FANSHAWE

INFO-6076

Web Security

Classification & Prioritization

Agenda

- Classification & Prioritization
  - CWE
  - CVE
  - CVSS
  - STRIDE
- OWASP
- Defense Approaches
- Lab 04 Overview

Classification & Prioritization

Classification

There are more threat vectors than time or money to protect against them

- Tools like the OWASP Top Ten help with this

Other threat classification and ranking systems:

- STRIDE
  - useful for the classification of general threats
- CWE (Common Weakness Enumeration)
  - useful for the classification of specific threats
- CVE (Common Vulnerabilities and Exposures)
  - useful for the classification of specific instances of a CWE
- CVSS (Common Vulnerability Scoring System)
  - useful for ranking threats

CWE

CWE: Common Weakness Enumeration

- A generic flaw that can lead to a unique vulnerability or exposure
- Formal list of software weaknesses
- This is more of a general classification

CVE

CVE: Common Vulnerabilities and Exposures

- A unique instance of a weakness (flaw) that can be used to access a system or network
- Provides unique identifiers for publicly known information security vulnerabilities
- Each CVE contains
  - CVE Identifier Number
  - Brief description of the security vulnerability or exposure
  - Pertinent references

Vulnerabilities

- To be considered a Vulnerability it must:
  - Allow an attacker to execute a command as another user
  - Allow an attacker access to data that is contrary to the specified access restrictions
  - Allow an attacker to pose as another entry
  - Allow an attacker to conduct a DoS attack

Exposures

- An exposure is a configuration issue or mistake in software that allows access to information or capabilities that can be used by a hacker as a stepping-stone into a system or network
  - Doesn’t directly allow compromise, but could be an important component of an attack
  - Exposures can be considered violations of a reasonable security policy
  - Is a primary point of entry that an attacker may attempt to use to gain access to the system or data
  - Allows attacker to conduct information gathering activities
  - Allows an attacker to hide their activities

CPE

CPE: Common Platform Enumeration

- Maintained by NIST, National Institute of Standards and Technology
- Structured naming scheme for information technology systems, software and packages
- Allows researchers to know that they are talking about the same platform

```
cpe:<cpe_version>:<part>:<vendor>:<product>:<version>:<update>:
<edition>:<language>:<sw_edition>:<target_sw>:<target_hw>:<other>
```

