

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.1525

(04/2015)

SERIES X: DATA NETWORKS, OPEN SYSTEM
COMMUNICATIONS AND SECURITY

Cybersecurity information exchange – Vulnerability/state
exchange

Common weakness scoring system

Recommendation ITU-T X.1525

ITU-T

ITU-T X-SERIES RECOMMENDATIONS
DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	X.1–X.199
OPEN SYSTEMS INTERCONNECTION	X.200–X.299
INTERWORKING BETWEEN NETWORKS	X.300–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	X.600–X.699
OSI MANAGEMENT	X.700–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	X.850–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	
General security aspects	X.1000–X.1029
Network security	X.1030–X.1049
Security management	X.1050–X.1069
Telebiometrics	X.1080–X.1099
SECURE APPLICATIONS AND SERVICES	
Multicast security	X.1100–X.1109
Home network security	X.1110–X.1119
Mobile security	X.1120–X.1139
Web security	X.1140–X.1149
Security protocols	X.1150–X.1159
Peer-to-peer security	X.1160–X.1169
Networked ID security	X.1170–X.1179
IPTV security	X.1180–X.1199
CYBERSPACE SECURITY	
Cybersecurity	X.1200–X.1229
Countering spam	X.1230–X.1249
Identity management	X.1250–X.1279
SECURE APPLICATIONS AND SERVICES	
Emergency communications	X.1300–X.1309
Ubiquitous sensor network security	X.1310–X.1339
PKI related Recommendations	X.1340–X.1349
CYBERSECURITY INFORMATION EXCHANGE	
Overview of cybersecurity	X.1500–X.1519
Vulnerability/state exchange	X.1520–X.1539
Event/incident/heuristics exchange	X.1540–X.1549
Exchange of policies	X.1550–X.1559
Heuristics and information request	X.1560–X.1569
Identification and discovery	X.1570–X.1579
Assured exchange	X.1580–X.1589
CLOUD COMPUTING SECURITY	
Overview of cloud computing security	X.1600–X.1601
Cloud computing security design	X.1602–X.1639
Cloud computing security best practices and guidelines	X.1640–X.1659
Cloud computing security implementation	X.1660–X.1679
Other cloud computing security	X.1680–X.1699

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T X.1525

Common weakness scoring system

Summary

Recommendation ITU-T X.1525 on the common weakness scoring system (CWSS) provides an open framework for communicating the characteristics and impact of information and communication technologies (ICT) weaknesses during development of software capabilities. The goal of this Recommendation is to enable ICT software developers, managers, testers, security vendors and service suppliers, buyers, application vendors and researchers to speak from a common language of scoring ICT weaknesses that could manifest as vulnerabilities when the software is used.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T X.1525	2015-04-17	17	11.1002/1000/12357

* To access the Recommendation, type the URL <http://handle.itu.int/> in the address field of your web browser, followed by the Recommendation's unique ID. For example, <http://handle.itu.int/11.1002/1000/11830-en>.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2015

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

Page

1	Scope.....	1
2	References.....	1
3	Definitions	1
	3.1 Terms defined elsewhere	1
	3.2 Terms defined in this Recommendation.....	2
4	Abbreviations and acronyms	2
5	Conventions	3
6	Use of CWSS.....	3
	6.1 CWSS description	3
	6.2 Workings of CWSS	4
	6.3 CWSS scoring	5
	6.4 CWSS users	6
7	Metric groups.....	7
	7.1 Metric group factors	7
	7.2 Values for uncertainty and flexibility.....	8
	7.3 Base finding metric group	9
	7.4 Attack Surface metric group.....	15
	7.5 Environmental Metric group	19
	7.6 CWSS score formula	24
	7.7 CWSS vectors, scoring examples and score portability.....	26
	Bibliography.....	30

Introduction

Software developers often face hundreds or thousands of individual bug reports for weaknesses that are discovered in their code. In certain circumstances, a software weakness can even lead to an exploitable vulnerability. Due to this high volume of reported weaknesses, stakeholders are often forced to prioritize which issues they should investigate and fix first. In short, people need to be able to reason and communicate about the relative importance of different weaknesses. While various scoring methods are used today, they are either ad hoc or inappropriate for application to the still-imprecise evaluation of software security. The common weakness scoring system (CWSS) provides a mechanism for prioritizing software weaknesses in a consistent, flexible, open manner while accommodating context for the various business domains and intended uses of the software. It is a collaborative, community-based effort that is addressing the needs of its stakeholders across government, academia and industry.

