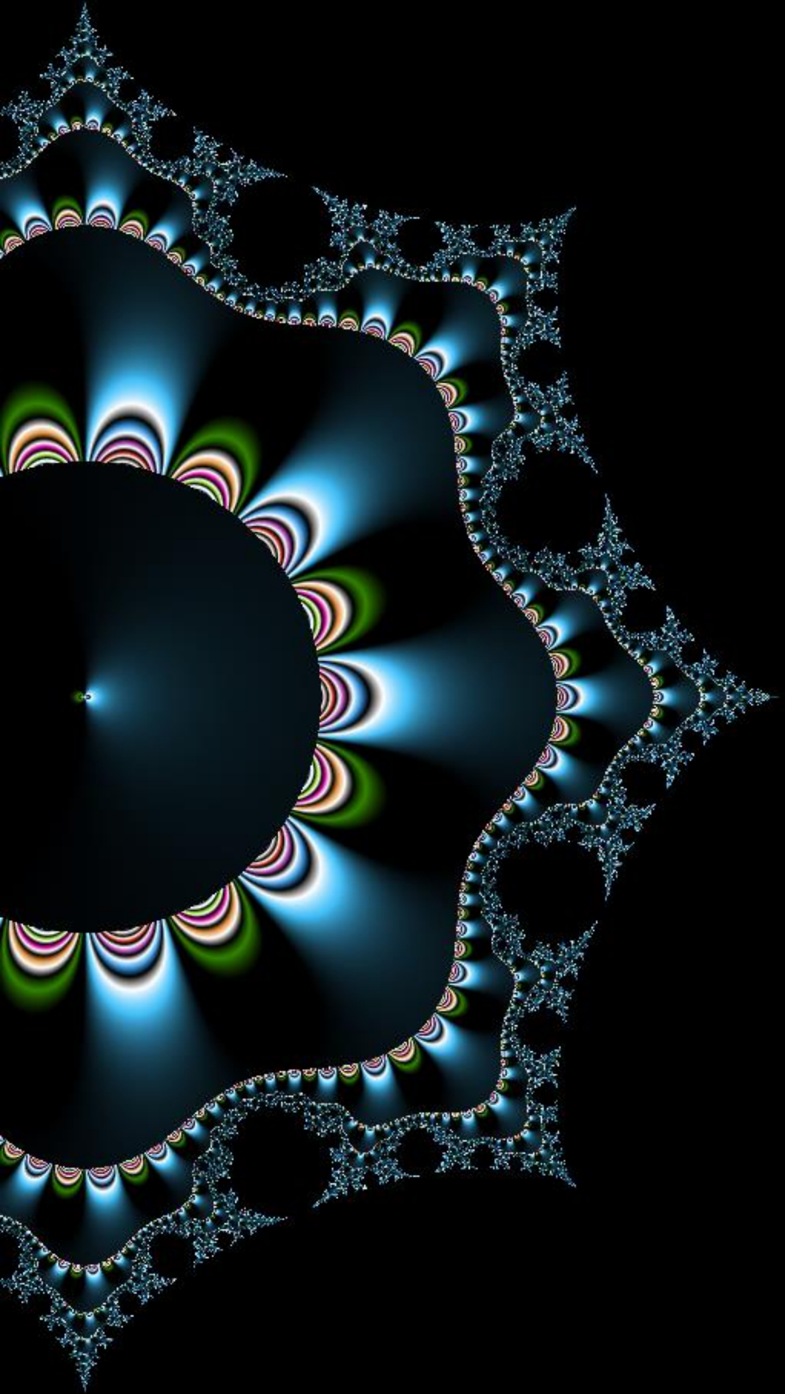


02 - Introduction

Security (of) softwares

Thibaut HENIN
www.arsouyes.org

What is Software Security ?

What is software Vulnerability ?

