Supply-Chain-Security: Wie Angriffe auf Zulieferer die IT-Sicherheit bedrohen

Supply-Chain-Angriffe gefährden zunehmend die IT-Sicherheit. Direkte Angriffe auf Zulieferer, Manipulationen in Open-Source-Projekten und Angriffe über Drittanbieter-Infrastrukturen erfordern angepasste Sicherheitsstrategien, um Unternehmen vor diesen Bedrohungen zu schützen.

Prof. Dr. Sebastian Höhn,
Lukas Platter

In einer zunehmend vernetzten Welt stehen Unternehmen vor neuen, oft unterschätzten Bedrohungen: Angriffe auf die Lieferkette. Doch was genau verbirgt sich hinter dem Begriff «Supply-Chain-Attack (SCA)» und wie gross ist die tatsächliche Gefahr? Bei SCAs zielen Cyberkriminelle darauf ab, Schwachstellen in der Lieferkette zu identifizieren, auszunutzen und dadurch einen Angriff auf viele Unternehmen gleichzeitig durchzuführen. Indem sie über vertrauenswürdige Partner agieren, versuchen sie, unentdeckt zu bleiben und sich Zugang zu den IT-Systemen der Zielunternehmen zu verschaffen. Die vergangenen Jahre haben durch eine Vielzahl von Vorfällen gezeigt, welch verheerende Auswirkungen solche Angriffe haben können. Um das Bewusstsein für diese Bedrohung zu schärfen und Unternehmen dazu anzuregen, ihre Sicherheitsstrategien und das Risikomanagement intensiv und kontinuierlich zu überprüfen, werden in diesem Artikel drei besonders bekannte Beispiele für SCAs beleuchtet. Abschliessend werden Empfehlungen gegeben, wie sich Unternehmen zukünftig besser schützen können.

Direkte Angriffe auf Zulieferer

Im Fall der Solarwinds-Attacke gelang es den Angreifern, den Entwicklungsprozess des Unternehmens zu kompromittieren und so Malware, in Form einer Backdoor, in ein Update der Netzwerkmanagement-Software Orion einzuschleusen. Dieses manipulierte Update wurde daraufhin an Kunden ausgeliefert, was den Angreifern unbemerkt Zugang zu zahlreichen IT-Systemen verschaffte. Die Folgen waren gravierend: Zur Wiederherstellung der Integrität ihrer Systeme mussten Unternehmen und Behörden diese umfassend überprüfen und Sicherheitslücken schliessen. Der Vorfall führte zu enormen Kosten und erschütterte das Vertrauen in Softwarelieferanten. Bis heute gilt dieser Angriff als einer der grössten in der Geschichte von SCAs.

Solche Angriffe zielen direkt auf die Systeme der Hersteller oder den Softwareentwicklungsprozess ab, um Schadcode zu verteilen. Sie sind besonders schwer zu erkennen, da die manipulierte Software von den meisten Sicherheitsmechanismen zunächst nicht entdeckt wird. Der Hersteller stellt beispielsweise auch gültige digitale Signaturen für das Update aus, da der Angriff zu diesem Zeitpunkt noch unbekannt war. Für die Kunden sah das Update aufgrund der gültigen digitalen Signatur völlig legitim aus. Diese Signatur bestätigt aber lediglich, dass die Software nach ihrer Freigabe durch den Hersteller nicht verändert wurde. Allerdings fand die Kompromittierung bereits beim Hersteller statt, wodurch die Schadsoftware schon in die signierte Version integriert war.

Indirekte Angriffe auf die Supply-Chain

XZ Utils ist ein beliebtes Tool zur Datenkompression, welches auf den meisten Linux Distributionen standardmässig vorinstalliert ist und auch im Linux-Kernel sowie in Netzwerkprotokollen wie z. B. SSH verwendet wird. Bei der als «XZ-Upstream» bezeichneten Attacke konnte sich ein Angreifer das Vertrauen der Open-Source-Community erarbeiten, bevor er bösartigen Code einschleuste. Bemerkenswert an diesem Fall ist die lange Dauer des Angriffs: Über einen Zeitraum von mehr als zwei Jahren leistete der Angreifer gute und verlässliche Arbeit für die Open-Source-Community, um schliesslich eine Schadfunktion einbauen zu können.

