Vulnerability Assessment

What is Vulnerability Assessment?
Process of defining, identifying, classifying and prioritizing vulnerabilities in computer systems, applications and network infrastructures and providing the organization doing the assessment with the necessary knowledge, awareness and risk background to understand the threats to its environment and react appropriately.

https://searchsecurity.techtarget.com/definition/vulnerability-assessment-vulnerability-analysis
Vulnerability Assessment

What is Vulnerability Assessment?
Process of defining, identifying, classifying and prioritizing vulnerabilities in computer systems, applications and network infrastructures and providing the organization doing the assessment with the necessary knowledge, awareness and risk background to understand the threats to its environment and react appropriately.

Why is Vulnerability Assessment Important?
To be able to respond quickly to mitigate potential threats – that is, before becoming a victim.
Recall persistent engagement.
A comprehensive vulnerability assessment program provides organizations with the knowledge, awareness, and risk background necessary to understand threats to their environment and react accordingly.

https://searchsecurity.techtarget.com/definition/vulnerability-assessment-vulnerability-analysis
Vulnerability Assessment

Vulnerability
A flaw or weakness in a system's design, implementation, operation or management that could be exploited to compromise the system's security objectives

Threat
Anything that may harm the assets owned by an application (resources of value, such as the data in a database or in the file system) by exploiting a vulnerability

Risk
Potential for loss due to some vulnerability being exploited by some threat

Test
An action to demonstrate that an application meets the security requirements of its stakeholders
Vulnerability Assessment

**Assessment is the Practice of**
Discovering vulnerabilities posed by an environment
Determining their negative risk impact
Documenting these observations for future planning.

**This may drive**
Modifications to a network or business practice to eliminate the vulnerability and reduce its exposure
Implementation of monitoring to notify in the event that an identified vulnerability is being exploited in the environment
Vulnerability Assessment

Assessment is the Practice of
  Discovering vulnerabilities posed by an environment
  Determining their negative risk impact
  Documenting these observations for future planning.

This may drive
  Modifications to a network or business practice to eliminate
  the vulnerability and reduce its exposure
  Implementation of monitoring to notify in the event that an
  identified vulnerability is being exploited in the environment

The Bottom Line
  Risk = Likelihood * Impact
Vulnerability Assessment

**OWASP**
Open Web Application Security Project
Non-profit, charitable organization (good guys)
Aim: improve security of software by making it visible so developers and organizations can make informed decisions about security risks.
[https://www.owasp.org/index.php/Main_Page](https://www.owasp.org/index.php/Main_Page)

**Products of OWASP**
Application security code of conduct for educational institutions
[https://www.owasp.org/images/6/6b/OWASP_Blue_Book-Educational_Institutions.pdf](https://www.owasp.org/images/6/6b/OWASP_Blue_Book-Educational_Institutions.pdf)

Risk Rating Methodology

OWASP Testing Guide
Vulnerability Assessment

App Security Code of Conduct for Educational Institutions

The educational institution must include application security content somewhere in the standard computer science curriculum.

The educational institution must offer at least one course dedicated to application security annually.

The educational institution must ensure that an OWASP chapter is available to their students and support it.
Vulnerability Assessment

OWASP Testing Guide

Web Application Security Testing

- Information gathering
- Configuration and deployment management testing
- Identity management testing
- Authentication testing
- Authorization testing
- Session management testing
- Input validation testing
- Testing for error handling
- Testing for weak cryptography
- Business logic testing
- Client side testing

Vulnerability Assessment

OWASP Testing Guide

**Tools** - https://www.owasp.org/index.php/Appendix_A:_Testing_Tools

- Document Object Module (DOM) XSS - https://www.owasp.org/index.php/DOM_Based_XSS
- AJAX - https://www.w3schools.com/xml/ajax_intro.asp
- Brute Force Password -
- Buffer Overflow -
- Fuzzer - https://www.owasp.org/index.php/Fuzzing