ICT software developers, managers, testers, security vendors and service suppliers, buyers, application vendors and researchers must identify and assess weaknesses in software that could manifest as vulnerabilities when the software is used. They then need to be able to prioritize these weaknesses and determine which to remediate based on which of them pose the greatest risk. When there are so many to fix, with each being scored using different scales, the various ICT community members, managers, testers, buyers and developers are left to their own methodologies to find some way of comparing disparate weaknesses and translating them into actionable information.

Because CWSS standardizes the approach for characterizing weaknesses, users of CWSS can invoke attack surface and environmental metrics to apply contextual information that more accurately reflects the risk to the software capability given the unique business context it will function within and the unique business capability it is meant to provide. This allows them to make more informed decisions when trying to mitigate risks posed by weaknesses.

CWSS leverages existing work from within the cyber security community such as the large number of diverse real-world publicly known vulnerabilities specified through [b-ITU-T X.1520] – common vulnerabilities and exposures (CVE) and the scoring system used for discussing the severity of those publicly known vulnerabilities through [b-ITU-T X.1521] – common vulnerability scoring system (CVSS) as well as the common weakness enumeration (CWE) list of weaknesses in the software's architecture, design, code or deployment. In constructing CWSS, the ability to allow reasonable default values for areas that may not be known, while providing for the tailoring based on business and technical context.

CWSS is one of a class of ITU-T Recommendations that comes from a large, existing, global development and user community that has written and evolved an open specification that is made available to the ITU-T for adoption with agreement that any changes or updates to the specification will be done in a manner that ensures full technical equivalency and compatibility will be maintained, that discussions about changes and enhancements will be done through the original user community processes, and includes explicit reference to the corresponding specific version maintained by the user community. Thus, at the time of initial adoption of Recommendation ITU-T X.1525, a due diligence verification and statement of equivalency will occur; and as changes are effected among the user community, timely reflection of those changes will be reflected in subsequent versions of the Recommendation through continued collaboration.

Recommendation ITU-T X.1525 - Common weakness scoring system (CWSS) has been developed on a collaborative basis with The MITRE Corporation bearing in mind the importance of maintaining, to the extent possible, technical compatibility between Recommendation ITU-T X.1525 – Common weakness scoring system (CWSS) and the "Common Weakness Scoring System (CWSS)", version 1.0.1, dated 5 September 2014 [https://cwe.mitre.org/cwss/cwss_v1.0.1.html].

Recommendation ITU-T X.1525

Common weakness scoring system

1 Scope

This Recommendation provides a standardized approach for communicating the characteristics and impacts of weaknesses during development of ICT software capabilities using attack surface and environmental metrics to apply contextual information. CWSS more accurately reflects the risk to the user of the software capability, given the unique business context it will function within for the user, and the unique business capability the software is providing to the user.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

None.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 access [b-ITU-T X.1521]: A subject's ability to view, modify, or communicate with an object. Access enables the flow of information between the subject and the object.

3.1.2 availability [b-ITU-T X.1521]: The reliability and timely access to data and resources by authorized individuals.

3.1.3 attack instance [b-ITU-T X.1544]: A specific detailed attack against an application or system targeting vulnerabilities or weaknesses in that system.

3.1.4 confidentiality [b-ITU-T X.1521]: A security principle that works to ensure that information is not disclosed to unauthorized subjects.

3.1.5 integrity [b-ITU-T X.1521]: A security principle that makes sure that information and systems are not modified maliciously or accidentally.

3.1.6 risk [b-ITU-T X.1521]: The relative impact that an exploited vulnerability would have to a user's environment.

3.1.7 threat [b-ITU-T X.1521]: The likelihood or frequency of a harmful event occurring.

3.1.8 vulnerability [b-ITU-T X.1500]: Any weakness that could be exploited to violate a system or the information it contains.

3.1.9 weakness [b-ITU-T X.1524]: A shortcoming or imperfection in the software code, design, architecture, or deployment that, could, at some point become a vulnerability, or could contribute to the introduction of other vulnerabilities.

