Engineering Secure Software

# COMMON VULNERABILITY SCORING SYSTEM

# How Bad is Bad?

- We've seen many vulnerabilities
  - Many of them *can* do catastrophic things
  - Danger really "depends on the situation"
- Many, many situational factors, such as:
  - Asset exposed, and its relative importance
  - Remotely, or locally exploitable?
  - Expertise needed to exploit the vulnerability?
  - Affects all deployments?
  - Impact on CIA properties
  - How good is the reporting as of now?

# CVSS

- Common Vulnerability Scoring System
  - Adopted by NIST
  - Required for reporting a vulnerability in CVE
- An open scoring system from FIRST
  - FIRST: Forum for Incident Response & Security Teams
  - http://www.first.org/cvss
  - A group of researchers & practitioners
- Three levels
  - **Base**: no changes over time or environment
  - **Environmental**: might vary in different deployments
  - **Temporal**: might change over time
  - Essentially a weighted average

# Base: Access Vector

- Regarding networking, what are the methods of exploiting?
- Levels:
  - (L) Local only
  - (A) Adjacent network (e.g. wi-fi, local IP subnet)
  - (N) Network: fully remotely exploitable
- Notes
  - More than one level affected? Go with the worse one
  - Client that opens stuff from an untrusted internet source? Go with Network (e.g. zip utility with a buffer overflow)
- XSS in a webapp?
- (N)
- Lack of SSL encryption on Facebook?
- (A)

# Base: Access Complexity

- How complex would the exploit be?
  - One step? e.g. buffer overflow
  - Multiple steps?
    e.g. convince an email user to download a sketchy attachment
- Levels
  - (H) High: Specialized access conditions
    - e.g. depends elaborate social engineering methods
    - e.g. depends on a strange, rare configuration
  - (M) Medium: somewhat specialized conditions
    - e.g. non-default configuration, but plausible
    - e.g. requires some information gathering to be possible
  - (L) Low: no specialized conditions
    - e.g. default configuration
    - e.g. requires little skill to perform
- Note: Low complexity is bad

# Base: Authentication

- Is authentication needed for exploit?
- Levels
  - (M) Multiple layers of authentication needed
    - Could be multiple systems (e.g. network and OS)
    - Includes multi-factor
  - (S) Single layer of authentication
  - (N) No authentication needed
- In an authentication system itself? Go with (N) e.g. Kerberos
- e.g. path traversal in photo upload for a Twitter client?
- (S)
- e.g. insecure PRNG for session IDs?
- (N)

# Base: CIA Impact

- Any impact on
  - confidentiality, integrity, and/or availability?
  - These are three separate metrics
- Levels (for each metric)
  - (N) None
  - (P) Partial
    - e.g. disclosing a few database tables
    - e.g. temporary DoS
  - (C) Complete
    - e.g. reading arbitrary memory locations is Complete Disclosure
    - e.g. full bypass of plug-in sandbox is Complete Integrity
    - e.g. root-level access? Complete on all three metrics
- e.g. hardcoded root credentials in blogging software?
- C = Complete, I=Complete, A=None

# Environ: Collateral & Targets

- What is the potential for *collateral damage*?
  - Loss of life, physical assets, productivity
  - Levels: None, Low, Low-Medium, Medium-High, High, Not defined
- Is there a *target distribution*?
  - What proportion of that product distribution is targeted?
  - Levels:
    - None: 0% of the environment is at risk, lab settings only
    - Low: 1%-25%
    - Medium: 26%-75%
    - High: 76%-100%
    - Not defined
- Also: CIA “requirements” is ignored for our purposes

# Temporal: Exploitability

- Is there a public exploit known?
- Levels
  - (U) Unproven, entirely theoretical exploit
  - (POC) Proof-of-concept exists out there, no known malicious exploits
  - (F) Functional exploit is available
  - (H) Functional Exploit is widely disseminated
  - (ND) Not defined (skip this part of the metric)
- Notes
  - Being temporal, this could change quickly
  - Many white hats will write exploits to make this score go up, so that it's fixed

