Lack of Understanding

- Hacking is seen as a black art
- There's a false sense of security behind firewalls, HTTPS and security controls
- But Its only a matter of time. No company is immune from a breach
- Security should be viewed as an investment with companies adopting a defense in depth approach
- Security should be not about building a wall but more like having an immune system in place to deal with todays threats

Skills Shortage

- What's Expected
  - Understand security architecture, design and development
  - Understand security controls & regulatory requirements (PCI, HIPPA etc.)
  - Perform penetration testing, Incident investigation and reporting of security vulnerabilities
  - Perform static and dynamic security scans
  - Document security controls, systems and develop tooling
  - Regular reporting and status updates to management
  - Provide secure development guidelines (education, awareness and tooling)
  - Identify, prioritise and manage security risks
  - Communicate concerns to management and contribute to a mitigation plan
  - Stay current with security trends, regulations and security standards

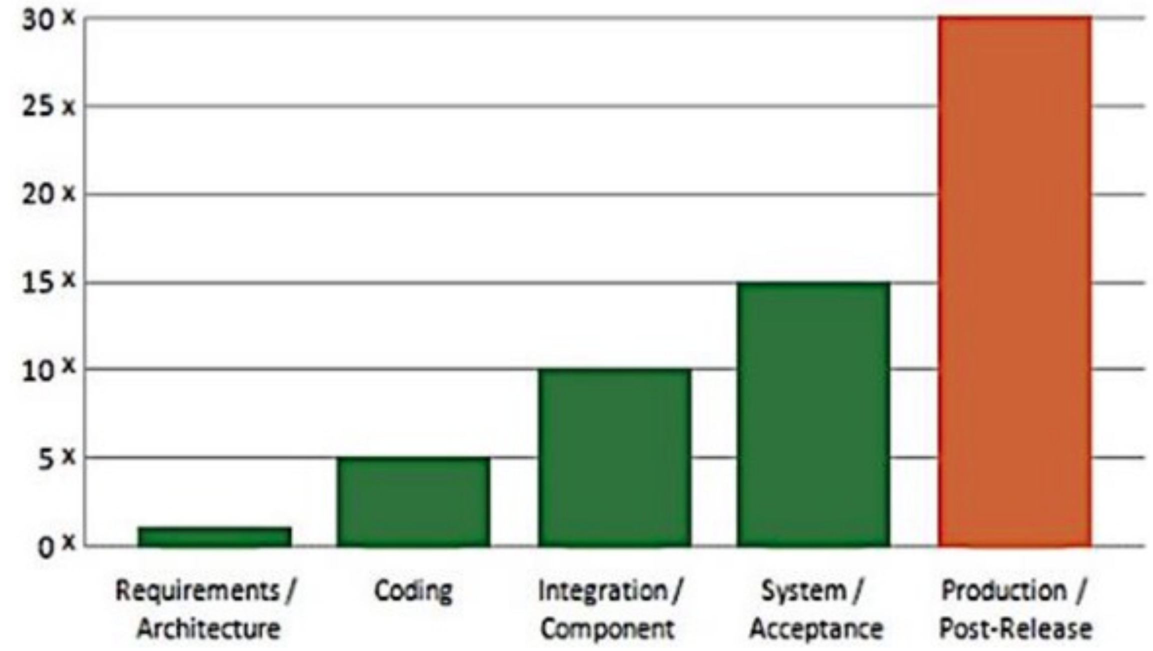
- Skills Needed
  - *Experience
  - Collaboration and communication skills
  - Knowledge of a defense in depth
  - Certs - CCNA Security, CEH, CISSP, CISA, CISM

Education, Awareness & Time

- Web Application and Cloud Security vulnerabilities predominately come from poor coding practices
- Others are a mix of unpatched systems or misconfigurations
- From universities to industry secure coding practices are not prioritised – security is often an afterthought
- Developers are under pressure to release and awareness often lies with a few members of the team

Cost of Fixing

- Research shows as you progress through the Software Development Lifecycle (SDLC) the cost to fix grows exponentially
- 30x more expensive to fix a vulnerability during post-production than during the design
- Bug Bounties from $50 - Tens of Thousands



Source: National Institute of Standards & Technology (NIST)

Constantly Evolving Landscape

- Black Hats & Security Researchers are at the forefront when it comes to emerging threats
- White Hats and the Industry are always a step behind constantly learning and evolving with emerging threats
- The fast-shifting landscape means security professionals need to stay on top of new exploits, vulnerabilities and methodologies
- You don't just learn how to hack it is more about a mindset and looking at things differently
- It can be frustrating, and it takes a type of person to remain interested and stay on top of things and is a real issue for industry.

