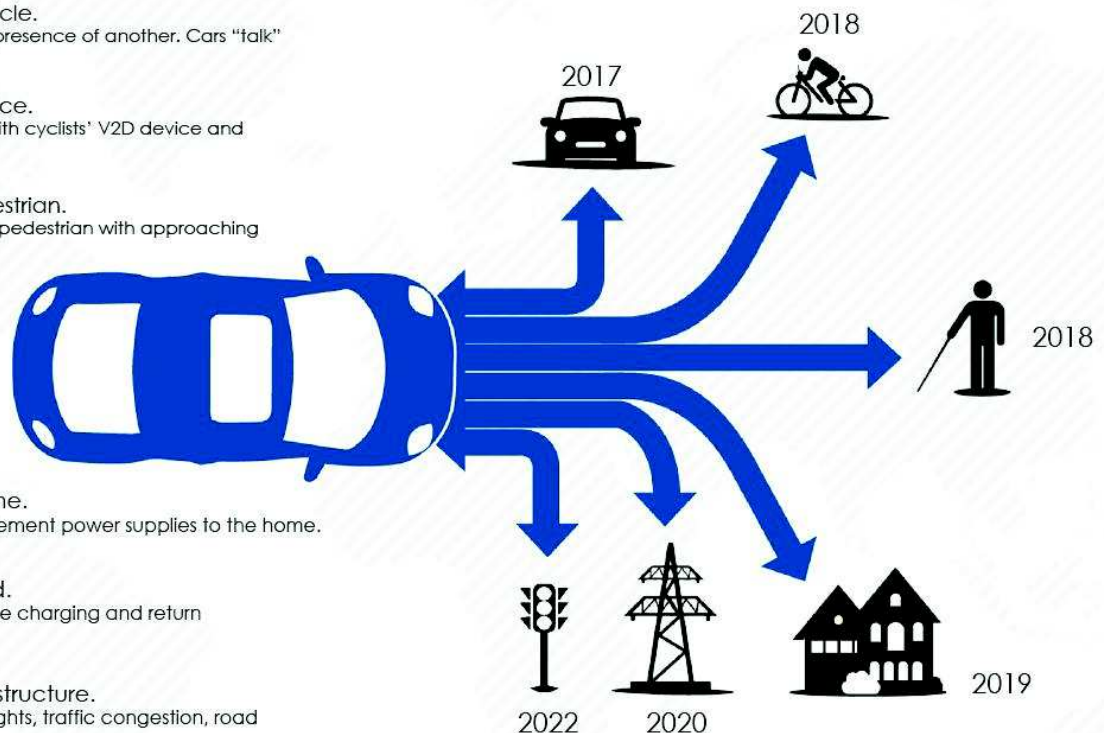
Vehicles will act as supplement power supplies to the home.

## V2G - Vehicle-to-Grid.

Smart grid controls vehicle charging and return electricity to the grid.

## V2I - Vehicle-to-Infrastructure.

Alerts vehicles to traffic lights, traffic congestion, road conditions, etc.



Mahibul Alam © 2016 All rights reserved

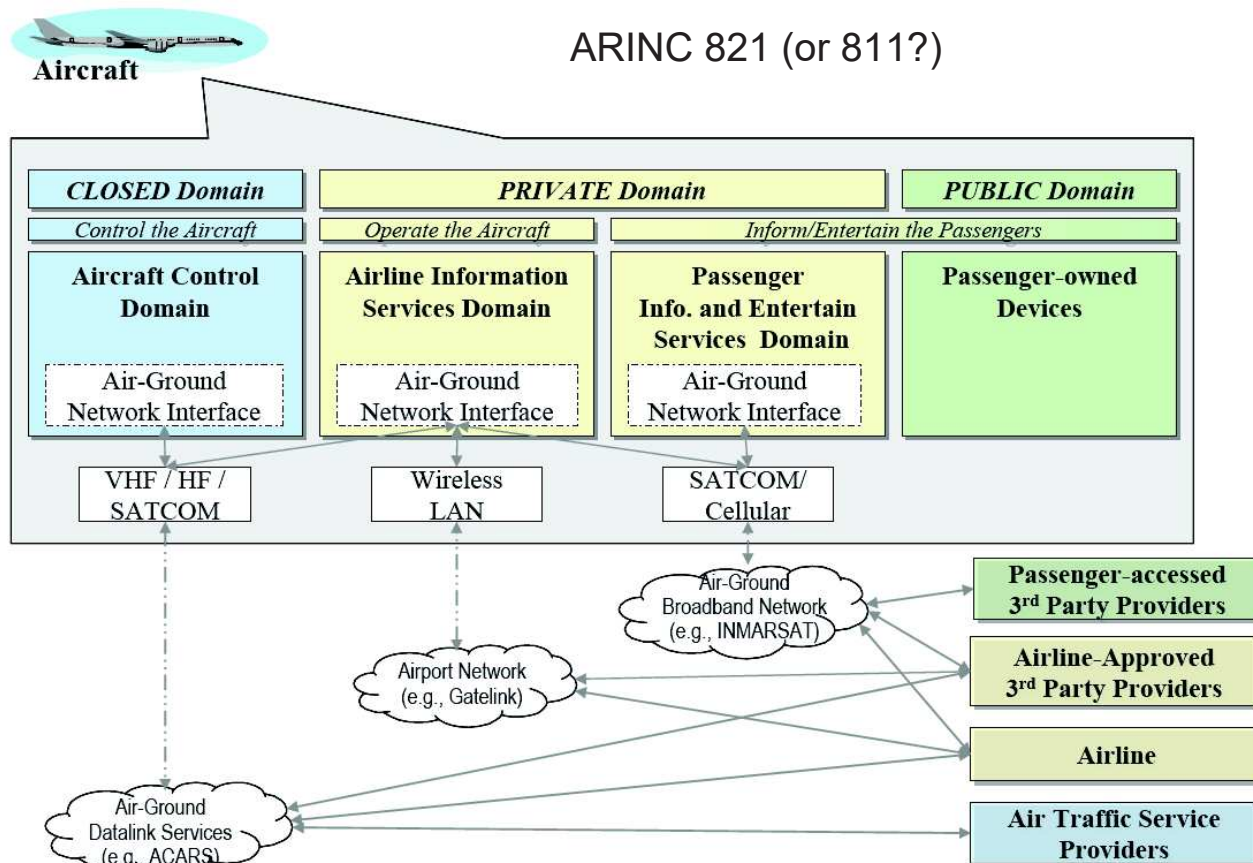
226

## New V2X wifi things

- V2X : Véhicule to everything
  - IEEE 801.11p (WLAN based)
  - C-V2X (cellular)
    - Esp. 5G
- PRESERVE security architecture  
<https://www.preserve-project.eu/>