https://en.wikipedia.org/wiki/Common_Platform_Enumeration

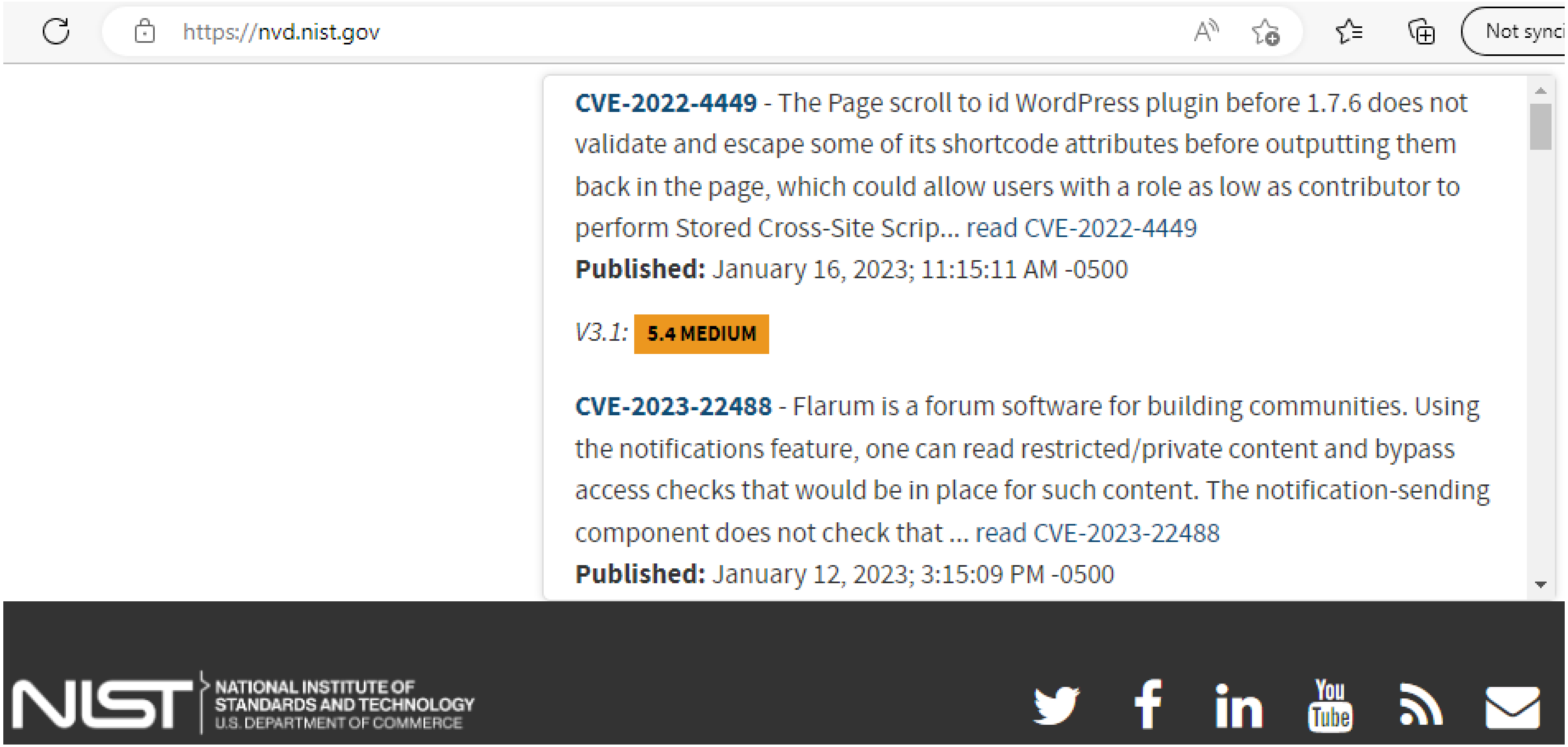
CWE, CVE Relationship

- A CWE will have many CVEs
- CVEs relate to a specific vulnerability under the same CWE umbrella
- CPEs are there to ensure the correct platform is listed in the CVE information

Resources:

- https://cve.mitre.org/
- https://nvd.nist.gov

CVE Example



CVE-2022-4449 - The Page scroll to id WordPress plugin before 1.7.6 does not validate and escape some of its shortcode attributes before outputting them back in the page, which could allow users with a role as low as contributor to perform Stored Cross-Site Scrip... read CVE-2022-4449