Diese Angriffe manipulieren das Vertrauen und die sozialen Beziehungen (a.k.a. Social-Engineering) der Entwickler-Community, was sie besonders gefährlich macht. Besonders problematisch ist es, wenn Angreifer bereit sind, über Jahre hinweg zuverlässig zu arbeiten und sich so einen guten Ruf in der Community aufbauen.

Kompromittierung der Infrastruktur

Der «Polyfill-Vorfall» verdeutlicht einen weiteren Angriffsvektor: In diesem Fall übernahm eine böswillige Entität die URL polyfill.io und konnte dadurch schädlichen Code in Webseiten integrieren. Polyfill ist eine weit verbreitete Bibliothek, die in Webapplikationen eingesetzt wird, um Unterschiede in den Implementierungen der verschiedenen Webbrowser mittels einer Java Script-Abstraktionsschicht zu eliminieren. Diese Bibliothek wird häufig über ein Content-Delivery-Network (CDN) bereitgestellt, sodass Anbieter von Webapplikationen sich nicht um die Verteilung kümmern müssen, sondern diese als Dienstleistung beziehen.

Im vorliegenden Fall wurde die URL der Bibliothek von der Community an eine chinesische Firma verkauft, die anschliessend über diesen Zugang Malware in die Browser einbinden konnte. Es ist wichtig festzuhalten, dass der Provider des CDNs, der das Open-Source-Projekt bis zum Verkauf durch die kostenlose Bereitstellung der Bibliothek unterstützt hat, nicht am Angriff beteiligt war und auch keine Möglichkeit hatte, den Angriff zu verhindern oder zu erkennen. Ob der Verkauf der URL durch Strohleute innerhalb der Community initiiert wurde, bleibt zum gegenwärtigen Zeitpunkt Spekulation.

Ein zweiter Angriffsvektor ist die sogenannte «Dependency-Confusion». Dabei machen sich Angreifer das Standardverhalten von Tools zur Verwaltung von Drittanbieter-Bibliotheken bei der Softwareentwicklung zunutze. Die grundlegende Idee dieser Tools ist es, stets aktuelle Versionen zu verwenden, um die Anwendungen immer auf dem neuesten Stand zu halten. Die meisten dieser Tools (z. B. die Paketmanager von Python und NodeJS) bevorzugen Quellcode aus öffentlichen Repositories. Viele Projekte verwenden neben diesen öffentlichen Bibliotheken auch interne Bibliotheken aus firmeneigenen Repositorys. Aufgrund der Standardkonfiguration vieler Tools kann ein Angreifer jedoch eine Bibliothek mit dem gleichen Namen wie eine interne Bibliothek in einem öffentlichen Repository veröffentlichen. Wenn er dieser Bibliothek eine ausreichend hohe Versionsnummer gibt, wird sie von den Tools automatisch heruntergeladen und in die Anwendung integriert. Dadurch können nahezu beliebige Schadfunktionen in die Applikation eingebaut werden.

Diese Angriffe nutzen die komplexen Infrastrukturen und Beziehungen zwischen Zulieferern, um die Lieferketten von Softwareanwendungen zu manipulieren.

Ausblick und Handlungsempfehlungen

Direkte, indirekte und auf die Infrastruktur gerichtete Angriffe unterscheiden sich zwar in ihrer Methodik, zielen jedoch alle auf die Kompromittierung der Integrität von Software ab.

Supply-Chain-Angriffe werden komplexer, was zeigt, dass herkömmliche Sicherheitsstrategien nicht mehr ausreichen. IT-Professionals müssen sich kontinuierlich weiterbilden und proaktiv Sicherheitslücken schliessen.

Eine Schwachstelle in der Lieferkette kann ausreichen, um erhebliche Schäden anzurichten. Um Supply-Chain-Angriffe zu verhindern, ist eine umfassende Sicherheitsstrategie entscheidend. Unternehmen sollten Zero-Trust-Prinzipien implementieren, um den Zugriff streng zu kontrollieren, und durch Schulungsprogramme das Cyberrisiko-Bewusstsein steigern. Meldepflichten, sorgfältige Lieferantenprüfungen durch Audits und Zertifizierungen sowie Risikoanalysen minimieren die Auswirkungen. Die fortlaufende Überwachung der Lieferkette mittels SIEM, EDR, Patch-Management und Schwachstellenscans ermöglicht eine frühzeitige Bedrohungserkennung. Ein robuster Incident Response Plan gewährleistet schnelles Handeln im Ernstfall. ■