# The ARINC overall model



## DOT/FAA/AR-08/31

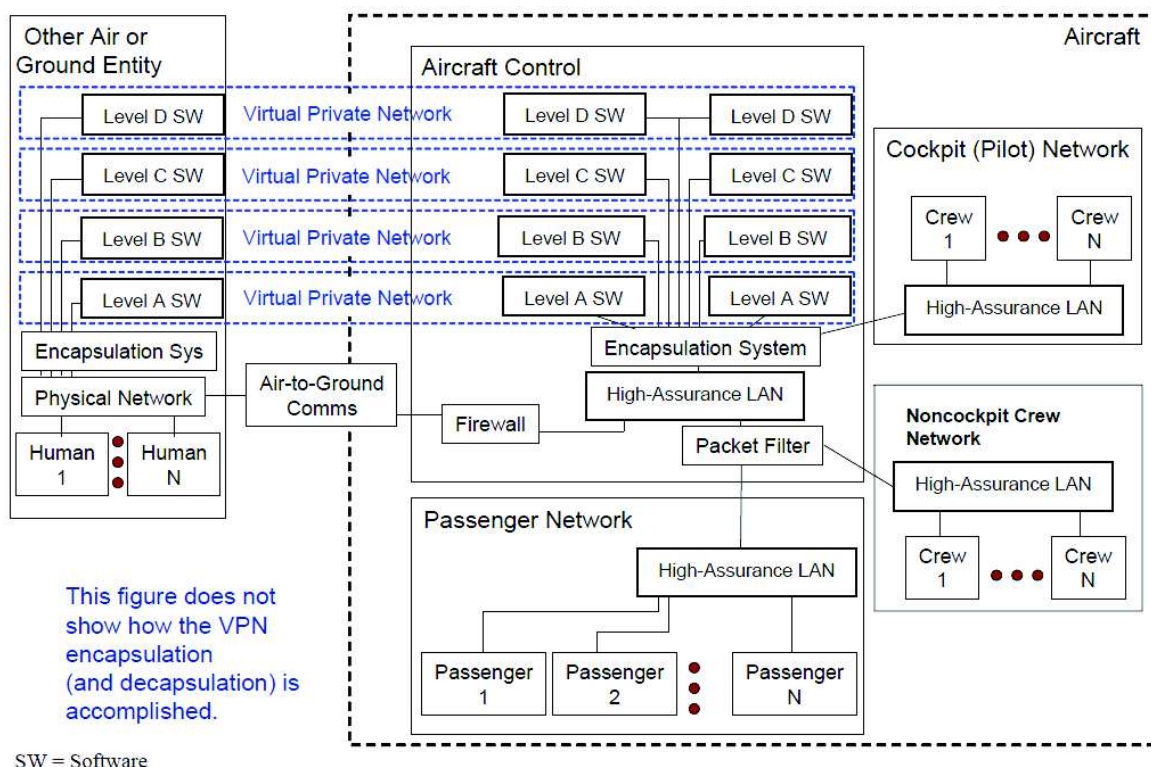


Figure 30. Secure Generic Airborne Network Design (High-Level View)

## Focus on maintenance (and software upload)

- Data loading on avionics systems
  - Uploading data and functional programs (software) from the data loader to airborne computers
  - Downloading data
  - ARINC 615
    - Uses ARINC 429 data bus for file transfers
  - ARINC 615A
    - Uses AFDX/A664 Ethernet for transfers
  - ARINC 665
    - Media set of multiple...
    - ...Loadable software parts (LSP)

Good ref.: [AviftechVideos](#)

238

## Current systems critique

- No explicit security needs formulation
- Confusion between security functions and regular system functionality
- No security requirements for certification (yet)
- Little authentication of users nor systems
- No distributed security services
- Lack of design security skills (for TCBs)
- Availability always safety-oriented
- Use of inadequate mechanisms for security
- Focus on perimetric security or verification
  - Questionable distribution of efforts
- No public liability of security mechanisms

# An astonishing counter-example

- Link 16
  - military tactical data link network used by NATO
  - Near-real time exchange of tactical picture, ie :
    - Identification
    - Location
    - Communications
    - Orders (command)
    - Friend
    - Training
    - not training (ie. War)
  - TDMA-based secure (with qualified crypto), jam-resistant, high speed (~10kb/s)
  - STANAG 5516 (J-type) messaging
  - Not only air : also land, surface, sea, etc.
  - Up to SECRET NATO or MISSION

240

ISAE – 2022/2023

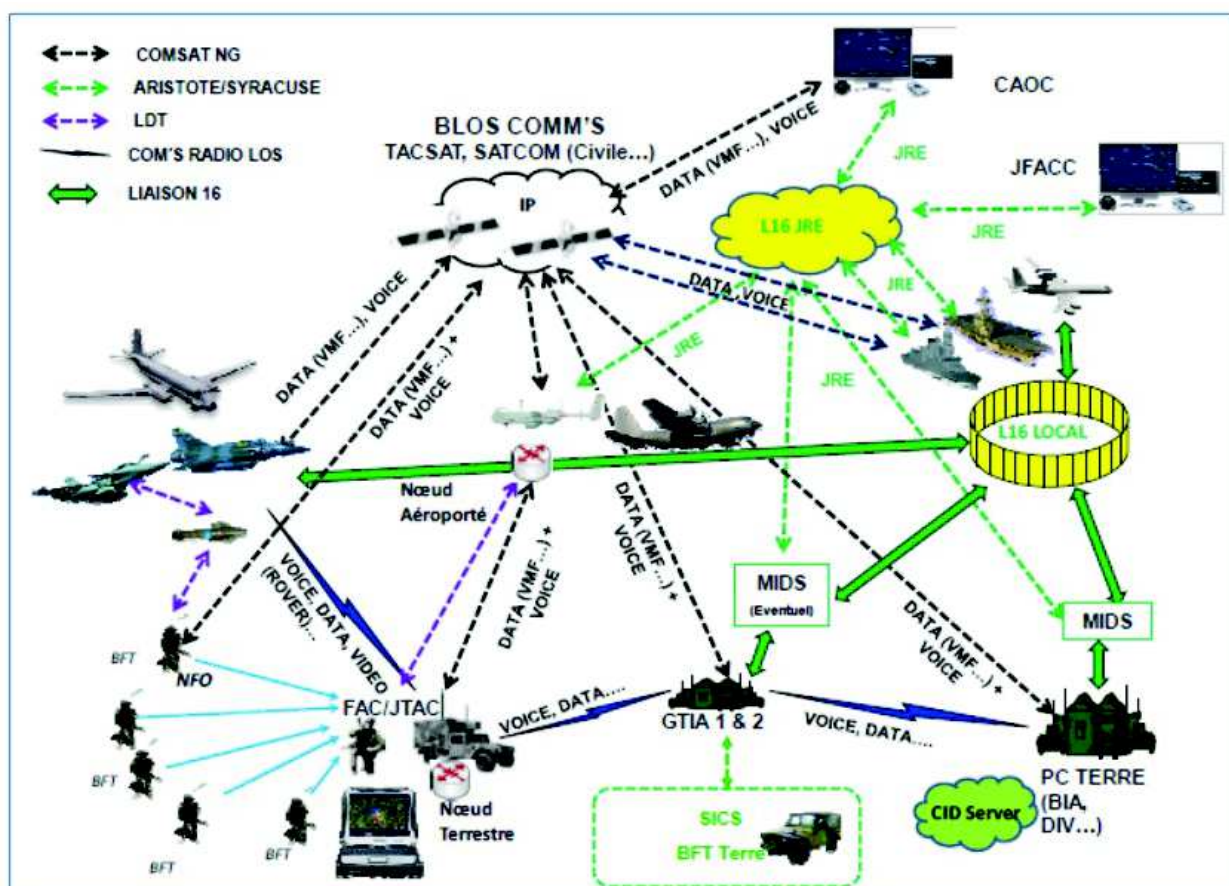


FIG. 4. – Schéma d'illustration du multi-liaisons (réf. OD DaCAS<sup>29</sup>).

241



*Les liaisons de données tactiques (LDT)*, Ministère des armées, République Française,  
Publication interarmées, PIA-3.50\_LDT(2017), N°109/DEF/CICDE/NP du 13 juin 2017.



FIG. 5. – La liaison 16.