Web Application Threa

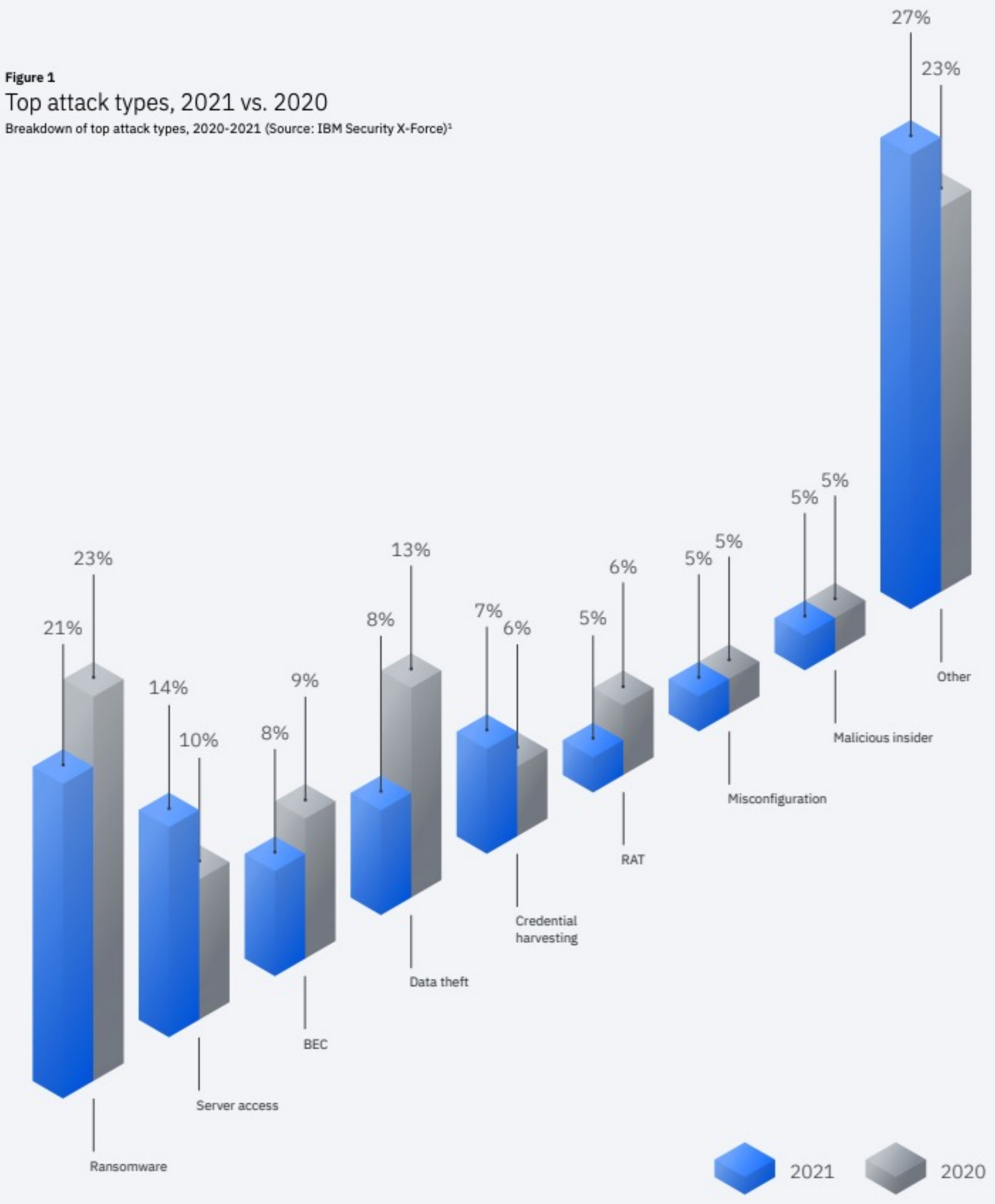
- Cloud and Web Application Security Threats are tightly linked
- Ransomware remains the most prevalent
- Broken Access Control causing the most harm

