Risk Modeling for Vulnerabilities

OWASP

Rishi Pande

The OWASP Foundation
http://www.owasp.org

Overview

- What is Risk Modeling?
- Why Risk Modeling?
- Overview of various risk models
- CVSS
- Operationalizing a risk model
- Takeaways

Caveats/ Warnings

- This is an Information Security Process Presentation - not a technical presentation (but I really hope you understand some technology)

- Risk modeling in this presentation refers to application security vulnerability risk modeling

- Any views or opinions presented are solely those of the author and do not necessarily represent those of my employers

- I/ We are not responsible for the consequences of any actions taken on the basis of the information provided

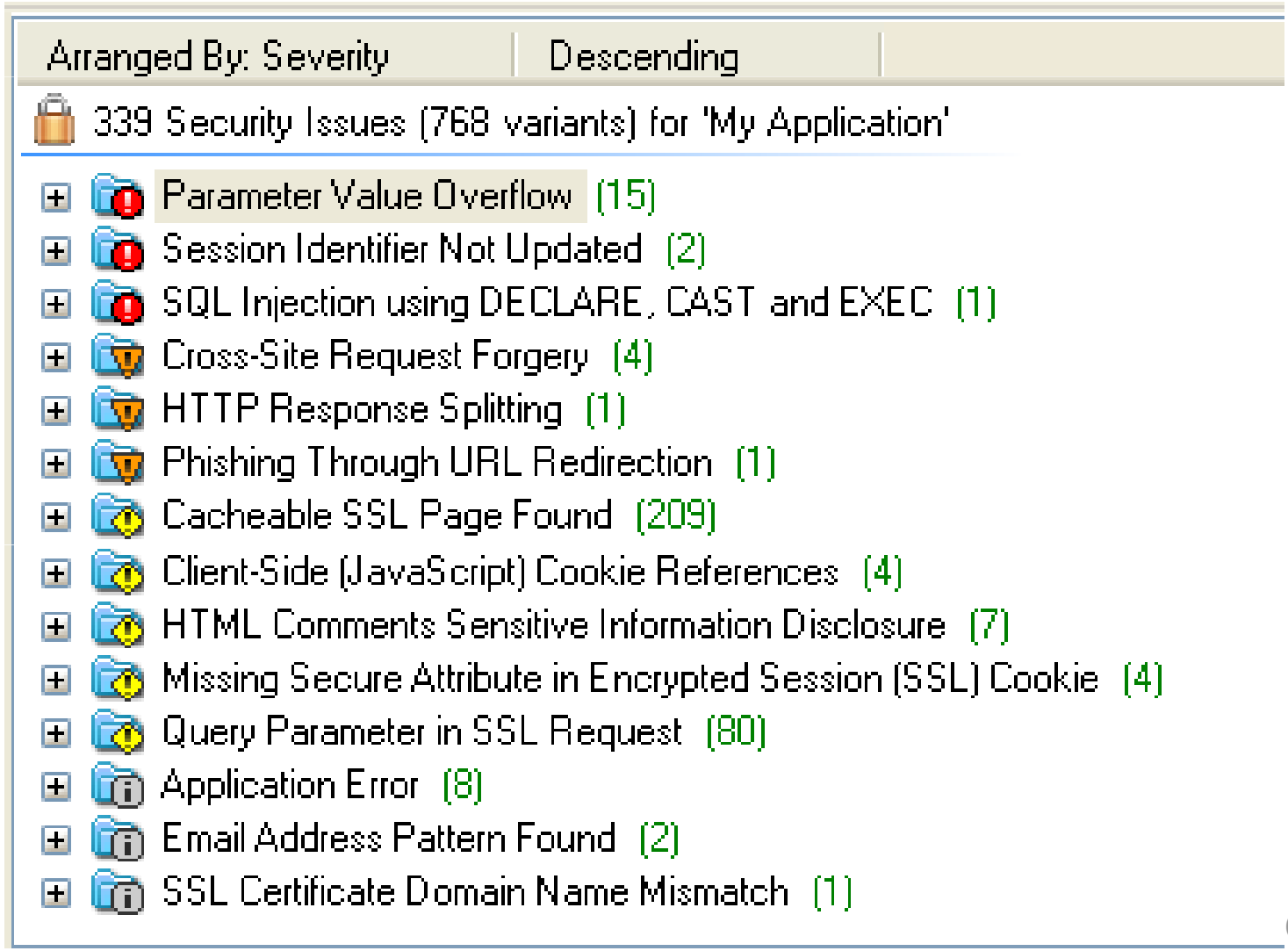
What is Risk Modeling?