242

ISAE – 2022/2023

## Big (secure) picture

- Legally valable always available inflight record
- Data or software update
- Component life cycle, certification and configuration management activities support
- Plane to plane opportunistic communications
- Plane to ground predictable communications
- Onboard ~~unsecure network~~ *Internet*
- Environment to plane cooperation
- Escadrille (vehicle cooperation) and fleet
- User authentication et rights management
- Remote control capability
- Emergency mode

243

## Requirements elicitation attempt - 1

- Physical security assumptions clarification
  - Should we count on hardware protection, or not ?
- Software installation integrity protection and exhaustive traceability
  - Software signature
  - Life long history
- Exhaustive auditing of all important actions with legally binding near realtime offlining
  - No onboard recorder needed anymore
  - No more fingerprinting or who did what questioning

245

## Requirements elicitation attempt - 2

- Opportunity to support several board/ground communication technologies
  - No link to specific wireless
- Adaptability to different onboard network architectures
  - Single network or multiple parallel networks
- Authentication without critical processes perturbation
  - Operators, environment, whatever
- Self authentication

246

## Requirements elicitation attempt - 3

- Configuration management
  - Secure and at least for certification
- Take into account civilian certification issues
  - No secret signature scheme, do sign forms...
- Enable fault-tolerance mechanisms integration
  - Not so easy to do variants authentication (eg)
- Do not compromise deterministic or realtime properties
  - Explicit guarantee that safety *also* rules

247

## Requirements elicitation attempt - 4

- Allow ARINC 811 four domains implementation
- Adapt to IMA idea (*idem*)
  - Aerospace issues
- Energy consumption control (?)
- Offer a good hardware resistance
  - With respect to existing computers
- Remote control service
  - Specific conditions acceptable

248

## Requirements elicitation attempt - 5

- Identity and authorization management functions (either internal or based on ground infrastructure)
  - Nb : a few thousand worldwide users...
- Compatible with existing systems
  - AFDX, Internet, L16, etc. (ok, isolated CAN...)
- Shared implementation opportunity
  - Reusable (if not open source), no lock-in, esp. with respect to protocols or access control

249

## Requirements elicitation attempt - 6

- Explicit phases of operation
  - Security guarantees with respect to transition
  - flying, rolling, etc.
  - sinking, hijacked, drunk-driven open to discussion
- Explicit irreversible phases of life
  - Secure proofs available at any time
  - being-built, in-operation, out-of-order, being-repaired, destroyed, for-sale
  - Whole system, ownership issues included
- Onboard secure maintenance documentation
  - May be mandatory
    - Unless proved really too little used after a (long) while.

250



## Requirements elicitation attempt - 7

- Physically Unforgeable Function
  - dunno but sounds cool and fit
- Possibility of secure cooperation link setup
  - Binding two trains together, escadrille, etc.
- Similar systems should also be able to cooperate securely
  - For less complex operations (collision avoidance, remote status relay, recommendation diffusion, etc.)
- Mandatory obsolescence control&planning
  - with link to hardware resistance

251

## Requirements elicitation attempt - 8

- Autonomous prioritization of critical functions
  - Especially in case of failure or complex security interactions, focus on critical functions if needed
    - esp. those involving realtime constraints
  - Availability-oriented requirement
    - In the context of malicious faults...
  - Do not disturb the program
- Emergency mode availability
  - In case of system failure
  - Security features should only enhance and never downgrade further any fail safe safety mode

252

## Requirements elicitation attempt - 9

- Provide public proofs of components security
  - Including security setup
  - In operation
  - Appropriately signed
    - also prints certification form on luxury paper for chief engineer confirmation signature
- Security minimization
  - Do not include any optional security functions
- (Option) Offer different operation modes
  - certified/test mode, public/confidential mode, etc.

253

## Architecture components - 0

- Security services
- Cryptographic services
- Data management services
- Data modeling functions
- Communication protocols
- Miscellaneous services

# Architecture components - 1

- Security services
  - Integrity
  - Authentication
  - Security states/phases management
  - Software installation
  - Secure logging
  - Remote control
- Cryptographic services
  - Conventional crypto.
  - Execution time evaluation
    - Focussed on appropriate functional subset

256

# Architecture components - 2

- Data management services
  - Security kernel data
    - Dedicated API, destruction and permanent storage functions, time-critical issues, etc.
  - State management data
  - Irreversible phase change
  - Users management
  - Internal documentation

257

## Architecture components - 3

- Data modelling services
  - Phases/states description language
  - System and configuration description language
  - Users and systems representation
  - Certification-related elements
  - Communication data (à la ASN.1)
  - Logging data representation
  - Documentation elements

258

## Architecture components - 4

- Communication protocols
  - (Secure) Conventional communication
  - Proximity communication
    - Limited security, short term ad hoc communication
  - Long term cooperation link
    - Establishment and usage
  - Remote control communication protocol

259

# Architecture components - 5

- Miscellaneous services
  - Embedded
    - Storage and internal communications (network)
      - With critical communication capabilities
    - Long range external communication
    - Configuration management
    - Physical access interfaces (and removable media)
    - Sensors
    - Positioning service
  - Infrastructure
    - Positioning service support
    - Communication infrastructure(s)
    - Version management
    - Certification verification and route control authority
    - Attack simulation (?)

260

ISAE – 2022/2023

## Overall presentation (2/2)

- Case studies
  - Wireless networks
  - New generation avionics systems
  - **Network appliances**
  - Mobile telephony
  - Gaming devices
- Wrap-up (on-demand)
  - IDS
  - Firewalls
  - Tripwire
  - Metasploit
  - Anti-virus

262

## Network *appliances*

- A common type of embedded systems
  - routers, switches
  - ADSL boxes
  - WiFi stations
  - ...
- Cisco OS
  - PIX
  - IOS

263

ISAE – 2022/2023

## A thrilling story

- 2002, Black Hat, Defcon X, other things
- Summer 2005, Black Hat conference
  - ***The Holy Grail: Cisco IOS Shellcode And Exploitation Techniques***
    - Michael Lynn, ISS
  - Cisco and ISS do act
    - complaint
    - on-site action (proceedings confiscated)
  - Michael Lynn, ex-ISS, speaks anyway
- November 2005
  - patch published by Cisco

264

## Random thoughts (true or false)

- Routers and switches use off-the-shelf CPU to run their software
  - hardware is not alone
- There are buffers and they overflow
  - there are no buffers overflow
- You cannot exploit them
  - you can exploit them
- Such exploits are portable
  - each piece of hardware is very different

Heavily based on Michael Lynn's Black Hat presentation

265

## IOS Basics

- Monolithic OS
  - no dynamic modules
  - all addresses are static
  - addresses differ from one build to another
- Realtime OS
  - as soon as you execute you control the CPU
  - exit cleanly (or fail miserably)
  - as soon as you execute you can keep the CPU
- Stability is valued over everything else
  - IOS would rather reboot than correct errors



## Code quality

- Much better than on other platforms
  - Heap internal integrity checks
  - Overflow runtime checks
  - Stack is rarely used
  - A process checks heap integrity
  - Very old code, very tested
- There are still bugs
  - But you need a lot of imagination

267

ISAE – 2022/2023

## *The Dreaded Check Heaps Process*

- Constantly walks the heap to spot bad links
  - Even for unfreed entries, it detects incorrect links
  - Executes every 30 or 60 seconds, depends on load
- It is the primary reason why heap overflow exploits are so hard

268

# Defeating the protection

- Code disassembly
- Lots of time and energy
- Few known tricks
  - pointers exchange
  - heap overflow
- Defeating the heap check process
  - Simulate a reboot (altering abort())
  - a CPU watchdog will kill the heap check process
- Use the available time to complete the exploit

269

ISAE – 2022/2023

## Impact?

- Cisco probably had a hard time
- A generic worm would have been very hard to develop
  - static addresses
  - a lot of different images in production
- But..., some also thought to
  - the Titanic
  - or Pearl Harbor

270

## Overall presentation (2/2)

- Case studies
  - Wireless networks
  - New generation avionics systems
  - Network appliances
  - **Mobile telephony**
  - Gaming devices
- Wrap-up (on-demand)
  - IDS
  - Firewalls
  - Tripwire
  - Metasploit
  - Anti-virus

271

ISAE – 2022/2023

## Mobile telephony (*before*)

- Windows CE (Microsoft)
- Symbian (Nokia)
- *open-source* (as much as possible)
  - Qtopia (TrollTech)
  - Android (Google, Motorola)
  - OpenMoko, OpenEmbedded (Sean, Koen, Harald, Mickey, etc.)



