Objective Security Evaluation

Possibly Feasible, or Feasibly Possible?

Andrew Jamieson
V201605190741

Opinions are my own, and do not necessarily reflect any official stance from UL

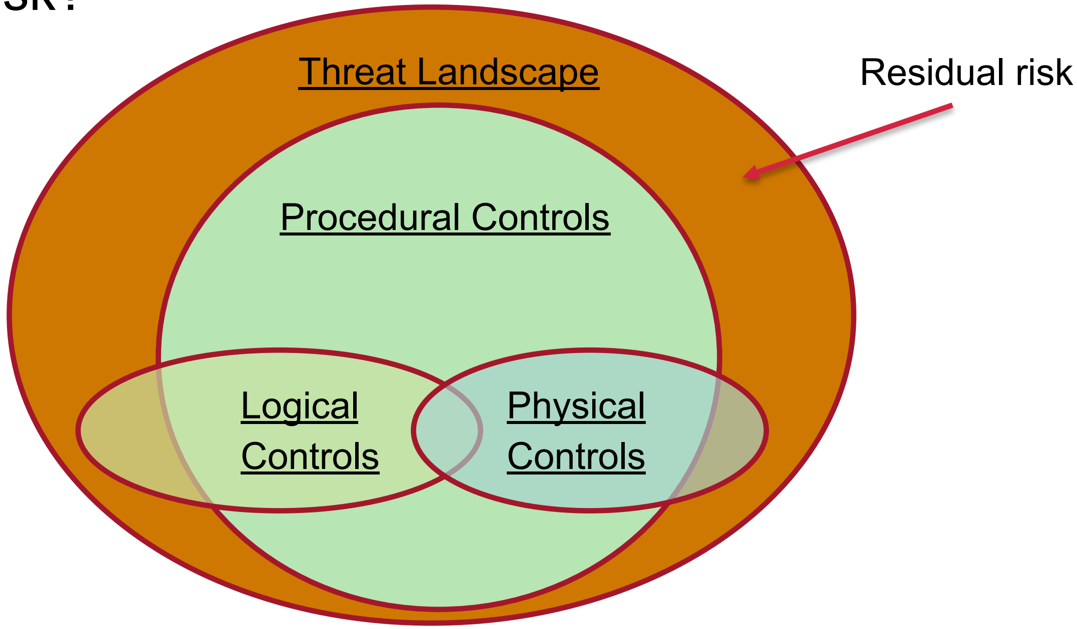
What is a ‘standard’?

“A required or agreed level of quality or attainment”

(Oxford Dictionary)