# Temporal: Remediation & Confidence

- What is the *level of remediation*?
  - How has the vendor reacted?
  - Levels
    - (O) Official Fix is available
    - (TF) Temporary fix is available
    - (W) Workaround is available.
      - Unofficial, non-vendor patches,
      - Temporary change in configuration
    - (U) Nothing is released yet
    - (ND) Not defined

- What is the *report confidence*?
  - (U) Unconfirmed by the source, or there are multiple conflicting reports
  - (UR) Uncorroborated reports from non-official sources
  - (C) Confirmed by the source
  - (ND) Not defined

# Scoring Tips

- Ignore interactions with other vulnerabilities, score each individually
- Emphasize targets to the host, not necessarily other users
  - E.g. XSS is a partial impact on integrity, but not full because it doesn't affect the host
- Assume the most common or default configuration of the server
- Score the greatest exploitation impact, if there are many

# Base Weights

- BaseScore = round_to_1_dec(
  - ((0.6*Impact)
  - +(0.4*Access)
  - -1.5)*f(Impact))
- Impact = 10.41*
  - (1-(1-ConfImpact)
  - *(1-IntegImpact)
  - *(1-AvailImpact))
- Access = 20*
  - AccessVector
  - *AccessComplexity
  - *Authentication
- f(impact)=0
  - if Impact=0,
  - 1.176 otherwise

| Metric | Level | Score |
|---|---|---|
| Access Vector | Local | 0.395 |
| | Adjacent | 0.646 |
| | Network | 1.0 |
| Access Complexity | High | 0.35 |
| | Medium | 0.61 |
| | Low | 0.71 |
| Authentication | Multiple | 0.45 |
| | Single | 0.56 |
| | None | 0.704 |
| ConfImpact, IntegImpact, AvailImpact | None | 0 |
| | Partial | .275 |
| | Complete | .660 |

# Weighting and Vectors

- Weights were derived from
  - Security experts got together (mostly industry)
  - Analyzed a bunch of vulnerabilities in their products
  - Agreed on all the labels for each vulnerability
  - Agreed on an overall ranking of many previous vulnerabilities
  - Adjusted the weights to match from there
- CVSS writers recommend
  - Using their calculator: http://nvd.nist.gov/cvss.cfm
  - Report each level along with the weighting
    - Weights may evolve
    - Comparing two vulnerabilities can be done at a finer level than just numbers

# Alternative: CWSS

- Common ***WEAKNESS*** Scoring System
  - Relatively recent (~2010) response to CVSS
  - More detailed, but not as widely-adopted
  - http://cwe.mitre.org/cwss/
  - Categories: Base, Attack Surface, Environmental

- Base Finding Metric Group
  - 5 metrics in this group
  - e.g. Acquired Privilege
    - *User-level access acquired. Admin?*
  - e.g. Acquired Privilege Layer
    - *Access to Network, App, entire Enterprise*
  - e.g. Internal Control Effectiveness
    - *Would our internal detection measures have been effective? Would we have known this was exploited?*

# Alternative: CWSS (cont.)

- Attack Surface Metric Group
  - 7 metrics in this group
  - e.g. Required Privilege AND Required Privilege Layer
    - *How much authentication was needed?*
  - e.g. Level of Interaction
    *How much social engineering is required?*
- Environmental Impact Group
  - 6 metrics in this group
  - e.g. Business Impact
  - e.g. Likelihood of Discovery
  - e.g. Likelihood of Exploit
  - e.g. Remediation Effort
    - *Is this a really difficult fix? Should we be worried about this coming up again or being incorrectly fixed?*

# Today's Activity

- Let's assess four vulnerabilities from industry
  - One from: PHP, Chromium, Tomcat, Linux kernel
    - Patches and reports are linked in
    - Feel free to use the internet to make your decisions (but don't look at any CVSS scorings online!)
  - CVSS base scores only
  - Also: two "detection" questions
    - Domain-specific
    - New code or changed code?
- In groups of 4-6
  - Planning Poker-like discussion
  - First: answer all four vulnerabilities for one question
  - Then: focus on one vulnerability at a time, for all questions