3.2 Terms defined in this Recommendation

This Recommendation defines the following term:

3.2.1 vignette: A vignette provides a shareable, formalized way to define a particular environment, the role that software plays within that environment and an organization's priorities with respect to software security.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AI	Authentication Instances
AL	Acquired privilege Layer
AP	Acquired Privilege
AS	Authentication Strength
ASLR	Address Space Layout Randomization
AV	Access Vector
BI	Business Impact
BVC	Business Value Context
CD	Compact Disc
CIO	Chief Information Officer
CSO	Chief Security Officer
CSRF	Cross-Site-Request-Forgery
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
CWRAF	Common Weakness Risk Analysis Framework
CWSS	Common Weakness Scoring System
DI	Likelihood of Discovery
DNS	Domain Name System
DS	Deployment Scope
EC	External Control effectiveness
EX	Likelihood of Exploit
FC	Finding Confidence
FTP	File Transfer Protocol
HTML	Hyper Text Markup Language
IC	Internal Control Effectiveness
ICT	Information Communication Technology
IN	level of Interaction
IP	Internet Protocol
NIST	National Institute of Standards and Technology
OS	Operating System

OWASP	Open Web Application Security Project
P	Prevalence
PCI DSS	Payment Card Industry Data Security Standard
RL	Required Privilege Layer
RP	Required Privilege
SAMATE	Software Assurance Metrics And Tool Evaluation
SANS	SysAdmin, Audit, Networking and Security
SQL	Structured Query Language
SSL	Secure Sockets Layer
TI	Technical Impact
TLS	Transport Layer Security
USB	Universal Serial Bus
XSS	Cross-Site Scripting

5 Conventions

None.

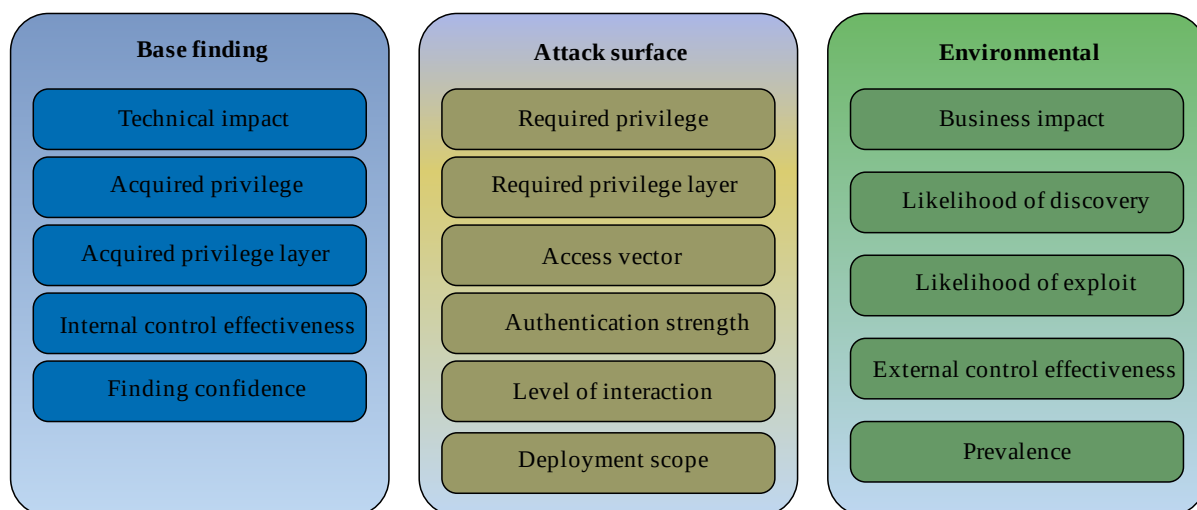
6 Use of CWSS

Currently, ICT software developers, managers, testers, security vendors and service suppliers, buyers, application vendors and researchers must identify and assess weaknesses in software that could manifest as vulnerabilities when the software is used. They then need to be able to prioritize these weaknesses and determine which to remediate based on which of them pose the greatest risk. When there are so many to fix, with each being scored using different scales, the various ICT community members, managers, testers, buyers and developers are left to their own methodologies to find some way of comparing disparate weaknesses and translating them into actionable information. The common weakness scoring system (CWSS) is an open framework that addresses this issue. It offers the following benefits:

- Quantitative measurements: CWSS provides a quantitative measurement of the unfixed weaknesses that might be present within a software application.
- Common framework: CWSS provides a common framework for prioritizing security errors ("weaknesses") that are discovered in software applications.
- Customized prioritization: in conjunction with the [Common Weakness Risk Analysis Framework \(CWRAF\) \[b-CWRAF\]](#), CWSS can be used by consumers to identify the most important types of weaknesses for their business domains, in order to inform their acquisition and protection activities as one part of the larger process of achieving software assurance.

