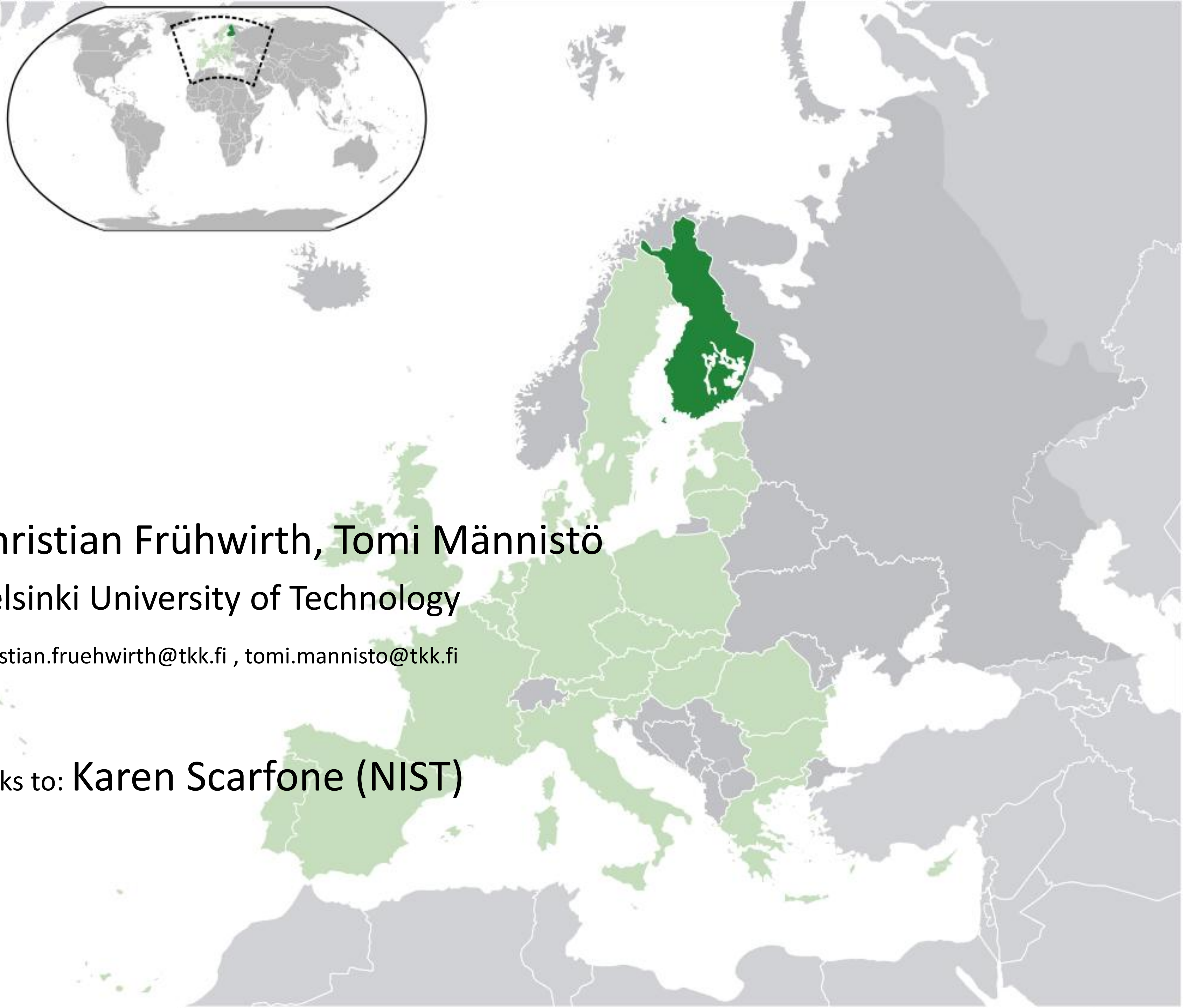


Authors: Christian Frühwirth, Tomi Männistö

Helsinki University of Technology

christian.fruehwirth@tkk.fi , tomi.mannisto@tkk.fi

Special thanks to: Karen Scarfone (NIST)

HELSINKI UNIVERSITY OF TECHNOLOGY

Improving CVSS-based vulnerability prioritization and response with context information

Presented at MetriSec09, Orlando FL

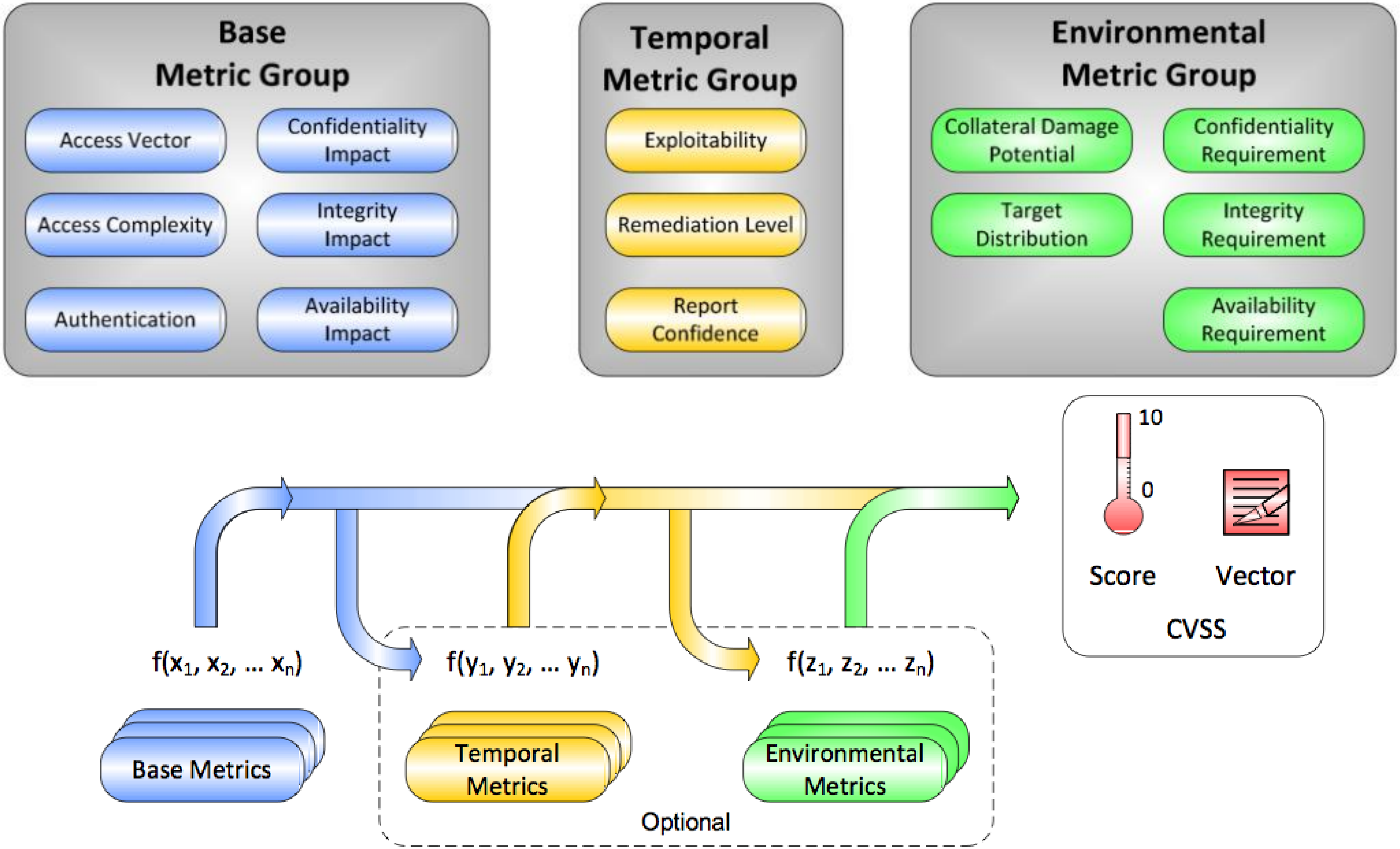
What is CVSS?