Published: January 16, 2023; 11:15:11 AM -0500

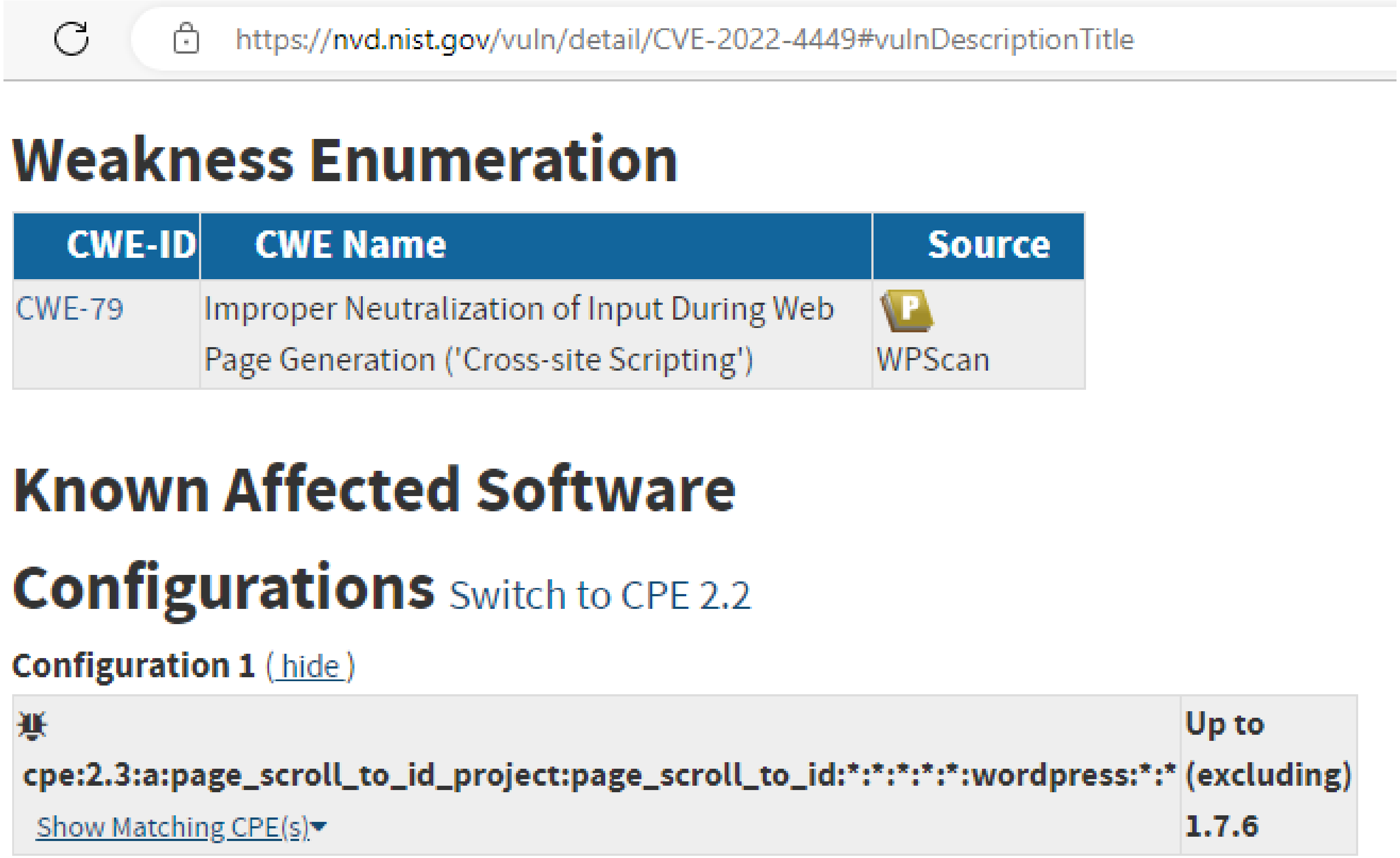
V3.1: **5.4 MEDIUM**

CVE-2023-22488 - Flarum is a forum software for building communities. Using the notifications feature, one can read restricted/private content and bypass access checks that would be in place for such content. The notification-sending component does not check that ... read CVE-2023-22488

Published: January 12, 2023; 3:15:09 PM -0500

NIST
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE
You Tube

CVE Example



https://nvd.nist.gov/vuln/detail/CVE-2022-4449#vulnDescriptionTitle

Weakness Enumeration

CWE-ID	CWE Name	Source
CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	P WPScan

Known Affected Software Configurations

Switch to CPE 2.2

Configuration 1 (hide)

cpe:2.3:a:page_scroll_to_id_project:page_scroll_to_id:*:*:*:*:*:wordpress:*:* Show Matching CPE(s)▾	Up to (excluding) 1.7.6

Classification & Prioritization of Threats

A **Threat Model** is a view of the application and its environment through security glasses

There are many Threat Models: STRIDE, PASTA, OCTAVE, CVSS, etc.

We focus on:

- STRIDE
  - Useful for the classification of general threats
- CVSS (Common Vulnerability Scoring System)
  - Useful for ranking threats

STRIDE

STRIDE is another example of a threat classification system originally developed by Microsoft

- **S**poofing
- **T**ampering
- **R**epudiation
- **I**nformation Disclosure
- **D**enial of Service
- **E**levation of Privilege

https://www.owasp.org/index.php/Threat_Risk_Modeling#STRIDE

STRIDE

Spoofing Vulnerabilities

- Allows an attacker to impersonate another user

Tampering Vulnerabilities

- Involves an attacker changing data they shouldn’t have access to

Repudiation Vulnerabilities

- Allows the attacker to deny they performed a given action
- Who did the damage?