Software vulnerability
definition

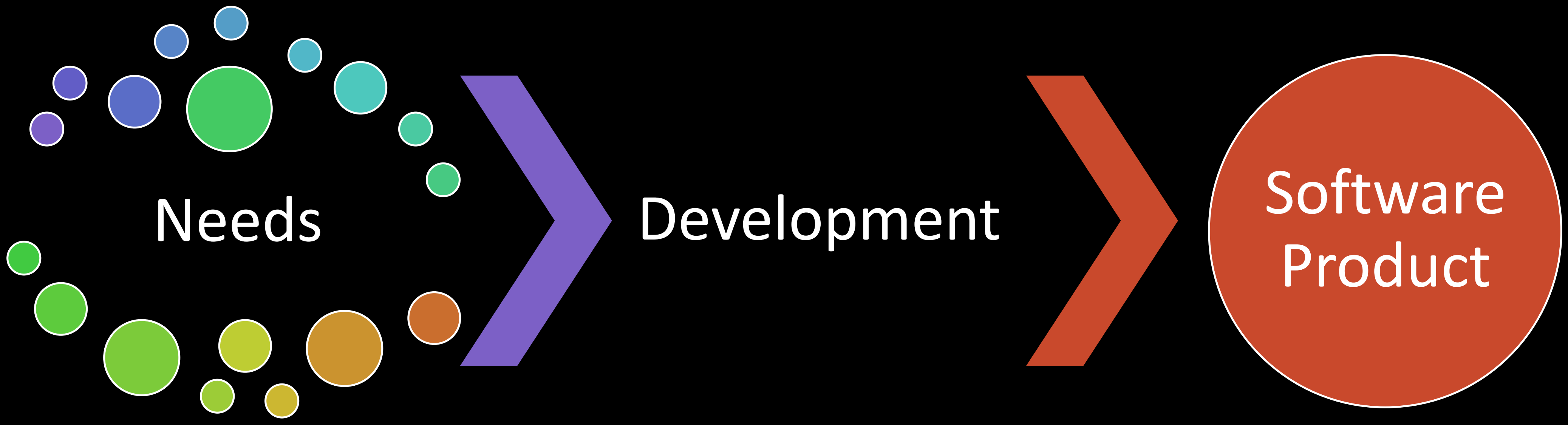
A defect

that allows

unauthorized actions

Software engineering

How we turn dreams into reality



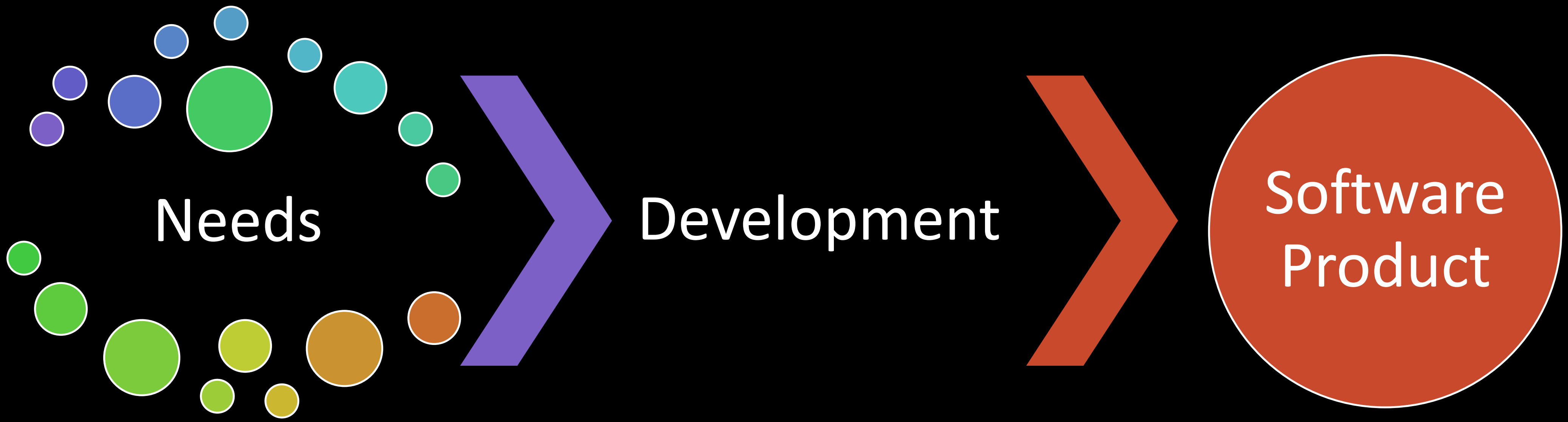
What we wants

What we get

Specifications

Bugs

Software Security
How we turn nightmares into reality



What we wants

What we get

Security Policy

Vulnerabilities

Two families

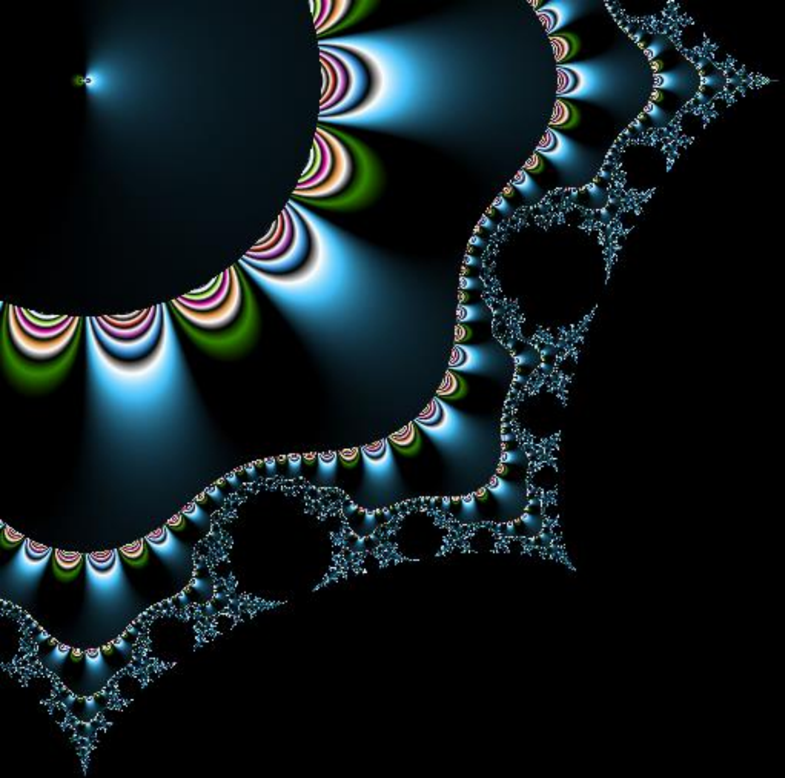
