Tricolour Alphanumerical Spaghetti

Application Vulnerability Severity Ranking

Do you know your "A, B, Cs" from your "1, 2, 3s"? Is "red" much worse than "orange", and why is "yellow" used instead of "green"? Just what is a "critical" vulnerability? Is "critical" the same as "very high"? How do PCI DSS "level 4 and 5" security scanning vulnerabilities relate to application weaknesses? Does a "tick" mean you passed? Are you using CWE and CVSS? Is a "medium" network vulnerability as dangerous as a "medium" application vulnerability? Can CWSS help?

- Colin Watson
- Watson Hall Ltd
  London, United Kingdom
- https://www.watsonhall.com

Scoping

Menu

What

- Not severity ranking
- Severity vs prioritisation

Why

- Purposes
- Audiences

How

- Calculation
- Desirable properties

Qualitative:

- Text
  - Info, Warning, Hot
  - Low, Moderate, Important, Critical
  - Low, Medium, High, Critical
  - Info, V.Low, Low, Low, Medium, High, V.High
  - Pass, Fail
- Numerical
  - 1, 2, 3
  - 1 – 5
  - 0.0 – 10.0
  - 1 – 180
- Alphabetical
  - I, C, B, A
- Quantitative
  - £, €, $, etc

Health warning

Purposes

The presentation today

- ✔ Individual application vulnerabilities
  - ✔ Severity
  - ✔ Prioritisation

Not today

- ✗ Target level of assurance
- ✗ Application portfolio risk ranking

Are we speaking the same language?

Software issue tracking - Bugzilla

- Status (e.g. unconfirmed, new, assigned, reopened, resolved, verified)
- Resolution (e.g. fixed, invalid, wontfix, duplicate, worksforme, duplicate)
- Priority and due date
- Severity measures "impact of a bug"

Blocker	Blocks development and/or testing work
Critical	Crashes, loss of data, severe memory leak
Major	Major loss of function
Normal	Regular issue, some loss of functionality under specific circumstances
Minor	Minor loss of function, or other problem where easy workaround is present
Trivial	Cosmetic problem like misspelled words or misaligned text
Enhancement	Request for enhancement

Event logging severity

- BSD syslog protocol (RFC 3164)
  0. Emergency (system is unusable)
  1. Alert (action must be taken immediately)
  2. Critical (critical conditions)
  3. Error (error conditions)
  4. Warning (warning conditions)
  5. Notification (normal but significant condition)
  6. Informational (informational messages)
  7. Debugging (debug-level message)

- Common Event Format (CEF)

  0 (least) – 10 (most significant)

- Custom e.g.
  - Fatal (an error that cannot be recovered from)
  - Error
  - Warning (anomalous condition)
  - Informational (non-error events)

Incident management

- Information Technology Infrastructure Library (ITIL)
  - Severity
  - Impact
  - Urgency
  - Priority
- NIST SP 800-61
  - Overall severity
    - Critical (7.50-10.00)
    - High (5.00-7.49)
    - Medium (3.75-4.99)
    - Low (2.50-3.74)
    - Minimal (1.00-2.49)
    - Low (0.00-0.99)

What does [.us].gov have to offer?

SP800-30 Guide for Conducting Risk Assessments

Risk

- The combination of the likelihood of a threat event's occurrence and its potential adverse impact

- Determine likelihood of threat event
  - Initiation/occurrence
  - Resulting in adverse impacts

- Determine relative impact on the target

CERT Secure Coding Standards

Each rule and recommendation has an assigned **Priority**. Priorities are assigned using a metric based on Failure Mode, Effects, and Criticality Analysis (FMECA) [IEC 60812]. Three values are assigned for each rule on a scale of 1 to 3 for

- **Severity** – how serious are the consequences of the rule being ignored

Value	Meaning	Examples of Vulnerability
1	low	denial-of-service attack, abnormal termination
2	medium	data integrity violation, unintentional information disclosure
3	high	run arbitrary code

- **Likelihood** – how likely is it that a flaw introduced by ignoring the rule can lead to an exploitable vulnerability

Value	Meaning
1	unlikely
2	probable
3	likely

- **Remediation Cost** – how expensive is it to comply with the rule

Value	Meaning	Detection	Correction
1	high	manual	manual
2	medium	automatic	manual
3	low	automatic	automatic

The three values are then multiplied together for each rule. This product provides a measure that can be used in prioritizing the application of the rules. These products range from 1 to 27, although only the following 10 distinct values are possible: 1, 2, 3, 4, 6, 8, 9, 12, 18, and 27. Rules and recommendations with a priority in the range of 1-4 are **Level 3** rules, 6-9 are **Level 2**, and 12-27 are **Level 1**.

- **Priorities** and **Levels**

Level	Priorities	Possible Interpretation
L1	12, 18, 27	High severity, likely, inexpensive to repair
L2	6, 8, 9	Medium severity, probable, medium cost to repair
L3	1, 2, 3, 4	Low severity, unlikely, expensive to repair

US-CERT Vulnerability Notes severity metric

- Used in Vulnerability Notes Database http://www.kb.cert.org/vuls/
- Method of calculation not publicly available
  - Public knowledge
  - Exploitability (preconditions and ease)
  - Currently being exploited
  - Impact on “the internet”
- Unequal weighting
- Score 0 to 180 (non linear scale)
- If >40, included in US-CERT alerts
- Vulnerability Notes published after 27 March 2012 use CVSS metrics instead