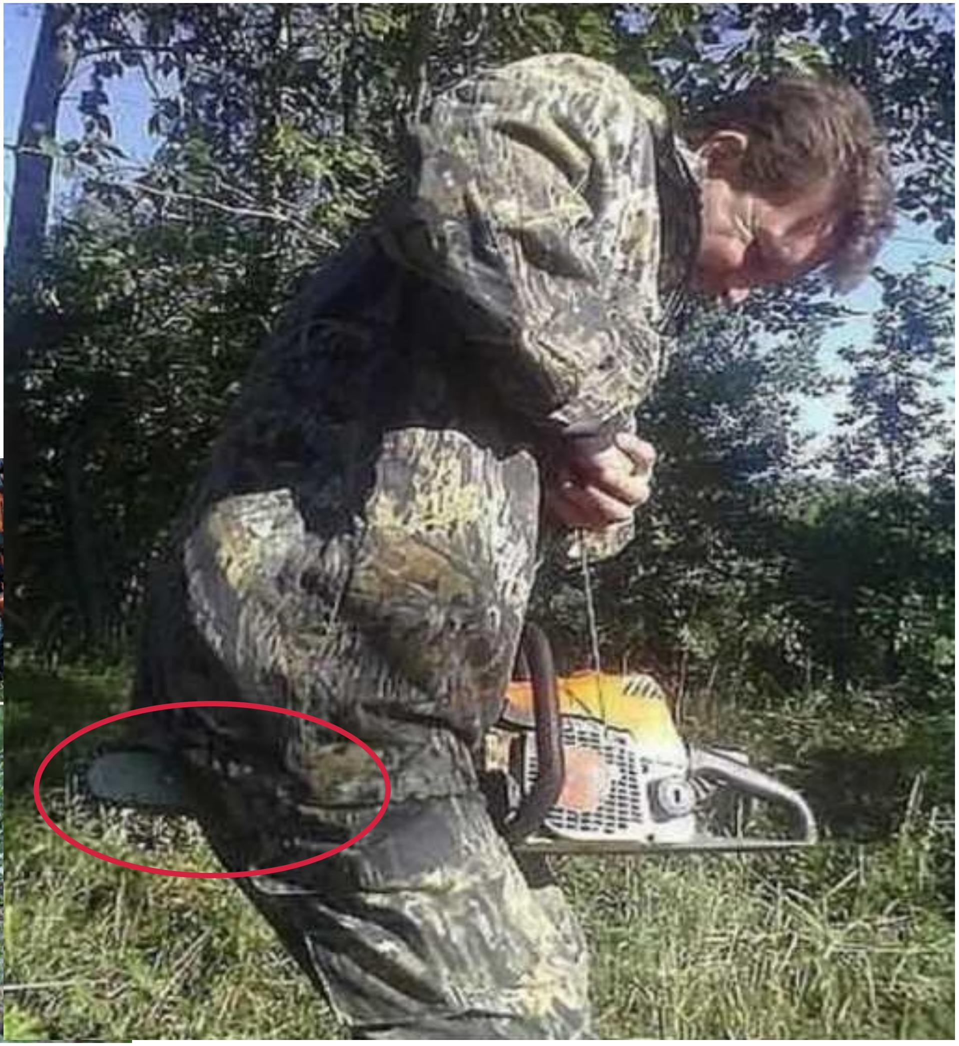
Agreeing on Security

- What do security standards try to 'attain'?
  - Security? Is anything ever really 'secure'? Is security binary?
- Security evaluations work within a 'threat landscape'
  - The goal of the evaluation is to assess the procedural, logical, and physical controls that minimise the remaining residual risk
  - What level of residual risk is acceptable?
  - How do we define risk?



Defining Risk

If left to the individual, risk is defined individually ...



How do security standards define risk?