Acceptance Testing

Runtime Analysis

Vulnerability Assessment

OWASP Risk Rating Methodology

Identify a risk in the system
A risk = threat agent + attack + vulnerability + impact

Estimate likelihood the risk is realized via attack
Threat agent factors – capability of attackers
Vulnerability factors – chance it will be discovered & used

Estimate impact – loss of value to the ‘customer’
Light to severe – coarse grained

Determine severity – effort needed to mitigate and fix problem
Low, medium, high

Decide what to fix

Customize the Risk Rating Model – change cutoff points
Vulnerability Assessment

OWASP Risk Rating Methodology

Identify a risk in the system
Gather information about the/all threat agent(s)

threat agent = capabilities + intentions + past activities

Types of threat agents to consider:

- non-target specific: computer viruses, worms, trojans
- employees: staff, contractors, maintenance security
- organized crime and criminals: bank accounts, IP, CCs
- corporations: competitive intelligence
- human, unintentional: accidents, carelessness.
- human, Intentional: insider, outsider.
- natural: flood, fire, lightning, meteor, earthquakes.
Vulnerability Assessment

OWASP Risk Rating Methodology

Identify a risk in the system
Gather information about the/all threat agent(s)
Develop a template for each threat agent

https://www.owasp.org/index.php/Logic/time_bomb

Logic Bomb

Description:
malicious code that is triggered when specific conditions are met. Objective is to delay execution of the code to give it time to spread

Risk Factors:
can affect any system accessible by attacker
internal developers or contractors are threat agents

Examples:
Siberian pipeline sabotage

Related Attacks: computer viruses
Related Vulnerabilities: ...
Vulnerability Assessment

OWASP Risk Rating Methodology

Identify a risk in the system
Gather information about the attack
Develop a template for each attack

http://gauss.ececs.uc.edu/Courses/c6056/lectures/PDF/csrf.pdf
https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)

Cross-Site Request Forgery (CSRF)

Overview:
forces end user to execute unwanted actions on a web application in which it's currently authenticated

Related Security Activities:
how to review code for CSRF vulnerabilities
how to test for CSRF vulnerabilities
how to prevent CSRF vulnerabilities

Description: ...
measures that do not work: ...

Examples:

Related Attacks: XSS

Related Controls: check referred head for correct site
Vulnerability Assessment

OWASP Risk Rating Methodology

Identify a risk in the system
Gather information about the impact
Develop a template for each impact
https://www.owasp.org/index.php/Business_Impact_template

Availability
Description:
  severity: highest
Risk Factors:
  power failure
  Denial of Service attack
Examples:
Related Impacts: integrity (e.g. service)
Related Controls: uninterruptible power supply
Vulnerability Assessment

OWASP Risk Rating Methodology
Identify a risk in the system
Gather information about the vulnerability
Develop a template for each vulnerability

Buffer Overflow
Description:
  consequences: ...
  required resources: any
  severity: very high
  likelihood of exploit: high to very high

Risk Factors:
  Language design

Examples:
Related Attacks: format string attack
Related Vulnerabilities: heap buffer overflow
Related Controls: use immune language (haskell?)
Related Technical Impacts:
Vulnerability Assessment

OWASP Risk Rating Methodology

Estimate likelihood the risk is realized (via attack)

Threat agent factors
- **Skill level**: penetration, network & programming, advanced
- **Motive**: low reward? High reward?
- **Opportunity**: resources needed for TA to exploit vuln
- **Size**: government? Mob? Loner?