272

# Symbian Devices



## Mobile telephony (*now*)

- Apple iPhone
- Google Android
- Not a phone anymore : a computer
  - a really portable one



# Android & the Droids

- Linux kernel-enforced sandboxing
  - Lots of « permissions » to request (refuse?)
- Application signing
  - Signature-level permissions
- User IDs and file-access
  - 2 applications have 2 UIDs
  - and/but there is « shareUserID »
- Declaring and enforcing permissions
  - Via the androidManifest.xml
- and per-URI permissions

Real-world usage examples?

278

ISAE – 2022/2023

## *Mobilife*

- [www.ist-mobilife.org](http://www.ist-mobilife.org)
- IST-FP6 project (2004-2006)
- End users needs
  - context awareness
  - group management
  - etc. (multimodal interactions, localization, ...)
- Reference architecture
  - ...
  - privacy & trust
  - group management

279

## TCG – Mobile Phone Use Cases (1/3)

- Platform integrity
  - Devices possess and run only authorized operating systems and hardware
- Device authentication
  - to assist in user authentication (hold keys)
  - prove the identity of the device itself
- Robust DRM implementation
- SIMLock / Device Personalisation
  - device remains locked to a particular network

284

## TCG – Mobile Phone Use Cases (2/3)

- Secure software download
  - application, patches, firmware updates, etc.
- Secure channel between device and UICC
  - Some security sensitive applications may be implemented partly in the UMTS Integrated Circuit Card (UICC) and partly in the device.
  - Sensitive (e.g. provisioning) data exchange
- Mobile ticketing
- Mobile payment
- Software use (security policies)

285

## TCG – Mobile Phone Use Cases (2/3)

- Proving platform and/or application integrity to end user
  - The end user wants to know that a Device or application can be trusted
- User Data Protection and Privacy
  - Personally identifiable information
  - Contact /Address books
  - Wallets, credentials, identity tokens

286

ISAE – 2022/2023

## GSM Security

- An old affair ?
- Not so good
  - <http://laforge.gnumonks.org/weblog/gsm/>
  - The network does not authenticate to the phone
  - A5 « private » ciphers family issues

287



# BYO SMS jamming

- « Blowing up the Celly »
  - PacSec 2014, DEFCON 22
  - Brian Gorenc, Matt Molinyawe (HP)
- OpenBTS-based
- RF test enclosure needed
- phone == target

288

ISAE – 2022/2023

## Needed hardware

### Our Bill of Materials

#### USRP and Accessories

USRP N210 Kit (782747-01) - \$1,717.00  
 WBX-40 USRP Daughterboard - \$480.00  
 USRP GPS-Disciplined Oscillator Kit - \$758.00  
 SMA-to-SMA Cable Assembly - \$30.00  
 VERT900 Vertical Antenna Dualband - \$35.00  
**Total: \$3,020.00**

#### RF Enclosure and Accessories

STE3000FAV - \$2,495.00  
 SMA Feedthrough Connectors  
 DB9 10 PF and DB9 100 PF Connectors  
 USB, RJ45 Adapter Kits  
**Total: \$3,096.00**

#### Cell Phones and SIMs

Unlocked Phones ~ \$500  
 Pre-paid SIMs ~ \$10-\$20  
 Micro SIM Cutter Tool ~ \$5  
**Total: ~\$550**



## Overall presentation (2/2)

- Case studies
  - Wireless networks
  - New generation avionics systems
  - Network appliances
  - Mobile telephony
  - **Gaming devices**
- Wrap-up (on-demand)
  - IDS
  - Firewalls
  - Tripwire
  - Metasploit
  - Anti-virus

290

ISAE – 2022/2023

## Gaming devices (>2000)

- Anti-piracy features
- Supplier-controlled software signature
- Protection architecture using hardware components (*hidden ROM*)
- XBOX example
  - Public key in PROM, private key at Bill's
  - Integrity checks starting from boot
  - Attack
    - *reverse engineering* and ROM exchange
    - Using James Bond, a *Mech* or a *sniper*...  
(third party vulnerable code)
- Sony problems

[www.xbox-linux.org](http://www.xbox-linux.org)

... a princess...

[www.wiibrew.org](http://www.wiibrew.org)

291

## Next step

- Multilevel security policy and mandatory access control ?
  - on a gaming device?
  - on a home video recorder? (Philips, DRM)
- OpenBSD : Old style (or not)?

292

ISAE – 2022/2023

## BadUSB

- SecurityResearchLabs study
  - Karsten Nohl, Sascha Krißler, Jakob Lell
  - PacSec Applied Security Conference

293

# BadUSB

- USB devices include a micro-controller and possibly flash storage
- Large family of possible attacks
  - Emulate keyboards
    - Device deregisters then register again as a different one
  - Spoof network card
    - DHCP magic overrides DNS or default gateway
  - « USB boot-sector » virus
  - *Hide data on stick of HDD*
  - *Rewrite data in-flight*
  - *Update PC BIOS*
  - *Spoof display*

294

ISAE – 2022/2023

# BadUSB

- Small hardware differences can determine vulnerability
  - Especially flash presence
- Exposure is probably growing
  - More devices, more complex and more programmable
- Effective defenses are missing
  - Simple ones (disable updates in hardware) are limited to new non upgradable devices
  - Secure crypto. sounds overkill for microcontrollers (though security guys may disagree)
  - Firmware scanning... can of worms
- **No responses**
  - Chip, peripheral or OS vendors alike

295

## Overall presentation (2/2)

- Case studies
  - Wireless networks
  - New generation avionics systems
  - Network appliances
  - Mobile telephony
  - Gaming devices
- Wrap-up (on-demand)
  - **IDS**
  - Firewalls
  - Anti-virus

