

Measuring the Root Cause of Incidents

Karen Scarfone

Computer Security Division

National Institute of Standards
and Technology (NIST)



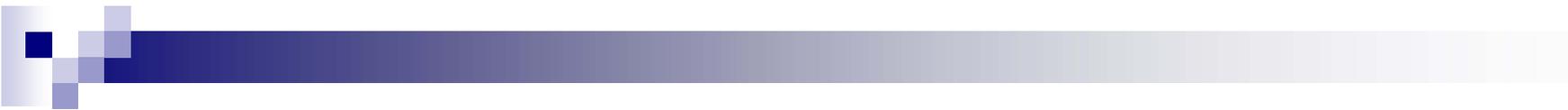
Overview

- Need quantitative measures of root cause vulnerabilities
 - Identify trends and common characteristics of incidents
 - Use as inputs to risk assessments and security control selection
- Base measures primarily on the Common Vulnerability Scoring System (CVSS) and related specifications
 - Provides consistent set of measures
 - Covers major classes of vulnerabilities
 - Can supplement with additional measures as needed



Outline

- Major classes of vulnerabilities
- Vulnerability measurement and scoring systems
- Additional measures not in the measurement and scoring systems
- Analysis and measurement of root causes
- Future work



Major classes of vulnerabilities

- Software flaws, security configuration issues, software feature/trust relationship misuse
- Example—use instant messaging to transfer unwanted files (malware) to the user's host
 - Software flaw: Coding flaw in IM client permits such transfers
 - Security configuration: IM client is configured to permit such transfers
 - Misuse: Social engineering tricks user into permitting such transfers; user mistakenly accepts transfer request; IM client does not offer a configuration option for restricting transfers

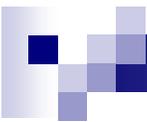


Root cause analysis example

A large malware incident started with one user.
Decompose the cause into its element vulnerabilities.

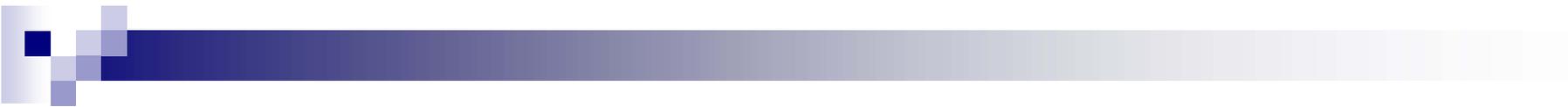
1. Misuse vulnerability: user clicked on something that he shouldn't have trusted
2. Configuration vulnerability: OS automatically executed a type of file that it should not have
3. Configuration vulnerability: user was running with administrator-level privileges that he should not have had
4. Software flaw vulnerability: OS had a flaw that the malware was able to exploit

Inherent value in decomposing causes this way



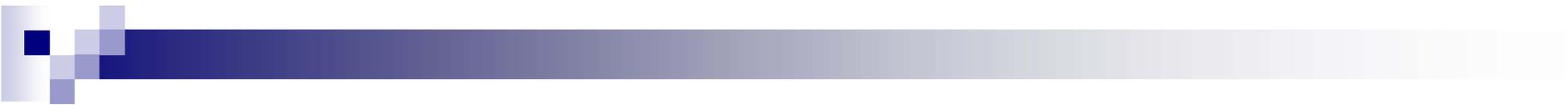
Vulnerability measurement and scoring systems

- Common Vulnerability Scoring System (CVSS) for software flaw vulnerabilities
 - Originally created to help prioritize patching
 - Various efforts to apply it for other purposes
- Common Configuration Scoring System (CCSS) for security configuration vulnerabilities
- Common Misuse Scoring System (CMSS) for software feature/trust relationship misuse vulnerabilities
- Collectively referred to as CxSS



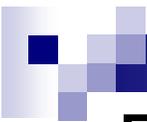
CVSS basics

- Three sets of vulnerability attribute measures
- Base: Constant over time and across all environments
 - Exploitability: Access Vector, Authentication, Access Complexity
 - Impact: Confidentiality, Integrity, Availability
- Temporal: Change over time but constant across all environments
 - Exploitability (Exploit Availability), Remediation Level, Report Confidence
- Environmental: Environment-specific
 - Collateral Damage Potential, Target Distribution, Security Requirements (Impact Bias)
- Produces 0-10 score for each set of measures