Severity	Include in alerts
41-180	Yes
0-40	No

Common Vulnerability Scoring Standard (CVSS) v2

Forum of Incident R… Teams (FIRST)

…erkan Özkan

- Standardised vul… between 0.0 and
- Associated "vecto… (AV:N/AC:L/Au:N… RC:C/CDP:L/TD:…
- No names (low, …
- Groups
  - **Base**
  - Temporal
  - Environmen…

Sponsored by DHS National Cyber Security Division/US-CERT

NIST National Institute of Standards and Technology

National Vulnerability Database

automating vulnerability management, security measurement, and compliance checking

Vulnerabilities | Checklists | 800-53/800-53A | Product Dictionary | Impact Metrics | Data Feeds | Statistics

Home | SCAP | SCAP Validated Tools | SCAP Events | About | Contact | Vendor Comments

CVSS Version 2 Scoring Page (TAS-XSS-3)

This page shows the components of the CVSS score for TAS-XSS-3 and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score. A concise form of this page is available to CVSS experts.

Update Scores | Reset Scores | View Equations

CVSS Base Score 5
Impact Subscore 2.9
Exploitability Subscore 10
CVSS Temporal Score 5
CVSS Environmental Score 5.7
Modified Impact Subscore 1.4
Overall CVSS Score 5.7

Base Score Metrics

These metrics describe inherent characteristics of the vulnerability. These scores have already been calculated for this vulnerability.

Exploitability Metrics

Related exploit range (AccessVector): Network
Attack complexity (AccessComplexity): Low
Level of authentication needed (Authentication): None

Impact Metrics

Confidentiality impact (ConfImpact): None
Integrity impact (IntegImpact): Partial
Availability impact (AvailImpact): None

Environmental Score Metrics

This section addresses metrics that describe the effect of a vulnerability within an organization's environment. These metrics must calculated separately for each organization.

General Modifiers

Organization specific potential for loss (CollateralDamagePotential): Low-Medium
Percentage of vulnerable systems (TargetDistribution): High (76-100%)

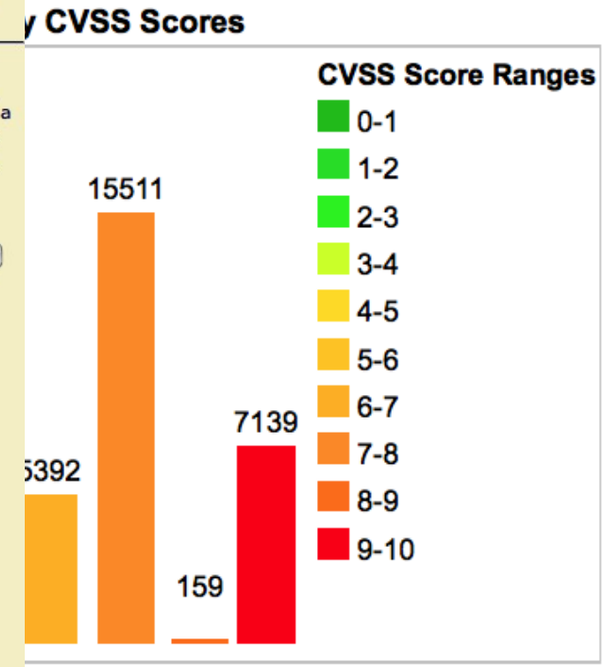
Impact Subscore Modifiers

System confidentiality requirement (draft proposal) (ConfidentialityRequirement): Medium
System integrity requirement (draft proposal) (IntegrityRequirement): Low
System availability requirement (draft proposal) (AvailabilityRequirement): High

Temporal Score Metrics

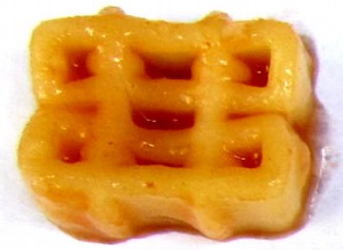
These metrics describe elements about the vulnerability that change over time. If all of these values are left as 'Undefined', the environmental score will be based on the base score.

Availability of exploit (Exploitability): High
Type of fix available (RemediationLevel): Unavailable
Level of verification that vulnerability exists (ReportConfidence): Confirmed



Severity	Detail
0.0 – 10.0	Vector

CxSS



Common Configuration Scoring System (CCSS)

- December 2010
- Use of security configuration settings that negatively affect the security of the software
- Vulnerabilities occur as a result of choosing to configure the software in a particular manner
- Score 0.0 – 10.0 and vector
- Examples
  - Kernel level auditing disabled
  - Account lockout duration set t less than required minimum
  - FTP service enabled

Common Misuse Scoring System (CMSS)

- July 2012
- Use of a software feature in an unintended manner in a way that provides an avenue to compromise the security of a system
- Vulnerabilities occur as the result of result of providing additional features
- Score 0.0 – 10.0 and vector
- Examples
  - Bypass file upload anti-virus scanning by changing file extension
  - Attacker can impersonate a valid user
  - User follows link to a spoofed website

Some vendors

Microsoft severity rating system

Rating	Definition
Critical	A vulnerability whose exploitation could enable the propagation of an Internet worm with little or no user action.
Important	A vulnerability whose exploitation could result in compromise of the confidentiality, integrity, or availability of users' data, or of the integrity or availability of processing resources.
Moderate	A vulnerability whose exploitation is mitigated to a significant degree by factors such as default configuration, auditing, or difficulty of exploitation.
Low	A vulnerability whose exploitation is extremely difficult, or whose impact is minimal.