296

ISAE – 2022/2023

## Vulnerabilities – Attacks – Alerts

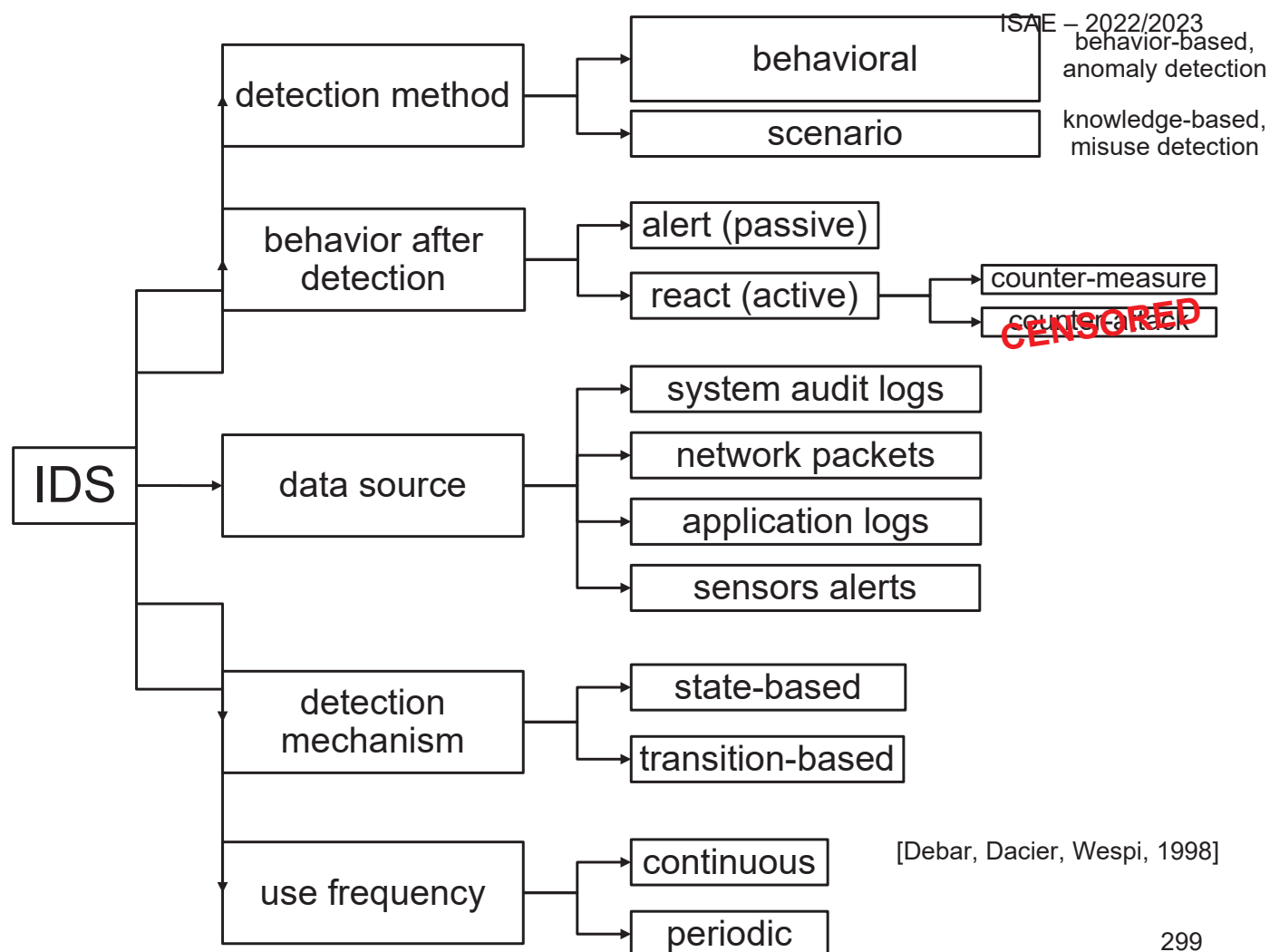
- Vulnerabilities
  - Many types : *buffer overflow*, CGI, permissive access rights, network session hijacking, privilege transfers, *social engineering*, cryptanalysis, etc.
- « Attack »
  - Exploitation of a single vulnerability
  - Elementary attack or intrusion scenario
  - Malicious vs. suspicious action
- Alerts
  - Message sent after detection of an attack
  - *IDMEF (XML): Intrusion Detection Message Exchange Format défini par l'IETF/IDWG*

297

# Alert generation (efficiency)

|                | No alert         | Alert            |
|----------------|------------------|------------------|
| No attack      | True negativef 😊 | False positive 😞 |
| Ongoing attack | False negative 😞 | True positive 😊  |

298



299

# Usable techniques

- Scenario-based approaches
  - Expert system (ES)
  - Signature analysis (SA)
  - Petri nets (PN)
- Behavioral approaches
  - Statistical (ST)
  - Expert system (ES)
  - Neural networks (NN)
  - Immunological approach (UII)

300

ISAE – 2022/2023

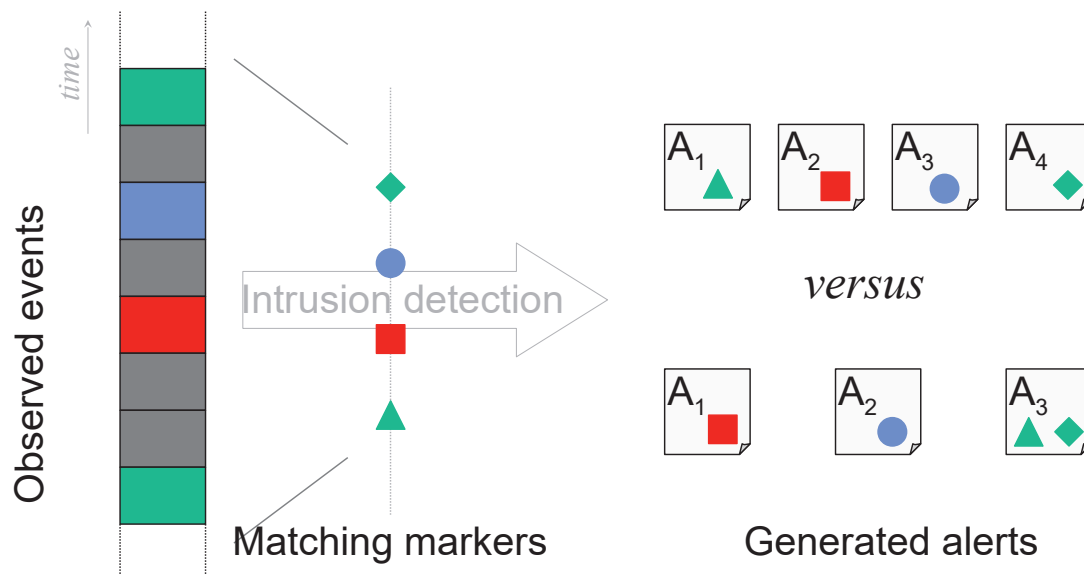
# Current trends

- A single technique per tool, usually
- Signatures-based techniques domine
  - Simpler implementation
  - Performances
- Behavioral approaches are seldomly used in commercial tools
- Reactive functions appear

301



# Multi-event analysis



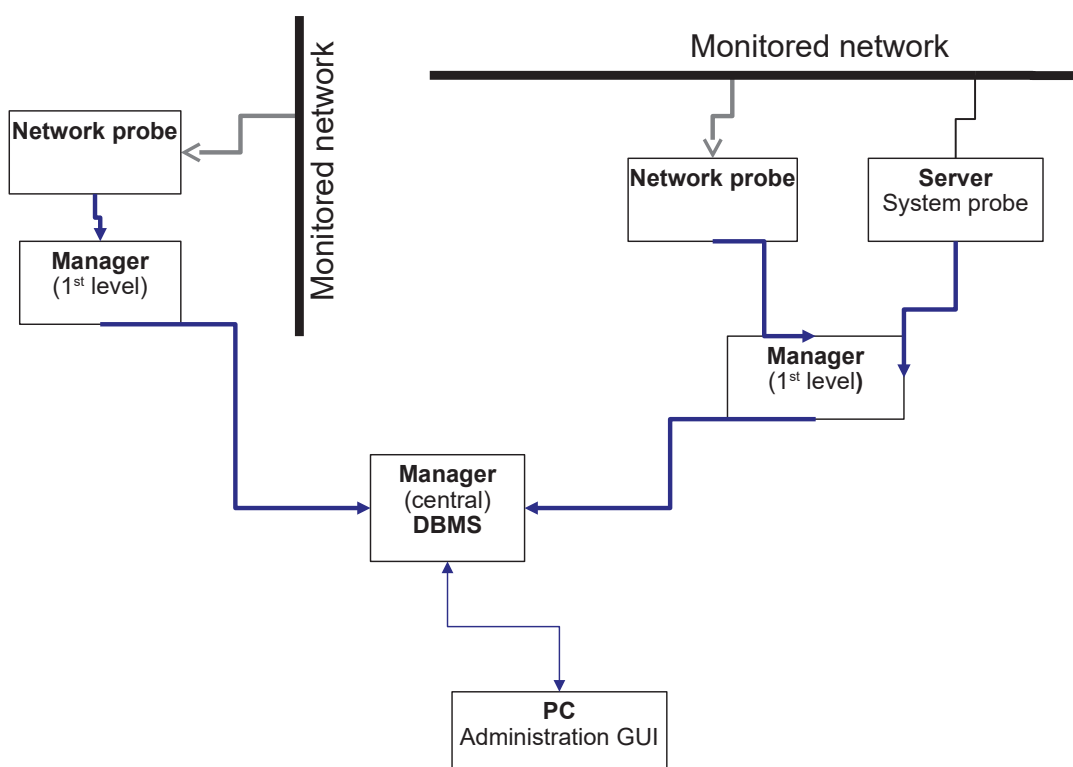
302

## Implementation considerations

- Probes
  - (Network) Monitoring
    - Situation choice
    - Issues with switched Ethernet (*mirroring* vs. *taps*)
  - System probes
  - Signature number (and CPU usage)
  - Signature accuracy and relevance
- Alerts management
  - Collectors
  - Secure exchange protocol
  - IDMEF exchange format (RFC 4765 plus 4766 & 4767)

