Schwachstellen-Radar

Möglichkeiten des Application-Security-Posture-Managements (ASPM)

Die Bewertung und Priorisierung von Schwachstellen aus verschiedenen Quellen ist für selbstentwickelte Anwendungen ohne Kontextinformationen zu den Anwendungen schwer vorzunehmen. Lösungen zum Application-Security-Posture-Management (ASPM) können hier sowohl der Informationssicherheit als auch den Entwicklungsabteilungen wertvolle Erkenntnisse liefern.

Von Benjamin Häublein, Heilbronn

Seit dem Log4Shell-Vorfall im November 2021, bei dem eine Schwachstelle im Logging-Framework Log4j bekannt wurde, die Remote-Code-Execution ermöglichte, hat das Thema „Drittanbieterkomponenten und Supply-Chain" auch in der Softwareentwicklung erhebliche Aufmerksamkeit erhalten.

Man könnte nun meinen, dass die Maßnahmen, die seitdem typischerweise ergriffen wurden, hinreichend sein müssten, um bei zukünftigen Vorfällen schnell handeln zu können. So ist zum Beispiel die Erstellung einer „Software Bill of Materials" (SBOM), die alle in einer Anwendung enthaltenen Drittanbieterkomponenten mit der jeweiligen Version auflistet, gut geeignet, um für eine Anwendung beispielsweise festzustellen, ob sie eine verwundbare Version von Log4j enthält. Wird die SBOM mit Informationen aus Schwachstellendatenbanken verknüpft (z. B. der National Vulnerability Database – NVD des NIST), lassen sich verwundbare Drittanbieterkomponenten tatsächlich bequemer als zuvor identifizieren und nötigenfalls austauschen.

Nach den Berichten über kompromittierte Drittanbieterpakete im letzten Jahr – zum Beispiel beim Shai-Hulud-2.0-Vorfall – geriet die Hoffnung jedoch ins Wanken, dass eine Behandlung solcher Ereignisse nunmehr deutlich einfacher sein würde als zuvor. Es zeigte sich vielmehr, dass trotz der Existenz von SBOMs weiterhin problematische Fragestellungen bleiben:

- Liegen SBOMs für alle Anwendungen vor?
- Liegen SBOMs für jeden Versionsstand vor?
- Wie steht es um Entwicklungszweige?
- Welche verschiedenen Versionen und Entwicklungszweige sind irgendwo in Betrieb?
- Wie stark sind Anwendungen beziehungsweise verschiedene Versionen und Entwicklungszweige exponiert?

All diese Informationen sind für alle Anwendungen notwendig, um innerhalb einer Organisation das Risiko durch eine bekannt gewordene Schwachstelle oder einen Supply-Chain-Angriff angemessen abschätzen zu können.

Überblick und Kontext

Um diese Bewertung vornehmen zu können, bedarf es somit auch einer zentralen Übersicht über alle entwickelten Anwendungen – zusammen mit Schwachstellen, welche diese Anwendungen betreffen, sowie Metainformationen zu den Anwendungen. Hier kommen Lösungen zum Application-Security-Posture-Management (ASPM) ins Spiel, die einen ganzheitlichen Blick auf selbstentwickelte Anwendungen und deren Sicherheitsstand ermöglichen.

Hierzu werden Softwareentwicklungsplattformen (z. B. GitHub oder GitLab) angebunden, um dort alle Projekte und Repositories eines Unternehmens zu erfassen. Durch die Anbindung von Plattformen, auf denen Anwendungen betrieben werden (z. B. Kubernetes-Cluster oder Cloud-Plattformen wie AWS, Microsoft Azure oder GCP) sind ASPM-Produkte in der Lage, Projekte in

den Entwicklungsplattformen mit den betriebenen Anwendungen in Verbindung zu bringen. So lässt sich etwa durch eine automatische Analyse der Plattform, auf der eine Anwendung betrieben wird, unter anderem feststellen, inwiefern diese über das Internet erreichbar ist. Über die Softwareentwicklungsplattform kann man wiederum erfassen, wie aktiv die Entwicklung einer Anwendung voranschreitet und wie viele Entwickler beteiligt sind, woraus (zumindest rudimentär) abzuleiten ist, welche Priorität die jeweilige Anwendung für eine Orgainsation hat.

Mit diesen Informationen ist die Beantwortung der letzten beiden oben genannten Fragen bereits in greifbare Nähe gerückt. Manche Lösungen können darüber hinaus noch weitere Informationen erfassen: Beispielsweise lässt sich aus im Code hinterlegten Datenbankzugriffen ableiten, welche Art von Daten verarbeitet werden – ob es sich etwa um Kreditkartendaten oder personenbeziehbare Informationen handelt.