Severity
Critical
Important
Moderate
Low

Microsoft exploitability index system

Rating	Definition
1	**Consistent exploit code likely**. This rating means that our analysis has shown that exploit code could be created in such a way that an attacker could consistently exploit that vulnerability. For example, an exploit would be able to cause remote code execution of that attacker's code repeatedly, and in a way that an attacker could consistently expect the same results. This would make it an attractive target for attackers, and therefore more likely that exploit code would be created. As such, customers who have reviewed the security bulletin and determined its applicability within their environment could treat this with a higher priority.
2	**Inconsistent exploit code likely**. This rating means that our analysis has shown that exploit code could be created, but an attacker would likely experience inconsistent results, when targeting the affected product. For example, an exploit would be able to cause remote code execution, but may only work 1 out of 10 times, or 1 out of 100 times, depending on the state of the system being targeted and the quality of the exploit code. While an attacker may be able to increase the consistency of their results by having better understanding and control of the target environment, the unreliable nature of this attack makes it a less attractive target for attackers. Therefore, it is likely that exploit code will be created, but it is unlikely that attacks will be as effective as other, more consistently exploitable, vulnerabilities. As such, customers who have reviewed the security bulletin and determined its applicability within their environment should treat this as a material update, but if prioritizing against other highly exploitable vulnerabilities, could rank this lower in their deployment priority.
3	**Functioning exploit code unlikely**. This rating means that our analysis has shown that exploit code which functions successfully is unlikely to be released. This means that it might be possible for exploit code to be released that could trigger the vulnerability and cause abnormal behavior, but it is unlikely that an attacker would be able to create an exploit that could successfully exercise the full impact of the vulnerability. Given that vulnerabilities of this type would require significant investment by attackers to be useful, the risk of exploit code being creating and used is much lower. Therefore, customers who have reviewed the security bulletin to determine its applicability within their environment could prioritize this update below other vulnerabilities within a release.

Redhat issue severity classification

Level	Description
Critical impact	This rating is given to flaws that could be easily exploited by a remote unauthenticated attacker and lead to system compromise (arbitrary code execution) without requiring user interaction. These are the types of vulnerabilities that can be exploited by worms. Flaws that require an authenticated remote user, a local user, or an unlikely configuration are not classed as critical impact.
Important impact	This rating is given to flaws that can easily compromise the confidentiality, integrity, or availability of resources. These are the types of vulnerabilities that allow local users to gain privileges, allow unauthenticated remote users to view resources that should otherwise be protected by authentication, allow authenticated remote users to execute arbitrary code, or allow local or remote users to cause a denial of service.
Moderate impact	This rating is given to flaws that may be more difficult to exploit but could still lead to some compromise of the confidentiality, integrity, or availability of resources, under certain circumstances. These are the types of vulnerabilities that could have had a critical impact or important impact but are less easily exploited based on a technical evaluation of the flaw, or affect unlikely configurations.
Low impact	This rating is given to all other issues that have a security impact. These are the types of vulnerabilities that are believed to require unlikely circumstances to be able to be exploited, or where a successful exploit would give minimal consequences.

Severity
Critical impact
Important impact
Moderate impact
Low impact

Approved Scanning Vendors (ASVs)

PCI Security Standards Council

Level	Severity	Description
5	Urgent	Trojan Horses; file read and writes exploit; remote command execution
4	Critical	Potential Trojan Horses; file read exploit
3	High	Limited exploit of read; directory browsing; DoS
2	Medium	Sensitive configuration information can be obtained by hackers
1	Low	Information can be obtained by hackers on configuration

Table 1 Vulnerability Severity Levels

"High-level vulnerabilities are designated as level 3, 4, or 5"

Level	Severity
5	Urgent
4	Critical
3	High
2	Medium
1	Low

ASVs (continued)

PCI Security Standards Council

Table 2: Vulnerability Severity Levels Based on the NVD and CVSS Scoring

CVSS Score	Severity Level	Scan Results	Guidance
7.0 through 10.0	High Severity	**Fail**	To achieve a passing scan, these vulnerabilities must be corrected and the environment must be re-scanned after the corrections (with a report that shows a passing scan). Organizations should take a risk-based approach to correct these types of vulnerabilities, starting with the most critical ones (rated 10.0), then those rated 9, followed by those rated 8, 7, etc., until all vulnerabilities rated 4.0 through 10.0 are corrected.
4.0 through 6.9	Medium Severity	**Fail**	
0.0 through 3.9	Low Severity	**Pass**	While passing scan results can be achieved with vulnerabilities rated 0.0 through 3.9, organizations are encouraged, but not required, to correct these vulnerabilities.

"Generally, to be considered compliant, a component must n
that has been assigned a CVSS base score equal to or highe

Severity	Colour	CVSS2 Base
High	Red	7.0 – 10.0
Medium	Red	4.0 – 6.9
Low	Green	0.1 – 3.9

Tenable Nessus

Severity	Risk	Colour	CVSS2 Base
4	Critical	Purple	10.0
3	High	Red	7.0 – 9.9
2	Medium	Orange	4.0 – 6.9
1	Low	Blue	0.1 – 3.9
0	None	Green	0.0

Secunia

Term: "Criticality" (Secunia's severity rating)

Extremely Critical (5 of 5)

Typically used for remotely exploitable vulnerabilities that can lead to system compromise. Successful exploitation does not normally require any interaction and exploits are in the wild.

These vulnerabilities can exist in services like FTP, HTTP, and SMTP or in certain client systems like email programs or browsers.

