Vulnerability Taxonomies

Objectives: Understand common vulnerbility taxonomies and how they relate.

- U.S. National Vulnerability Database ***NVD***
- Common Platform Enumeration ***CPE***
- Common Vulnerabilities and Exposures ***CVE***
- Common Vulnerability Scoring System ***CVSS***
- Common Weakness Enumeration ***CWE***
- Common Attack Pattern Enumeration and Classification ***CAPEC***
- Adversarial Tactics, Techniques & Common Knowledge ***ATT&CK***
- Open Web Application Security Project ***OWASP***

U.S. National Vulnerability Database ***NVD***

- Created by the National Institute of Standards and Technology (NIST)
- repository of standards based Vulnerability management data
- Includes multiple databases of security checklists, security related software flaws, misconfigurations, product names, and impact metrics.
- CVE, CVSS, and others are all a part of the NVD

nvd.nist.gov

Common Platform Enumeration ***CPE***

- Basically just an official naming and versioning scheme for IT systems, software, and packages.
- Try using it on one of your projects sometime!
- Contains a dictionary of platform names and versions to automate decisions based on known vulnerabilities.

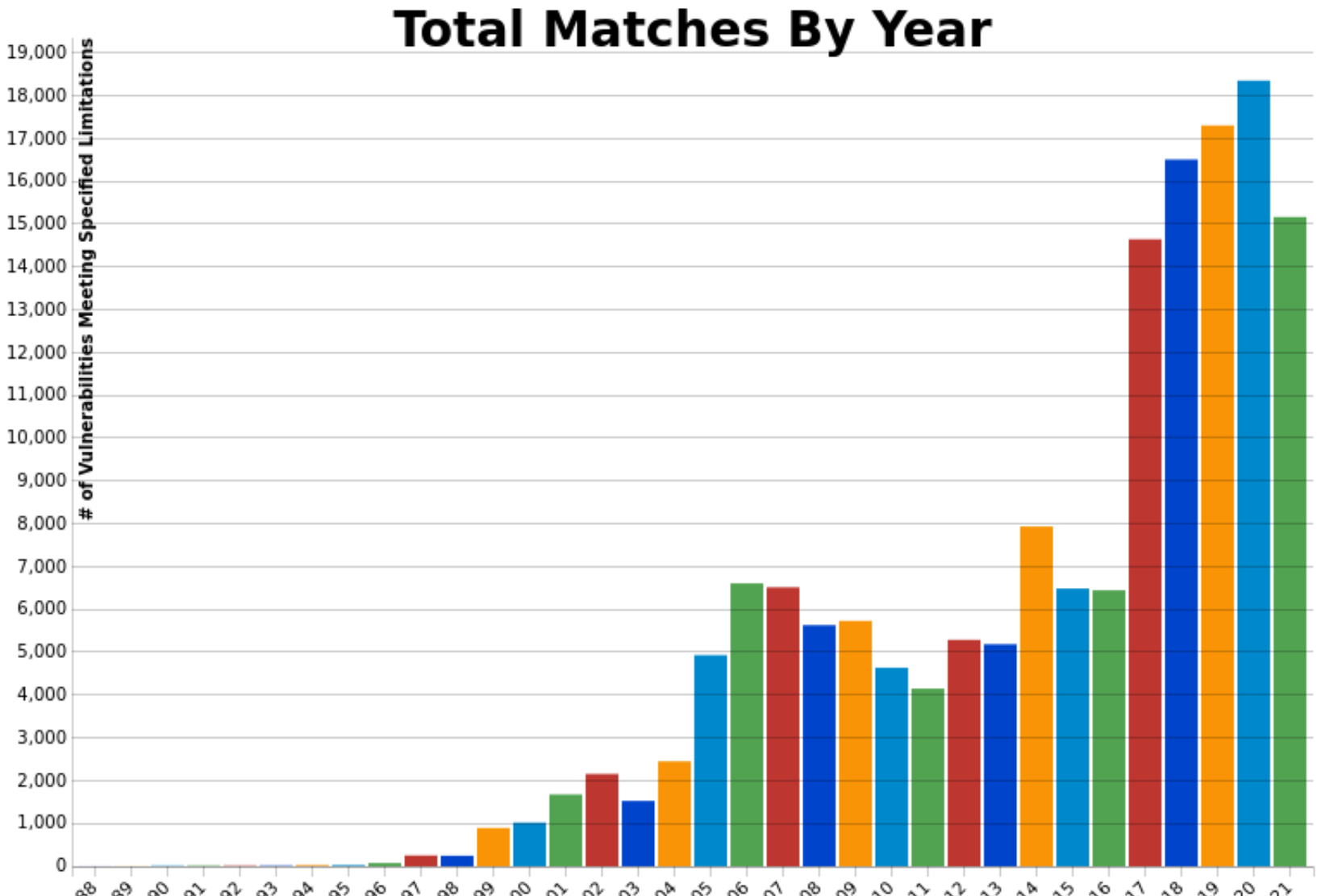
CPE

Common Vulnerabilities and Exposures ***CVE***

- Reference method for publicly known information security vulnerabilities and exposures
- A CVE name/number/ID is a unique identifier for a a single vulnerbility
- Only CVE Numbering Authorities (CNA) can issue CVE's
  - MITRE is the primary CNA
  - Various companies can assign CVE numbers for their own products (Microsoft, Oracle, Red Hat, etc.)
- CVE database contains several specified fields

CVE Further information and search

- CVE Wikipedia
- NVD CVE Lookup
- MITRE CVE Lookup



Common Vulnerability Scoring System ***CVSS***

- Given the growing number of CVE's each year we need a way to focus on the most important ones
- CVSS is a means of assigning a numerical score based on the ***severity*** of a given CVE
- Scores range from 0 to 10, low being not very important and 10 being a critical security vulnerability
- Several changes to this scoring metric have occured, be sure you are comparing similar versions of CVSS scores