IBM X-Force Threat Intelligence Index 2022: https://www.ibm.com/downloads/cas/ADLMYLAZ

Figure 1

Top attack types, 2021 vs. 2020

Breakdown of top attack types, 2020-2021 (Source: IBM Security X-Force)[1]



[1] Other attacks include adware, banking trojans, botnets, cryptominers, defacements, fraud, DDoS, point of sale malware, spam, webscripts, webshells, and worms.

Industry Standards

- Open Web Application Security Project (OWASP)
- SysAdmin, Audit, Network & Security (SANS)
- MITRE - not an acronym
  - Common Weakness Enumeration (CWE)
  - Common Vulnerability Exposure (CVE)

Open Web Application Security Project (OWASP)

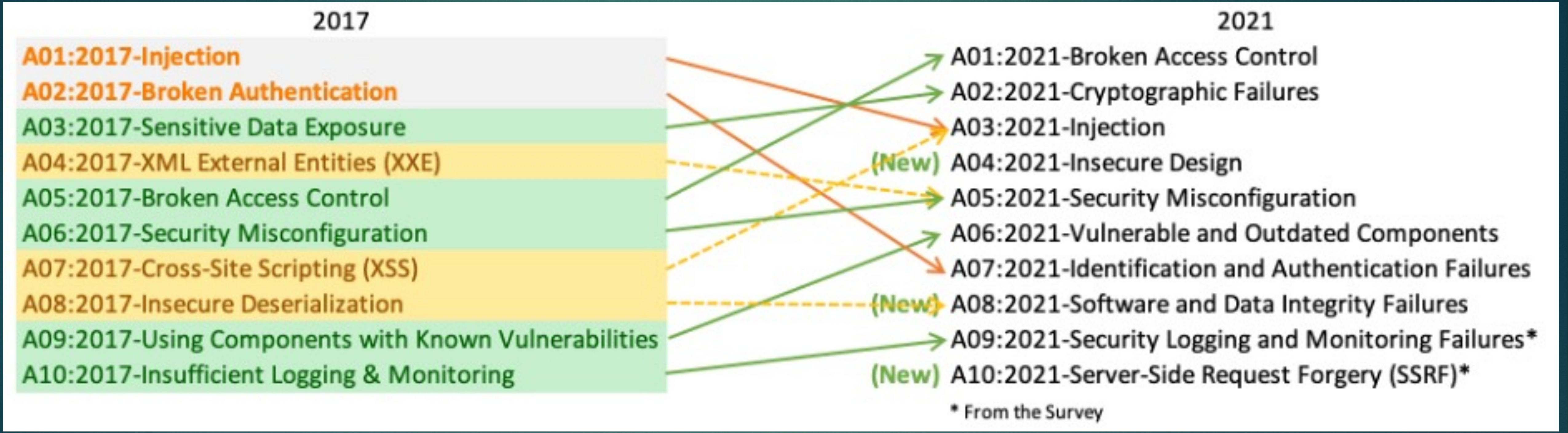
- The Open Web Application Security Project is an online community which creates freely-available articles, methodologies, documentation, tools, and technologies in the field of web application security.
- Not-for-profit charitable organization
- Local Chapters Around the Globe
  - Dublin https://www.owasp.org/index.php/Ireland-Dublin

OWASP
Open Web Application
Security Project

Utilising OWASP Flagship Projects

- We're going to utilise four projects to;
  - Understand the top risks
  - Learn the what, why, when, where, and how of testing web applications
  - Gain hands-on learning
  - Utilise the all-round tool for testing
- These 4 projects are;
  - OWASP Top 10
  - OWASP Testing Guide
  - Security Shepherd
  - Zed Attack Proxy or Burp Suite (not OWASP)