Highly Critical (4 of 5)

Typically used for remotely exploitable vulnerabilities that can lead to system compromise. Successful exploitation does not normally require any interaction but there are no known exploits available at the time of disclosure.

Such vulnerabilities can exist in services like FTP, HTTP, and SMTP or in client systems like email programs or browsers.

Moderately Critical (3 of 5)

Typically used for remotely exploitable Denial of Service vulnerabilities against services like FTP, HTTP, and SMTP, and for vulnerabilities that allow system compromises but require user interaction.

This rating is also used for vulnerabilities allowing system compromise on LANs in services like SMB, RPC, NFS, LPD and similar services that are not intended for use over the Internet.

Less Critical (2 of 5)

Typically used for cross-site scripting vulnerabilities and privilege escalation vulnerabilities.

This rating is also used for vulnerabilities allowing exposure of sensitive data to local users.

Not Critical (1 of 5)

Typically used for very limited privilege escalation vulnerabilities and locally exploitable Denial of Serv

This rating is also used for non-sensitive system information disclosure vulnerabilities (e.g. remote applications).

Criticality		Colour
5	Extremely critical	Red
4	Highly critical	Orange
3	Moderately critical	Yellow
2	Less critical	Greeny-yellow
1	Not critical	Green

Qualys Qualysguard

Vulnerabilities

Vulnerabilities are design flaws, programming errors, or mis-configurations that make your web application and web application platform susceptible to malicious attacks. Depending on the level of the security risk, the successful exploitation of a vulnerability can vary from the disclosure of information to a complete compromise of the web application and/or the web application platform. Even if the web application isn't fully compromised, an exploited vulnerability could still lead to the web application being used to launch attacks against users of the site.

SEVERITY	LEVEL	DESCRIPTION
	Minimal	Basic information disclosure (e.g. web server type, programming language) might enable intruders to discover other vulnerabilities, but lack of this information does not make the vulnerability harder to find.
	Medium	Intruders may be able to collect sensitive information about the application platform, such as the precise version of software used. With this information, intruders can easily exploit known vulnerabilities specific to software versions. Other types of sensitive information might disclose a few lines of source code or hidden directories.
	Serious	Vulnerabilities at this level typically disclose security-related information that could result in misuse or an exploit. Examples include source code disclosure or transmitting authentication credentials over non-encrypted channels.
	Critical	Intruders can exploit the vulnerability to gain highly sensitive content or affect other users of the web application. Examples include certain types of cross-site scripting and SQL injection attacks.
	Urgent	Intruders can exploit the vulnerability to compromise the web application's data store, obtain information from other users' accounts, or obtain command execution on a host in the web application's architecture.

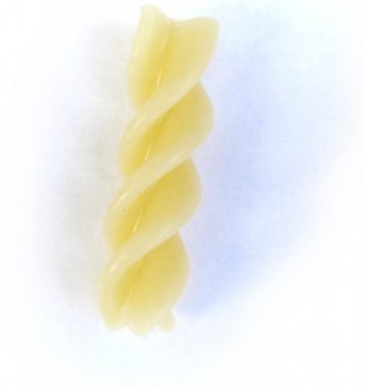
Information Gathered

Information Gathered includes visible information about the web a
include information about users of the web application.

SEVERITY	LEVEL	DESCRIPTION
	Minimal	Intruders may be able to retrieve sensitive infor
	Medium	Intruders may be able to retrieve sensitive infor application.
	Serious	Intruders may be able to detect highly sensitive of the web application.

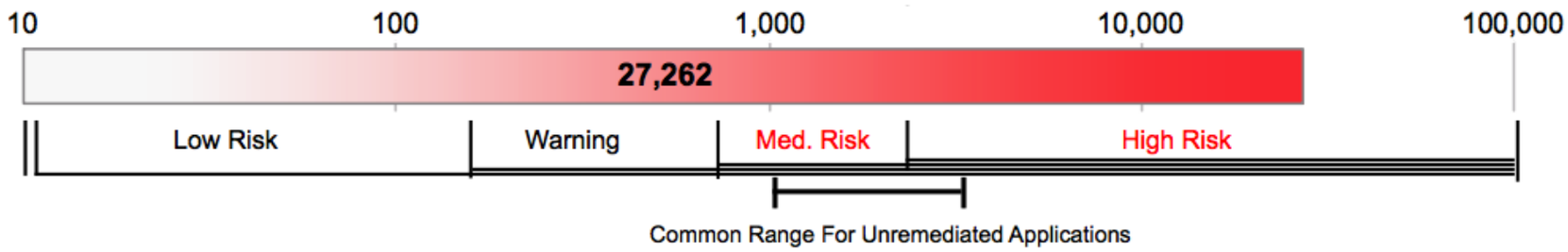
Severity		Colour 1	Class	Colour 2	CVSS2 Base
5	Urgent	All Red	High	Red	7.0 – 10.0
4	Critical	Lots of red			
3	Serious	Very red	Medium	Orange	4.0 – 6.9
2	Medium	Fairly red	Low	Yellow	0.0 – 3.9
1	Minimal	A bit of red			

Cenzic



HARM Scoring

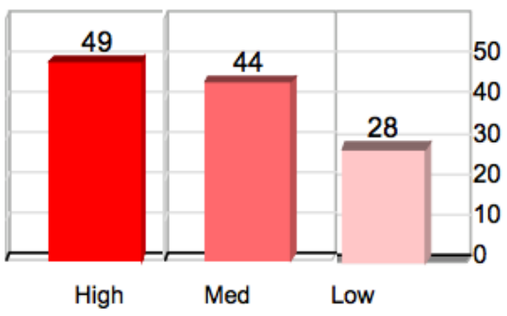
Total HARM™ Score: 27262 = 27262 (Raw Scores) x 1.0 (App Risk Factor)