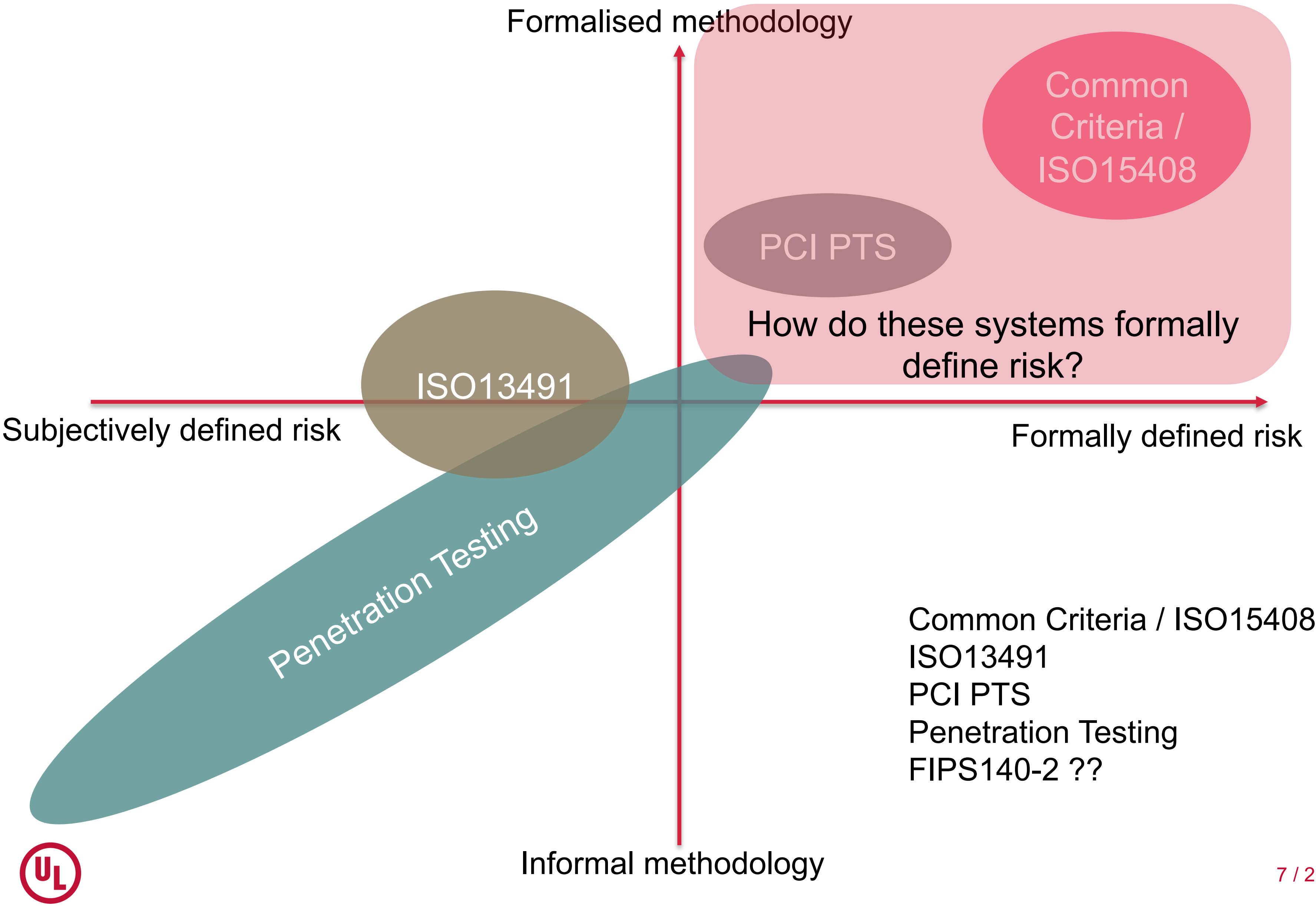
Formalising Risk

AS05.41: (Multiple-Chip Embedded – Level 4) The cryptographic module components shall be covered by potting material or contained within an enclosure encapsulated by a tamper detection envelope (e.g., a flexible mylar printed circuit with a serpentine geometric pattern of conductors or a wire-wound package or a non-flexible, brittle circuit or a strong enclosure) that shall detect tampering by means such as cutting, drilling, milling, grinding, or dissolving of the potting material or enclosure to an extent sufficient for accessing plaintext secret and private keys cryptographic keys or CSPs.

AS07.02: (Levels 1, 2, 3, and 4) Public keys shall be protected within the cryptographic module against unauthorized modification and substitution.

How to determine the feasibility of ‘shall’?
What risk are we trying to mitigate?
What methodology to use for ‘agreement’?

Defining Security Evaluation



How do these systems formally define risk?

Common Criteria / ISO15408
ISO13491
PCI PTS
Penetration Testing
FIPS140-2 ??

Factors	Identification		Exploitation	
Elapsed time	**CC**	**PCI**	**CC**	**PCI**
< one hour	0	0	0	0
< 8 hours	NA	2	NA	2
< one day (24h)	1	3	3	3
< one week (40h)	2	3.5	4	3.5
< 80 hours	NA	4	NA	4
< one 160h	3	5	6	5
> one 160h	5	5.5	8	5.5
Not practical	*	N/A	*	N/A
Expertise	**CC**	**PCI**	**CC**	**PCI**
Layman	0	0	0	0
Proficient	2	1.5	2	1.5
Expert	5	4	4	4
Multiple Expert	7	N/A	6	N/A
Knowledge of TOE	**CC**	**PCI**	**CC**	**PCI**
Public	0	0	0	0
Restricted	2	2	2	2
Sensitive	4	3	3	3
Critical	6	NA	5	NA
Very critical	9	NA	NA	NA

Factors	Identification		Exploitation	
Equipment	**CC**	**PCI**	**CC**	**PCI**
None	0	0	0	0
Standard	1	1	2	1
Specialised	3	3	4	3
Bespoke	5	5	6	5
Multi Bespoke (CC) / Chip level (PCI)	7	7	8	7
Open Samples	**CC**	**PCI**	**CC**	**PCI**
Public	0	NA	NA	NA
Restricted	2	NA	NA	NA
Sensitive	4	NA	NA	NA
Critical	6	NA	NA	NA
Access to TOE	**CC**	**PCI**	**CC**	**PCI**
< 10 samples	0	NA	0	NA
< 30 samples	1	NA	2	NA
< 100 samples	2	NA	4	NA
> 100 samples	3	NA	6	NA
Not practical	*	NA	*	NA