OWASP Top 10

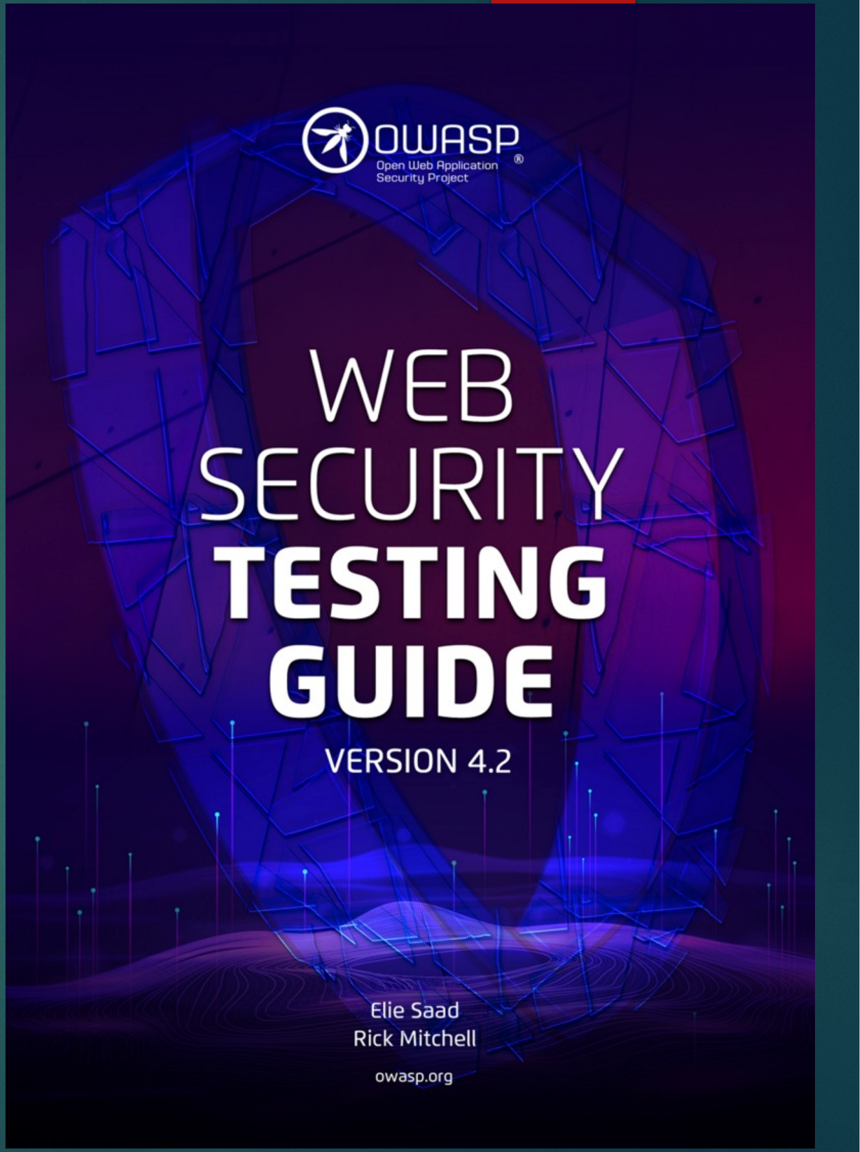


2017		2021
A01:2017-Injection		A01:2021-Broken Access Control
A02:2017-Broken Authentication		A02:2021-Cryptographic Failures
A03:2017-Sensitive Data Exposure		A03:2021-Injection
A04:2017-XML External Entities (XXE)	(New)	A04:2021-Insecure Design
A05:2017-Broken Access Control		A05:2021-Security Misconfiguration
A06:2017-Security Misconfiguration		A06:2021-Vulnerable and Outdated Components
A07:2017-Cross-Site Scripting (XSS)		A07:2021-Identification and Authentication Failures
A08:2017-Insecure Deserialization	(New)	A08:2021-Software and Data Integrity Failures
A09:2017-Using Components with Known Vulnerabilities		A09:2021-Security Logging and Monitoring Failures*
A10:2017-Insufficient Logging & Monitoring	(New)	A10:2021-Server-Side Request Forgery (SSRF)*

* From the Survey

OWASP Testing Guide v4.2

- Testing Framework - "Not just a checklist"
- Application Framework Areas
  - 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