i.e. ISO-27000 : 2005 *versus* 2013

Check Lists

i.e. PCI-DSS, ISO-21434 (road vehicles), mehari,…

Risk Analysis

i.e. Common Criteria (ISO-15408), CSPN, EBIOS,…



CSPN

Short Introduction

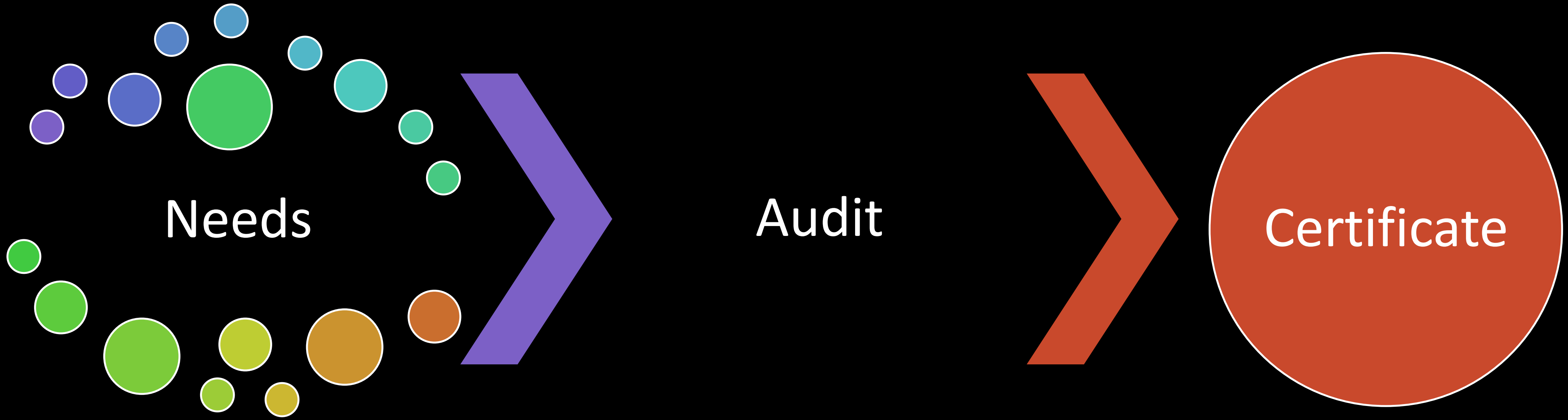
Simplification of Common Criteria

i.e. ISO-27000 in 2005 or 2013

	Common Criteria	CSPN
Perimeter	Everything involved	Software only
Workload	No limit	30 days of work
Recognition	World Wide	Made in France