Formalising Risk

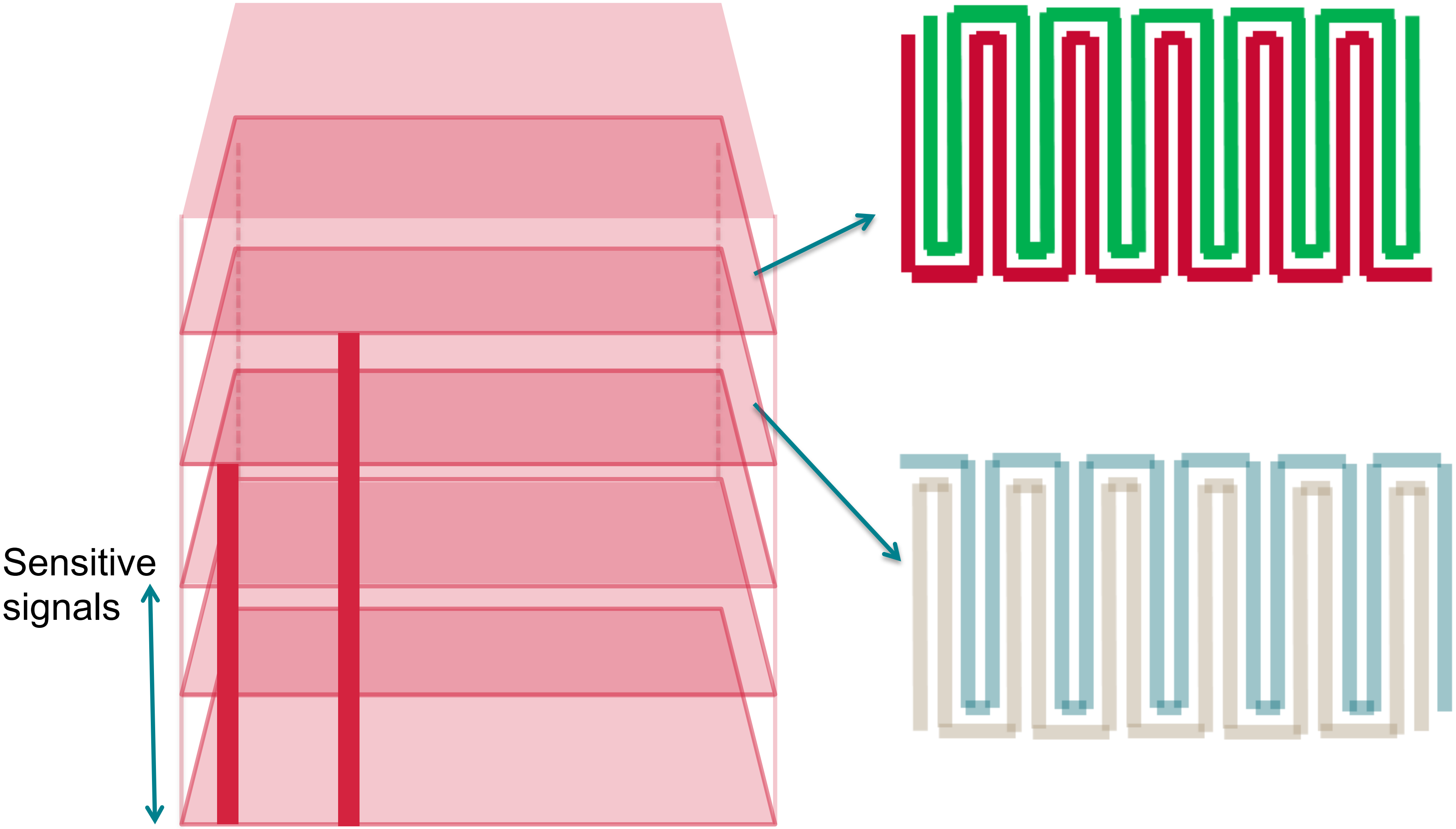
How long does an attack take?

How much skill?