STRIDE

Information Disclosure Vulnerabilities

- Involves an attacker being able to read data they shouldn’t have access to

Denial of Service Attack Vulnerabilities

- Prevents valid users from accessing the application

Elevation of Privilege Vulnerabilities

- Allows attackers to perform actions they shouldn’t be able to perform
- Actions with higher privileges, such as those of an administrator

STRIDE Threat Model: SQL Injection

Threat Type	SQL Injection Example
Spoofing	•Retrieve and use another user’s credentials •Modify Author value for messages
Tampering	•Modify product stock information •Change any other data in the database
Repudiation	•Delete transaction records •Delete database event logs
Information disclosure	•Obtain saved credit card numbers •Gain insight into internal design of app
Denial of service	•Run resource-intensive SQL queries •Kill sqlservr.exe process
Elevation of privilege	•Retrieve and use administrator credentials •Run shell commands

CVSS: Common Vulnerability Scoring System

- Current version 3.1 is maintained by **FIRST**:
  (Forum of Incident Response and Security Teams)

- Ranks vulnerabilities on a scale of 1 to 10, ten being the highest risk

Severity Ratings:

- None (0)
- Low (0.1-3.9)
- Medium (4.0-6.9)
- High (7.0-8.9)
- Critical (9.0-10.0)

CVSS: Common Vulnerability Scoring System

Uses three main factors to determine score:

Base Score: inherent characteristics of vulnerability

Temporal Score: characteristics that change over time (new exploits, mitigation available)

Environmental Score: characteristics specific to your organization (use of SQL databases)

https://owasp.org/

OWASP

Open Web Application Security Project

- Non-for-profit charitable organization

OWASP Top Ten Project

- Identifies the top 10 most critical web application security risks at the time of release

http://www.owasp.org

Web Application Vulnerabilities

Web applications are uniquely vulnerable

- It is estimated that up to 70 percent of attacks come through web applications
- This stems from the fact that user traffic needs to pass through the firewall to the web application

Firewalls alone are an ineffective defense for attacks against web applications

- Unfortunately, most companies spend much more resources on network defense, than on building or configuring their web applications properly