CSPN

Two phases



What we wants

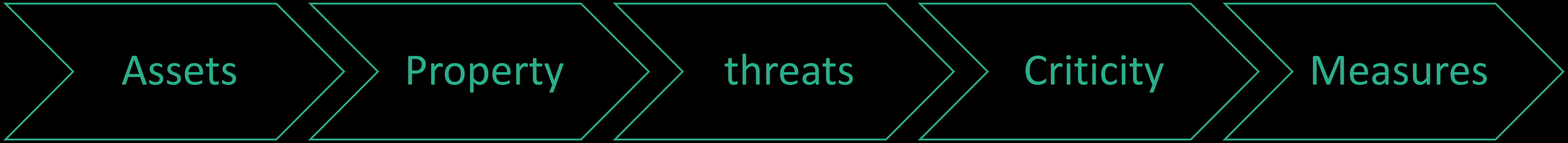
What we get

Security Target

Security Evaluation

Security Target

(from risk management)



Step 0 – the product

Who it is

Identification

(name, version, editor, …)

Description

(features / use cases, users, prerequisites, …)

Step 0 – the product

Example

```
Organisation :    Speed-e-dev
Product :         Speed-e-blog
Version number :  2.0
Category :        miscelaneous
```

Step 1 - Assets
Definition

A resource

(information, data, hardware, functionnality, …)

That need to be protected

(against malicious agent)

Step 1 - Assets
Example

Business assets	Support assets
A1 - Articles	*A4 - Passwords*
A2 - Nicknames	*A5 - Files – configuration*
A3 - Web browsers	*A6 – Files – source code*
	A7 - Servers

Step 2 - Security Properties

Three main ones

Confidentiality

(only authorized agend can read)

Integrity

(only authorized agent can write)

Availability

(asset can be accessed)

Step 2 - Security Properties

Other usefull ones

Authenticity

(the resource is the one that have been sent)

Traceability

(access are recorded on a log)

Non repudiation

(nobody can say « it’s not me » or « it’s someone else »)

Step 2 - Coverage matrix

Assets and properties

Assets	Confidentiality	Availability	Integrity
A1 - Articles			✓
A2 - Nicknames			✓
A3 - Web browsers	✓		✓
A4 - Passwords	✓		✓
A5 - Files - configuration	✓		✓
A6 - Files – source code			✓
A7 - Servers	✓		✓

Step 3 – Threats

Definition

Feared event

(what wrong can happen)

Step 3 – Threats
Example

T1 – Fraudulent modification of article

T2 – Execution on browser

T3 – Fraudulent deletion of article

T4 – Impersonation of writers

T5 – Password theft

T6 – Theft of account

T7 – Fraudulent access to files

T8 – Fraudulent modification of files

T9 – Execution on server

Step 3 – Coverage matrix

Assets by threats

Threats	A1 Articles	A2 Nicknames	A3 Browsers		A4 Passwords		A5 Files Config		A6 Files Source code	A7 Servers	
	I	I	C	I	C	I	C	I	I	C	I
T1 - Modification article	✓	✓									
T2 - Execution, browser	✓	✓	✓	✓							
T3 - Deletion article	✓	✓				✓					
T4 - Impersonation	✓	✓									
T5 - Password theft	✓	✓			✓						
T6 - Account theft	✓	✓				✓					
T7 - File access					✓		✓			✓	
T8 - File changes	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
T9 - Execution, server	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Step 4 – Criticity (optionnal)

Product of two parameters

Severity - Consequences on the asset

e.g. if articles are defaced, the branding of the editor is hurt

Probability - Ease of the threat

e.g. access to writers’ password database

Step 4 – Criticity

Visually

For sure	4	8	12	16
Probable	3	6	9	12
May occurs	2	4	6	8
Not expected	1	2	3	4
	no effect	It hurts	Low damage	High damage

Step 4 – Criticity (simplification)

For software (since we use booleans)

	no effect	Some effects
Possible	0	1
Impossible	0	0

Step 5 - Measures

a.k.a. security function / security features

Things to mitigate the risks

Eg. Access control, backups, updates, training, monitoring, …

Step 5 - Coverage matrix

Threats by measures

Mesure	Article modification	Password theft	Execution on server
Authentication & access control	✓		
Secure storage of password		✓	
Input data filtering			✓