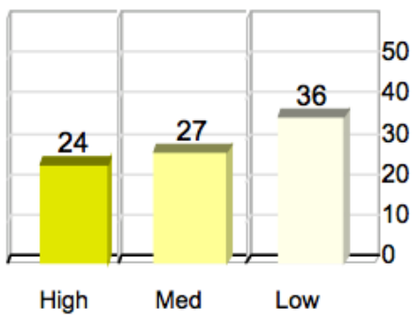
The **'Total HARM Score'**, above, is a sum of the HARM scores for all the SmartAttack assessments included in this report. SmartAttacks have different HARM scores based on the risks associated with each kind of vulnerability . The charts reflect the raw HARM scores without application specific risk adjustments.

Severity of Findings

Severity Drill Down Severity Drill-down w/Info Items



Vulnerabilities



Warnings

Pages Tested	101
Attack Count	14878

Note: **High, Med. & Low** relate to the severity of the findings. **Warnings** are findings for which there is less confidence of being real vulnerabilities.

Severity	Colour
High	Red
Medium	Salmon
Low	Pink

VeraCode

Veracode Detailed Report
Application Security Report
for Example Company

Application Assessed: WebGoat

Veracode Level: VL1
Rated: Jul 28, 2011

Business Criticality:	Very High	Application Version:	5.0 - Java
Target Level:	VL4	Published Rating:	DD

Executive Summary

This report contains a summary of the security flaws identified in the application using automated static, automated dynamic and/or manual security analysis techniques. This is useful for understanding the overall security quality of an individual application or for comparisons between applications.

Application Business Criticality: BC5 (Very High)

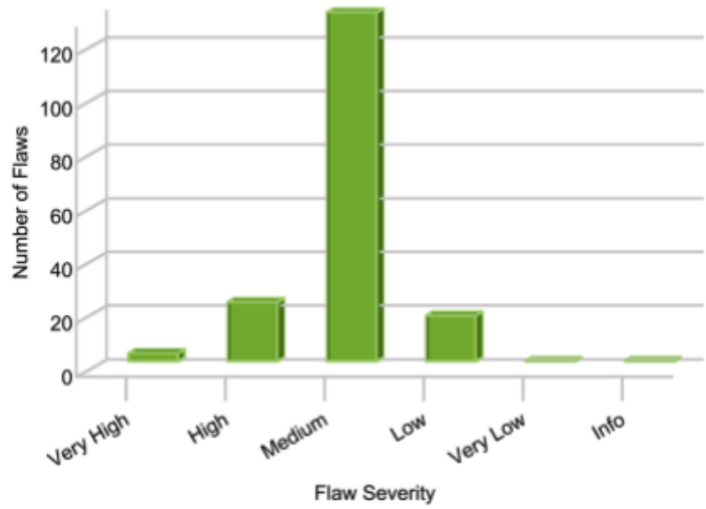
Impacts:Operational Risk (High), Financial Loss (High)

An application's business criticality is determined by business risk factors such as: reputation damage, financial loss, operational risk, sensitive information disclosure, personal safety, and legal violations. The Veracode Level and required assessment techniques are selected based on the policy assigned to the application.

Analyses Performed vs. Required

	Any	Static	Dynamic	Manual
Performed:		●	●	○
Required:	○	●	○	○

Summary of Flaws Found by Severity



Severity
Very High
High
Medium
Low
Very Low
Info

Some other rating methodologies

(STRIDE and) DREAD

STRIDE

- Method to help identify threats

DREAD

- Classification scheme for quantifying, comparing and prioritising the risk presented by each identified threat
- Calculation (score 1-3 or 1-10 for each):
  - Damage potential
  - Reproducibility
  - Exploitability
  - Affected users
  - Discoverability

Severity
High
Medium
Low

OWASP Risk Rating Methodology

Likelihood Impact

- Threat
  - S entiality
  - M ity
  - C ilability
  - S ountability
- Vulnera
  - E
  - E dam
  - A ance
  - Intrusion detection
  - Privacy violation

Description :

Likelihood factors		Impact factors	
Threat Agent Factors		*Technical Impact Factors*	
Skills required	Some technical skills [3]	Loss of confidentiality	Minimal non-sensitive data disclosed [2]
Motive	Low or no reward [1]	Loss of Integrity	Minimal slightly corrupt data [1]
Opportunity	Full access or expensive resources required [0]	Loss of Availability	Not Applicable [0]
Population Size	Anonymous Internet users [9]	Loss of Accountability	Not Applicable [0]
Vulnerability Factors		*Business Impact Factors*	
Easy of Discovery	Easy [7]	Financial damage	Damage costs less than to fix the issue [1]
Ease of Exploit	Easy [5]	Reputation damage	Not Applicable [0]
Awareness	Hidden [4]	Non-Compliance	Clear violation [5]
Intrusion Detection	Not Applicable [0]	Privacy violation	Not Applicable [0]
Calculate	Likelihood score: **Medium** Formula: 3.625 = (3+1+0+9+7+5+4+0)/8	Calculate	Impact score: **Low** Formula: 1.125 = (2+1+0+0+1+0+5+0)/8

Overall Risk Severity : Low

	Impact		
Likelihood	**-> Low**	Medium	High
Low	Note	Low	Medium
-> Medium	**-> Low <-**	Medium	High
High	Medium	High	Critical

Copy/paste THIS URL (<CTRL>+D) and all factors are selected as the current settings. This might save you time for future use.