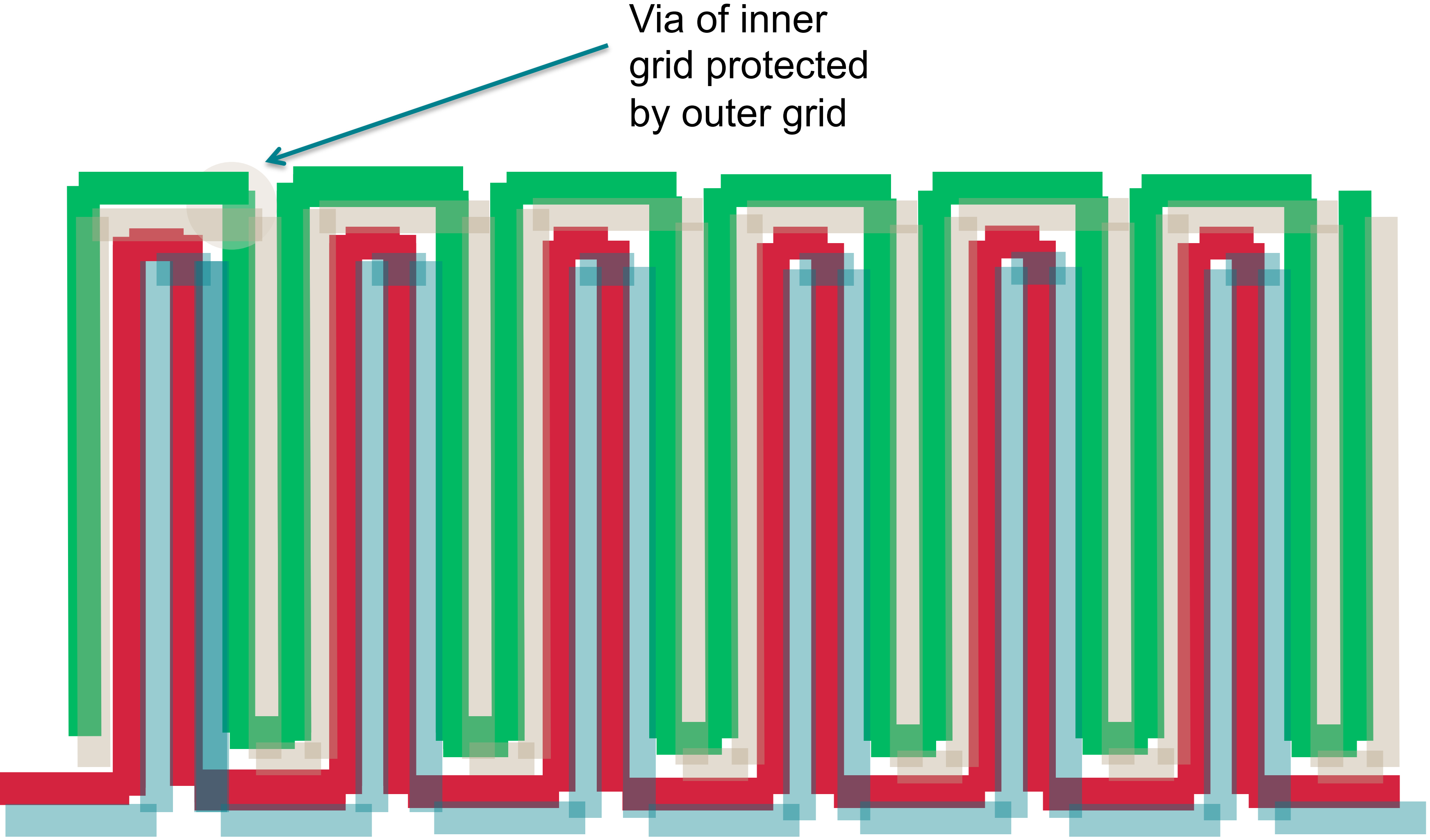
What type of equipment?

AS05.41: (Multiple-Chip Embedded – Level 4) The cryptographic module components shall be covered by potting material or contained within an enclosure encapsulated by a tamper detection envelope (e.g., a flexible mylar printed circuit with a serpentine geometric pattern of conductors or a wire-wound package or a non-flexible, brittle circuit or a strong enclosure) that shall detect tampering by means such as cutting, drilling, milling, grinding, or dissolving of the potting material or enclosure to an extent sufficient for accessing plaintext secret and private keys cryptographic keys or CSPs.