Vulnerability factors
- What is the likelihood vuln is discovered and exploited?
  - **Ease of discovery**: easy, difficult, practically impossible...
  - **Ease of exploit**: easy, difficult, theoretical ...
  - **Awareness**: how well known is vuln to threat agents?
  - **Detection**: how likely is it that attack can be detected?
Vulnerability Assessment

OWASP Risk Rating Methodology

Estimate impact

Technical Impact Factors

Loss of confidentiality: how sensitive is the data?
Loss of integrity: how much of the data can be damaged?
Loss of availability: how vital is this service?
Loss of accountability: are threat agents' actions traceable to an individual

Business impact

Some companies have an asset classification guide and business impact reference to formalize the potential loss

Financial damage: less than cost to fix the vulnerability?
Reputation damage: loss of major accounts?
Non-compliance: how much exposure does non-compliance introduce? high-profile violation?
Privacy violation: how much personal info can be exposed?
Vulnerability Assessment

OWASP Risk Rating Methodology

Determine severity of the risk

Likelihood and impact estimates are put together

Use a scale from 0-9 for both

Example for likelihood:

<table>
<thead>
<tr>
<th>Threat Agent Factors</th>
<th>Skill level</th>
<th>Motive</th>
<th>Opportunity</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skill level</td>
<td>5</td>
<td>2</td>
<td>7</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Impact Factors</th>
<th>Ease of Discovery</th>
<th>Ease of Exploit</th>
<th>Awareness</th>
<th>Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease of Discovery</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>2</td>
</tr>
</tbody>
</table>

Overall likelihood: 4.375 (average of all the numbers)
Vulnerability Assessment

OWASP Risk Rating Methodology
Determine severity of the risk
Likelihood and impact estimates are put together
Use a scale from 0-9 for both
Example for impact:

<table>
<thead>
<tr>
<th>Technical Impact Factors (loss of)</th>
<th>Confidentiality</th>
<th>Integrity</th>
<th>Availability</th>
<th>Accountability</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>7</td>
<td>5</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

Overall technical impact: 7.25 (average of all the numbers)

<table>
<thead>
<tr>
<th>Business Impact Factors</th>
<th>Financial Damage</th>
<th>Reputation Damage</th>
<th>Non-compliance</th>
<th>Privacy Violation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

Overall business impact: 2.25 (average of all the numbers)
Vulnerability Assessment

OWASP Risk Rating Methodology

Determine severity of the risk
Likelihood and impact estimates are put together
Let 0-2 be 'Low', 3-5 be 'Medium', 6-9 be 'High'
Calculation of Risk Severity:

<table>
<thead>
<tr>
<th>Impact</th>
<th>Overall Risk Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td>High</td>
</tr>
<tr>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td>High</td>
</tr>
<tr>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td>High</td>
</tr>
<tr>
<td>Worry?</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td>High</td>
</tr>
</tbody>
</table>

Worry? Means maybe it is not worth taking care of the risk
Vulnerability Assessment

OWASP Risk Rating Methodology

Decide what to fix

Prioritize risks based on risk analysis
Assign $ costs to repair and loss for each risk
Find a point below which there is no net gain to fixing the vulnerabilities – consider fixing those above
Or find the highest point in the list above which the cost to fix vulnerabilities is greater than the funds available to do so – consider fixing those one above that point
Vulnerability Assessment

OWASP Risk Rating Methodology

Customize the Risk Rating Model

The model must be tailored to the organization

Add factors:
  e.g. military org may add a casualty impact factor
  encryption algorithm strength as a likelihood factor

Customize options:
  e.g. financial damage may be different for different
  departments, so one can assign different ratings
  accordingly

Weight the factors:
  In the above, all factors have the same weight
Vulnerability Assessment

Microsoft Risk Rating Methodology

Microsoft Risk Rating Methodology

Identify and Define Security Objectives

**Identity**: does the application protect user identity from abuse are there adequate controls to ensure evidence of identity

**Financial**: what cost is org willing to absorb in remediation

**Reputation**: loss incurred for misuse or attack success

**Privacy and Regulatory**: to what extent does the application have to protect associated data. e.g. tax prep software is subject to privacy laws in most countries

**Availability guarantees**: is the application subject to availability guarantees per service level agreement, e.g?

**ISO IEC 27000 standards met?**

http://www.praxiom.com/iso-27002.htm

https://www.sans.org/security-resources/policies
Microsoft Risk Rating Methodology

Application Overview

Identify components, data flows, trust

**Trust Boundaries:** because data flows across boundaries must be carefully analyzed, not so inside same boundary

Boundary example: cloud service consumer trusts cloud service provider
Microsoft Risk Rating Methodology

Application Overview

Trust Boundaries: because data flows across boundaries must be carefully analyzed, not so inside same boundary.