Schwachstellenanalyse

Derartige Meta-Informationen sind zwar ein Kern von ASPM, das namensgebende Thema „Security" wird damit aber nur periphär abgedeckt. Hierzu bieten ASPM-Produkte die Möglichkeit, auch Schwachstellen für die behandelten Anwendungen zu erfassen; bei dieser Funktionalität findet man allerdings wesentliche Unterschiede bei den Produkten auf dem Markt. Die meisten sind zumindest in der Lage, andere Tools einzubinden, um Schwachstellen von dort zu importieren: So lassen sich zum Beispiel Lösungen zur statischen Codeanalyse anbinden und die hierdurch identifizierten Schwachstellen in die ASPM-Lösung zu übernehmen. Ein anderes Beispiel ist die Integration von Lösungen, die Schwachstellen in Software-Containern identifizieren, die zum Beispiel das Basisimage oder Berechtigungen zur Laufzeit betreffen. Tools, die Anwendungen mit simulierten Angriffen zur Laufzeit analysieren, oder Cloud-Plattformen analysieren, um Konfigurationsschwachstellen zu identifizieren, können ebenfalls dabei helfen, Schwachstellen in einer ASPM-Lösung zentral zu aggregieren.

Darüber hinaus bringen manche ASPM-Produkte auch eigene Tools mit, die in der Lage sind, Anwendungen auf Schwachstellen zu prüfen. In vielen Fällen sind dann Open-Source-Tools eingebettet, um einen schnellen und vor allem einfachen Weg zu bieten, Schwachstellen in eigenen Anwendungen zu identifizieren. Häufig sind das Tools zur statischen Codeanalyse, die direkt aus dem ASPM auf die verwalteten Projekte anwendbar sind. Die ASPM-Lösung selbst orchestriert dann meist die Scans, erfasst die Schwachstellen, kontextualisiert sie mit Metainformationen und ermöglicht die weitere Verfolgung der Schwachstellenbehandlung.

Gerade der letzte Schritt ist eine wesentliche Verbesserung gegenüber dem direkten Einsatz von Open-Source-Lösungen, die häufig nur eine Momentaufnahme über die Kommandozeile bieten. Analog unterstützen manche ASPM-Lösungen zum Beispiel auch Open-Source-Tools, die „Infrastructure as Code" (IaC) überprüfen, oder im Code hinterlegte Secrets identifizieren – Nachverfolgung und Kontextualisierung erfolgen hier ebenfalls in der Plattform.

Einige ASPM-Lösungen bieten zusätzlich zu oder anstelle von Open-Source-Tools auch eigene Scanner zur Analyse der Anwendungen. Die Ziele der Hersteller sind

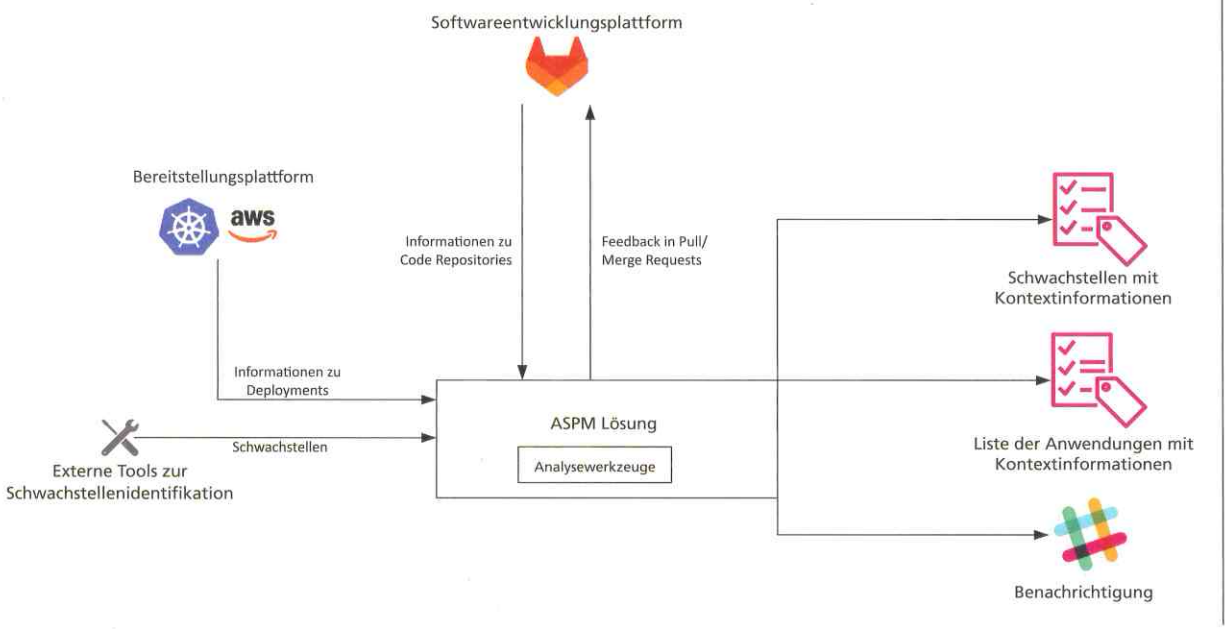

Abbildung 1: Typische Anbindungen von ASPM-Lösungen

dabei, eine bessere Integration in ihre Lösung oder eine bessere Qualität der Ergebnisse zu erreichen.

