Commentary on ‘Known Vulnerabilities’

By Eric Hill, Synopsys

Discussion

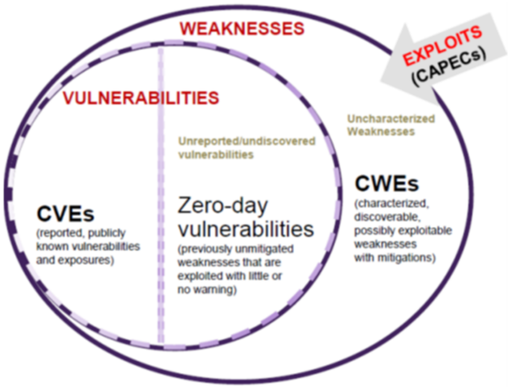
Synopsys’ software verification suite (Black Duck, Coverity, Seeker, Defensics & Code DX) offers SCA, SAST, IAST, DAST and ASOC capabilities. Experience in the field has afforded the company an industry wide perspective on software verification best practices in managing defects while maintaining software development velocity.

It is noticed that the phrase “known vulnerabilities” is utilized several times in the “DRAFT Baseline Criteria for Consumer Software Cybersecurity Labeling”. The key attestation noted is to declare software “Free from Known Vulnerabilities”.

EO14028 has resulted in software verification software and practices being rapidly & widely adopted. There is confusion in industry as to what constitutes a CVE versus a CWE (see diagram). Reported vulnerabilities (ie. ‘known’ publicly) will have been entered in the National Vulnerability database & can be mapped to 1 or more CWE’s. Our goal in applying software verification as a practice, and current patterns offer a Software Factory configuration, is to apply CWE avoidance in releasing software. Thus, DevSecOps teams lower the risk of CVE’s being reported in the future against their released software.

CWE/CVE/CAPEC Venn Diagram

(Synopsys SIG Tool Reporting Using Standards-Based Security)


WEAKNESSES
EXPLOITS
(CAPECs)
VULNERABILITIES
Uncharacterized Weaknesses
Unreported/undiscovered vulnerabilities
CVEs
(reported, publicly known vulnerabilities and exposures)
Zero-day vulnerabilities
(previously unmitigated weaknesses that are exploited with little or no warning)
CWEs
(characterized, discoverable, possibly exploitable weaknesses with mitigations)

Recommendation

Consider being more precise via a simple definition of ‘known vulnerabilities’ using the terms CVEs with an additional note to include ‘CWEs that are weaknesses that represent sources for future exploitable vulnerabilities.’