- Answers the question: “What is the risk of a particular vulnerability to your organization?”

- Assumes that your organization has already discovered the vulnerability in the application

What is Risk Modeling?

- How does Appscan know “Parameter Value Overflow” is a high risk issue?



Why Risk Modeling?

- Allows organizations to determine risk level arising from a particular vulnerability to the organization, based on its own criteria

- Provides organizations with a ranked list of vulnerabilities to determine correct controls and produce effective countermeasures

- Provides a structured thinking methodology for rating application vulnerabilities to development , audit / assurance, and business

- Allows for translation of vulnerabilities to business risk

What you need to know

- Vulnerability
- Application usage in business context
- Application architecture and data flow
- Application's Information Security requirements
- The threat vector (type of attacker) you are defending against:
  - Curious Attacker
  - Script Kiddies
  - Motivated Attacker
  - Organized Crime

Overview of different risk models

I. OH-SHIT

II. STAR

III. STRIDE

IV. DREAD

V. OWASP

VI. CVSS

I. OH- SHIT model

- AKA "we need a model" model
- AKA "everything is a high" model
- AKA "security auditors know best" model
- Business input tends to be ignored
- No prioritization of risks
- Highly dependent on the background of the individuals involved in the rating of the risk

II. STAR model

- Security Targeting and Analysis of Risks
- Analyzes processes instead of vulnerabilities or systems
- Asks a series of questions arising from a particular vulnerability to determine needed controls
- Builds a matrix of process controls and system severity based on stakeholder input
- May lead to high operational overhead
- Pioneered by Virginia Tech in 2002
- Popular in Educational Institutions

III. STRIDE - Overview

- Classification scheme for vulnerabilities in the following categories:
  - Spoofing Identity
  - Tampering Data
  - Repudiation
  - Information disclosure
  - Denial of Service
  - Elevation of Privilege
- Optimal Usage in software development
- Decomposes system into components based on data flow diagrams
- Analyzes individual components for susceptibility to threats
- Controls added, components reanalyzed

III. STRIDE - Considerations

- No rating scheme for vulnerabilities identified
- Process could go into endless loop
- System integration could result in new (or unforeseen) vulnerabilities that were not identified earlier
- One vulnerability could be placed in different classifications, e.g., XSS could be placed in almost every category

IV. DREAD - Overview

- **Damage Potential**
  - If a threat exploit occurs, how much damage will be caused?
- **Reproducibility**
  - How easy is it to reproduce the threat exploit?
- **Exploitability**
  - What is needed to exploit this threat?
- **Affected Users**
  - How many users will be affected?
- **Discoverability**
  - How easy is it to discover this threat?
- **Risk_DREAD** = (DAMAGE + REPRODUCIBILITY + EXPLOITABILITY + AFFECTED USERS + DISCOVERABILITY) / 5

IV. DREAD – Pros & Cons

- Each vector has a numerical value between 1 to 10 assigned to it, depending on severity

- Damage potential value:
  - 0 = Nothing
  - 5 = Individual user data is compromised or affected.
  - 10 = Complete system or data destruction

- Final output is quantitative, which can be used to prioritize the risks to be addressed

- Quantitative values too wide: difficult to differentiate between a 7 and 8 for damage potential

- "Neither of them (STRIDE or DREAD) were developed with any real academic rigor, and from a scientific standpoint, neither of them tend to hold up very well" - David LeBlanc

V. **OWASP**- Overview

- Risk = Likelihood * Impact
- Individual calculations for the severity of Likelihood and Impact are combined
- Likelihood is measured by:
  - Threat Agent factors
  - Vulnerability factors

Threat agent factors					**Vulnerability factors**			
Skill level	Motive	Opportunity	Size		Ease of discovery	Ease of exploit	Awareness	Intrusion detection
5	2	7	1		3	6	9	2
Overall likelihood=4.375 (MEDIUM)								

- Impact is measured by:
  - Technical Impact
  - Business Impact

Technical Impact					**Business Impact**			
Loss of confidentiality	Loss of integrity	Loss of availability	Loss of accountability		Financial damage	Reputation damage	Non-compliance	Privacy violation
9	7	5	8		1	2	1	5
Overall technical impact=7.25 (HIGH)					Overall business impact=2.25 (LOW)			