Formalising Risk



Formalising Risk


Via of inner grid protected by outer grid

Formalising Risk

Costing Example

Identification Phase	Value	
Attack Time	5.5	> One hundred and sixty hours
Expertise	4	Expert
Knowledge	0	Public
Access Costs	3	One mechanical and one functional sample without keys
Equipment required	0.5	Standard (shared with exploit)
Specific Parts	1	Standard
Identification Total	13.5	
Exploitation Phase	Value	
Attack time	3	≤ Twenty four hours
Expertise	4	Expert
Knowledge	0	Public
Access Costs	4	Functional sample with working keys and software
Equipment required	0.5	Standard
Specific Parts	1	Standard
Exploitation Total	12.5	
Grand Total	26	

Objective assessment of security is a delicate balance between subjectivity and granularity – management of this balance is the duty of the certification body.

At least we have methods for hardware – what about evaluation of software security?

Defining Software Security

- Security systems are increasingly reliant on software
  - ... and that software continues to become increasingly complex
  - Is this a bad thing?
- Security is not a measurable absolute
  - It’s both subjective and (non-linearly) mutable over time
  - New vulns introduce step-changes to the threat landscape
  - Often because they invalidate assumptions made
    - Power analysis, ROP, rowhammer, etc
    - As system complexity increases the scope for both ‘traditional’ vulnerabilties and ‘new’ types of vulnerabitilites increases exponentially

How do we measure software security objectively?

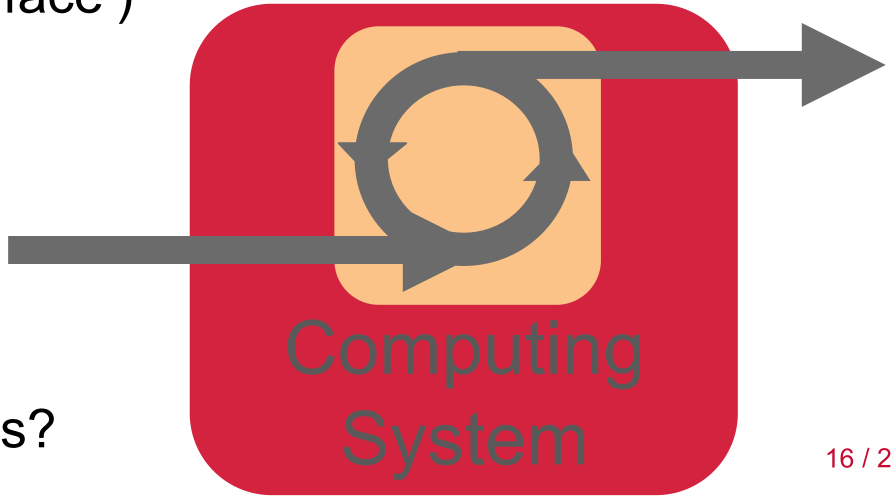
Defining Security

Keep it simple, stupid!

- Devices can be defined by three things
  - Interfaces (Input / Output)
  - Processing attack surface
  - System architecture

} ← Vulnerability Surface

- The more interfaces, and larger attack surface, the less secure a system can objectively be considered
- Specifics of the architecture either help or hinder security (reducing the ‘vulnerability surface’)
- Then we ‘just’ need to wrap metrics around this process!