- The “Common vulnerability scoring system”
- A severity metric for security vulnerabilities in software products
- A widely used, de-facto standard. (e.g. at NVD)

CVSS

- Assigns vulnerabilities a score of 0-10 (10 = most critical)
- Scores are based on collections of metrics e.g. the vuln. exploitability, impact on information confidentiality, etc.
- The CVSS metrics are divided in 3 groups: *Base*, *Temporal* and *Environmental*.

The common vulnerability scoring system



Base Metric Group

- Access Vector
- Access Complexity
- Authentication
- Confidentiality Impact
- Integrity Impact
- Availability Impact

Temporal Metric Group

- Exploitability
- Remediation Level
- Report Confidence

Environmental Metric Group

- Collateral Damage Potential
- Target Distribution
- Confidentiality Requirement
- Integrity Requirement
- Availability Requirement

$f(x_1, x_2, \ldots x_n)$
Base Metrics
$f(y_1, y_2, \ldots y_n)$
Temporal Metrics
$f(z_1, z_2, \ldots z_n)$
Environmental Metrics
Optional
10
0
Score
Vector
CVSS

CVSS usage in the industry

CVSS scores are often used to ***prioritize*** vulnerability responses

- Apply bug fixes
- Roll out patches
- Build workarounds
- …

CVSS usage in the industry

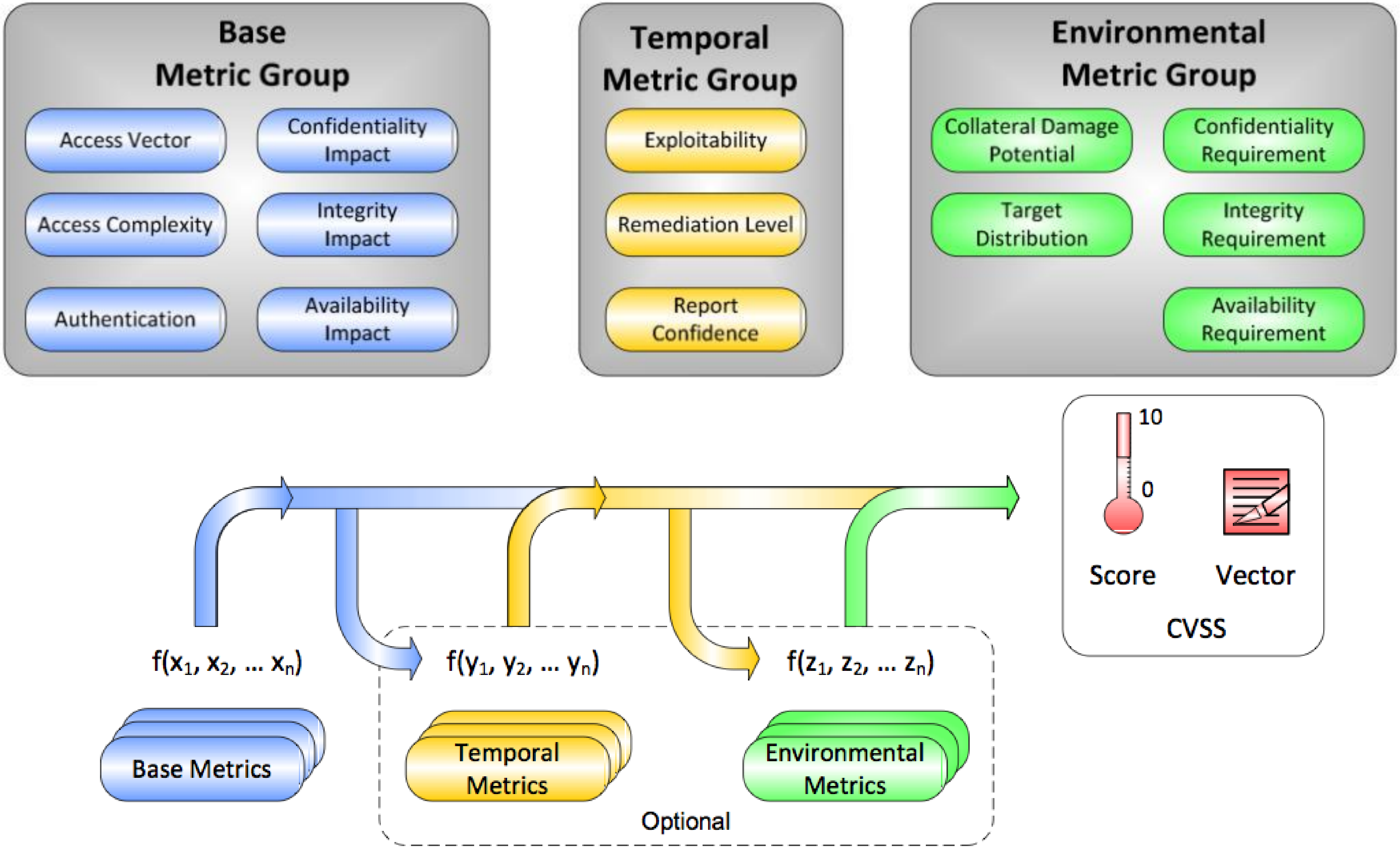
Problem: Many use CVSS information (e.g. provided by the NVD) “as-is”

→Leaving out temporal metrics (e.g. Exploitability)