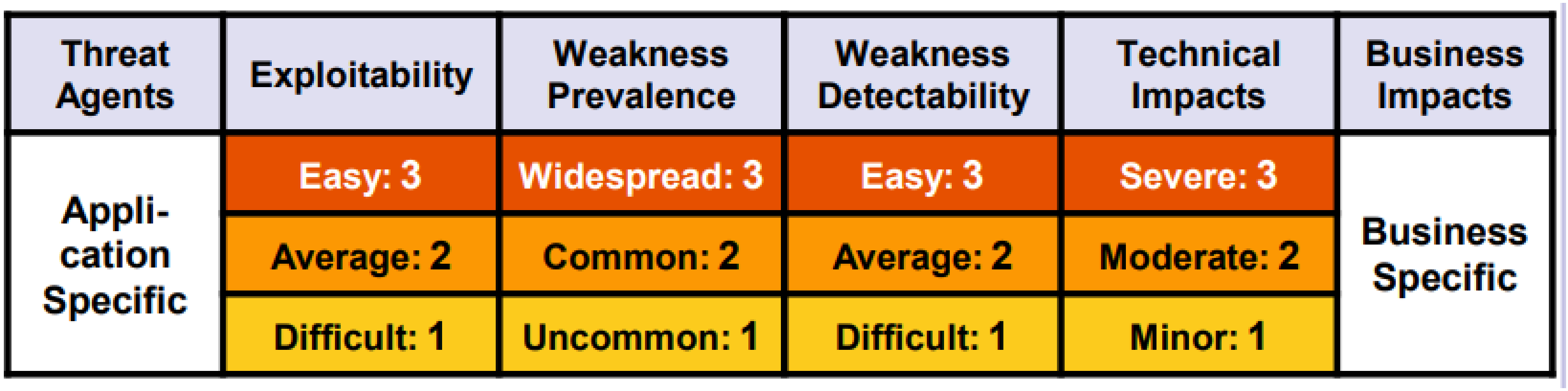
Web Application Security Risks

Attacker can use many paths through a web application to harm an organization

- Each of these paths represents a risk

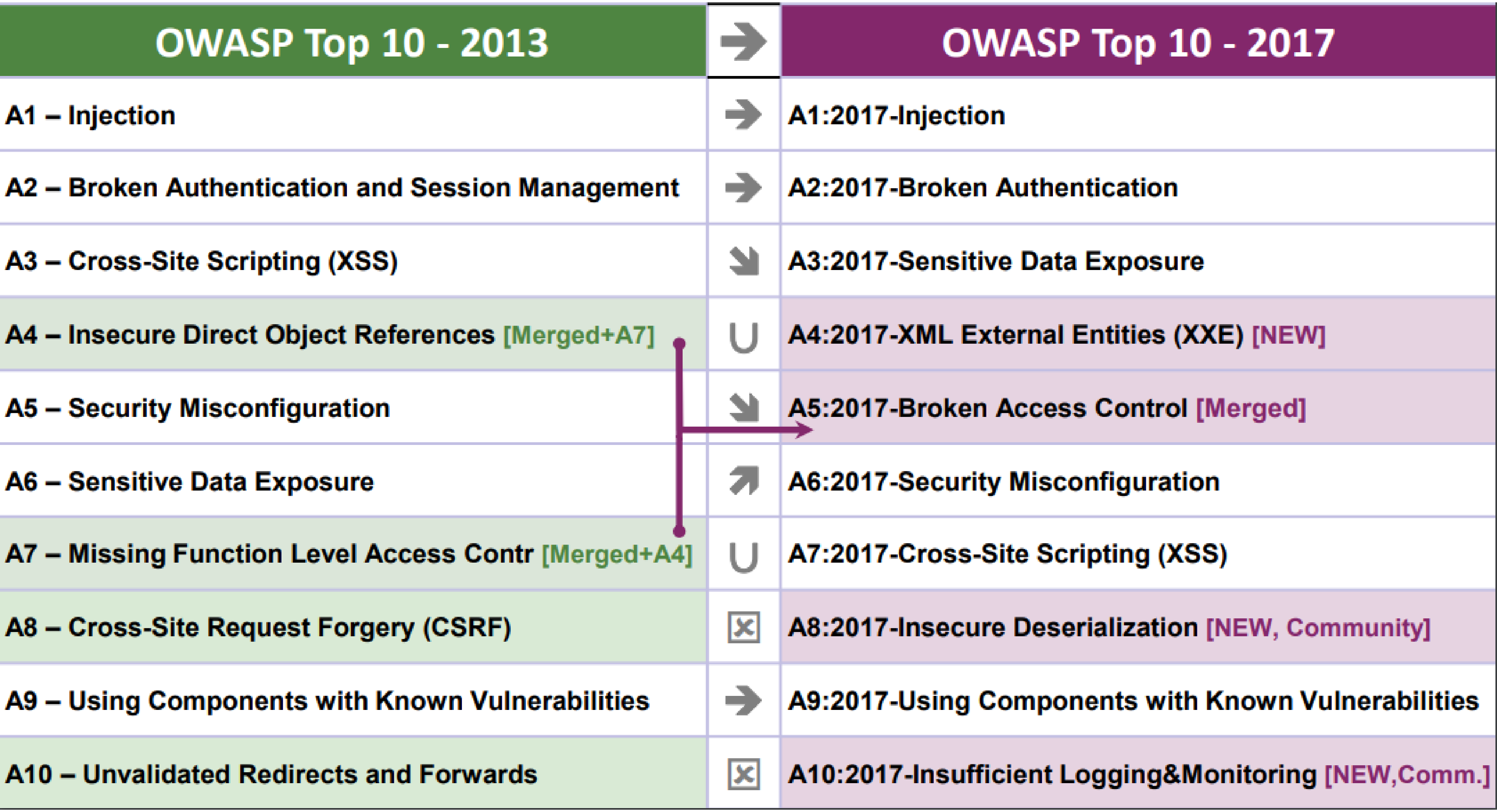
The OWASP top 10 project attempts to identify the most dangerous risks