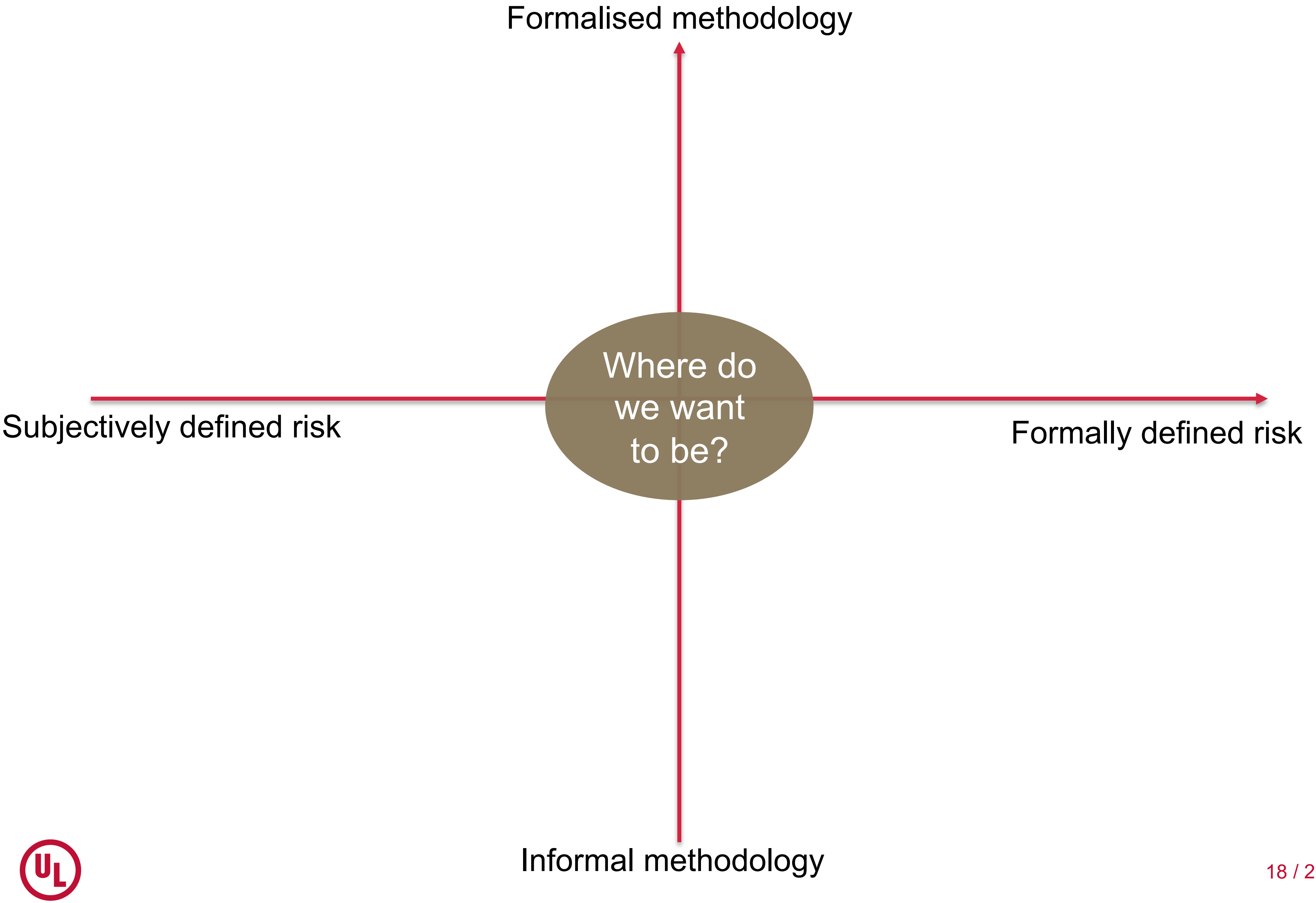


How do we define the metrics?

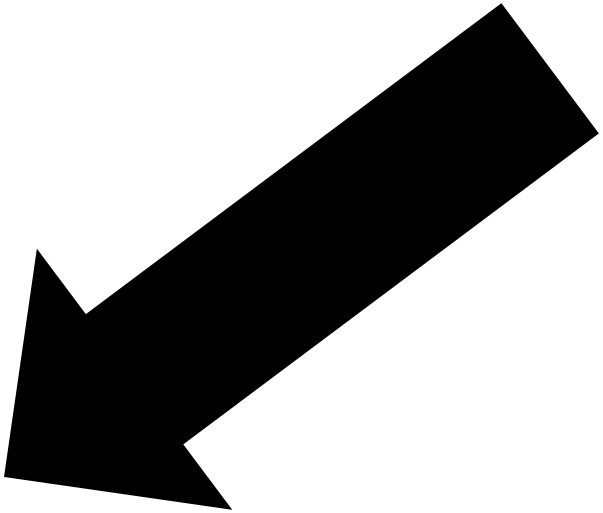
Metric – Logical Security Posture

- Based on a points system
  - Points are assigned for security features the system has
    - DEP, MAC, separate execution environments, etc
  - Points are deducted for increasing attack surface
    - Logical and physical interfaces, OS type / size, processor architecture
- Most computing vulnerabilities have similar root causes
  - Lack of randomness where needed
  - Default configurations / passwords / cryptographic keys
  - ‘Over privilaged’ (and vulnerable) code
  - Insecure updates and communication methods
  - Little to no logical protections – security is just not thought about

Defining Security Evaluation



You're with stupid!



Thank you