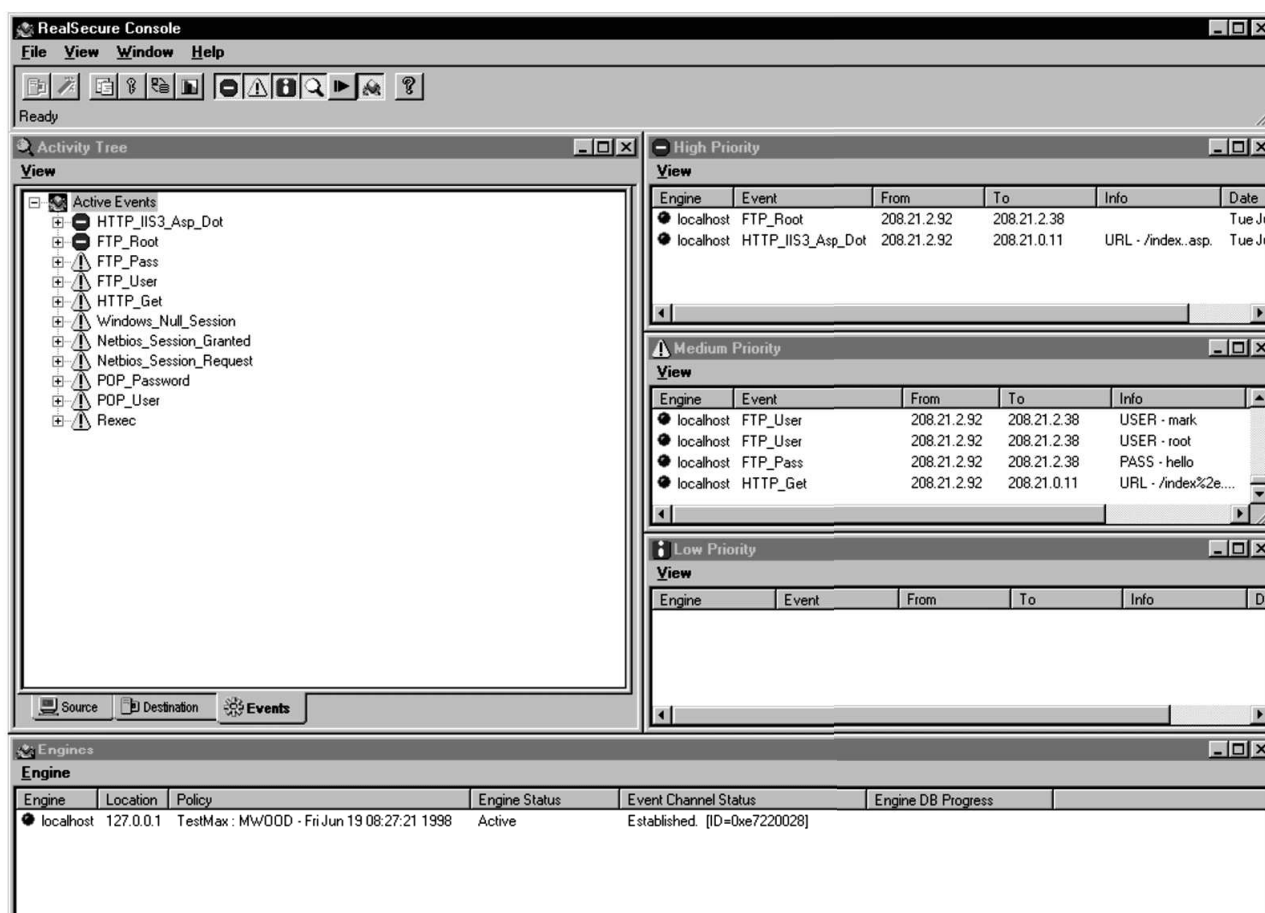
303

# Possible architecture



304

Ex. : ISS RealSecure GUI



305

# Signatures – Snort (1)

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SID</b>                  | 1800                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Message</b>              | VIRUS Klez Incoming                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Signature</b>            | alert tcp \$EXTERNAL_NET any -> \$SMTP_SERVERS 25 (msg:"VIRUS Klez Incoming"; flow.to_server,established; dsize:>120; content:"MIME"; content:"VGhpcyBwcm9"; classtype:misc-activity; sid:1800; rev:3;)                                                                                                                                                                                                                                                              |
| <b>Summary</b>              | This event is generated when an incoming email containing the Klez worm is detected.                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Impact</b>               | System compromise and further infection of target hosts.                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Detailed Information</b> | <p>W32/Klez.h@MM exploits the vulnerability in Microsoft Internet Explorer (ver 5.01 or 5.5 without SP2), enabling it to execute email attachments.</p> <p>Once executed, it can unload several processes including Anti-virus programs.</p> <p>The worm is able to propagate over the network by copying itself to network shares (assuming sufficient permissions exist). Target filenames are chosen randomly, and can have single or double file extensions.</p> |
| <b>Affected Systems</b>     | Microsoft Internet Explorer (ver 5.01 or 5.5 without SP2)                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Attack Scenarios</b>     | This virus can be considered a blended threat. It mass-mails itself to email addresses found on the local system, then exploits a known vulnerability, spreads via network shares, infects executables on the local system.                                                                                                                                                                                                                                          |
| <b>Ease of Attack</b>       | Simple. This is worm activity.                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>False Positives</b>      | Certain binary file email attachments can trigger this alert.                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>False Negatives</b>      | None known.                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Corrective Action</b>    | <p>Apply the appropriate vendor supplied patches.</p> <p>Block incoming attachments with .bat, .exe, .pif, and .scr extensions</p>                                                                                                                                                                                                                                                                                                                                   |
| <b>Contributors</b>         | <p>Sourcefire Research Team</p> <p>Brian Caswell &lt;bmc@sourcefire.com&gt;</p>                                                                                                                                                                                                                                                                                                                                                                                      |

306

# Signatures – Snort (2)

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SID</b>                  | 2251                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Message</b>              | NETBIOS DCERPC Remote Activation bind attempt                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Signature</b>            | alert tcp \$EXTERNAL_NET any -> \$HOME_NET 135 (msg:"NETBIOS DCERPC Remote Activation bind attempt"; content:" 05 "; distance:0; within:1; content:" 0b "; distance:1; within:1; byte_test:1,&1,0,relative; content:" B8 4A 9F 4D 1C 7D CF 11 86 1E 00 20 AF 6E 7C 57 "; distance:29; within:16; reference:cve,CAN-2003-0352; classtype:attempted-admin; reference:url,www.microsoft.com/technet/security/bulletin/MS03-026.asp; reference:cve,CAN-2003-0715; sid:2251; rev:1;)                                                                                                                                                                                                                                                       |
| <b>Summary</b>              | This event is generated when an attempt is made to exploit a known vulnerability in Microsoft RPCSS service for RPC.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Impact</b>               | Denial of Service. Possible execution of arbitrary code leading to unauthorized remote administrative access.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Detailed Information</b> | <p>A vulnerability exists in Microsoft RPCSS Service that handles RPC DCOM requests such that execution of arbitrary code or a Denial of Service condition can be issued against a host by sending malformed data via RPC.</p> <p>The Distributed Component Object Model (DCOM) handles DCOM requests sent by clients to a server using RPC. A malformed request to the host running the RPCSS service may result in a buffer overflow condition that will present the attacker with the opportunity to execute arbitrary code with the privileges of the local system account. Alternatively the attacker could also cause the RPC service to stop answering RPC requests and thus cause a Denial of Service condition to occur.</p> |
| <b>Affected Systems</b>     | <p>Windows NT 4.0 Workstation and Server</p> <p>Windows NT 4.0 Terminal Server Edition</p> <p>Windows 2000</p> <p>Windows XP</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

307

Prelude IDS Web Front-End - Filter builder [guest] - Microsoft Internet Explorer

Fichier Edition Affichage Favoris Outils ? Adresse http://tspsecur/piwi/Filters.pl?priv\_name=&trigger=&pa

| Alert List     | HeartBeat           | Top 20 Attackers | Top 20 Attacks | Statistics  |
|----------------|---------------------|------------------|----------------|-------------|
| Filter Factory | Edit current filter | None             |                | Load filter |

**Severity filter**  
☒ high  
☒ medium  
☒ low

**Sort by**  
☒ timestamp  
☐ group by key

**Results per page**

**Group by**  

Classification  
Source address  
Target address  
Target port

**Order**  
☒ Desc.  
☐ Asc.

**Since**

submit

<-- re-open sensor\_tree

23 results for those filters. Page 4/4.