OWASP
Open Web Application Security Project
WEB SECURITY TESTING GUIDE
VERSION 4.2
Elie Saad
Rick Mitchell
owasp.org

Aims of the Testing Guide

- Security testing based on the principles of engineering and science that offers a consistent, repeatable and defined approach to testing web applications
- You can't build a secure application without performing security testing
- The guide aims to provide organisations with a guide to testing in the hope of removing the varying degrees of quality and rigor that web applications go through when being tested.
- However, by itself security testing is not a good stand alone measure of how secure an application is.
- There are an infinite number of ways an attacker might be able to break an application. "We can't hack ourselves secure"
- White hat hackers have a limited time to test and defend where an attacker does not have such constraints.
- The testing guide should be used in conjunction with a well-defined security procedure during the SDLC with secure design principals, secure code practices and code review

- The OWASP Security Shepherd project is a web and mobile application security training platform.
- Security Shepherd has been designed to foster and improve security awareness among a varied skill-set demographic.
- The aim of this project is to take AppSec novices or experienced engineers and sharpen their penetration testing skillset to security expert status.

Web Application Pentesting Tools

- Not just an HTTP Proxy!
- Zed Attack Proxy (ZAP)
  - The OWASP Zed Attack Proxy (ZAP) is one of the world’s most popular free security tools and is actively maintained by hundreds of international volunteers*. It can help you automatically find security vulnerabilities in your web applications while you are developing and testing your applications. Its also a great tool for experienced pentesters to use for manual security testing.
- Burp Suite
  - Burp Suite is an integrated platform for performing security testing of web applications. Its various tools work seamlessly together to support the entire testing process, from initial mapping and analysis of an application's attack surface, through to finding and exploiting security vulnerabilities.

SysAdmin, Audit, Network & Security (SANS)

- The SANS Institute is a private U.S. for-profit company founded in 1989 that specialises in information security and cybersecurity training.
- Training includes cyber and network defenses, penetration testing, incident response, digital forensics, and audit.
- Courses are developed through a consensus process involving administrators, security managers, and information security professionals
- They developed the SANS 25 for developers to avoid common coding errors

SANS 25: Developers

- Insecure Interaction Between Components
  - Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
  - Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
  - Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
  - Unrestricted Upload of File with Dangerous Type
  - Cross-Site Request Forgery (CSRF)
  - URL Redirection to Untrusted Site ('Open Redirect')

CWE and SANS Institute

TOP 25 MOST DANGEROUS SOFTWARE ERRORS

SANS 25: Developers

- Risky Resource Management
  - Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')
  - Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
  - Download of Code Without Integrity Check
  - Inclusion of Functionality from Untrusted Control Sphere
  - Use of Potentially Dangerous Function
  - Incorrect Calculation of Buffer Size
  - Uncontrolled Format String
  - Integer Overflow or Wraparound

CWE and SANS Institute

TOP 25 MOST DANGEROUS SOFTWARE ERRORS

SANS 25: Developers

- Porous Defenses
  - Missing Authentication for Critical Function
  - Missing Authorization
  - Use of Hard-coded Credentials
  - Missing Encryption of Sensitive Data
  - Reliance on Untrusted Inputs in a Security Decision
  - Execution with Unnecessary Privileges
  - Incorrect Authorization
  - Incorrect Permission Assignment for Critical Resource
  - Use of a Broken or Risky Cryptographic Algorithm
  - Improper Restriction of Excessive Authentication Attempts
  - Use of a One-Way Hash without a Salt

CWE and SANS Institute

TOP 25 MOST DANGEROUS SOFTWARE ERRORS

MITRE

- American not-for-profit organisation

- It manages Federally Funded Research and Development Centers (FFRDCs);
  - Department of Defense (DOD),
  - Federal Aviation Administration (FAA),
  - Internal Revenue Service (IRS),
  - Department of Veterans Affairs (VA),
  - Department of Homeland Security (DHS),
  - Administrative Office of the U.S. Courts on behalf of the Federal Judiciary,
  - Centers for Medicare and Medicaid Services (CMS),
  - National Institute of Standards and Technology (NIST).
- Runs
  - CWE & CVE