V. OWASP- Calculations

- The following scale is used to measure likelihood and impact levels:
  - 0 to < 3 Low
  - 3 to < 6 Medium
  - 6 to 9 High

- The following matrix is then used to calculate the risk:

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

V. **OWASP** – Pros & Cons

- Takes reputational impact, repudiation, and privacy violations into account
- Does not give a quantitative overall risk score
- Impact and likelihood vector ranges too wide 0-9
- All factors have the same weight

VI. CVSS

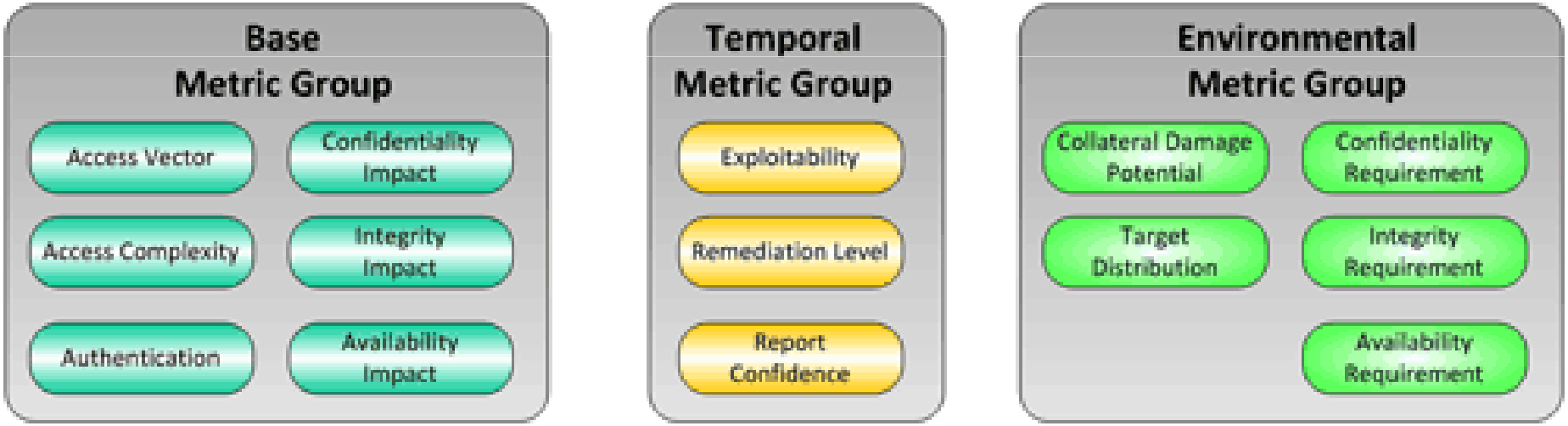
Rishi Pande

CVSS - Overview

- Common Vulnerability Scoring System
- Commissioned by NIAC / Maintained by FIRST
- Quickly becoming the *standard* for application vulnerability risk modeling
- Provides a score as well as equation that quickly tells the reader how the score was determined:
  - CVSS2:5.9(AV:L/AC:L/Au:S/C:C/I:C/A:N/E:H/RL:OF/RC:C/CDP:ND/TD:ND/CR:H/IR:H/AR:H)

CVSS – Metric Groups

- CVSS is composed of three metric groups: Base, Temporal, and Environmental, each consisting of a set of metrics



Base: represents the intrinsic and fundamental characteristics of a vulnerability that are constant over time and user environments.