→Leaving out environmental (context) metrics: Security requirements

→**CVSS is not used to its full potential**

The common vulnerability scoring system



Base Metric Group

Access Vector

Confidentiality Impact

Access Complexity

Integrity Impact

Authentication

Availability Impact

Temporal Metric Group

Exploitability

Remediation Level

Report Confidence

Environmental Metric Group

Collateral Damage Potential

Confidentiality Requirement

Target Distribution

Integrity Requirement

Availability Requirement

10

0

Score

Vector

CVSS

$f(x_1, x_2, \ldots x_n)$

$f(y_1, y_2, \ldots y_n)$

$f(z_1, z_2, \ldots z_n)$

Base Metrics

Temporal Metrics

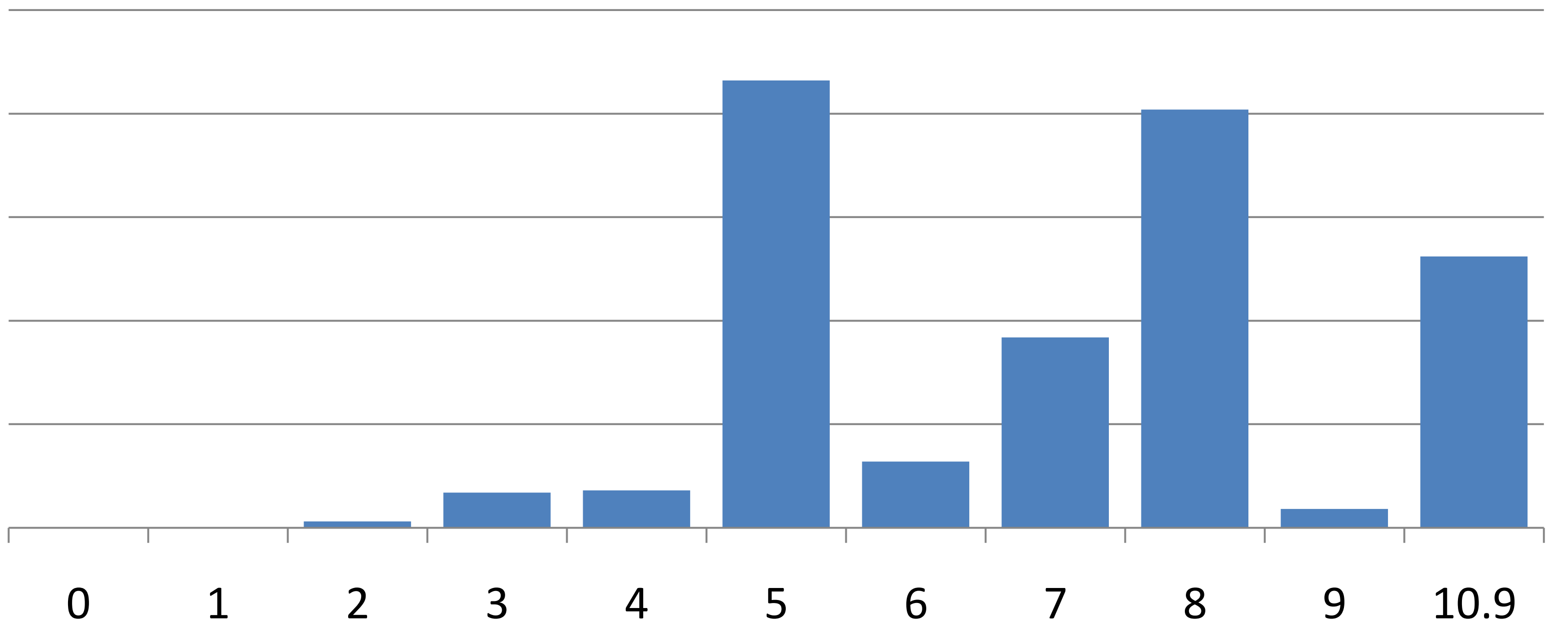
Environmental Metrics

Optional

No context info in CVSS

Using only the base metric group results in too many vulnerabilities with the same scores

Nr. of vulnerabilities in 3 months of VND records with a score of:



Example: NVD entry “CVE-2009- 0609”

Denial-of-service (DoS) vulnerability in the Sun Java System Directory Server.

Base score of **7.8** points. (Categorization: High)

If a company, has a high *requirement for availability* and *exploits for the vulnerability are already available*, the score changes to:

Score of **10** points. (Categorization: Critical)

Different scores, so what?

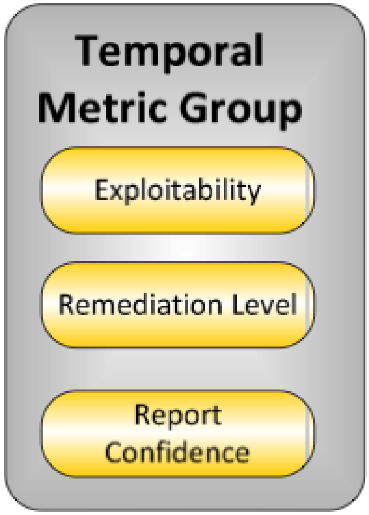
Companies use scores to *categorize* and *prioritize* vulnerability response processes

Execution costs of vuln. response processes can vary:

- Critical vulnerabilities need faster response times
- May require unscheduled reboots that affect productivity.
- Indirect costs when patches with potential side effects on other systems have to be rolled-out without prior testing.
- Lower priority response processes can be executed during regularly scheduled system maintenance windows.

Invest in gathering context information

Using CVSS built in context metrics can improve overall vulnerability prioritization, response and save costs.



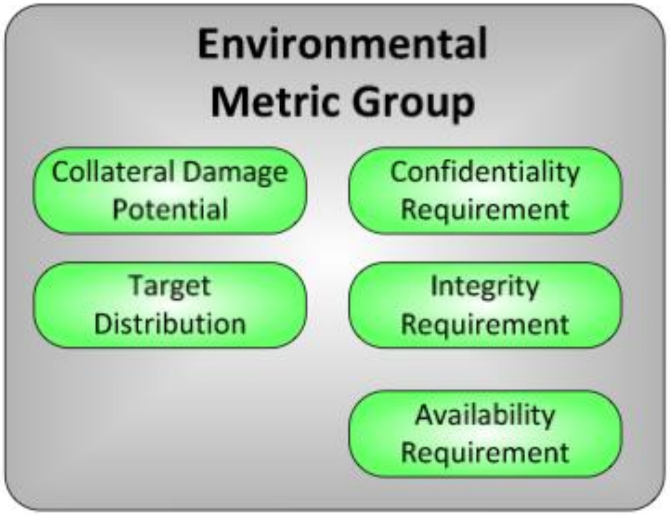
Temporal Metric Group

Exploitability

Remediation Level

Report Confidence

Problem: “Gathering context info is expensive, how can we estimate whether it will be worth it?”



Environmental Metric Group

Collateral Damage Potential

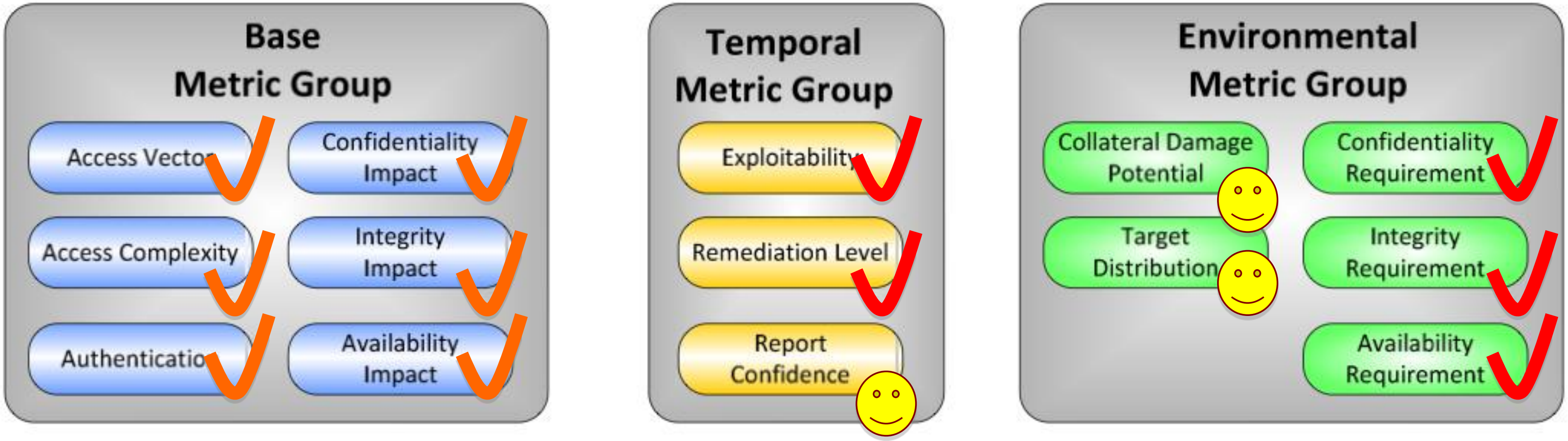
Confidentiality Requirement

Target Distribution

Integrity Requirement

Availability Requirement

A little experiment with available and artificial data



Base Metric Group

- Access Vector ✓
- Confidentiality Impact ✓
- Access Complexity ✓
- Integrity Impact ✓
- Authentication ✓
- Availability Impact ✓