CWE & CVE

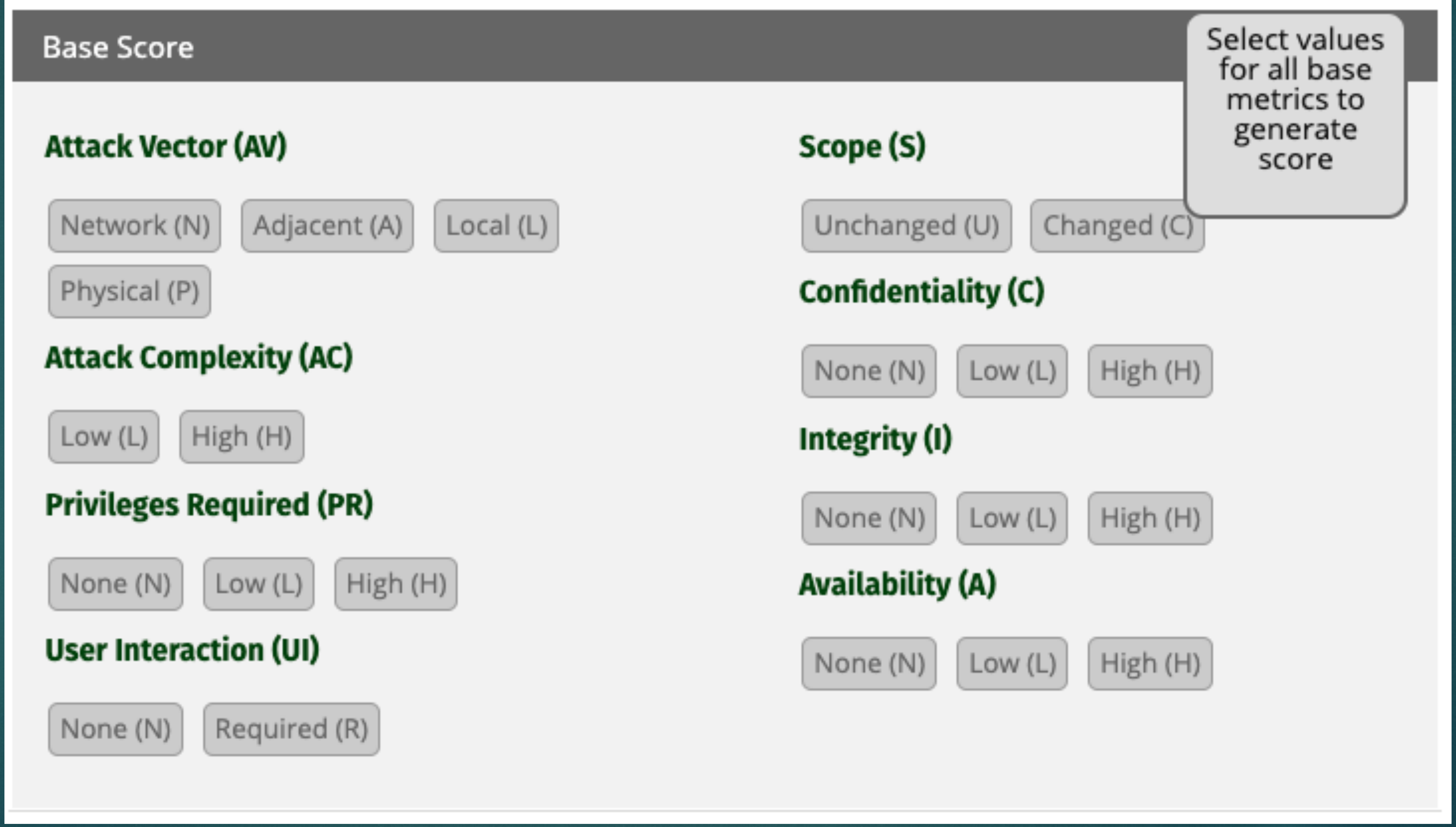
- Common Weakness Enumeration (CWE™) is a community-developed list of common software security weaknesses.
  - It serves as a common language, a measuring stick for software security tools, and as a baseline for weakness identification, mitigation, and prevention efforts.
- CWE™ is used in conjunction with SANS 25 where each of the top 25 is assigned a CWE number, e.g., SQL Injection CWE-89
- Common Vulnerability and Exposures (CVE®) International in scope and free for public use, CVE is a dictionary of publicly known information security vulnerabilities and exposures.
- If you find an exploit in the wild your bug will be assigned a CVE number
- Good Resource: https://www.cvedetails.com/