- Serious enough to warrant attention



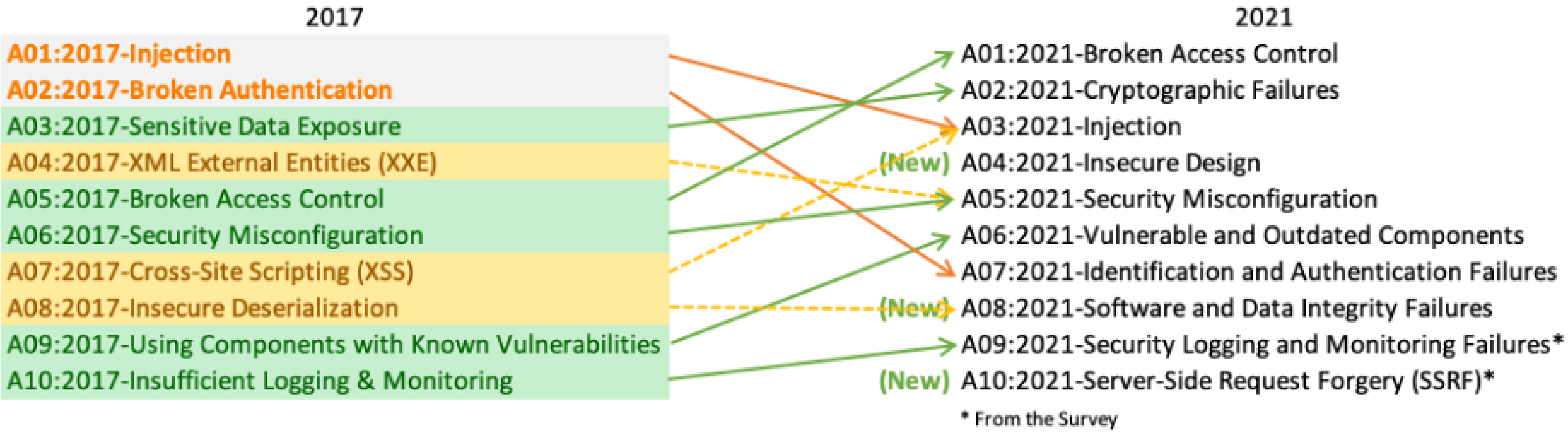
Threat Agents	Exploitability	Weakness Prevalence	Weakness Detectability	Technical Impacts	Business Impacts
Application Specific	Easy: 3	Widespread: 3	Easy: 3	Severe: 3	Business Specific
	Average: 2	Common: 2	Average: 2	Moderate: 2	
	Difficult: 1	Uncommon: 1	Difficult: 1	Minor: 1	

OWASP Top 10 - 2013/2017



OWASP Top 10 - 2013	→	OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	∪	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	∪	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

OWASP Top 10 - 2017/2021



2017

A01:2017-Injection
A02:2017-Broken Authentication
A03:2017-Sensitive Data Exposure
A04:2017-XML External Entities (XXE)
A05:2017-Broken Access Control
A06:2017-Security Misconfiguration
A07:2017-Cross-Site Scripting (XSS)
A08:2017-Insecure Deserialization
A09:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging & Monitoring

2021

A01:2021-Broken Access Control
A02:2021-Cryptographic Failures
A03:2021-Injection
(New) A04:2021-Insecure Design
A05:2021-Security Misconfiguration
A06:2021-Vulnerable and Outdated Components
A07:2021-Identification and Authentication Failures
(New) A08:2021-Software and Data Integrity Failures
A09:2021-Security Logging and Monitoring Failures*
(New) A10:2021-Server-Side Request Forgery (SSRF)*