Each part of an application has a role, and you cannot cross a trust boundary without some form of validation.

https://www.owasp.org/index.php/Input_Validation_Cheat_Sheet
Decompose Application

decompose application architecture to identify features and modules with a security impact that need to be evaluated.

**Example:** when investigating the authentication module, it is necessary to understand
- how data enters the module
- how the module validates and processes the data
- where the data flows
- how the data is stored
- what fundamental decisions and assumptions are made by the module.
Microsoft Risk Rating Methodology

Identify Threats

Use a threat graph

Vulnerability Assessment

Attacker may be able to read other users' messages

User may not have logged off on a shared computer

Data validation may fail, allowing SQL injection

Authorization may fail, allowing unauthorized access

Browser cache may contain contents of message

Implement data validation

Implement authorization checks

Implement anti-caching HTTP headers

If risk is high, use SSL
Microsoft Risk Rating Methodology

Identify Threats

Understand who might attack the application

**Accidental Discovery:** ordinary user stumbles across a functional mistake using a web browser, gains access to privileged information or functionality.

**Automated Malware:** programs or scripts, search for known vulnerabilities, then report back to collection site

**Curious Attacker:** security researcher or user notices something wrong and explores further

**Script Kiddies:** common renegades seek to compromise or deface applications for collateral gain, notoriety, or political agenda

**Motivated Attacker:** disgruntled staff member with inside knowledge or paid professional attacker.

**Organized Crime:** criminals crack e-commerce or corporate banking applications for financial gain.
Microsoft Risk Rating Methodology


Classification scheme for characterizing known threats according to the kinds of exploit that are used or motivation of the attacker

**Spoofing Identity:** a user should not be able to become another user or assume attributes of another user

**Tampering with Data:** data obtained only from within the app should not be sent to a user. App should check data received from a user for sanity before storing it

**Repudiation:** app should employ audit trails when needed to prevent a user from making a false claim about the results of the application: e.g. “I transferred the money”

**Information Disclosure:** app minimizes info sent to user

**Denial of Service:** app design should prevent DoS possibility

**Elevation of Privilege:** only permitted roles can access privileged functionality
Vulnerability Assessment

Microsoft Risk Rating Methodology

DREAD

Classification scheme for quantifying, comparing and prioritizing the amount of risk presented by each evaluated threat

Risk = (damage+reproducibility+exploitability+affected-users+Discoverability)/5

Damage Potential: 0-10 (0=nothing, 10=complete destruction)

Reproduce Threat: 0-10 (0=can't, 10=easily)

Exploitability: 0-10 (0=difficult, 10=easy)

Affected Users: 0-10 (0=none, 10=everyone)

Discoverability: 0-10 (0=impossible, 10=right there in browser)

Vulnerability Assessment

Performing threat modeling provides a far greater return than most any other control

Other threat modeling approaches:
https://www.owasp.org/index.php/Threat_Risk_Modeling
Vulnerability Classification

Recall Vulnerability Factors to Determine Risk:
What is the likelihood vuln is discovered and exploited?
   Ease of discovery: easy, difficult, practically impossible...
   Ease of exploit: easy, difficult, theoretical ...
   Awareness: how well known is vuln to threat agents?
   Detection: how likely is it that attack can be detected?