Temporal Metric Group

- Exploitability ✓
- Remediation Level ✓
- Report Confidence ☺

Environmental Metric Group

- Collateral Damage Potential ☺
- Confidentiality Requirement ✓
- Target Distribution ☺
- Integrity Requirement ✓
- Availability Requirement ✓

How can we estimate whether it will be worth it?

Use available data:

Step 1: Assign a cost factor to the execution of each category of vulnerability response processes (low, med, high, critical)

Step 2: Gather publicly available vulnerability data (e.g. NVD)

Add artificially created data:

Step 3: Estimate the likelihood of patch and exploit availability based on models developed in the literature

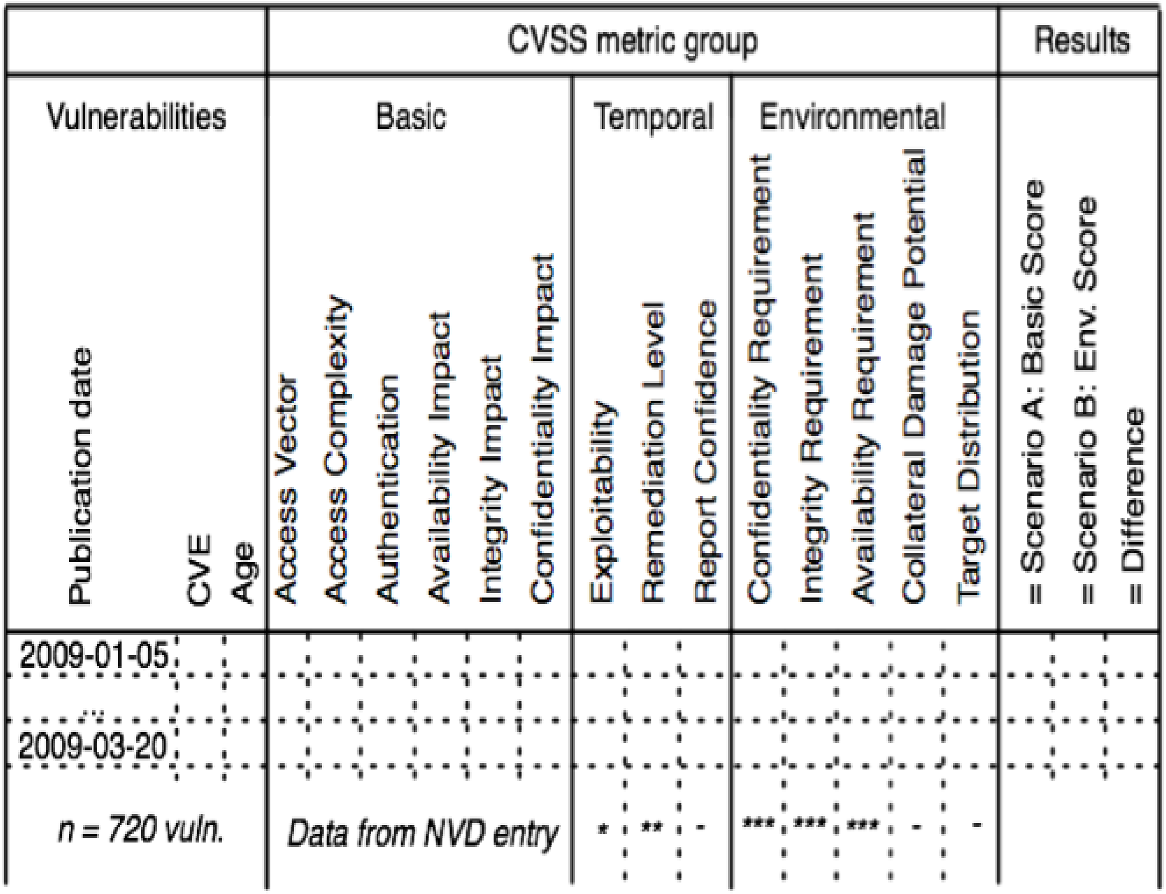
Step 4: Elicit high level security requirements in the organization. Use them to determine the likelihood of high, med, low requirements for individual systems.

Step 5: Calculate new scores and categorize vuln. accordingly

Step 6: Calculate anticipated costs for vuln. response processes (using 2 scenarios)

Step 7: Compare costs between scenarios

In practice:
Compare scores in 2 scenarios



			CVSS metric group														Results		
Vulnerabilities			Basic						Temporal			Environmental							
Publication date	CVE	Age	Access Vector	Access Complexity	Authentication	Availability Impact	Integrity Impact	Confidentiality Impact	Exploitability	Remediation Level	Report Confidence	Confidentiality Requirement	Integrity Requirement	Availability Requirement	Collateral Damage Potential	Target Distribution	= Scenario A: Basic Score	= Scenario B: Env. Score	= Difference
2009-01-05																			
...																			
2009-03-20																			
n = 720 vuln.			*Data from NVD entry*						*	**	-	***	***	***	-	-			

* ... *IF [Pareto(age,alpha,k) > Rand()]*
THEN "HIGH", ELSE "UNPROVEN"