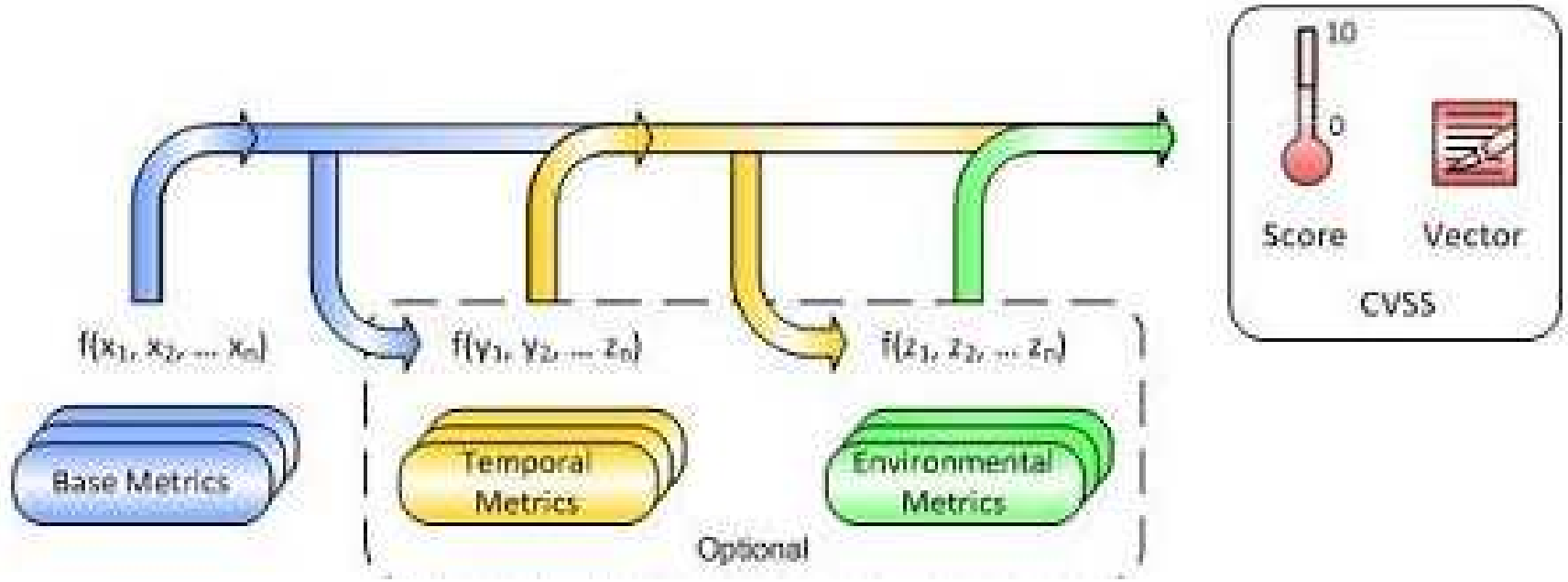
Temporal: represents the characteristics of a vulnerability that change over time but not among user environments.

Environmental: represents the characteristics of a vulnerability that are relevant and unique to a particular user's environment.

Please refer to the following link for more information: http://www.first.org/cvss/cvss-guide.html

CVSS – Group Interaction

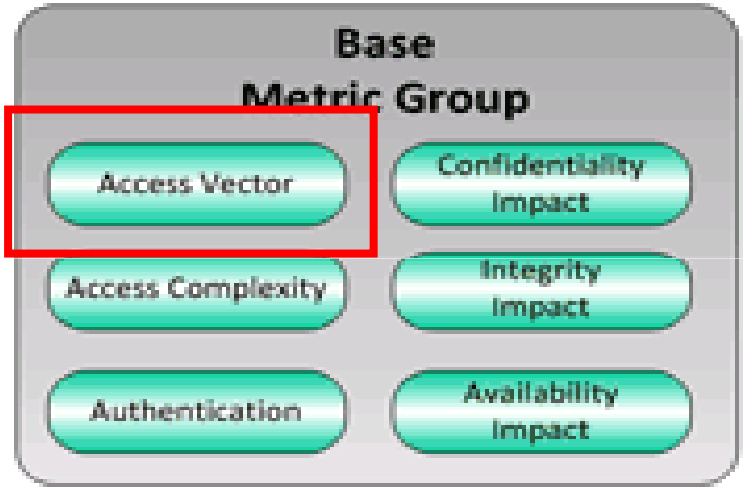
- How do the three groups interact?



- If you are unable to calculate metrics for one particular group, the model will assume default values to determine the overall calculation

Base Metrics – Access Vector

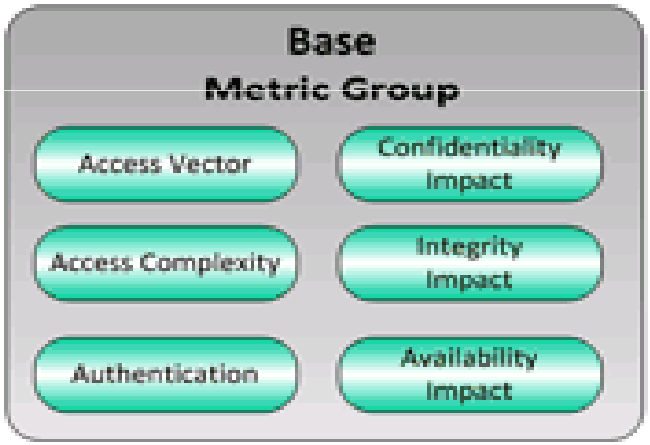
- Access Vector defines the location from which a vulnerability can be exploited.
- The more remote the location, the greater its impact on the score.