6.1 CWSS description

CWSS is organized into three *metric groups*: Base Finding, Attack Surface and Environmental, as shown in Figure 1. Each group contains multiple metrics – also known as *factors* – that are used to compute a CWSS score for a weakness.



X.1525(15) F01

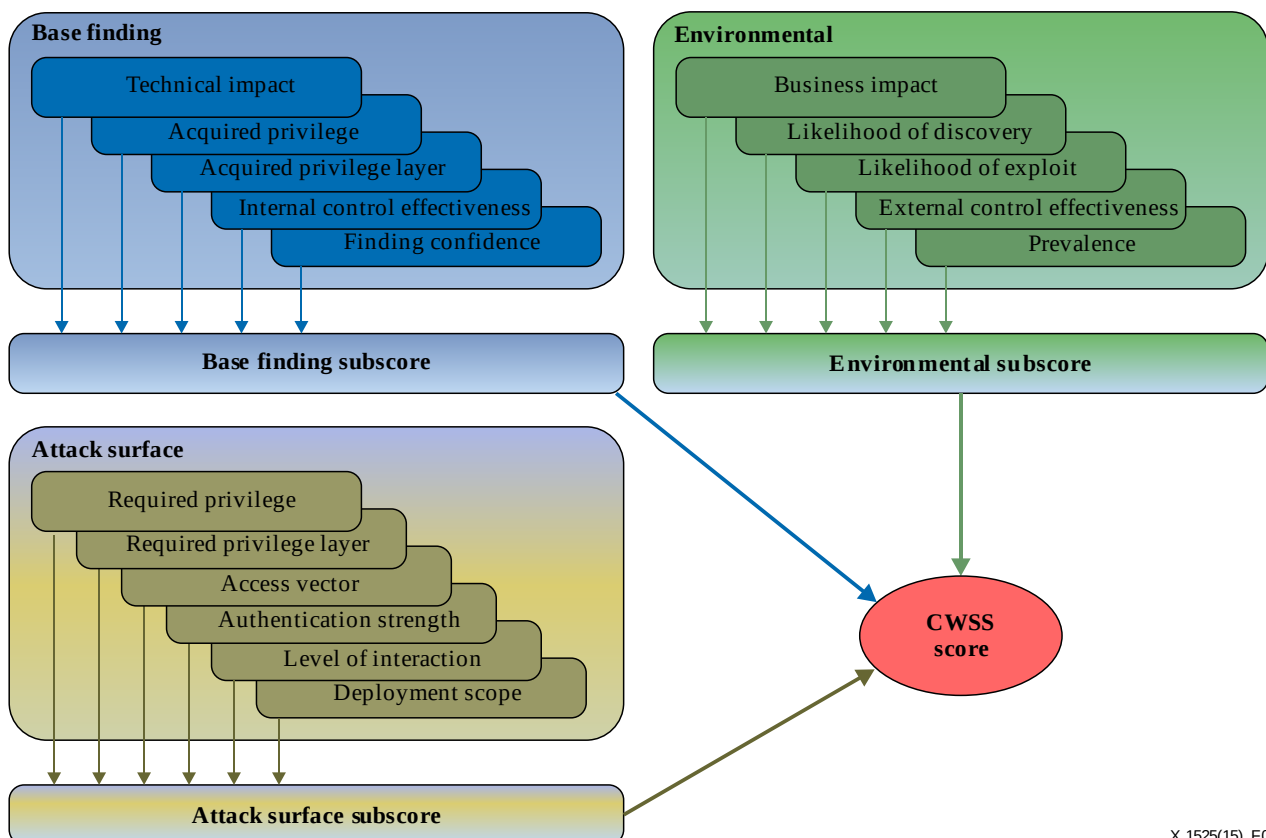
Figure 1 – CWSS metric groups

These metric groups are described as follows:

- Base Finding metric group: captures the inherent risk of the weakness, confidence in the accuracy of the finding and strength of controls. Base Finding metric group is discussed in clause 7.3.
- Attack Surface metric group: the barriers that an attacker must overcome in order to exploit the weakness. Attack Surface metric group is discussed in clause 7.4.
- Environmental metric group: characteristics of the weakness that are specific to a particular environment or operational context. Environmental metric group is discussed in clause 7.5.

6.2 Workings of CWSS

Each factor in the Base Finding metric group is assigned a value. These values are converted to associated weights, and a Base Finding subscore is calculated. The Base Finding subscore can range between 0 and 100. The same method is applied to the Attack Surface and Environmental metric group; their subscores can range between 0 and 1. Finally, the three subscores are multiplied together, which produces a CWSS score between 0 and 100, as illustrated below in Figure 2.