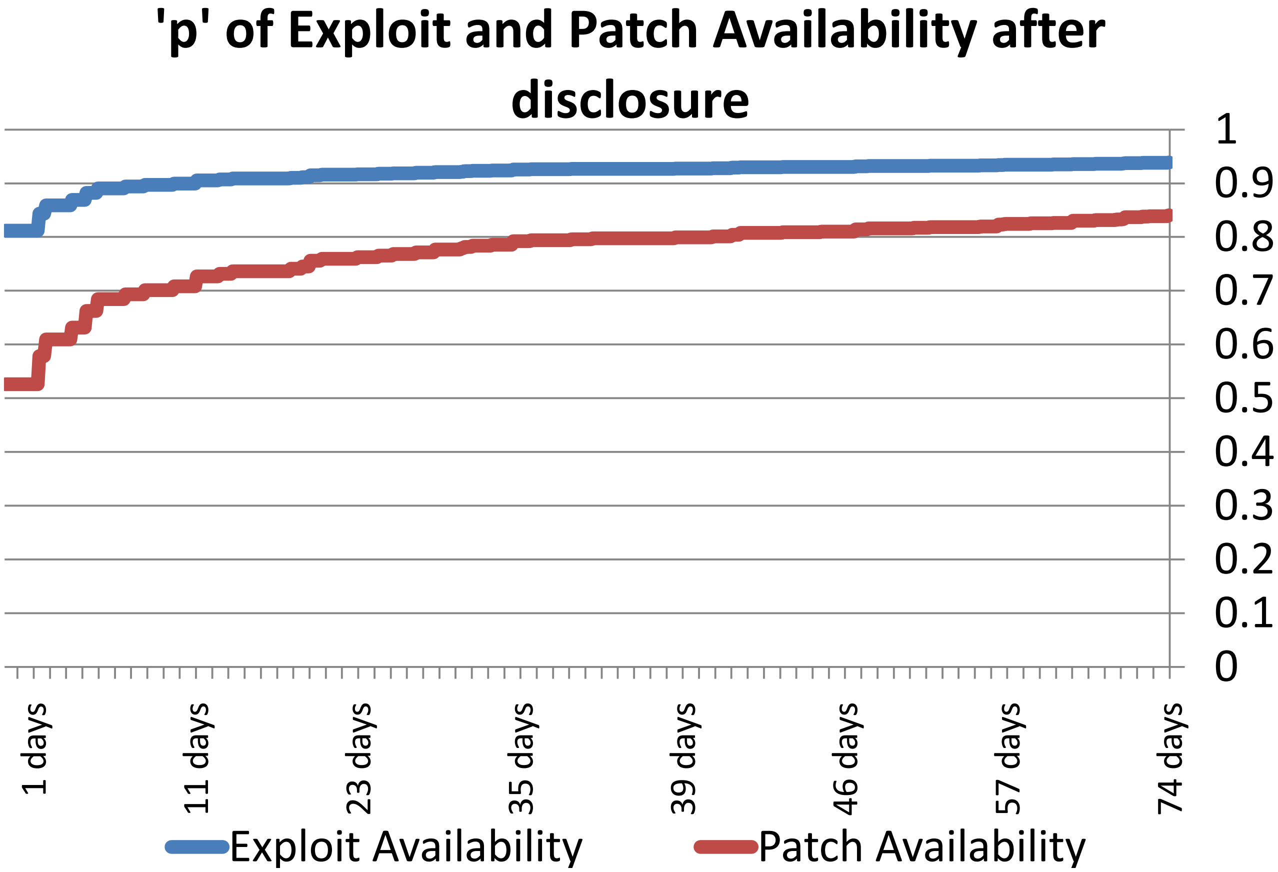
** ... *IF [Weibull(age,lambda,k) > Rand()]*
THEN "OFFICIAL-FIX", ELSE "UNAVAILABLE"

***... *IF [IntervieweePercentage > Rand()]*
THEN "HIGH", ELSE "LOW"

- ... *Left in default state*

n=720

Estimating temporal metrics with distribution model



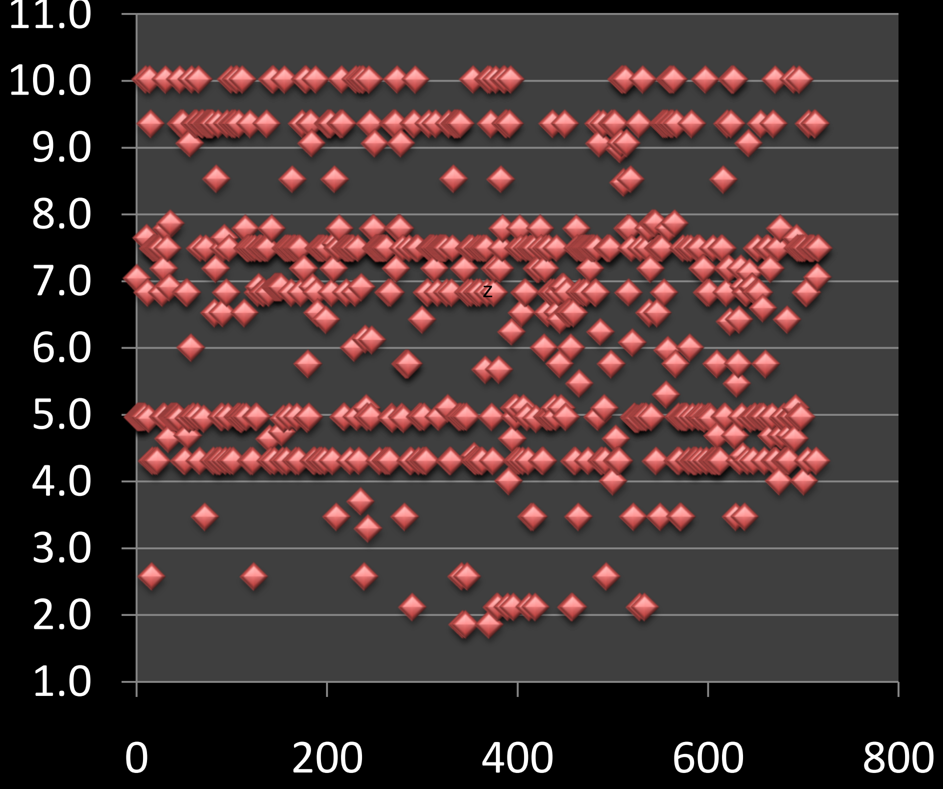
n=720

Based on: S. Frei, M. May, U. Fiedler, and B. Plattner, “Large- scale vulnerability analysis,” Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense, Pisa, Italy: ACM, 2006, pp. 131-138.

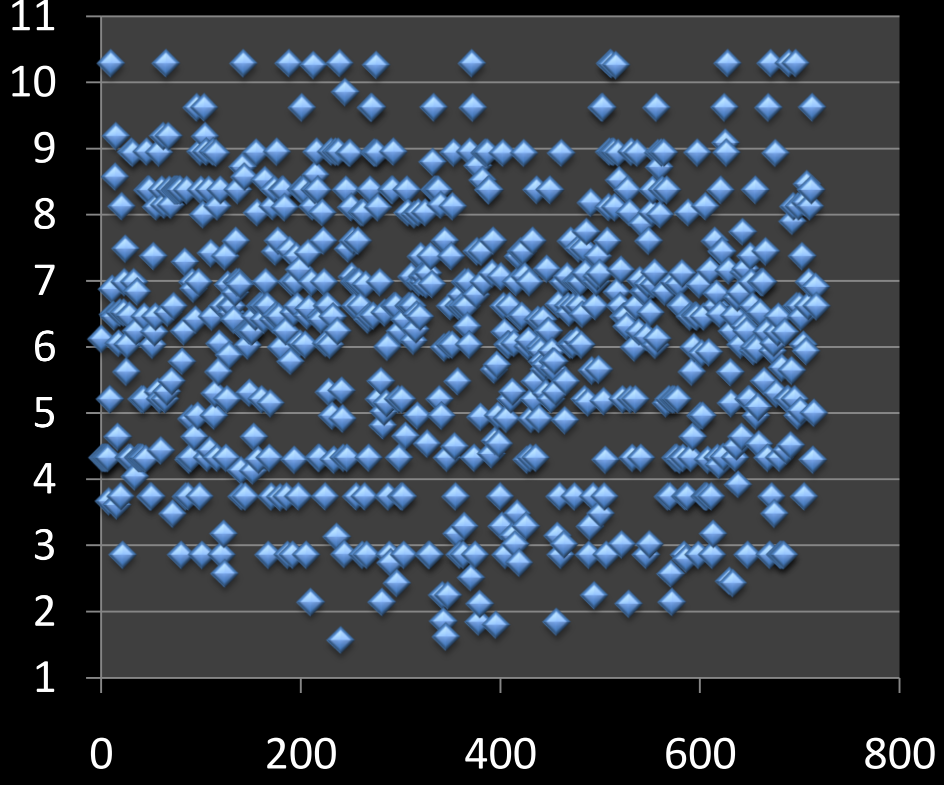
Comparing Base-Scores with environmental score