Step 5b - Residual risks

Value after measure take effects

Mesure	New Probability	New Severity	New Risk
Article modification → Access control	1 → 0	1	1 → 0
Password theft → Secure storage	1	1 → 0	1 → 0
Execution on server → Input filtering	1 → 0	1	1 → 0

Security Target

Definition

Document that tells :

« This is how this software claims to be secure »

(all previous content)

Security Policy
Definition

Document that tells :

« How we claims to be secure »

(same but for everything beyond software)

Security Audit
Definition

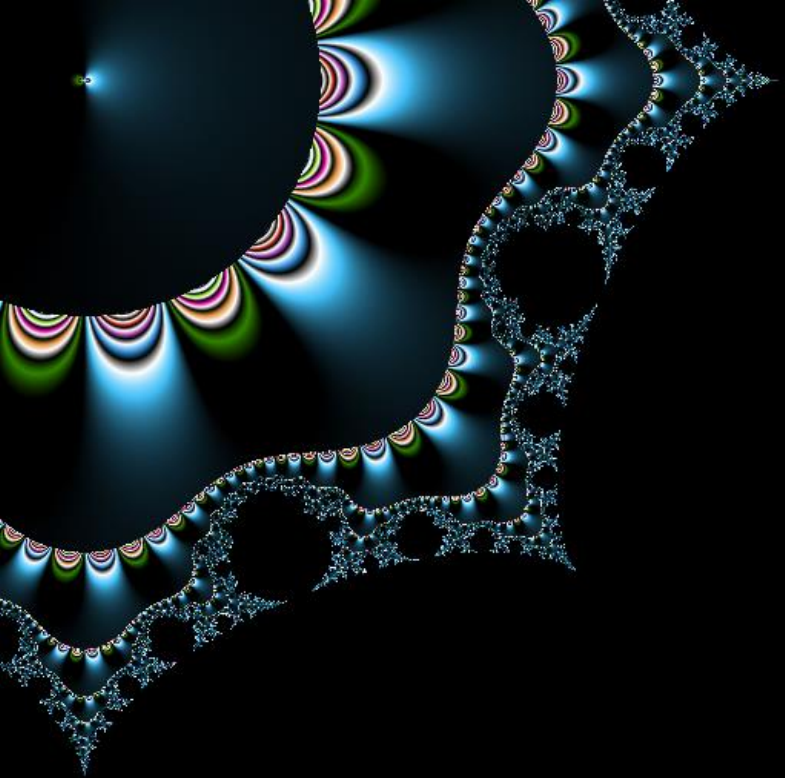
Procedure that check :

« The claims are effectives »

So what is a software vulnerability ?

Vulnerability

Bypass of Security Policy



Birth of vulnerabilities

Why are they so common ?

By Negligence

« Don’t touch what works »

By Conservatism

« We've always done that way! »

Technical debt

« It takes to much time to do it properly »

« We’ll fix it later »

By Incompetence

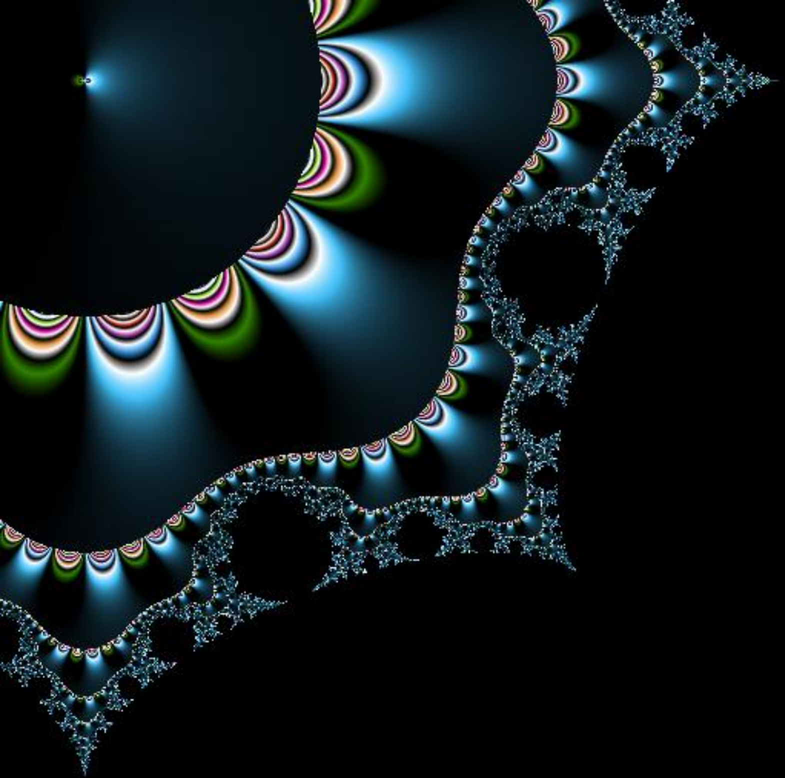
« I didn’t know »