Delicious Bookmark this page on Delicious

Feedback is welcome. Let us know at info@paradoslabs.nl.

Severity	Colour
Critical	Purple
High	Red
Medium	Orange
Low	Yellow
Note	Green

CVSS environmental and temporal groups

Environmental

- Collateral damage potential
  None, low, low-medium, medium-high, high, not defined
- Target distribution
  None, low, medium, high, not defined
- Security requirements for each of confidentiality, integrity and availability
  None, low, medium, high, not defined

Temporal

- Exploitability
  Unproven, proof of concept, functional, high, not defined
- Remediation level
  Official fix, temporary fix, workaround, unavailable, not defined
- Report confidence
  Unconfirmed, uncorroborated, confirmed, not defined

Warning: Vegetarians look away now

Why it's sometimes even more of a dog's breakfast

Chained issues

- A sequence of two or more separate weaknesses that can be closely linked together within software
- One weakness, X, can directly create the conditions that are necessary to cause another weakness Y
- Example: XSS via Shared User-Generated Content
  - SVG file type not included in banned file types
    CWE-184: Incomplete Blacklist
  - Can upload SVG files
    CWE-434: Unrestricted Upload of File with Dangerous Type
  - Malicious JavaScript code can be executed in user-uploaded SVG file
    CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

Composite issues

- A combination of two or more separate weaknesses that can create a vulnerability, but only if they all occur all the same time
- One weakness, X, can be "broken down" into component weaknesses Y and Z
- By eliminating any single component, a developer can prevent the composite from becoming exploitable
- Example: Application Worm
  - “Add a Friend” susceptible to CSRF
    CWE-352: Cross-Site Request Forgery (CSRF)
  - “Add a Friend” susceptible to Type 2 (Stored) XSS
    CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
  - “Add a Friend” usage Increases Exponentially
    CWE-799: Improper Control of Interaction Frequency

Aggregation and automation

Channel specific severity ratings

- Vendor vulnerability announcements
- Manual and automated source code analysis
- Manual and automated dynamic analysis
- Operational issue detection (e.g. web application firewalls, configuration monitoring, host intrusion detection, file integrity monitoring systems, event correlation engines, continuous and manual audit processes)
- Notification by customers/clients/citizens

- Export
  - Vulnerability findings exchange
  - Benchmarking

Counting vulnerabilities

Identity

- Per affected line of code
- Per entry form / page / screen
- Per application

Generic

- “All”
- “Every form”
- “Every page for authenticated users”

Groups

- Aggregated
- Chained

Comparison

- Consistency
- Equality

Infrastructure vs. application

- CVSS
  - Helps score vulnerabilities in deployed software
  - Repeatable
  - Scores inconsistent where
    - there is missing information
    - there is a desire to achieve a certain value
    - guidance is not followed
  - Doesn't take into account mandatory requirements
- Software
  - Many weaknesses, but not all necessarily vulnerabilities that are also exploitable
  - Scoring based on impact on the system

Compliance thresholds

"Standards"

- Corporate policies
- CWE/SANS Top 25 Top 25 Most Dangerous Software Errors
- Federal Information Processing Standard (FIPS) 200
- NIST Special Publication 800-37
- OWASP Top 10 Risks
- OWASP Application Security Verification Standard (level?)

Contractual

- PCI SSC (e.g. Data Security Standard)
- Non disclosure agreements
- Contractual clauses and SLAs

Legislation and regulations

Rating

- Binary choice?
  - Pass
  - Fail
- Not quite black and white
  - Degree of confidence
  - Coverage
- Accepted non-compliance
  - Emergency
  - Business as usual
  - Ignored

Read the label

CVSS considerations

Calculations

- Range of scores
- Application vulnerabilities even narrower range
- Environmental group
- Consider the lifecycle of a vulnerability

Presentation

- Base score
- Vector
- Colours

Interpretation

- Scoping
- Over-reliance on numerical score
- Disconnect with code weaknesses

Coming soon?

- CVSS v3

Another way? CWSS

Common Weakness Scoring System

- Scoring software application weaknesses
- Built around Common Weakness Enumeration (CWE)
- Account for incomplete information
- Three metric groups:
  - Base finding
  - Attack surface
  - Environmental group

CWSS metric groups

- Base finding
  - Technical Impact (TI)
  - Acquired Privilege (AP)
  - Acquired Privilege Layer (AL)
  - Internal Control Effectiveness (IC)
  - Finding Confidence (FC)
- Attack surface
  - Required Privilege (RP)
  - Required Privilege Layer (RL)
  - Access Vector (AV)
  - Authentication Strength (AS)
  - Authentication Instances (AI)
  - Level of Interaction (IN)
  - Deployment Scope (SC)
- Environmental
  - Business Impact (BI)
  - Likelihood of Discovery (DI)
  - Likelihood of Exploit (EX)
  - External Control Effectiveness (EC)
  - Remediation Effort (RE)
  - Prevalence (P)

CWSS metric groups comparison with CVSS

- Base finding
  - CIA impacts & security requirements and CDP
  - -
  - -
  - Access Complexity & Remediation Level
  - Report Confidence
- Attack surface
  - Access Complexity
  - Access Complexity
  - Access Vector (AV)
  - -
  - Authentication Instances (AI)
  - Access Complexity
  - Access Complexity & Target Distribution
- Environmental
  - Collateral Damage Potential
  - -
  - -
  - Access Complexity
  - -
  - Target Distribution
- Exploitability

