L-018

攻撃者による SBOM を用いた脆弱性管理に関する検討
Study on vulnerability management using SBOM by attackers

五反田 和也† 八槇 博史†
Kazuya Gotanda Hirofumi Yamaki

1. はじめに

システム管理者は SBOM を導入することで、ソフトウェアサプライチェーン上のコンポーネントやライブラリを容易に把握や管理ができる。SBOM の普及に伴い、SBOM 関連ツールが多数登場した。この一連の動きが、攻撃者に与える有利な側面も検討すべきである。本研究では、攻撃者が SBOM 関連ツールを攻撃用ツールとしてサイバー攻撃を行うシナリオを想定する。攻撃者による SBOM 関連ツールの悪用方法と実現可能性、攻撃を成立させるために必要な要件について検討する。そして、その攻撃への対策について考察する。

2. 想定シナリオと SBOM の活用

攻撃者による SBOM を用いた脆弱性管理のシナリオとして以下を想定する。

攻撃者は攻撃対象である企業に対してサプライチェーン攻撃により攻撃する。サプライチェーン攻撃とは、攻撃対象の企業の子会社や取引先といった関連する企業にも攻撃を仕掛ける攻撃である。関連企業にも攻撃することにより、業務停止によって経済的被害を与えることや、当該企業を踏み台とし新な攻撃へとつなげることが可能となる。

攻撃に使用する脆弱性は、ゼロデイ脆弱性を用いたゼロデイ攻撃を想定する。ゼロデイ攻撃とは、新たに発見された脆弱性を用いて、パッチが適用される前に攻撃を仕掛ける手法である。そのため、ゼロデイ攻撃を成功させるためには、パッチ適用までの時間内に一連の攻撃をする必要がある。

シナリオにおけるゼロデイ攻撃の工程を述べる。まず、サプライチェーン上の企業がゼロデイ脆弱性を含んでいるかを偵察する。偵察結果を受け、当該脆弱性を含むサービスが存在するならば、攻撃を仕掛ける。先述したようにゼロデイ攻撃を成功させるためには、この一連の動作をパッチ適用前に完了しなければならない。そのため、この一連の動作に要する時間を削減することで、攻撃の成功確率が向上すると考えられる。

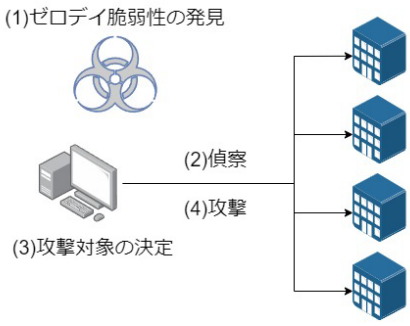

図 1　シナリオにおける一連の攻撃

ここで、攻撃者は前もって攻撃対象のサービスのソフトウェア構成を把握していると仮定する。ソフトウェア構成情報を把握することで、ゼロデイ脆弱性を含むサービスが存在するかの偵察行為を行うことなく、攻撃対象となるサービスを特定することができる。そのため、ゼロデイ脆弱性を発見した段階で、迅速に攻撃フェーズへと移行することができる。攻撃者にとって攻撃対象のサービス構成情報は有益である。

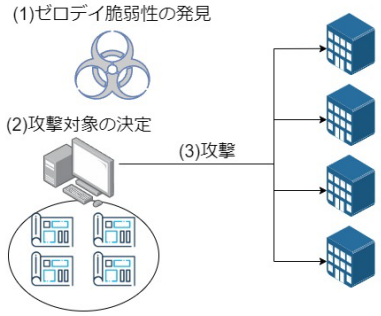

図 2　シナリオにおける SBOM を用いた一連の攻撃

しかし、攻撃対象サービスのソフトウェア構成情報を攻撃者が管理することは容易ではない。本実験では、攻撃者が外部から攻撃対象サービスの SBOM を作成することにより、サービスのソフトウェア構成情報の管理を図る手法を提案する。SBOM を作成することによって、サービスの構成要素となるライブラリやコンポーネントを把握することが可能となる。SBOM 管理ツールは多数存在し、大量の SBOM の一元管理も容易である。そのため、攻撃者があらかじめ SBOM を所持していることにより一連の攻撃の高速化を図ることが可能となる。

外部から SBOM を作成するために必要な要件や、本手法の実用上の有用性について次項より検討する。

3. 外部からの SBOM 作成

外部から SBOM を作成するためには、サービス名とバージョン情報が必要である。

攻撃者が SBOM を作成する手順を述べる。まず、サービス名とバージョン情報を取得するために偵察行為を行う。これらの情報を元に、サービスを再構築する。SBOM 作成ツールを用いて当該サービスの SBOM を作成する。この一連の動作により、外部から SBOM を作成することが可能となる。

しかし、偵察行為によって得られた情報に誤りがある場

† 東京電機大学大学院 システムデザイン工学研究科 情報システム工学専攻 (Information System Engineering, Graduate School of System Design and Technology, Tokyo Denki University)