Neben der Identifikation von Schwachstellen im Anwendungscode bieten die meisten ASPM-Lösungen auch Mechanismen, um einerseits selbst eine SBOM für alle in der Lösung verwalteten Anwendungen zu erstellen und andererseits auch erkannte Drittanbieterkomponenten auf bekannte Schwachstellen hin zu prüfen. Damit sind dann auch die ersten drei Fragen aus der eingangs angeführten Liste beantwortet: Die anwendungsspezifische SBOM liegt in der ASPM-Lösung vor und wird mit Informationen zu bekannten Schwachstellen in den Komponenten verknüpft.

Transparenz und Priorisierung

Sobald Metainformationen zu Anwendungen und Informationen zu Schwachstellen in den Anwendungen (oder enthaltenen Drittanbieterkomponenten) vorliegen, kann ein ASPM-Produkt seine volle Wirkung entfalten:

_____ Wenn einer Schwachstelle Eintrittswahrscheinlichkeit und ein allgemeines Schadenspotenzial sowie
_____ der betroffenen Anwendung Metainformationen (Etwa Interneterreichbarkeit oder Priorität) zugewiesen sind,
dann ermöglicht die Kombination dieser Informationen ein differenziertes Bild über alle Anwendungen und Schwachstellen hinweg.

Eine Schwachstelle in einer Anwendung, die nicht über das Internet erreichbar ist, führt dabei dann nicht zum gleichen Risiko wie bei einer netzoffenen Anwendung – eine Anwendung, die vertrauliche Daten verarbeitet, ist dann relevanter als eine reine Marketingwebsite. So kann man mithilfe der ASPM-Lösung gezielt priorisieren – oder die Priorisierung erfolgt sogar anhand der Metainformationen direkt durch die Lösung.

Steht einem ein derart mächtiges Werkzeug zur Verfügung und eine neue weit reichende Schwachstelle wie Log4Shell wird bekannt, dann ist man fundamental handlungsfähiger, als viele Unternehmen es zuvor waren:

_____ Erstens verfügt man über eine Übersicht aller betroffenen Anwendungen und
_____ zweitens ist für jede Anwendung bekannt, ob sie über das Internet erreichbar ist.

Die Priorität zur Behandlung der verschiedenen Anwendungen lässt sich direkt aus diesen Informationen ableiten und damit die Behandlung effektiv steuern.

So lässt sich nun auch erkennen, warum die primäre Zielgruppe der ASPM-Lösungen das Security-Management ist, da durch diese Lösungen eine gewisse Transparenz in die Softwareentwicklung gebracht wird. Dennoch bleibt ASPM auch für die Entwickler selbst hilfreich: Denn diese Produkte unterstützen nicht nur reaktiv – viele Lösungen ermöglichen eine direkte Integration in die Entwicklungsumgebungen mit dem Ziel, mögliche Schwachstellen frühzeitig zu erkennen und idealerweise schon zu beheben, bevor sie überhaupt in die Softwareentwicklungsplattform gelangen.

Aber auch danach gibt es noch Hilfe durch ASPM-Lösungen: Zum Beispiel können sie sogenannte Pull- oder Merge-Requests untersuchen, bei denen ein Entwickler Code bereitstellt, der dann in den Hauptzweig des Anwendungscodes übernommen werden soll. Je nach Lösung kann dann bei der Identifikation von Schwachstellen entweder ein entsprechender Kommentar zum Merge-Request hinterlegt oder der Merge sogar unterbunden werden, bis der problematische Code entfernt oder explizit akzeptiert wird. So gelangen (erkannte) neue Schwachstellen gar nicht erst in die produktive Anwendung.

Ein unmittelbares Feedback erleichtert Entwicklern ihre tägliche Arbeit, da sie Informationen zu eventuellen Schwachstelle bearbeiten können, während das Wissen zur entsprechenden Codestelle noch frisch ist. Darüber hinaus lässt sich der „Schmerz", den etwa Ergebnisse eines Penetrationstests kurz vor dem Release einer Anwendung verursachen, etwas lindern, da hoffentlich einige der ansonsten erst dann gefundenen Schwachstellen bereits zuvor identifiziert und behoben werden konnten.

Fazit

ASPM-Lösungen bieten eine wertvolle Unterstützung, indem sie einen Überblick über eine große Anzahl von Anwendungen, ihre Schwachstellen und Rahmenbedingungen geben. Das erleichtert die Einordnung und Priorisierung identifizierter Schwachstellen oder macht das überhaupt erst fundiert möglich. Das kann zum Beispiel für die Erfüllung von Anforderungen des EU Cyber-Resilience-Act (CRA) an die Softwareentwicklung sehr hilfreich sein. Möglichkeiten, um Schwachstellen bereits frühzeitig im Entwicklungszyklus zu finden und zu beseitigen, runden das Bild weiter ab. ■

Benjamin Häublein ist Senior Berater bei cirosec.