CWRAF

Common Weakness Risk Analysis Framework (CWRAF)

- Business value context
- Technical impact scoresheets

1 -10	Modify data
1 -10	Read data
1 -10	DoS: unreliable execution
1 -10	DoS: resource consumption
1 -10	Execute unauthorised code or commands
1 -10	Gain privileges / assume identity
1 -10	Bypass protection mechanism
1 -10	Hide activities

E-commerce drivers

Requirement 6.2

PCI Security Standards Council

Requirement	Guidance
6.2 Establish a process to identify and assign a risk ranking to newly discovered security vulnerabilities. ***Notes:*** *Risk rankings should be based on industry best practices. For example, criteria for ranking "High" risk vulnerabilities may include a CVSS base score of 4.0 or above, and/or a vendor-supplied patch classified by the vendor as "critical," and/or a vulnerability affecting a critical system component.* *The ranking of vulnerabilities as defined in 6.2.a is considered a best practice until June 30, 2012, after which it becomes a requirement.*	The intention of this requirement is that organizations keep up-to-date with new vulnerabilities that may impact their environment. While it is important to monitor vendor announcements for news of vulnerabilities and patches related to their products, it is equally important to monitor common industry vulnerability news groups and mailing lists for vulnerabilities and potential workarounds that may not yet be known or resolved by the vendor. Once an organization identifies a vulnerability that could affect their environment, the risk that vulnerability poses must be evaluated and ranked. This implies that the organization has some method in place to evaluate vulnerabilities and assign risk rankings on a consistent basis. While each organization will likely have different methods for evaluating a vulnerability and assigning a risk rating based on their unique CDE, it is possible to build upon common industry accepted risk ranking systems, for example CVSS. 2.0, NIST SP 800-30, etc. Classifying the risks (for example, as "high", "medium", or "low") allows organizations to identify and address high priority risk items more quickly, and reduce the likelihood that vulnerabilities posing the greatest risk will be exploited.

Requirement 6.5.6

PCI Security Standards Council

Requirement	Guidance
6.5 Develop applications based on secure coding guidelines. Prevent common coding vulnerabilities in software development processes, to include the following: ***Note**: The vulnerabilities listed at 6.5.1 through 6.5.9 were current with industry best practices when this version of PCI DSS was published. However, as industry best practices for vulnerability management are updated (for example, the OWASP Guide, SANS CWE Top 25, CERT Secure Coding, etc.), the current best practices must be used for these requirements.*	The application layer is high-risk and may be targeted by both internal and external threats. Without proper security, cardholder data and other confidential company information can be exposed, resulting in harm to a company, its customers, and its reputation. As with all PCI DSS requirements, Requirements 6.5.1 through 6.5.5 and 6.5.7 through 6.5.9 are the *minimum* controls that should be in place. This list is composed of the most common, accepted secure coding practices at the time that this version of the PCI DSS was published. As industry accepted secure coding practices change, organizational coding practices should likewise be updated to match. The examples of secure coding resources provided (SANS, CERT, and OWASP) are suggested sources of reference and have been included for guidance only. An organization should incorporate the relevant secure coding practices as applicable to the particular technology in their environment.
6.5.6 All "High" vulnerabilities identified in the vulnerability identification process (as defined in PCI DSS Requirement 6.2). ***Note**: This requirement is considered a best practice until June 30, 2012, after which it becomes a requirement.*	Any high vulnerabilities noted per Requirement 6.2 that could affect the application should be accounted for during the development phase. For example, a vulnerability identified in a shared library or in the underlying operating system should be evaluated and addressed prior to the application being released to production.

Risk ranking of vulnerabilities

- Avoid using the terms “low”, “medium” or “high”
- Triage (for PCIDSS)
    - “bad” = affects the protection of cardholder data
    - “not bad”
    - out of scope
- Prioritise but flag all the issues that can impact on the security of the cardholder data environment

Build your own

The most important points

Clarity

- Make it understandable
- Define terminology
- Avoid numbers post calculation
- Don't get hung up on precise names
- Scoring is not that accurate, so think about fuzzy ranges
- Train users