First Prev Last

| P | Id   | Classification                                  | Impact | Completion | Source         | Destination            | Class            | Timestamp           |
|---|------|-------------------------------------------------|--------|------------|----------------|------------------------|------------------|---------------------|
|   | 1161 | SIMPLE Windows Event ID [560]: security FAILURE | user   | failed     | unknown        | 50.128.146.178         | Prelude LML/HIDS | 2003-10-31 16:46:50 |
|   | 1160 | SIMPLE Windows Event ID [560]: security FAILURE | user   | failed     | unknown        | 50.128.146.178         | Prelude LML/HIDS | 2003-10-31 16:45:59 |
|   | 1159 | SSH Remote user logging                         | user   | succeeded  | 50.128.146.178 | 127.0.0.1 22/tcp (ssh) | Prelude LML/HIDS | 2003-10-31 16:48:33 |
|   | 1158 | SSH Remote user logging                         | user   | succeeded  | 50.128.146.178 | 127.0.0.1 22/tcp (ssh) | Prelude LML/HIDS | 2003-10-31 16:40:24 |
|   | 1157 | Root login                                      | admin  | succeeded  | unknown        | 127.0.0.1              | Prelude LML/HIDS | 2003-10-31 16:35:27 |

Intranet local

308

ISAE – 2022/2023

## Intrusion detection shortcomings (currently)

- Low detection rate
  - False negative alerts
- Too many alerts
  - False alerts : False positive
  - Several thousand alerts per week (busy site)
- Insuficient alert semantic
  - No global view
  - Detection of a distributed attack is very hard
- It is difficult to detect unknown attacks
  - This is an advantage of behavior-based methods

309

Exemple : alertes générées par Dragon

# Too many details

[\*\*] [1:1256:2] WEB-IIS CodeRed v2 root.exe access [\*\*]  
 07/20-13:59:32.291193 64.165.187.170:4515 -> 193.54.194.111:80  
 [\*\*] [1:1002:2] WEB-IIS cmd.exe access [\*\*]  
 07/20-13:59:33.059882 64.165.187.170:4533 -> 193.54.194.111:80  
 [\*\*] [1:1002:2] WEB-IIS cmd.exe access [\*\*]  
 07/20-13:59:33.576217 64.165.187.170:4566 -> 193.54.194.111:80

|        |                  |                                                                                                                                                                                                                                                                               |
|--------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 07/20- | <b>SID</b>       | 1256                                                                                                                                                                                                                                                                          |
| 07/20- | <b>Message</b>   | WEB-IIS CodeRed v2 root.exe access                                                                                                                                                                                                                                            |
| 07/20- | <b>Signature</b> | alert tcp \$EXTERNAL_NET any -> \$HTTP_SERVERS \$HTTP_PORTS (msg:"WEB-IIS CodeRed v2 root.exe access"; flow:to_server,established; uricontent:"/root.exe"; nocase; classtype:web-application-attack; reference:url,www.cert.org/advisories/CA-2001-19.html; sid:1256; rev:7;) |

07/20-13:59:34.817953 64.165.187.170:4593 -> 193.54.194.111:80  
 [\*\*] [1:1002:2] WEB-IIS cmd.exe access [\*\*]  
 07/20-13:59:35.219711 64.165.187.170:4601 -> 193.54.194.111:80  
 [\*\*] [1:1002:2] WEB-IIS cmd.exe access [\*\*]  
 07/20-13:59:35.607048 64.165.187.170:4603 -> 193.54.194.111:80  
 [\*\*] [1:1002:2] WEB-IIS cmd.exe access [\*\*]  
 07/20-13:59:35.607048 64.165.187.170:4603 -> 193.54.194.111:80

|  |                  |                                                                                                                                                                                                       |
|--|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <b>SID</b>       | 1002                                                                                                                                                                                                  |
|  | <b>Message</b>   | WEB-IIS cmd.exe access                                                                                                                                                                                |
|  | <b>Signature</b> | alert tcp \$EXTERNAL_NET any -> \$HTTP_SERVERS \$HTTP_PORTS (msg:"WEB-IIS cmd.exe access"; flow:to_server,established; content:"cmd.exe"; nocase; classtype:web-application-attack; sid:1002; rev:5;) |

310

ISAE – 2022/2023

Exemple : alertes générées par Dragon

# Too many details

[\*\*] [1:1256:2] WEB-IIS CodeRed v2 root.exe access [\*\*]  
 07/20-13:59:32.291193 64.165.187.170:4515 -> 193.54.194.111:80  
 [\*\*] [1:1002:2] WEB-IIS cmd.exe access [\*\*]  
 07/20-13:59:33.059882 64.165.187.170:4533 -> 193.54.194.111:80  
 [\*\*] [1:1002:2] WEB-IIS cmd.exe access [\*\*]  
 07/20-13:59:33.576217 64.165.187.170:4566 -> 193.54.194.111:80  
 [\*\*] [1:1002:2] WEB-IIS cmd.exe access [\*\*]  
 07/20-13:59:33.999027 64.165.187.170:4582 -> 193.54.194.111:80

07/20-13:59 [\*\*] [1:1002:2] WEB-IIS cmd.exe access [\*\*]  
 07/20-13:59 [\*\*] [1:1002:2] WEB-IIS cmd.exe access [\*\*]  
 07/20-13:59 [\*\*] [1:1002:2] WEB-IIS cmd.exe access [\*\*]

**Nimda attack from 64.165.187.170  
towards 193.54.194.111**

[\*\*] [1:1002:2] WEB-IIS cmd.exe access [\*\*]  
 07/20-13:59:35.607048 64.165.187.170:4603 -> 193.54.194.111:80  
 [\*\*] [1:1002:2] WEB-IIS cmd.exe access [\*\*]  
 07/20-13:59:35.607048 64.165.187.170:4603 -> 193.54.194.111:80

311

Exemple : alertes générées par Dragon

## Poor semantics

07/20-13:59:32.291193 64.165.187.170:4515 -> 193.54.194.111:80  
 07/20-13:59:33.059882 64.165.187.170:4533 -> 193.54.194.111:80  
 07/20-13:59:33.576217 64.165.187.170:4566 -> 193.54.194.111:80  
 07/20-13:59:33.90027 64.165.187.170:4582 -> 193.54.194.111:80  
 07/20-13:59:35.607048 64.165.187.170:4603 -> 193.54.194.111:80

*Nimda* attack from 64.165.187.170  
 towards 193.54.194.111,  
 193.54.194.111 not vulnerable

312

ISAE – 2022/2023

## Alert correlation opportunities

- Correlation techniques
- Integration of system information
- Next step? : Grouping and alert fusion functions inside existing tools

313

## Overall presentation (2/2)

- Case studies
  - Wireless networks
  - New generation avionics systems
  - Network appliances
  - Mobile telephony
  - Gaming devices
- Wrap-up (on-demand)
  - IDS
  - **Firewalls**
  - Anti-virus

314

ISAE – 2022/2023

## Firewalls and Network protection

- Several design principles
  - (TCP,UDP) « state-based » firewalls
  - proxy firewalls
- Several security levels associated to DMZs
- Access control based on network flow characteristics
  - IP addresses : source, destination)
  - TCP/UDP : source port, destination port = protocol
  - action : drop, deny, allow, nat, trap, encrypt, ...