Out of laziness

« It’s too booooooring »

The error is human

« Oups, I didn’t notice »



Discoveries

By whom ? Why and How ?

It’s a job

Rather twice than once

Security Audit

Planned for certification

Bug Hunting

Opportunistic discoveries (unplanned)

Selling your time

By editor, user or agency

Selling vulnerabilities

To editors (bounties), agencies or mafia

No Disclosure

Keep it secret

Avoid wild exploitation

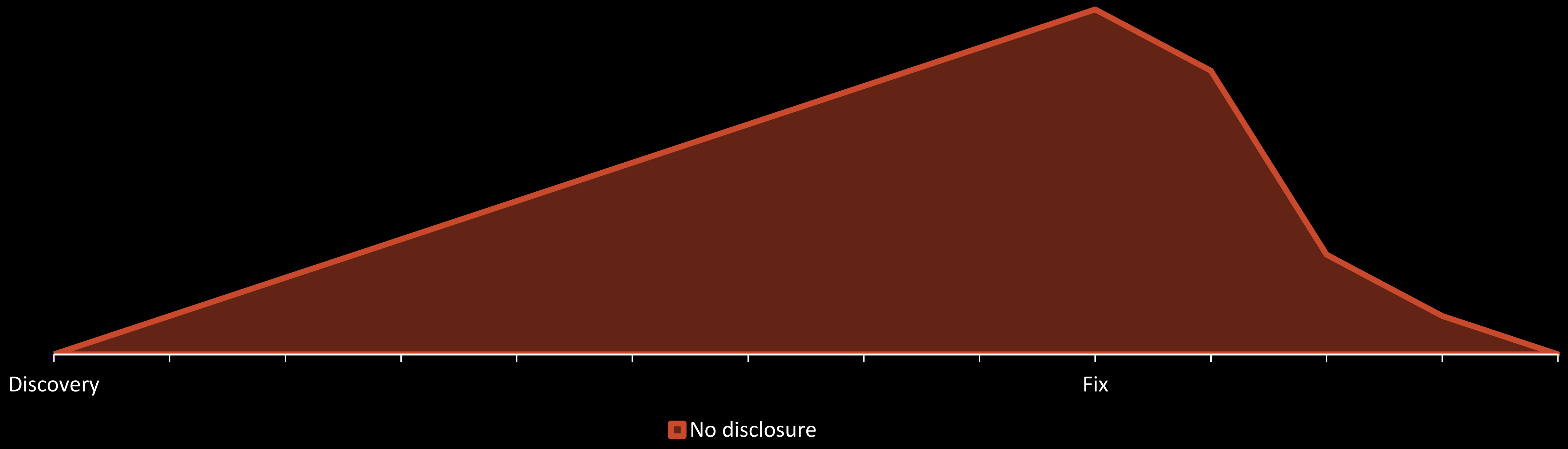
(minimize damages)

Sell exploits

(no fix means high exploit value)