Additional measures not in CxSS

- Level of access gained
 - Root-level, user-level, process-level, etc.
- More specific vulnerability types
 - Common Weakness Enumeration (CWE)—
software weakness identifiers
- Can add others



Root cause measurement example: vulnerability #1 (user clicked on malicious attachment)

■ Base exploitability

- Attacker can exploit vulnerability remotely
- No authentication required by attacker
- Medium attack complexity (dependent on successful social engineering)

■ Temporal exploitability

- Exploit level is high (type of attack happens often)
- Remediation level is medium (somewhat mitigate through awareness and antimalware technologies)

■ Environmental exploitability

- Vulnerability prevalence is high (many users would click)
- Perceived target value is low
- Remediation level is low (mitigation strategies are less effective in this environment than in typical environments)



Root cause measurement example (cont.): vulnerability #1 (user clicked on malicious attachment)

■ Base impact

- Limited impact on host integrity
- No direct impact on availability or confidentiality

■ Environmental impact

- Confidentiality and integrity have medium importance, availability has low importance
- Collateral damage potential is low



Analyzing the measures

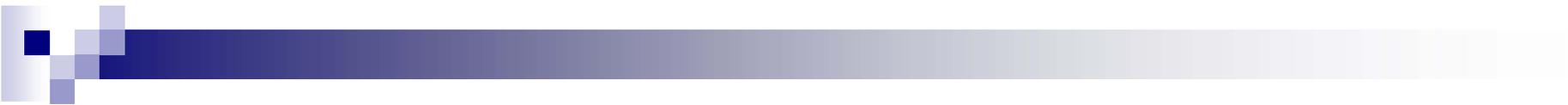
- Identify patterns
 - Characteristics of vulnerabilities
 - Chain of vulnerabilities for each incident
- Determine how incident prevention and handling can be improved
 - Which types of vulnerabilities need stronger mitigation—which links in the chain can be cut
 - How detection, containment, and other processes may need to be changed



Future work

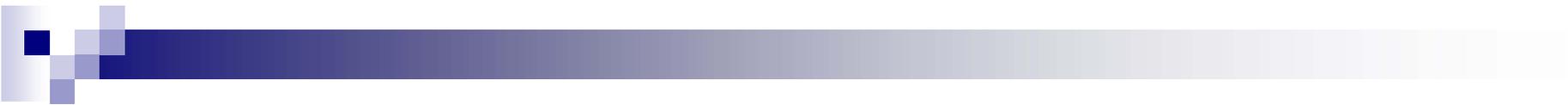
- Finalize CCSS and CMSS specifications
- Create misuse vulnerability taxonomy for CMSS
 - CVSS has CVE, CCSS has CCE
- Evaluate the soundness of CxSS measures and scores using real-world data

We welcome your input on what measures you have found useful, what we can do to improve our specifications, and what else we can do to help with root cause analysis and measurement



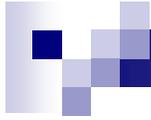
Conclusion

- Quantitative measurement of security is increasingly desired for decision making
- Proposed measures are a possible way of improving root cause analysis and better integrating it into other parts of security
- Substantial reference data is already available for proposed measures
- Need empirical data to verify which measures are feasible and valuable



Additional Information

- CVSS: Mell, P., Scarfone, K., and Romanosky, S., *A Complete Guide to the Common Vulnerability Scoring System Version 2.0*, June 2007, <http://first.org/cvss/cvss-guide.html>
- CCSS: Scarfone, K. and Mell, P., Draft NIST Interagency Report 7502, *The Common Configuration Scoring System (CCSS)*, May 2008, <http://csrc.nist.gov/publications/PubsNISTIRs.html>
- CMSS: Van Ruitenbeek, E. and Scarfone, K., Draft NIST Interagency Report 7517, *The Common Misuse Scoring System (CMSS): Metrics for Software Feature Misuse Vulnerabilities*, February 2009, <http://csrc.nist.gov/publications/PubsNISTIRs.html>



Thank you! Questions?

karen.scarfone@nist.gov