Vulnerabilities - Intro

Software Security

Goals…

- Describe how vulnerabilities are categorized
- Describe how individual vulnerabilities are tracked
- Explain how a vulnerability's severity is calculated

Terms

- Vulnerability
  - “the quality or state of being exposed to the possibility of being attacked or harmed”
- Exploit
  - “a software tool designed to take advantage of a flaw in a computer system, typically for malicious purposes such as installing malware.”
- Taxonomy
  - “a scheme of classification”

Some Vuln Classes We'll Cover

- SQL Injection
- Cross Site Scripting
- Buffer Overflows
- Integer Overflows
- Command Injection
- Information Leakage
- Weak Password Systems
- …and more

Vulnerability Taxonomies & Databases

- CVE
  - Common Vulnerabilities and Exposures
- CWE
  - Common Weakness Enumeration
- OSVDB
  - Open Source Vulnerability Database
- CAPEC
  - Common Attack Pattern Enumeration and Classification

CVE

- Provides ID numbers to specific vulnerabilities
  - CVE + Year + Number
- Description of the vulnerability
- Links to other references
- https://cve.mitre.org/index.html

- …that's it
- Maintained by MITRE, funded by DHS
- 111 organizations are CVE Numbering Authorities (CNAs)
  - Mostly vendors (~99)

National Vulnerability Database

- US Government (NIST)
- Vulnerability management data in a common format
  - SCAP – Security Content Automation Protocol
  - Automation of vulnerability management is a big goal
- Provides a list of CVEs
- Uses CVSS to score the vulnerabilities
  - 0 to 10, 10 is most severe

CVSS

- Common Vulnerability Scoring Subsystem
- Open industry standard for rating vulnerabilities
- Assigns severity scores
  - Allows security professionals to prioritize the handling of vulnerabilities
- Three components
  - Base score
  - Temporal score
  - Environmental score
- Current version is 3.1

OSVDB

- Open Source Vulnerability Database
- Started in 2005 by HD Moore, et. al
- Supported by the Open Security Foundation

- Provided more analysis to the vulnerability than NVD

- Shut down in 2016

CAPEC

- Common Attack Pattern Enumeration and Classification
- Maintained by MITRE
- Known methods adversaries use to exploit known weaknesses

- Doesn’t really document vulnerabilities, but rather how they are exploited

CWE

- Common Weakness Enumeration
- Provides categories to weaknesses and vulnerabilities
- Also maintained by MITRE with financial support from DHS

- Not specific vulnerabilities in specific pieces of software
- Over 800 weaknesses cataloged

- https://cwe.mitre.org/

CWE-89

- Improper Neutralization of Special Elements used in an SQL Command
  - SQL Injection
- https://cwe.mitre.org/data/definitions/89.html

CVSS Details & Examples

Software Security

Vulnerability Rating Metrics

- Attack Vector
  - Network, Adjacent, Local, Physical
  - The more remote, the higher the score
- Attack Complexity
  - Low – special conditions do not exist, attack is successfully repeatable
  - High – success depends on conditions beyond attacker's control
- Privileges Required
  - None – Attacker can perform the attack unauthorized
  - Low – Only basic user privileges are required for the attack
  - High – Significant, often administrative, permissions are required

Vulnerability Rating Metrics

- User Interaction
  - None – the system can be exploited without user interaction
  - Required – successful exploitation requires a user to take some action
- Scope
  - Unchanged – exploiting a vulnerability will only allow access to the vulnerable component
  - Changed – exploiting a vulnerability will allow access beyond the vulnerable component
- Confidentiality Impact
  - High – total loss of confidentiality (ex. Password is exposed)
  - Low – some loss of confidentiality
  - None – no loss of confidentiality