n=720

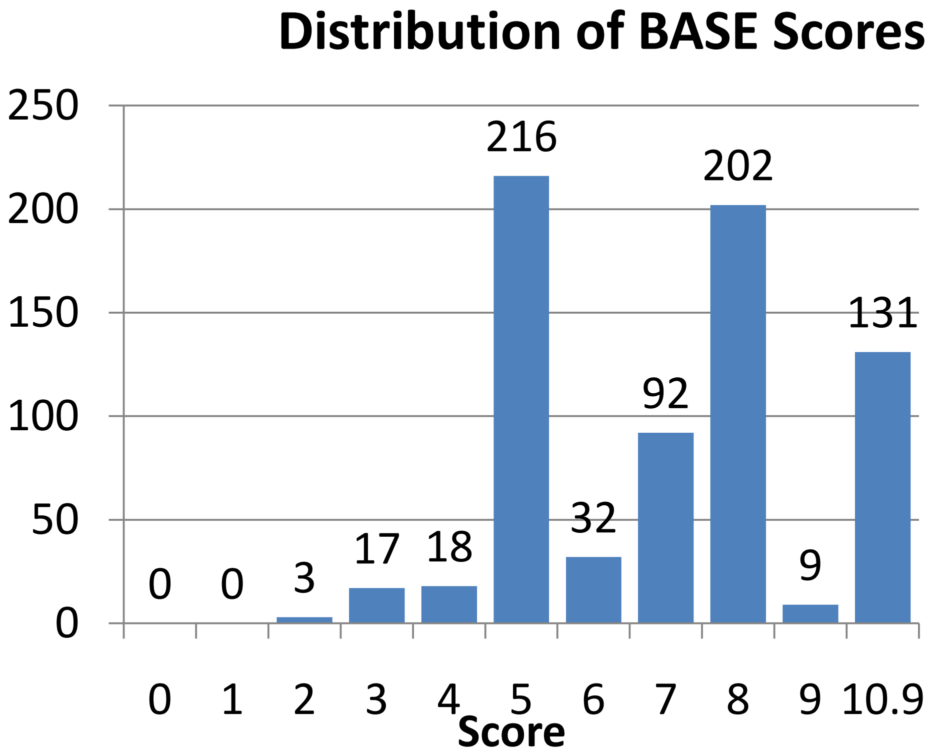
Base SCORE

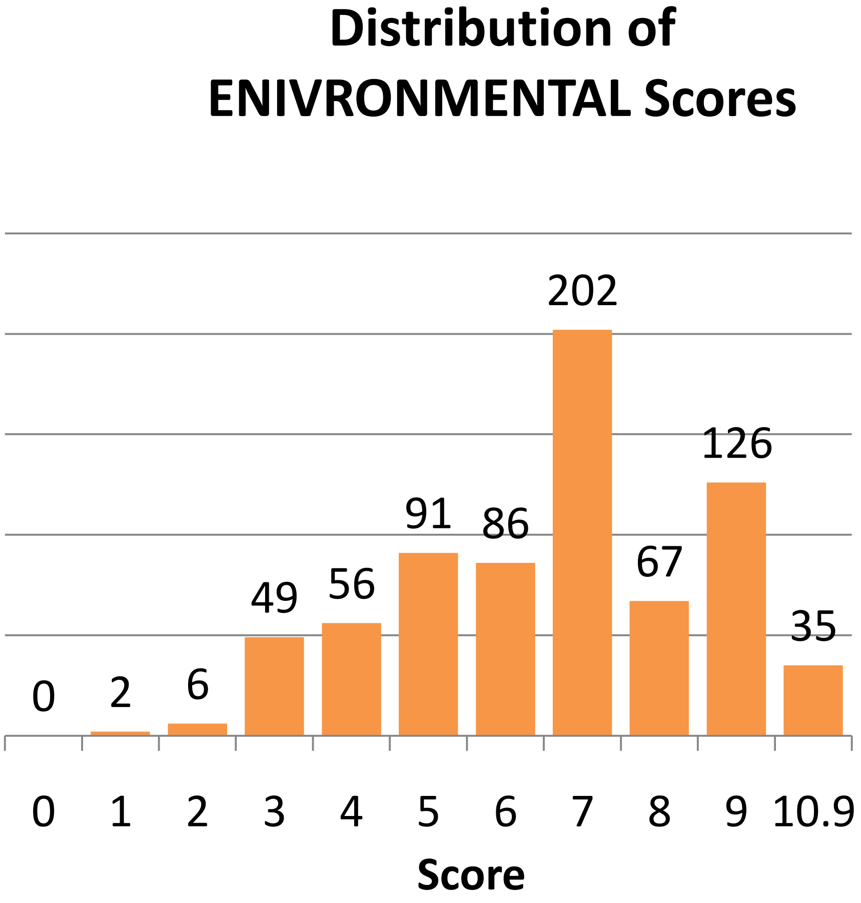


IMPROVED SCORE



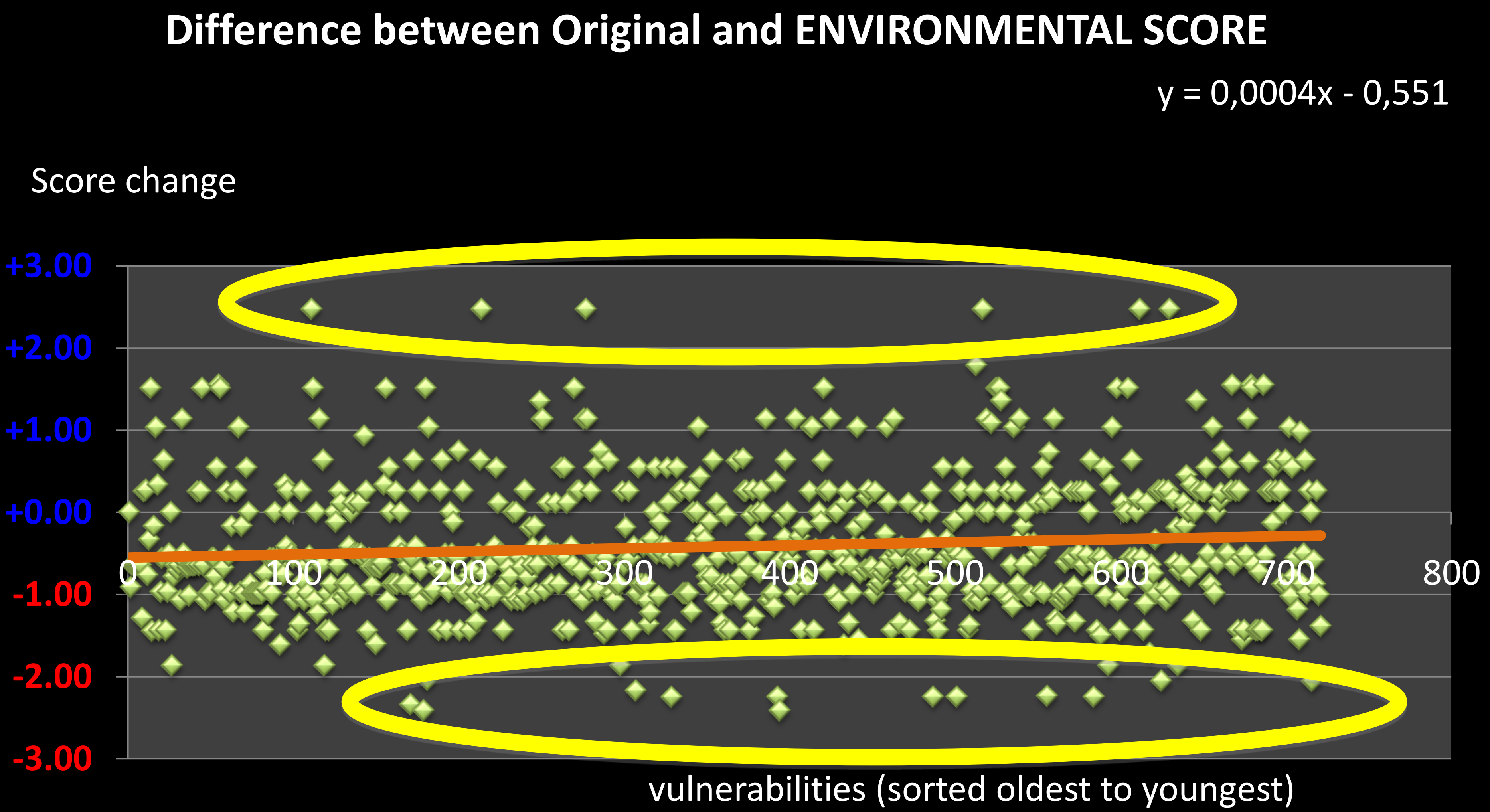
Distribution of scores



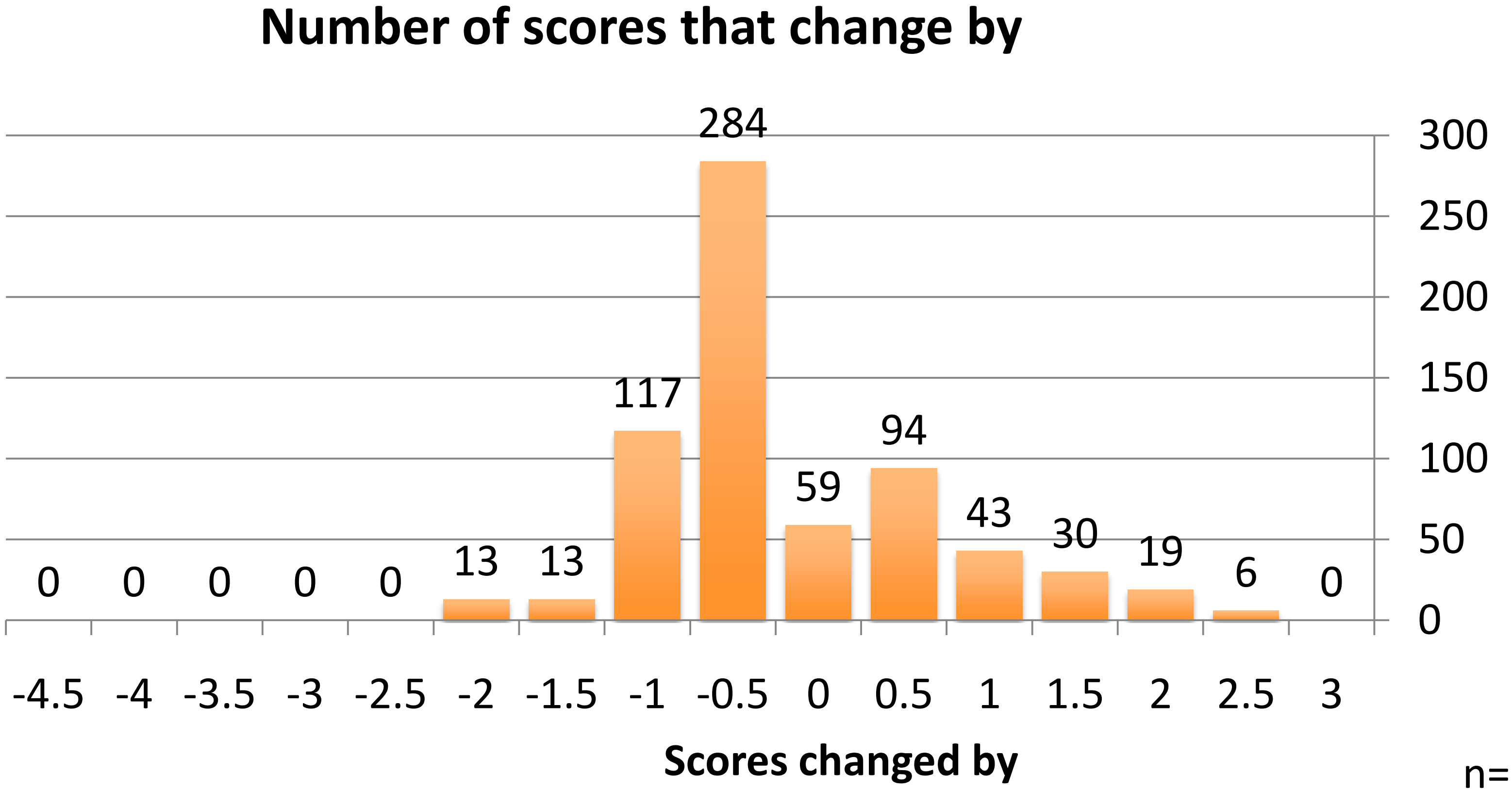


n=720

The of context info on scores

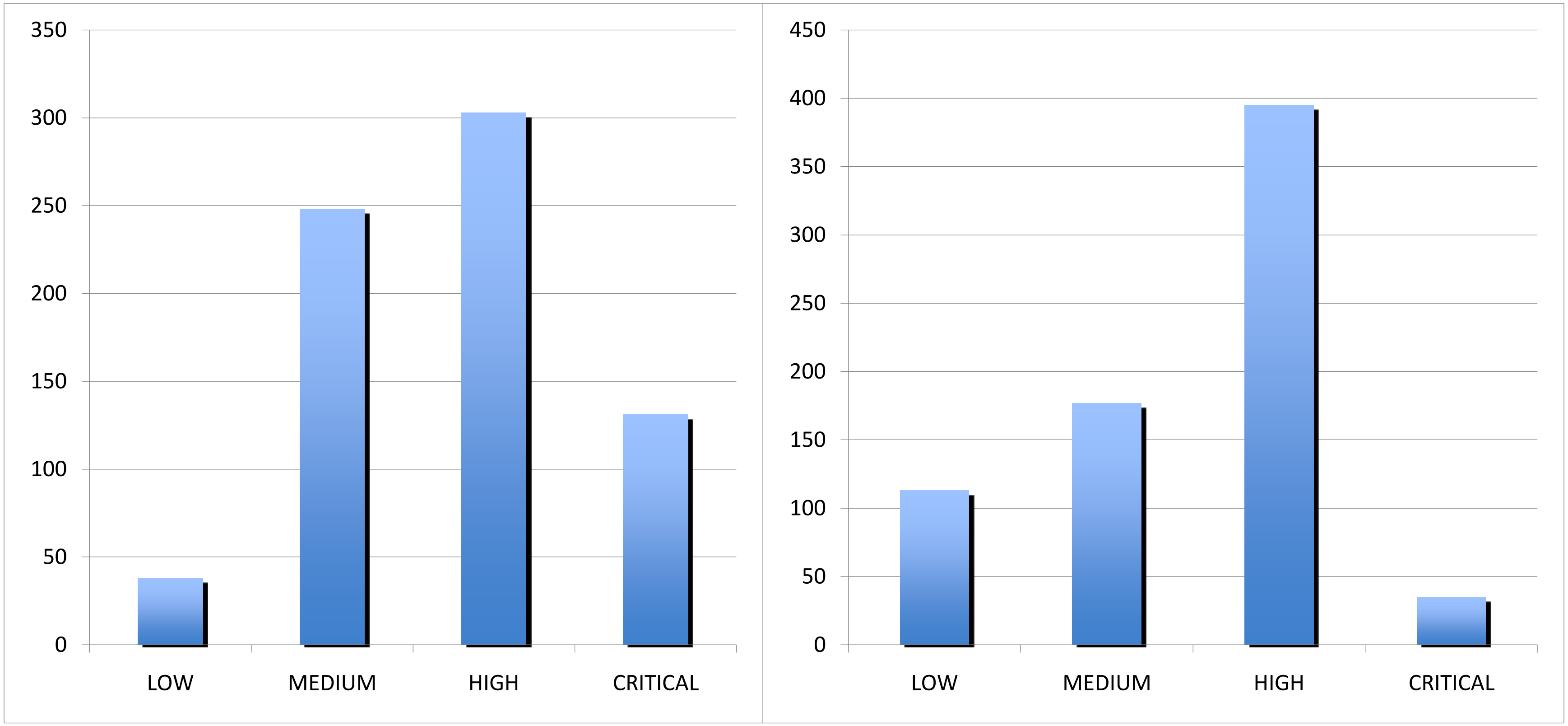


Impact of context info on scores



n=720

Impact of score changes on classification



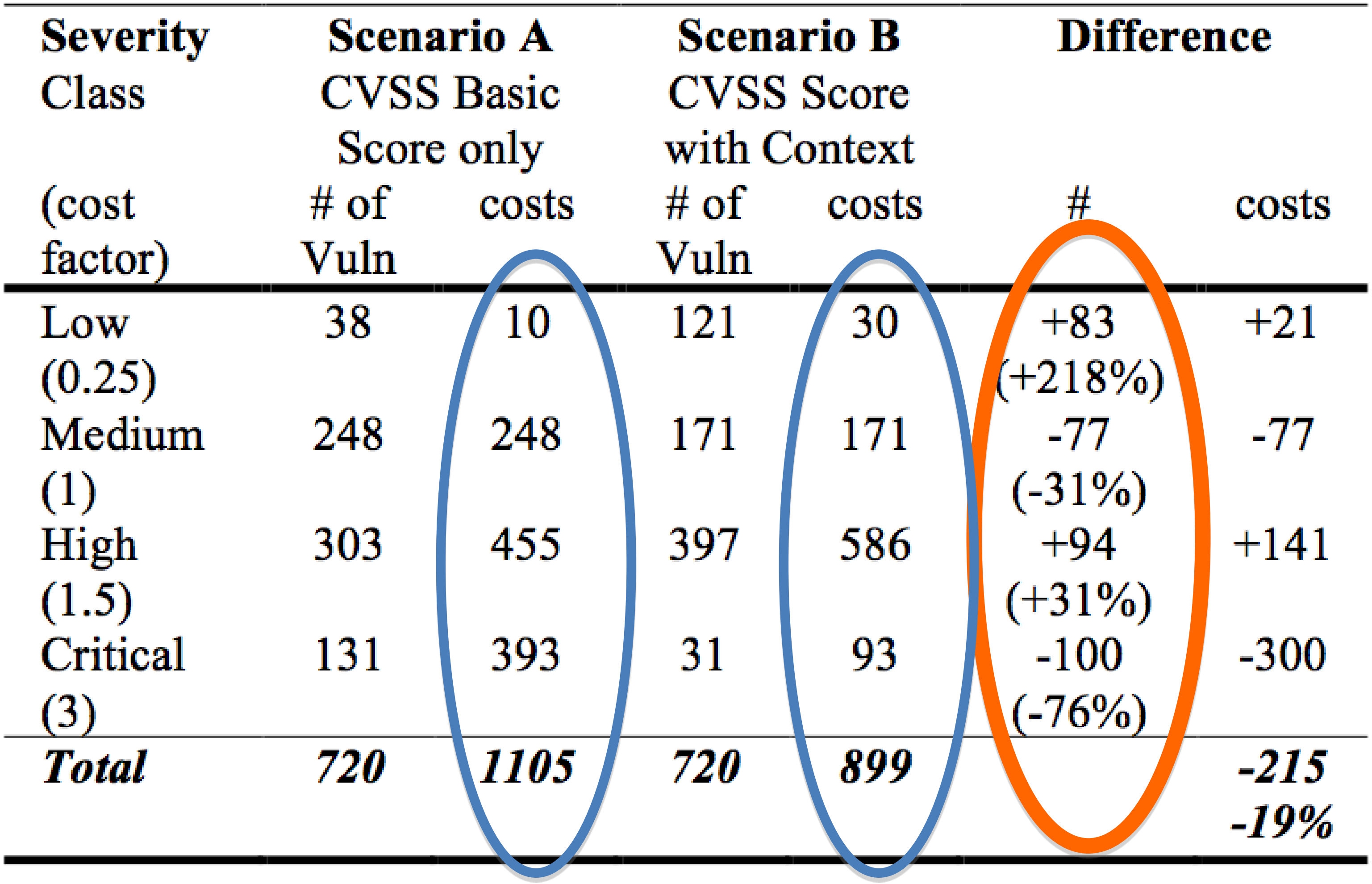
Categorization based only on BASE score

n=720

Based on environmental score

Scores below 5 were classified as ‘Low’ ≥ 5: ‘Medium’, ≥7: ‘High‘ and ≥9: ‘Critical’.

Impact of score changes on anticipated costs



Severity Class	**Scenario A** CVSS Basic Score only		**Scenario B** CVSS Score with Context		**Difference**	
(cost factor)	# of Vuln	costs	# of Vuln	costs	#	costs
Low (0.25)	38	10	121	30	+83 (+218%)	+21
Medium (1)	248	248	171	171	-77 (-31%)	-77
High (1.5)	303	455	397	586	+94 (+31%)	+141
Critical (3)	131	393	31	93	-100 (-76%)	-300
Total	***720***	***1105***	***720***	***899***		***-215*** ***-19%***

Lessons learned

- CVSS is a powerful tool, if used right
- Using CVSS from sources like NVD scores as-is produces sub-optimal prioritization and categorization results
- Estimation can help to estimate improvement potential

What’s next?

- Real world test are underway to compare estimations with actually realized cost savings.

Goal:

- A method to align the security/vulnerability management practices with business goals

Thank you!

- Christian Frühwirth
  - PhD student at Helsinki University of Technology
  - christian.fruehwirth@tkk.fi
- Tomi Männistö
  - Prof. of Software Engineering at Helsinki University of Technology
  - tomi.mannisto@tkk.fi