315



# How do you define a rule, in practice?

- Given an application
  - vlc (what's this?)
  - <http://maffreebox.freebox.fr/freeboxtv/playlist.m3u> (starting to understand)
- which « does not work », « Port number? »
- First steps

```
ortalo@hurricane:~$ ping -c 1 maffreebox.freebox.fr
PING freeplayer.freebox.fr (212.27.38.253) 56(84) bytes of data.
64 bytes from freeplayer.freebox.fr (212.27.38.253): icmp_seq=1 ttl=64
time=1.16 ms
--- freeplayer.freebox.fr ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.168/1.168/1.168/0.000 ms
ortalo@hurricane:~$ tethereal -i eth1 host 212.27.38.253
...nothing...
```

316

- Find (all) sources and destinations involved
  - IP<sub>eth1</sub> and 212.27.38.253 (hmm...)
- Experimental approach : monitor drops one after the other while checking the network traffic

```
DROPPED IN= OUT=eth1 SRC=81.56.84.23 DST=212.27.38.253 LEN=52 TOS=0x00
PREC=0x00 TTL=64 ID=48783 DF PROTO=TCP SPT=1047 DPT=80 SEQ=1610765695
ACK=0 WINDOW=5840 RES=0x00 SYN URGP=0 OPT (020405B40101040201030300)
DROPPED IN= OUT=eth1 SRC=81.56.84.23 DST=212.27.38.253 LEN=52 TOS=0x00
PREC=0x00 TTL=64 ID=48784 DF PROTO=TCP SPT=1047 DPT=80 SEQ=1610765695
ACK=0 WINDOW=5840 RES=0x00 SYN URGP=0 OPT (020405B40101040201030300)
DROPPED IN= OUT=eth1 SRC=81.56.84.23 DST=212.27.38.253 LEN=52 TOS=0x00
PREC=0x00 TTL=64 ID=1506 DF PROTO=TCP SPT=1048 DPT=80 SEQ=1611201085
ACK=0 WINDOW=5840 RES=0x00 SYN URGP=0 OPT (020405B40101040201030300)
```

317

- Let's allow outbound HTTP

```
DROPPED IN= OUT=eth1 SRC=81.56.84.23 DST=212.27.38.253 LEN=52 TOS=0x00
PREC=0x00 TTL=64 ID=22928 DF PROTO=TCP SPT=1082 DPT=554 SEQ=2534727009
ACK=0 WINDOW=5840 RES=0x00 SYN URGP=0 OPT (020405B40101040201030300)
DROPPED IN= OUT=eth1 SRC=81.56.84.23 DST=212.27.38.253 LEN=52 TOS=0x00
PREC=0x00 TTL=64 ID=22929 DF PROTO=TCP SPT=1082 DPT=554 SEQ=2534727009
ACK=0 WINDOW=5840 RES=0x00 SYN URGP=0 OPT (020405B40101040201030300)
```

- and TCP/554 inbound (?)

```
DROPPED IN=eth1 OUT= MAC=00:50:bf:29:e7:88:00:07:cb:05:ec:fc:08:00
SRC=212.27.38.253 DST=81.56.84.23 LEN=1356 TOS=0x00 PREC=0xE0 TTL=57
ID=18727 DF PROTO=UDP SPT=32803 DPT=1044 LEN=1336
DROPPED IN=eth1 OUT= MAC=00:50:bf:29:e7:88:00:07:cb:05:ec:fc:08:00
SRC=212.27.38.253 DST=81.56.84.23 LEN=1356 TOS=0x00 PREC=0xE0 TTL=57
ID=18982 DF PROTO=UDP SPT=32803 DPT=1044 LEN=1336
```

- TV selection list available
- We allow UDP inbound (>1025)

```
hurricane:~# dmesg | grep 212
DROPPED IN= OUT=eth1 SRC=81.56.84.23 DST=212.27.38.253 LEN=80 TOS=0x00
PREC=0x00 TTL=64 ID=6 DF PROTO=UDP SPT=1065 DPT=32769 LEN=60
DROPPED IN= OUT=eth1 SRC=81.56.84.23 DST=212.27.38.253 LEN=44 TOS=0x00
PREC=0x00 TTL=64 ID=7 DF PROTO=UDP SPT=1065 DPT=32769 LEN=24
```

- The show begins...

318

- Channels keep on changing (!!?)

```
hurricane:~# dmesg | grep 212
DROPPED IN= OUT=eth1 SRC=81.56.84.23 DST=212.27.38.253 LEN=80 TOS=0x00
PREC=0x00 TTL=64 ID=6 DF PROTO=UDP SPT=1065 DPT=32769 LEN=60
DROPPED IN= OUT=eth1 SRC=81.56.84.23 DST=212.27.38.253 LEN=44 TOS=0x00
PREC=0x00 TTL=64 ID=7 DF PROTO=UDP SPT=1065 DPT=32769 LEN=24
```

- We allow outbound UDP on the port range 32000-33999

- « It works. »

```
hurricane:~# dmesg | grep 212
hurricane:~# iptraf
hurricane:~#
```

| Total           |       | Incoming          |                     | Outgoing |       |      |
|-----------------|-------|-------------------|---------------------|----------|-------|------|
| Packets         | Bytes | Packets           | Bytes               | Packets  | Bytes |      |
| Total:          | 32980 | 45043248          | 32918               | 45037593 | 62    | 5655 |
| IP:             | 32980 | 44581462          | 32918               | 44576675 | 62    | 4787 |
| TCP:            | 68    | 5064              | 41                  | 2968     | 27    | 2096 |
| UDP:            | 32911 | 44576302          | 32876               | 44573611 | 35    | 2691 |
| ICMP:           | 1     | 96                | 1                   | 96       | 0     | 0    |
| Other IP:       | 0     | 0                 | 0                   | 0        | 0     | 0    |
| Non-IP:         | 0     | 0                 | 0                   | 0        | 0     | 0    |
| Total rates:    |       | 2466.2 kbits/sec  | Broadcast packets:  |          | 0     |      |
|                 |       | 225.2 packets/sec | Broadcast bytes:    |          | 0     |      |
| Incoming rates: |       | 2466.0 kbits/sec  |                     |          |       |      |
|                 |       | 225.0 packets/sec |                     |          |       |      |
| Outgoing rates: |       | 0.2 kbits/sec     | IP checksum errors: |          | 0     |      |
|                 |       | 0.2 packets/sec   |                     |          |       |      |
| Elapsed time:   |       | 0:02              |                     |          |       |      |
| X-exit          |       |                   |                     |          |       |      |

- By the way... where is the documentation?

319

## One last note...

*« The final step (...) simply adds a second Trojan horse to the one that already exists. The second pattern is aimed at the C compiler. The replacement code is a (...) self-reproducing program that inserts both Trojan horses in the compiler. (...) First we compile the modified source with the normal C compiler to produce a bugged binary. We install this binary as the official C. We can now remove the bugs from the source of the compiler and the new binary will reinsert the bugs whenever it is compiled. Of course, the login command will remain bugged with no trace in source anywhere. »*

320

ISAE – 2022/2023

## Morale

*« You can't trust code that you did not totally create yourself.  
(Especially code from companies that employ people like [him].) »*

Ken Thomson, [Reflections on Trusting Trust](#), Turing award lecture, in *Communications of the ACM*, vol.27, no.8, pp.761-763, August 1984.