Environment-specific

- Technical
- Business

Consistency

- Differentiation (spread of scores)
- Reproducible

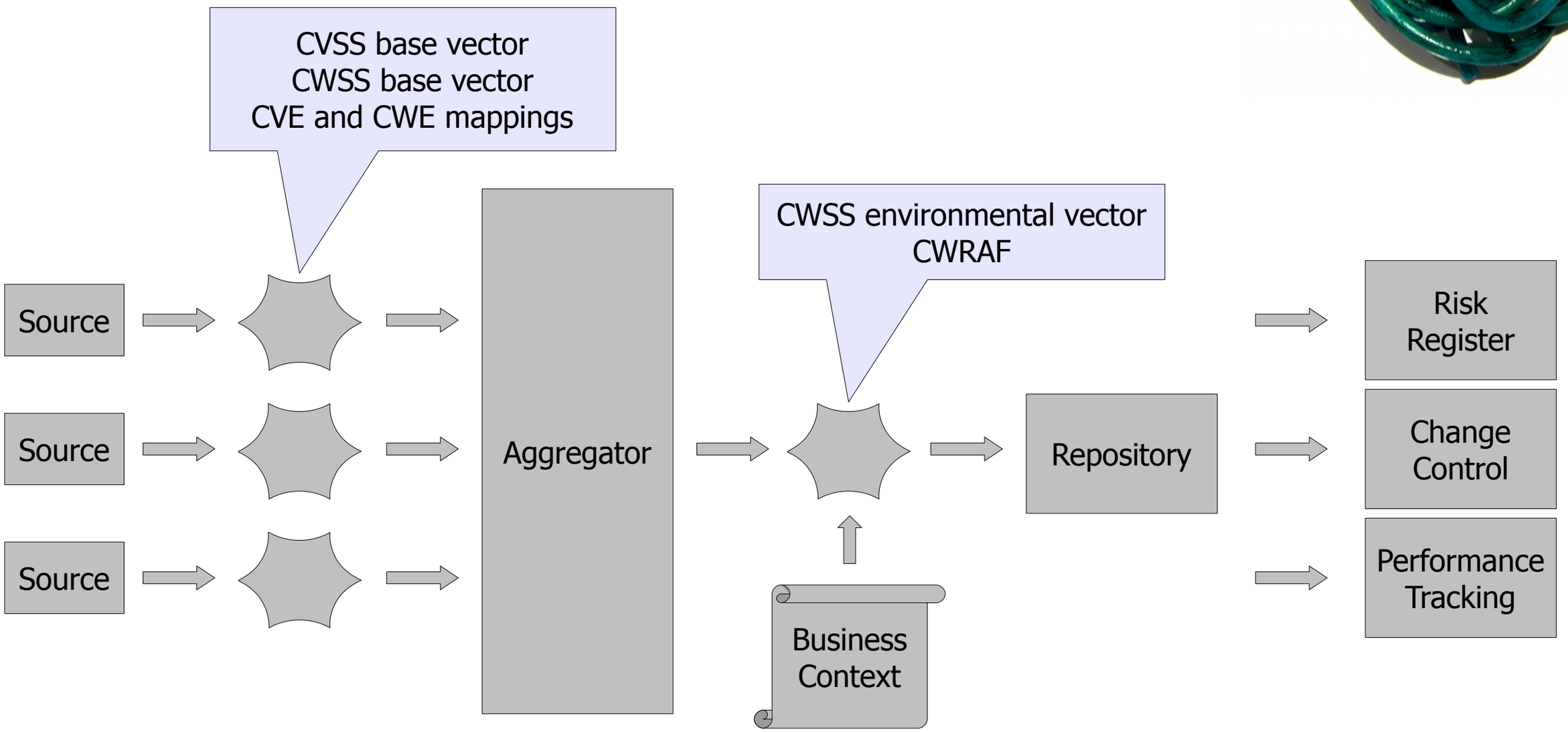
Test the scheme

- Test plan
- Edge cases
  - Unconfirmed vulnerability
  - Unexploitable vulnerability
  - Exploitable but negligible impact
  - Exploitable but extremely improbable

Prepare for / enable automation

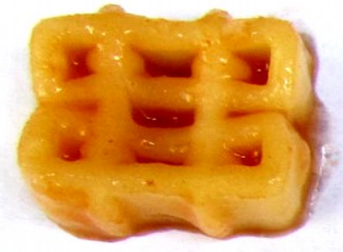
- Identification
- Interoperability
- Mappings (one to many)
- Level of confidence
- Time dependent data
- Out-of-scope results

A proposed risk assessment framework



Conclusions

Engagement with other parties



As the recipient (e.g. development manager, application owner, business manager)

- Define the objectives of the verification activity
- Discuss in advance what pass or fail means
- Understand the scoring/rating methodology being used, whether this has changed and especially what impact target is being assumed
- If CVSS is used, insist upon having both the score and the vector
- Ask for more than a generic description of the vulnerability and a severity rating

As a provider (e.g. design/code reviewer, pen test company, ASV, software analysis vendor)

- Understand the client's business and risks
- Ask the client if they have a preferred rating methodology
- Find out what the client's objectives are
- Know what threshold(s) the client will be sensitive to
- Be open about the ranking process used
- Use CWE identifiers in findings
- Consider CCSS, CMSS and CWSS too
- Provide recommendations and discuss mitigating measures and considerations

Assess yourself

1. Is "red" much worse than "orange"?
2. Why is "yellow" used instead of "green"?
3. Just what is a "critical" vulnerability?
4. Is "critical" the same as "very high"?
5. How do PCI DSS "level 4 and 5" security scanning vulnerabilities relate to application weaknesses?
6. Does a "tick" mean you passed?
7. Are you using both CWE and CVSS?
8. Is a "medium" network vulnerability as dangerous as a "medium" application vulnerability?
9. Can CWSS help?
10. Does risk ranking equate to prioritisation?

1. *Not necessarily*
2. *Green could suggest no risk*
3. *It depends on your own definition*
4. *(as above)*
5. *Your "Risk Ranking Process" created to meet PCI DSS requirement 6.2 needs to define this*
6. *Not always*
7. *Yes*
8. *It might be – it depends what you mean by "danger" – but this should be a comparison of likelihood & impact*
9. *Yes*
10. *No, but they are related*

Credits 1

Colin Watson

- colin.watson(at)owasp.org

Credits 2


HEINZ
1869
alphabetti
pasta in our classic
HEINZ
Pasta shapes
HEINZ
HEINZ
1869
numberetti
pasta in our classic
HEINZ
4Kids
Wholewheat pasta shapes in tomato sauce
SCOOBY-DOO!
No Artificial Additives
No Artificial Colours
Wholewheat Pasta
Low in Sugar & Fat
HEINZ
HELLO KITTY
Pasta shapes
in tomato sauce
1 of your 5 a day