Life of a vulnerability threat

No Disclosure



Full Disclosure

Publish everything

Force editors to fix

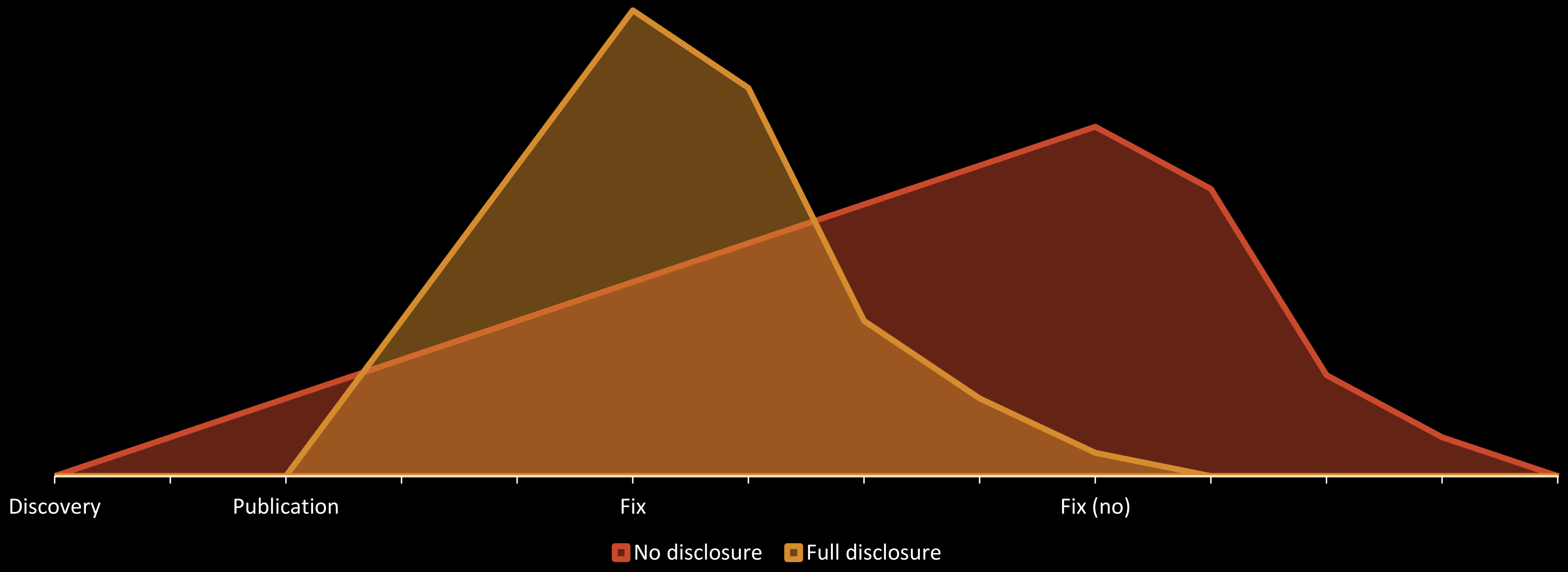
(avoid further exploitation)

Be credited

(and becomes famous)

Life of a vulnerability threat

Full Disclosure



Responsible Disclosure

Brave new world

Tell editor

(negociate a delay)

Then publish

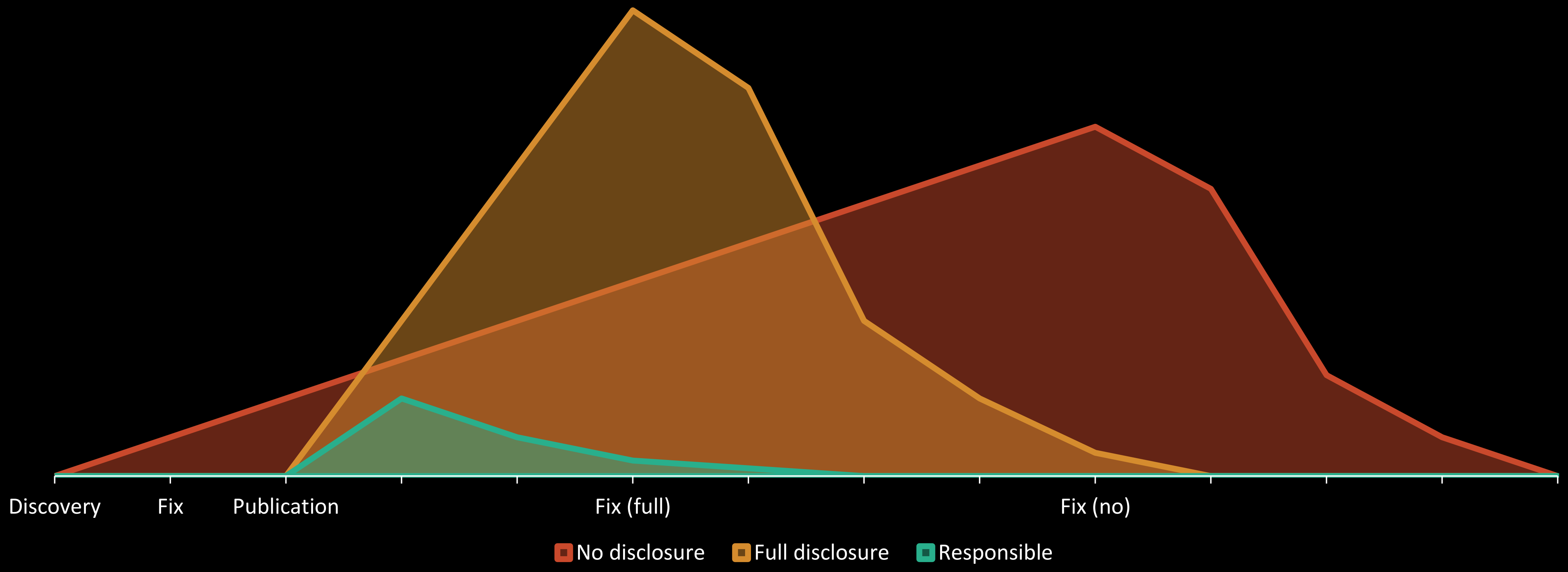
(and becomes famous)