X.1525(15)_F02

Figure 2 – CWSS scoring

6.3 CWSS scoring

The stakeholder community is collaborating with MITRE to investigate several different scoring methods that might need to be supported within the CWSS framework. The current four scoring methods are:

- | | |
|-------------------------|--|
| Targeted | Score individual weaknesses that are discovered in the design or implementation of a specific ("targeted") software package, e.g., a buffer overflow in the username of an authentication routine in line 1234 of server.c in an FTP server package. Automated tools and software security consultants use targeted methods when evaluating the security of a software package in terms of the weaknesses that are contained within the package. |
| Generalized | Score classes of weaknesses independent of any particular software package, in order to prioritize them relative to each other (e.g., "buffer overflows are higher priority than memory leaks"). This approach is used by the CWE/SANS Top 25, OWASP Top Ten and similar efforts, but also by some automated code scanners. The generalized scores could vary significantly from the targeted scores that would result from a full analysis of the individual occurrences of the weakness class within a specific software package. For example, while the class of buffer overflows remains very important to many developers, individual buffer overflow bugs might be considered less important if they cannot be directly triggered by an attacker and their impact is reduced due to operating system (OS)-level protection mechanisms such as address space layout randomization (ASLR). |
| Context-adjusted | Modify scores in accordance with the needs of a specific analytical context that may integrate business/mission priorities, threat environments, risk tolerance, etc. These needs are captured using vignettes that link inherent characteristics of weaknesses |

with higher-level business considerations. This method could be applied to both targeted and generalized scoring.

Aggregated Combine the results of multiple, lower-level weakness scores to produce a single, overall score (or "grade"). While aggregation might be most applicable to the targeted method, it could also be used in generalized scoring, as occurred in the 2010 CWE/SANS Top 25.

Note that the current focus for most discussions about CWSS is on the Targeted scoring method and a framework for context-adjusted scoring. Methods for aggregated scoring will follow. Generalized scoring is being developed separately, primarily as part of the 2011 Top 25 and CWRAF.

CWSS scores can be automatically calculated, e.g., by a code analysis tool, or they can be manually calculated by a software security consultant or developer. Since automated analysis is not likely to have certain information available – such as the application's operating environment – CWSS scoring could possibly be conducted in multiple rounds: a tool first automatically calculates CWSS scores, then a human analyst manually adds additional details and recalculates the scores.

6.4 CWSS users

To be most effective, CWSS supports multiple usage scenarios by different stakeholders who all have an interest in a consistent scoring system for prioritizing software weaknesses that could introduce risks to products, systems, networks and services. Below are some examples of the primary stakeholders:

- **Software developers:** Developers often operate within limited time frames, due to release cycles and limited resources. As a result, they are unable to investigate and fix every reported weakness. They may choose to concentrate on the worst problems, the easiest-to-fix. In the case of automatic weakness findings, they might choose to focus on the findings that are least likely to be false positives.
- **Software development managers:** Development managers create strategies for prioritizing and removing entire classes of weaknesses from the entire code base, or at least the portion that is deemed to be most at risk, possibly by defining custom "Top-N" lists. They must understand the security implications of integrating third-party software, which may contain its own weaknesses. They may need to support distinct security requirements and prioritization for each product line.
- **Software acquirers:** Customers, including acquisition personnel, want to obtain third-party software with a reasonable level of assurance that the software provider has performed due diligence in removing or avoiding weaknesses that are most critical to the acquirer's business and mission. Related stakeholders include chief information officers (CIOs), chief security officers (CSOs), system administrators and end users of the software.
- **Enterprise security managers:** Enterprise security managers seek to minimize risk within their enterprise, both for well-known vulnerabilities in third-party products, as well as vulnerabilities (or weaknesses) in their own in-house software. They may wish to use a scoring mechanism that can be integrated with other security management processes, such as combining third-party vulnerability scanning results (for known third-party vulnerabilities) with custom application analysis (for in-house software) to help assess the overall risk to an asset.
- **Code analysis vendors and consultants:** Vendors and consultants often have their own custom scoring techniques, but they want to provide a consistent, community-vetted scoring mechanism for different customers.
- **Evaluators of code analysis capabilities:** Evaluators analyse and measure the capabilities of code analysis techniques (e.g., NIST SAMATE). They could use a consistent weakness