合や、情報が十分に得られない場合には、再構築するサービスに差異が生じる。そのため、本来の SBOM を完全に再現することはできない。

サービス名とバージョン情報を取得する手段として、能動的偵察アプローチと OSINT アプローチを提案する。

3.1 能動的偵察アプローチ

能動的偵察アプローチは、ネットワークスキャンにより能動的にサービス名やバージョン情報を偵察する手法である。

3.2 OSINT アプローチ

OSINT アプローチは、OSINT ツールを用いてサービス名やバージョン情報を偵察する手法である。OSINT とはオープンな情報源からの情報収集により知見を得る手法である。能動的偵察アプローチは、攻撃グループが独自の手法を使用することにより、攻撃グループが特定可能な痕跡が残る可能性がある。OSINT ツールを用いることで特定可能な痕跡を残さずに偵察することができる。

4. 実験

攻撃対象サービスを Docker コンテナ上に WordPress を用いて構築した。サービス名とバージョン情報を、能動的偵察アプローチと OSINT アプローチによって取得した。偵察により得られた情報を元に Docker 上にサービスを再構築した。攻撃対象のサービスと再構築したサービスそれぞれの SBOM を作成した。それぞれの SBOM の脆弱性情報の一致率を求めることで、外部から SBOM を作成する手法の実用上の有用性を評価した。SBOM のフォーマットに SPDX と CycloneDX を用いた。SBOM 作成ツールに Syft、脆弱性スキャンツールに Grype を用いた。能動的偵察アプローチのネットワークスキャンツールに WhatWeb を用いた。OSINT アプローチの OSINT ツールに Wappalyzer を用いた。

5. 結果

能動的偵察アプローチによって得られた情報は、Apache httpd/2.4.59、Debian、WordPress/6.5.2、PHP/8.2.18 であった。 OSINT アプローチによって得られた情報は Apache httpd/2.4.59、Debian、WordPress/6.5.2、PHP/8.2.18 であった。両アプローチによって得られた情報は同一だった。そのため、コンテナ上に再構築するサービスも等しくなり、両アプローチの結果は等しくなった。

再構築したサービスから作成した SBOM と、攻撃対象のサービスから作成した SBOM の脆弱性情報を比較し、一致した脆弱性と当該 CVSS スコアの表を表 1 に示す。

表内 CVSS 列の” (2”表記は CVSS バージョン 2 の結果を使用していることを表し、それ以外は CVSS バージョン 3 の結果を用いている。また、”n/a”は当該 CVSS のスコアリングがなされていない脆弱性を表す。

表 1　一致した脆弱性と当該 CVSS スコア

CVE IDs	CVSS	CVE IDs	CVSS
CVE-2007-0086	7.8(2	CVE-2023-29383	3.3
CVE-2007-2728	5.0(2	CVE-2023-3164	5.5
CVE-2007-3205	5.0(2	CVE-2023-45918	n/a
CVE-2007-4596	7.5(2	CVE-2023-50495	6.5
CVE-2016-2781	6.5	CVE-2023-5363	7.5
CVE-2017-13716	5.5	CVE-2023-5678	5.3
CVE-2018-10126	6.5	CVE-2023-6129	6.5
CVE-2018-6952	7.5	CVE-2024-0727	5.5
CVE-2021-45261	5.5	CVE-2024-2236	n/a
CVE-2022-27943	5.5	CVE-2024-26458	n/a
CVE-2022-3219	3.3	CVE-2024-26461	n/a
CVE-2022-4900	5.5	CVE-2024-26462	n/a
CVE-2024-28835	n/a	CVE-2024-28834	n/a

攻撃対象のサービスに含まれる脆弱性は 165 個あった。そのうち、攻撃者が作成した SBOM によって推定できた脆弱性は 26 個であった。一致率の結果は 15.7%となった。ここでの一致率とは、攻撃対象のサービスをもとに作成した SBOM から検出可能な脆弱性のうち、攻撃者によって推定が成功した脆弱性の個数の割合とする。

SBOM フォーマットによる結果の差異はなかった。

6. 考察

実験結果より、外部からの SBOM 作成により脆弱性情報を管理することは実現可能であることがわかった。つまり、攻撃者は攻撃対象サービスのサービス名とバージョン情報により、攻撃対象の脆弱性情報とソフトウェアの構成情報を管理することが可能となる。

脆弱性の一致率は、攻撃者の偵察スキルとサービスの再構築スキルに依存する。実験では Debian ではなく、Debian 系 OS の Ubuntu 上にサービスを構築した。それに起因し、再構築したサービスの SBOM に差異が生じた。偵察結果を元に、高い精度でサービスの再構築が可能ならば、より一致率の高い SBOM が作成可能であることが考察される。

攻撃者に外部から SBOM を作成されないためには、バージョン情報の秘匿化が重要である。具体的には、問い合わせに対してバージョン情報を返答しない設定や、エラーページで不必要な情報を公開しない設定にするといった対策があげられる。

7. おわりに

攻撃者が脆弱性管理に SBOM を外部から作成する手法は有用であり、実現可能であることを示した。一致率の結果は、攻撃者の偵察スキルとサービスの再構築スキルに依存する。そのため、一定以上のスキルを持つ攻撃者であればある程度 SBOM を推定することができる。外部から SBOM を作成させないようにするためには、バージョン情報を秘匿化する対策法が有効である。

参考文献

[1]面 和毅, 中村 行宏,”サイバー攻撃から企業システムを守る! OSINT 実践ガイド”,株式会社日経 BP,2023,442p