Get a bounty

(and becomes rich)

Life of a vulnerability threat

Responsible disclosure



World Wide Vulnerability Database

CVE

(Common Vulnerability and Exposure)

Unique Identifier

(CVE-AAAA-NNNN)

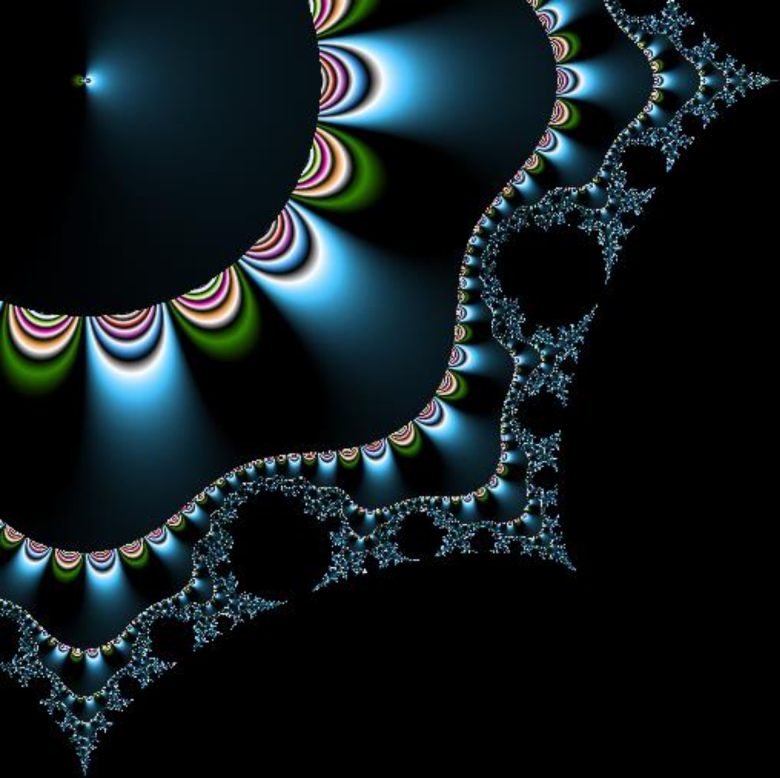
Edited by MITRE

`https://www.cve.org/`

Common Vulnerability Scoring System

Score /10

	Base score	Options
Likelihood	Exploitation Vector, Complexity, Authentication	Temporal Exploit, fix, confidence
Criticity	Impact Confidentiality, Integrity, Availability	Environment Use’s context



How this is handled ?

The real world is full of human beings

Full of Politics

It’s not a vulnerability (it’s a feature)

When auditing : Scope

Marketing vs reality

When bug bounting : Scope and Score

Marketing and Budget vs bounties

Which side of the force ?

Asymetrical confrontation

	Blue team (defense)	Red team (attacks)
Defeat	Bad consequences	-
Victory	-	Positive consequences

Optimistic Ostrich

It works, everything’s good

We’ll see later

Nobody wants to attack us

Paranoid perfectionist

Everything must be perfect

A vulnerability is a proof of incompetence

There always remain a risk

Constructive humility

Where are the weaknesses ?

How can we fix them ?

Continuous improvement

« It’s not perfect but we work toward »