* From the Survey

The latest list was established in 2021

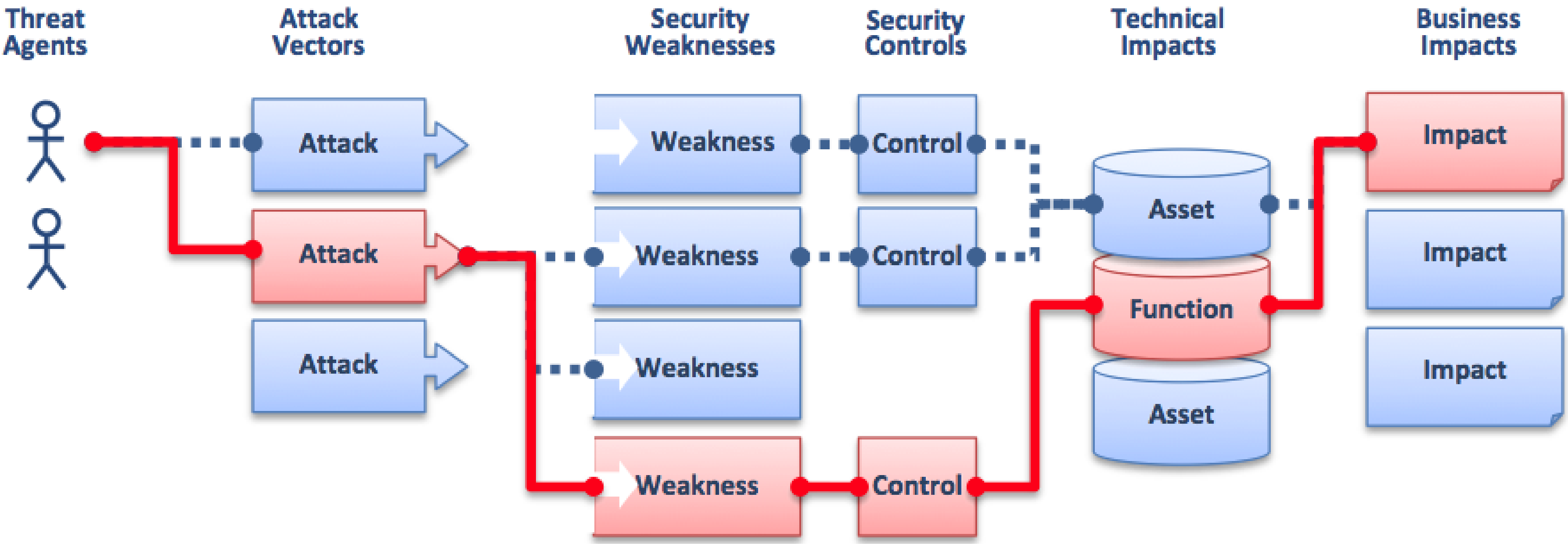
Source: https://owasp.org/www-project-top-ten/

Web Application Security Risks

Each risk has detailed information

- **Threat Agents**
  - Where will these attacks originate?
- **Exploitability**
  - How easy is it to perform the attack?
- **Weakness Prevalence**
  - How Common is the weakness?
- **Weakness Detectability**
  - How easy is it to detect the weakness?

Web Application Security Risks



By Neil Smithline - http://www.owasp.org/index.php/File:2010-T10-ArchitectureDiagram.png, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=12312894

Threat Agents

Threat Agents

How technically skilled is this group of threat agents?

- Security penetration skills
- Network and programming skills
- Advanced computer user
- Some technical skills
- No technical skills

Threat Agents

How motivated is this group of threat agents to find and exploit this vulnerability?

- Low or no reward
- Possible reward
- High reward

Threat Agents

What resources and opportunities are required for this group of threat agents to find and exploit this vulnerability?

- Full access or expensive resources required
- Special access or resources required
- Some access or resources required
- No access or resources required

Threat Agents

How large is this group of threat agents?

- Developers
- System administrators
- Intranet users
- Partners
- Authenticated users
- Anonymous Internet users

Exploitability

Exploit Discovery

How easy is it for a group of threat agents to discover this vulnerability?

- Practically impossible
- Difficult
- Easy
- Automated tools available

Ease of Exploit

How easy is it for a group of threat agents to actively exploit this vulnerability?

- Theoretical
- Difficult
- Easy
- Automated tools available

Ease of Exploit

How well known is this vulnerability to this group of threat agents?

- Unknown
- Hidden
- Obvious
- Public knowledge

Exploit Detection

How likely is an exploit to be detected?

- Active detection in application
- Logged and reviewed
- Logged without review
- Not logged

Web Application Security Risks

Technical Impacts

- How severe will the attack be on the infrastructure?

Business Impacts

- What will be the varied costs to the business if a successful attack takes place?

Technical Impacts

Loss of confidentiality

- How much data could be disclosed and how sensitive is it?