Metric Value	Description
Local (L)	A vulnerability exploitable with only *local access* requires the attacker to have either physical access to the vulnerable system or a local (shell) account.
Adjacent Network (A)	A vulnerability exploitable with *adjacent network access* requires the attacker to have access to either the broadcast or collision domain of the vulnerable software.
Network (N)	A vulnerability exploitable with *network access* means the vulnerable software is bound to the network stack and the attacker does not require local network access or local access. Such a vulnerability is often termed "remotely exploitable".

Base Score - Calculations

- BaseScore =round_to_1_decimal(((0.6*Impact) + (0.4*Exploitability)–1.5)*f(Impact))
  - **Impact =**
    - 10.41*(1-(1-ConfImpact)*(1-IntegImpact)*(1-AvailImpact))
  - **Exploitability =**
    - 20* AccessVector*AccessComplexity*Authentication
  - **f(impact)=**
    - 0 if Impact=0, 1.176 otherwise



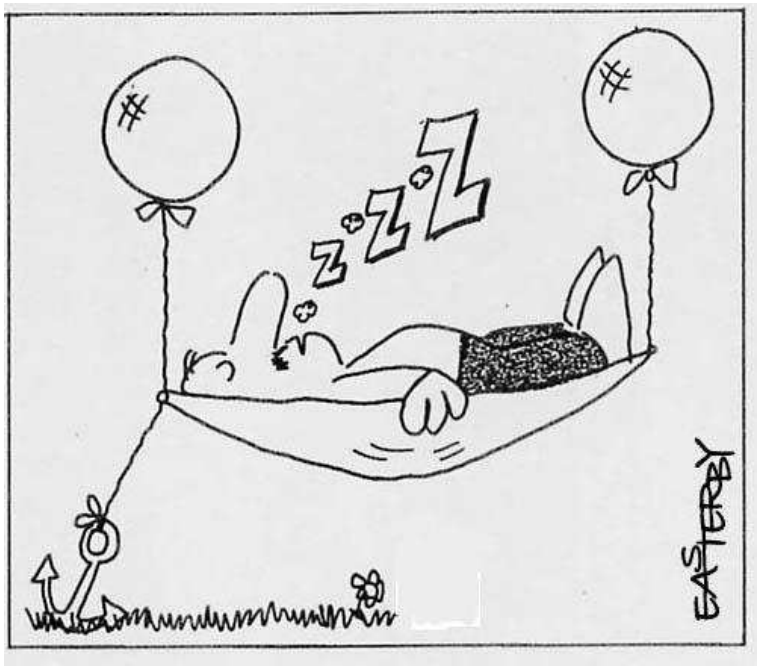
CVSS - Overall Score

- **TemporalScore =**

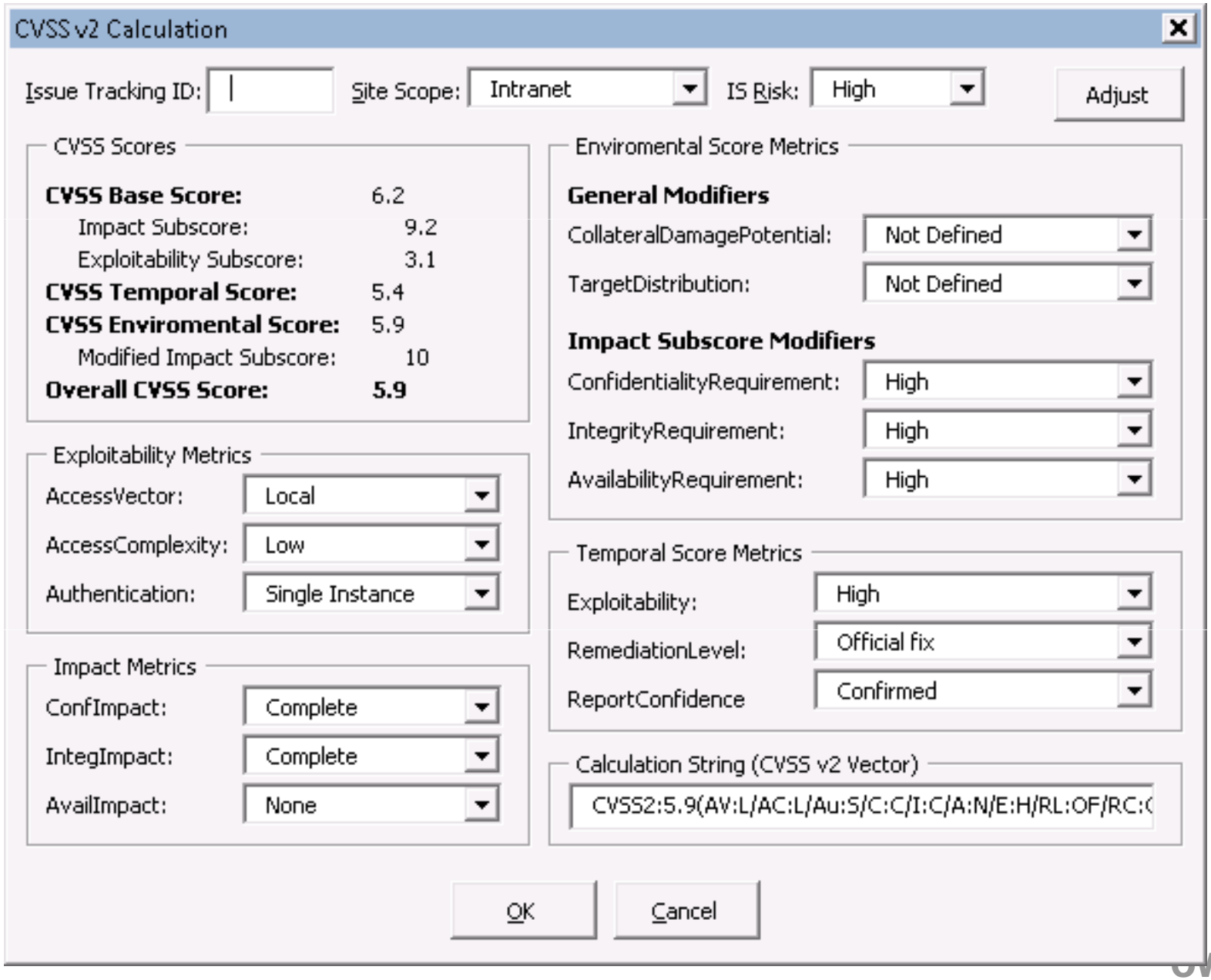
  Round_to_1_decimal (BaseScore*Exploitability *RemediationLevel*ReportConfidence)

- **EnvironmentalScore =**

  Round_to_1_decimal((AdjustedTemporal + (10-Adjusted Temporal)*CollateralDamagePotential)*TargetDistribution)



CVSS - Calculator



CVSS - Conclusions

- Calculators provided by NIST
- Provides a score between 0 and 10. NIST standard proposes to use the following rating scheme:
  - Low 0.0 – 3.9
  - Medium 4.0 – 6.9
  - High 7.0 – 10.0
- Used by several agencies and vendors to report their findings:
  - National Vulnerability Database (NVD)
  - Cisco, Qualys, ISS publish vulnerabilities with CVSS scores
- Supported by Vulnerability Scanning tools such as Appscan, WebInspect, etc. (as of 2009)
- Organizations should adapt vectors to application specific scenarios

Operationalizing a Risk Models

- Determine business environment
- Determine available input variables
- Allow stakeholders to provide data to different parts of the model where they possess domain knowledge
- Security auditors --> CIA *compromise* of the vulnerability
- Business --> CIA *requirement* for the application

Takeaways

- Having any quantitative repeatable risk model is better than none at all
- Consider and understand the operational requirements for each model prior to final selection
- Adapt the chosen model to meet your company's needs prior to implementation (avoid scope creep)
- Ensure that all stakeholders understand the chosen risk model and their roles in providing input
- CVSS has proven to be the most popularly used risk model because it’s of its NIST standard, quantitativeness, relative ease of comprehension, and repeatability

Discussions

Rishi Pande