Grasping these for risk analysis is assisted with the use of a vulnerability classification scheme
   Simple example: OWASP top 10 Privacy Risks
   http://gauss.ececs.uc.edu/Courses/c6056/extra/OWASP-priv-risks.pdf

Lots: https://cwe.mitre.org/about/sources.html
# Vulnerability Classification

<table>
<thead>
<tr>
<th>No.</th>
<th>Title</th>
<th>Frequency</th>
<th>Impact</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Web Application Vulnerabilities</td>
<td>High</td>
<td>Very high</td>
<td>Vulnerability is a key problem in any system that guards or operates on sensitive user data. Failure to suitably design and implement an application, detect a problem or promptly apply a fix (patch) is likely to result in a privacy breach. This risk also encompasses the OWASP Top 10 List of web application vulnerabilities and the risks resulting from them.</td>
</tr>
<tr>
<td>P2</td>
<td>Operator-sided Data Leakage</td>
<td>High</td>
<td>Very high</td>
<td>Failure to prevent the leakage of any information containing or related to user data, or the data itself, to any unauthorized party resulting in loss of data confidentiality. Introduced either due to intentional malicious breach or unintentional mistake e.g. caused by insufficient access management controls, insecure storage, duplication of data or a lack of awareness.</td>
</tr>
<tr>
<td>P3</td>
<td>Insufficient Data Breach Response</td>
<td>High</td>
<td>Very high</td>
<td>Not informing the affected persons (data subjects) about a possible breach or data leak, resulting either from intentional or unintentional events; failure to remedy the situation by fixing the cause; not attempting to limit the leaks.</td>
</tr>
<tr>
<td></td>
<td>Insufficient Relation</td>
<td></td>
<td></td>
<td>Failure to effectively and/or timely delete personal</td>
</tr>
</tbody>
</table>
# Vulnerability Classification

**Another example:**

<table>
<thead>
<tr>
<th>OWASP Top 10 2007</th>
<th>OWASP Top 10 2004</th>
<th>SANS CWE/25</th>
<th>WASC 24(+2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CWE-78: Improper Sanitization of special elements used in an OS Command</td>
<td>4.4 OS Commanding</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CWE-94: Failure to Control Generation of Code ('Code Injection')</td>
<td>4.6 SSI Injection</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.7 XPath Injection</td>
</tr>
<tr>
<td>A8. Insecure Cryptographic Storage</td>
<td>A8. Insecure Storage</td>
<td>CWE-327 Use of a Broken or Risky Cryptographic Algorithm</td>
<td></td>
</tr>
<tr>
<td>A5. Cross Site Request Forgery (CSRF)</td>
<td></td>
<td>CWE-352 Cross-Site Request Forgery (CSRF)</td>
<td>1.4 CSRF *</td>
</tr>
<tr>
<td>A6. Information Leakage and Improper Error Handling</td>
<td>A7. Improper Error Handling</td>
<td>CWE-209 Error message Information leak</td>
<td>5.2 Information leakage</td>
</tr>
</tbody>
</table>
Vulnerability Classification

What do we want to get out of a classification?

Is the vulnerability due to an OS problem or app
Is (can) the vulnerability (be) intentionally implanted
Is the vulnerability in hardware or software – which part
Is the vulnerability caused by network protocol weakness
At what part in the system design cycle can it be caught
Is the vulnerability triggered by social engineering
Is the vulnerability due to a configuration error
Is the vulnerability due to a language design problem
How does the vulnerability impact business
How can the vulnerability be found and isolated

https://cwe.mitre.org/about/sources.html
Vulnerability Classification

What are some (popular) types of classification?

Software Development Life Cycle (SDLC)

Development life cycle:
- feasibility study, requirements, design, implementation, integration, testing, operations and maintenance

Classification:
- on when the vulnerability appears in the develop life cycle analysis, design, implementation, deployment, maintenance

Usefulness:
- timing of code reviews could be set according to when the vulnerability in question appears

Notes:
- Vulnerability may be due to bad implementation
- If due to algorithm it will show up in the design phase
- Vulns introduced in operations are configuration probs
- Vulns in maintenance phase may be due to bug fixes
Vulnerability Classification

What are some (popular) types of classification?

Classification by Genesis

Classes: Inadvertent, Intentional flaws
Subclasses: malicious, not malicious
Sub-subclasses: …

Problem:
no known procedure to class a flaw according to above

Usefulness:
can state which scanning tools may be used to catch the flaw

http://gauss.ececs.uc.edu/Courses/c6056/pdf/weber.pdf
Vulnerability Classification

What are some (popular) types of classification?