Loss of integrity

- How much data could be corrupted and how damaged is it?

Loss of availability

- How much service could be lost and how vital is it?

Loss of accountability

- Are the threat agents' actions traceable to an individual?

Business Impacts

Financial damage

- How much financial damage will result from an exploit?

Reputation damage

- Would an exploit result in reputation damage that would harm the business?

Non-compliance

- How much exposure does non-compliance introduce?

Privacy violation

- How much personally identifiable information could be disclosed?

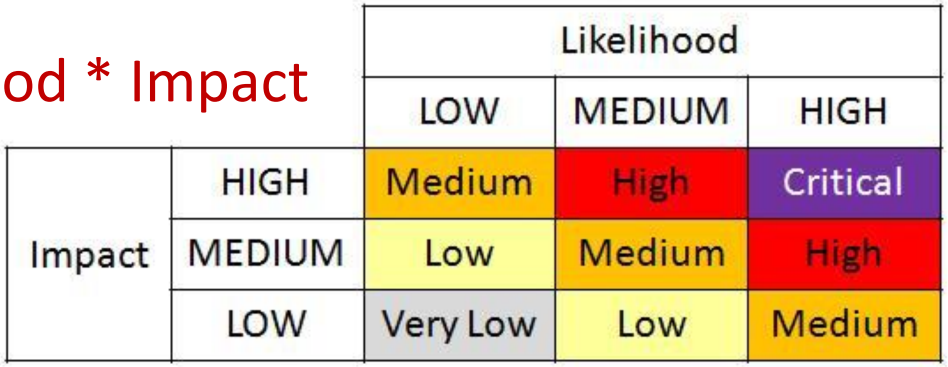
The OWASP Risk Rating Methodology

Vulnerability that is critical to one organization may not be very important to another.

OWASP Risk Rating Methodology is a basic framework that should be *customized* for the particular organization.

Threat agent factors likelihood of a successful attack				Vulnerability factors likelihood of the particular vulnerability			
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)							

Risk Severity = Likelihood * Impact



		Likelihood		
		LOW	MEDIUM	HIGH
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Very Low	Low	Medium

How skills and exploits affect the risk?

The more skills required the less the risk.

The more exploits available, the greater the risk.

Defense Approaches

Defense Approaches

There are three primary defense approaches when it comes to most web application security issues:

Input Validation

- Never trust the user

Access / Attack Surface Reduction

- Don’t give users access to functionality they don’t need, or even better, don’t enable functionality that isn’t needed

Classification and Prioritization of Threats

- Know which risks are most relevant to your organization of focus your attention on them

Input Validation

There are two primary types of input validation:

Blacklist Validation

- Involves listing out all the input that should not come from a user, then blocking it

Whitelist Validation

- Involves listing out the input that should come from a user, then allowing it

Challenges

Difficulties with blacklist validation:

- It is extremely difficult to anticipate everything that should be blocked
- This is especially true when you take character encoding into account
  - All the following inputs reference the same page:
    - my page.html
    - My Page.html
    - MY PAGE.html
    - my%20PAGE.html

Challenges

Difficulties with whitelist validation:

- You need to make sure you have whitelisted any potentially valid inputs
- Not all valid inputs are easy to define
    - Usernames, email addresses, etc.
- Regular expressions can be used to handle more complicated input validation
    - Can be difficult to write
    - You can use tools such as Regex Buddy or Regex Magic

Attack Surface Reduction

Involves controlling the code and functionality users can access

- If a user doesn’t need access to a feature don’t give it to them
- You can allow users to opt into additional functionality as they need it

A non web application example of this would be current versions of Windows Server

- You add roles and features as needed

Logging and Detection

Is there a central log server?

Do the logs get reviewed for suspicious activity?

- Network Intrusion Detection Systems
  - Firewall / Network Security Appliance
- Host-Based Intrusion Detection Systems
  - OSSEC

Are logs kept for a minimum of 90 days?

Lab Overview

LAB-04: Overview

Lab-04: OWASP Top Ten Attacks

- IIS 10.0 & FTP Role set up on Windows Server 2016
- OWASP Juice Shop Installation
- Burp Suite Setup
- Directory Traversal
- XML External Entity Injection