A06:2021 – Anfällige und veraltete Komponenten

Faktoren

CWEs kartiert	Maximale Inzidenzrate	Durchschnittliche Inzidenzrate	Maximale Abdeckung	Durchschnittliche Abdeckung	Durchschnittlich gewichteter Exploit	Durchschnittliche gewichtete Auswirkung
3	27,96 %	8,77 %	51,78 %	22,47 %	5.00	5.00

Überblick

Es belegte Platz 2 in der Top-10-Community-Umfrage, verfügte aber auch über genügend Daten, um es datentechnisch in die Top 10 zu schaffen. Gefährdete Komponenten sind ein bekanntes Problem, das wir nur schwer testen und bewerten können. Sie stellen die einzige Kategorie dar, in der den enthaltenen CWEs keine Common Vulnerability and Exposures (CVEs) zugeordnet sind. Daher wird eine standardmäßige Exploit-/Auswirkungsgewichtung von 5,0 verwendet. Bemerkenswerte CWEs sind *CWE-1104: Verwendung nicht gewarteter Drittanbieterkomponenten* und die beiden CWEs aus den Top 10 2013 und 2017.

Beschreibung

Sie sind wahrscheinlich gefährdet:

- Wenn Sie nicht die Versionen aller von Ihnen verwendeten Komponenten kennen (sowohl clientseitig als auch serverseitig). Dazu gehören Komponenten, die Sie direkt verwenden, sowie verschachtelte Abhängigkeiten.
- Wenn die Software anfällig, nicht unterstützt oder veraltet ist. Dazu gehören das Betriebssystem, der Web-/Anwendungsserver, das Datenbankverwaltungssystem (DBMS), Anwendungen, APIs und alle Komponenten, Laufzeitumgebungen und Bibliotheken.
- Wenn Sie nicht regelmäßig nach Schwachstellen suchen und Sicherheitsbulletins für die von Ihnen verwendeten Komponenten abonnieren.
- Wenn Sie die zugrunde liegende Plattform, Frameworks und Abhängigkeiten nicht risikobasiert und zeitnah reparieren oder aktualisieren. Dies geschieht häufig in Umgebungen, in denen das Patchen eine monatliche oder vierteljährliche Aufgabe unter Änderungskontrolle ist, wodurch Unternehmen tage- oder monatelang unnötig mit behobenen Schwachstellen konfrontiert werden.
- Wenn Softwareentwickler die Kompatibilität aktualisierter, aktualisierter oder gepatchter Bibliotheken nicht testen.
- Wenn Sie die Konfigurationen der Komponenten nicht sichern (siehe A05:2021-Security Misconfiguration).

Wie man etwas vorbeugt

Es sollte ein Patch-Management-Prozess vorhanden sein, um:

- Entfernen Sie ungenutzte Abhängigkeiten, unnötige Funktionen, Komponenten, Dateien und Dokumentation.
- Kontinuierliche Bestandsaufnahme der Versionen sowohl der clientseitigen als auch der serverseitigen Komponenten (z. B. Frameworks, Bibliotheken) und ihrer Abhängigkeiten mithilfe von Tools wie Versionen, OWASP Dependency Check, retire.js usw. Überwachen Sie kontinuierlich Quellen wie Common Vulnerability and Exposures (CVE). und National Vulnerability Database (NVD) für Schwachstellen in den Komponenten. Verwenden Sie Software-Tools zur Analyse der Zusammensetzung, um den Prozess zu automatisieren. Abonnieren Sie E-Mail-Benachrichtigungen zu Sicherheitslücken im Zusammenhang mit den von Ihnen verwendeten Komponenten.
- Beziehen Sie Komponenten nur von offiziellen Quellen über sichere Links. Bevorzugen Sie signierte Pakete, um die Wahrscheinlichkeit zu verringern, dass eine modifizierte, bösartige Komponente enthalten ist (siehe A08:2021 – Software- und Datenintegritätsfehler).
- Überwachen Sie, ob Bibliotheken und Komponenten nicht gewartet werden oder keine Sicherheitspatches für ältere Versionen erstellen. Wenn das Patchen nicht möglich ist, erwägen Sie die Bereitstellung eines virtuellen Patches zur Überwachung, Erkennung oder zum Schutz vor dem entdeckten Problem.

Jede Organisation muss einen fortlaufenden Plan für die Überwachung, Triage und Anwendung von Updates oder Konfigurationsänderungen während der gesamten Lebensdauer der Anwendung oder des Portfolios sicherstellen.

Beispielhafte Angriffsszenarien

Szenario Nr. 1: Komponenten werden normalerweise mit denselben Berechtigungen wie die Anwendung selbst ausgeführt, sodass Fehler in einer Komponente schwerwiegende Auswirkungen haben können. Solche Fehler können zufällig (z. B. ein Codierungsfehler) oder absichtlich (z. B. eine Hintertür in einer Komponente) sein. Einige Beispiele für entdeckte ausnutzbare Komponentenschwachstellen sind:

- CVE-2017-5638, eine Schwachstelle in Struts 2 zur Remotecodeausführung, die die Ausführung beliebigen Codes auf dem Server ermöglicht, wurde für schwerwiegende Verstöße verantwortlich gemacht.
- Während das Internet der Dinge (IoT) häufig nur schwer oder gar nicht gepatcht werden kann, kann es von großer Bedeutung sein, sie zu patchen (z. B. bei biomedizinischen Geräten).

Es gibt automatisierte Tools, die Angreifern helfen, nicht gepatchte oder falsch konfigurierte Systeme zu finden. Die Shodan IoT-Suchmaschine kann Ihnen beispielsweise dabei helfen, Geräte zu finden, die immer noch unter der im April 2014 behobenen Heartbleed-Sicherheitslücke leiden.

Verweise

- OWASP Application Security Verification Standard: V1 Architektur, Design und Bedrohungsmodellierung
- OWASP-Abhängigkeitsprüfung (für Java- und .NET-Bibliotheken)
- OWASP-Testleitfaden – Kartenanwendungsarchitektur (OTG-INFO-010)
- Best Practices für das virtuelle OWASP-Patching
- Die unglückliche Realität unsicherer Bibliotheken
- MITRE Common Vulnerabilities and Exposures (CVE)-Suche
- Nationale Schwachstellendatenbank (NVD)
- Retire.js zum Erkennen bekanntermaßen anfälliger JavaScript-Bibliotheken
- GitHub-Beratungsdatenbank
- Datenbank und Tools für Sicherheitshinweise der Ruby-Bibliotheken
- SAFECode Software-Integritätskontrollen [PDF]

Liste der zugeordneten CWEs

CWE-937 OWASP Top 10 2013: Verwendung von Komponenten mit bekannten Schwachstellen

CWE-1035 2017 Top 10 A9: Verwendung von Komponenten mit bekannten Schwachstellen

CWE-1104 Verwendung nicht gewarteter Komponenten von Drittanbietern