Location in Object Modules
Classes: which object module a vulnerability belongs to

Problem:
vulnerability could depend on interactions between modules – OK, so add some more classes

Example: networking layers are the modules
Recall layers: physical, data link, IP, TCP, application
Data link switch: segments network, provides bridges
IP layer switch: provide fast IP routing
Suppose: router can do IP to data-link layer broadcast
Result: SMURF attack – spoofed pings are broadcast, get multiplied by router, traffic becomes too great (DoS)

Usefulness:
browsers should limit security risks posed by vulnerable plugins and malicious scripts
Vulnerability Classification

What are some (popular) types of classification?

Affected Technologies

**Classes:** design limitations in technologies that create difficult or tricky situations for developers

**Examples:**
- C language
  - Format string vulnerabilities
    - write data to arbitrary locations via %n
  - Polyvariadic functions
    - functions have no way of knowing # arguments
- Meta character vulnerabilities
  - SQL injection
  - Characters with syntactic meaning
    - ';' used to separate commands
Vulnerability Classification

What are some (popular) types of classification?

Resource Exhaustion

**Classes:** limited computer resources that can be “hogged”

**Examples:**
- SYN-flood attacks
- memory leaks
- attacks on algorithm complexity - DoS
  - Billion Laughs attack on XML parsers
def 10 entities as consisting of 10 of previous entities
document has single instance of largest entity
which expands to one billion copies of the first one
the first one is often the string “lol”
Vulnerability Classification

What are some (popular) types of classification?

Errors or Mistakes

Classes: errors known to have led to vulnerabilities

Examples:
- failure to store and protect data securely
- use of weak password based system
- free memory more than once

Usefulness:
- indicate the nature of the impact of the error
- type of change needed to fix or eliminate the error
- the cause of errors
Vulnerability Classification

What are some (popular) types of classification?

Enabled Attack Scenario

**Classes:** sets of vulnerabilities that enable a particular kind of attack

**Example:**

XSS vulnerabilities enable injection of scripting code into content served to web browsers

these enable other attacks and this one category is a good place to start looking at these (as the cause)

**Note:**

DoS would not be such a class because it would be the result of an enabled attack that could be achieved in many different ways
Vulnerability Classification

What are some (popular) types of classification?

Network Protocol Vulnerabilities

**Feature:** simultaneously offers an attack taxonomy and countermeasures (Pothamsetty and Akyol)

**Classes – defined in terms of attacks they enable (below):**
- clear text communication
- non-robust protocol state handling
- insecure protocol state handling
- inability to handle abnormal packet rates
- vulnerabilities arising from replay and reuse
- protocol field authentication
- entropy problems

**Test or attack techniques:**
- packet sniffing, packet flooding, replay, reuse, comm initiation, protocol field fuzzing, encryption and random # check, protocol field spoofing, comm termination, info retrieval, special and reserved packets, packet size/number variations...
Vulnerability Classification

What are some (popular) types of classification?

Seven Kingdoms

Classes: software security errors

Top Level Classes:
input validation and representation
API abuse
security features
time and state
ero error handling
code quality
encapsulation
configuration

Note:
classification includes causes, consequences, bad practices

Usefulness:
provides rules for code scanning software
security education – conveys secure programming concepts

https://cwe.mitre.org/about/sources.html
Vulnerability Classification

What are some (popular) types of classification?

System Vulnerabilities, Configuration Issues

**Note:** a vulnerability in a system under one security policy may not be a vulnerability under a different security policy.

**Example:** the system may be fine in a restricted environment but vulnerable when restrictions are out.

**Note:** a system vulnerability may exist even if all components have no vulnerabilities!

**Example:** use RSA for encryption and authentication but if the same key is used for both there will be a problem.

**Note:** several legitimate acts may be combined to create a security policy violation.

**Example:** sending a message is legitimate but sending too many (SPAM) is abusive.