Measuring Risk: CVSS

- The Common Vulnerability Scoring System is a free and open industry standard for assessing the severity of computer system security vulnerabilities.
- CVSS attempts to assign severity scores to vulnerabilities, allowing responders to prioritize responses and resources according to threat.
- Scores are calculated based on a formula that depends on several metrics that approximate ease of exploit and the impact of exploit.
- Scores range from 0 to 10, with 10 being the most severe.
- Most utilise only the CVSS Base score for determining severity

The CVSS Tool v3

https://www.first.org/cvss/calculator/3.0



Select values for all base metrics to generate score

CVSS Base Score

- Attack Vector (AV)
  - Where is vulnerability exploitation possible. Score Increases as you move from physical to network
- Attack Complexity (AC)
  - Conditions beyond the attacker's control that must exist in order to exploit the vulnerability. Do you need more information or have the system in a certain configuration for it to be exploited.
- Privileges Required (PR)
  - The level of privileges an attacker must possess before successfully exploiting the vulnerability. Score increases the fewer privileges needed
- User Interaction (UI)
  - Does the attacker require another user to interact with something for a successful attack. Can it be exploited solely or does a separate user need to perform an action.
- Scope (S)
  - Does the vulnerability impact resources beyond its means or privileges. Can the attack escalate their privileges or can the exploit break out of a VM and effect the host OS

CVSS Base Score; CIA Triad

- Confidentiality (C)
  - Is information disclosed to an unauthorised user?
- Integrity (I)
  - Is the trustworthiness and veracity of information maintained?
- Availability (A)
  - Is the component available, e.g., is a web, database, or email server brought down and unreachable even partially by the exploit?

1
AVAILABILITY
INFORMATION SECURITY
2
INTEGRITY
3
CONFIDENTIALITY

Attack Vector

- Network
  - The vulnerable component is bound to the network stack and the attacker's path is through OSI layer 3. "remotely exploitable”
- Adjacent
  - The attack is limited to the same shared physical (e.g. Bluetooth, IEEE 802.11), or logical (e.g. local IP subnet) network
- Local
  - Not bound to the network stack. Either the attacker has logged in locally or needs user interaction to execute the exploit.
- Physical
  - Requires the attacker to physically touch or manipulate the vulnerable component

Attack Complexity & Privileges Required

- AC – Low
  - The exploit just works no conditions or special circumstances need to be in place
- AC – High
  - The exploit depends on conditions to be just right and can be beyond the attackers control.
- PR – None
  - The exploit is successful with no authorisation. The attacker does not need to login
- PR – Low
  - The attacker needs to login and is authorised with basic user privileges
- PR – High
  - The attacker needs a high level of access e.g. admin to carry out the exploit

User Interaction & Scope

- UI – None
  - The exploit doesn't need a user to interact with anything for the attack to be successful
- UI – Required
  - A user has to take some action before the exploit is successful
- Scope – Unchanged
  - The exploit only affects resources within the confines of the exploited system or user authorisation.
- Scope – Changed
  - The exploit affects resources beyond the authorisation of the vulnerable component.

Confidentiality, Integrity & Availability (CIA)

- None
  - There is no loss of CIA
- Confidentiality – Low
  - Some information is leaked but the attacker does not have control over what information, the amount or what kind. Information leaked may also not be seriously impacting
- Confidentiality – High
  - All information in the impacted resource is divulged or only some information is leaked but it is seriously impacting.
- Integrity – Low
  - Modification of data is possible but the attacker has no control over the amount or consequence. The modification does not have serious impact.
- Integrity – High
  - Total loss of integrity where the attacker can modify all files or some files but it has serious impact.
- Availability – Low
  - Reduced performance or an interruption to the service but the exploit does not completely take it offline. Also an impacted component could be taken completely offline but the impact is not serious
- Availability – High
  - The system is totally taken offline where it is persistent or some availability is impacted but this has serious consequences.

CVSS Scoring Results

Rating	CVSS Score
None	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0