Vulnerability Rating Metrics

- Integrity Impact
  - High – complete loss of integrity, attacker is able to modify files
  - Low – can modify data, but modification may not have a direct impact
  - None – no loss of integrity
- Availability Impact
  - High – total loss of availability, attacker can fully deny access
  - Low – reduced performance or availability interruptions
  - None – no impact to availability

Example…

- *Campus Security Cameras are configured with a default password*
- https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator
- **Attack Vector** – Network, Adjacent, Local, Physical
- **Attack Complexity** – Low, High
- **Privileges Required** – None, Low, High
- **User Interaction** – None, Required
- **Scope** – Unchanged, Changed
- **Confidentiality Impact** – High, Low, None
- **Integrity Impact** – High, Low, None
- **Availability Impact** – High, Low, None

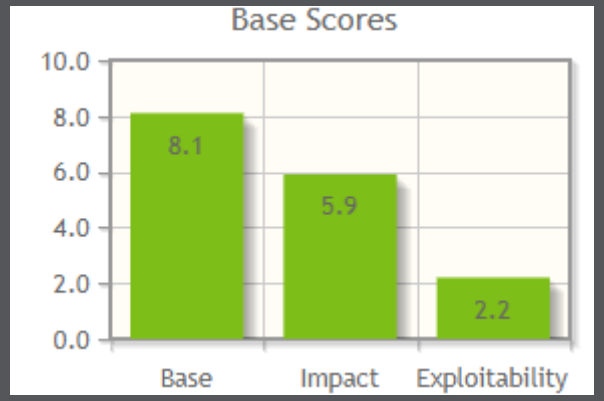
Example…

- *Campus Security Cameras are configured with a default password*
- **Attack Vector –** Network, Adjacent, Local, Physical
- **Attack Complexity –** Low, High
- **Privileges Required –** None, Low, High
- **User Interaction –** None, Required
- **Scope –** Unchanged, Changed
- **Confidentiality Impact –** High, Low, None
- **Integrity Impact –** High, Low, None
- **Availability Impact –** High, Low, None

Example…

- *Campus Security Cameras are configured with a default password*

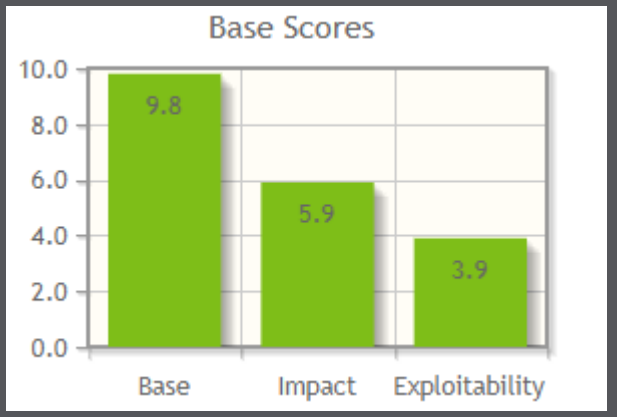
CVE-2017-0143



- MS-017-010 - https://cve.mitre.org/cve/cna.html
  - “The most severe of the vulnerabilities could allow remote code execution if an attacker sends specially crafted messages to a Microsoft Server Message Block 1.0 (SMBv1) server.”

- https://nvd.nist.gov/vuln/detail/CVE-2017-0143
- https://docs.microsoft.com/en-us/security-updates/securitybulletins/2017/ms17-010

CVE-2019-5523



- VMSA-2019-0004
  - VMware vCloud Director for Service Providers update resolves a Remote Session Hijack vulnerability
  - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-5523
  - https://www.vmware.com/security/advisories/VMSA-2019-0004.html
  - https://nvd.nist.gov/vuln/detail/CVE-2019-5523

How do we find vulnerabilities?

- Known vulnerabilities
  - Vulnerability scanner
  - Nessus
  - OpenVAS
  - …others
- Unknown vulnerabilities
  - Source Code Analysis
  - Binary Code Analysis
  - Static/Dynamic Code Analysis
  - Fuzzing