CVSS Wikipedia

Common Weakness Enumeration ***CWE***

- Category system for software and hardware weaknesses and vulnerabilities
- Over 600 categories including
  - Buffer Overflow
  - path/directory traversal errors
  - hard-coded passwords
  - insecure random numbers. . . etc.

Vulnerbility cange by year MITRE about CWE CWE Top 25

Common Attack Pattern Enumeration and Classification

CAPEC

- Public catalog of common attack patterns to help users understand how weaknesses are exploited
- Based on Software Design Patterns
- Relates weaknesses (CWE) and vulnerbilities (CVE).
- Similar to CWE, the same CAPEC may apply to many CVEs
- CAPEC-139: Relative Path Traversal

Attack Patterns Wikipedia CAPEC Website

Adversarial Tactics, Techniques & Common Knowledge

ATT&CK

- Knowledge base of adversarial tactics
- The more you know (or a more theatrical: know your enemy)

MITRE ATT&CK

CAPEC & ATT&CK

Use CAPEC for:

- Application threat modeling
- Developer training and education
- Penetration testing

Use ATT&CK for:

- Comparing computer network defense capabilities
- Defending against the Advanced Persistent Threat
- Hunting for new threats
- Enhancing threat intelligence
- Adversary emulation exercises

Open Web Application Security Project ***OWASP***

MITRE is just one (pretty big) organization. There are others that attempt to classify similar things.

OWASP is a community that attempts similar classification for just web applications.

OWASP Wikipedia

OWASP.org

Homework

Properly Formatted `Yourname.md` uploaded to pilot. Style counts, I will be reading this in Github!.

- Read this: top 25 software CWE's
- Choose 1 of the top 25 (or honestly any known CSE) that you have personally put in code you used/submitted. Do a deep dive on that CWE (read all about it).
- Write up (at least) three paragraphs on the CWE, how your code was vulnerable to it, and how you could have changed the code to not be vulnerable.
- Be sure to include:
  - What it is, in your own words
  - At least one CVE with an explanation of what it was
  - What your personal experience is with the CWE
  - How you could have fixed it (what would you need to have done to not implement this weakness in your code)