THE SOLARWINDS が もたらした転換点

digicert®

目次

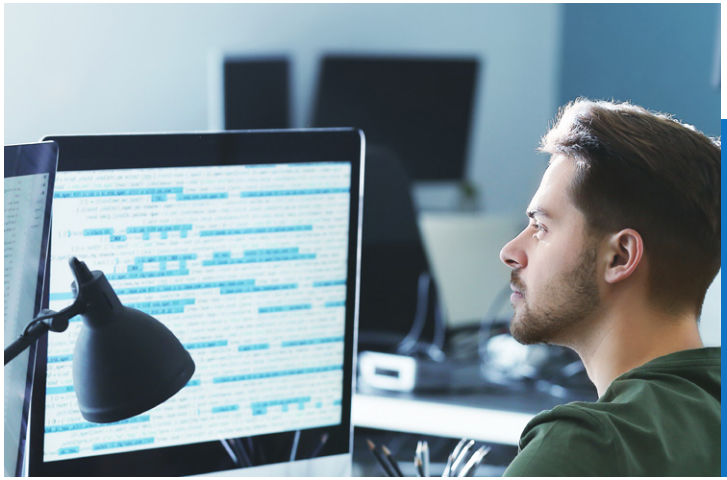
大規模なサプライチェーン強盗

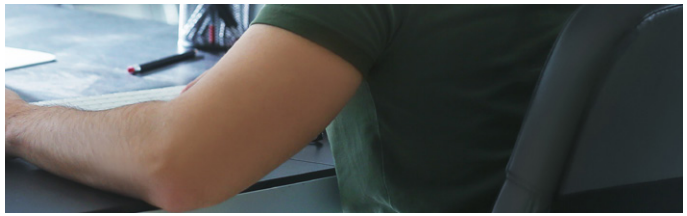
2020 年 12 月 13 日（日）、米国政府は連邦政府の複数のコンピューターネットワークがハッカーの侵入を受けていたことを認めました。その後の数週間で、その巧妙な攻撃はロシアの諜報機関によるものにほぼ間違いないとの発表があり、何カ月間にもわたって密かに機密データが流出し続けていたことが判明しました。さらに、専門家の調査によって侵害の範囲についても明らかになりました。これは、ただの諜報活動の成功事例ではなく、ここ数年で最大規模のマルウェア侵入でした[1]。

攻撃の元を追っていくと、サイバーセキュリティ企業の SolarWinds（米テキサス州）が開発、販売した IT 管理プログラム、Orion に辿り着きました。SolarWinds は瞬く間にデータ侵害の代名詞となりました。しかし、セキュリティの損害に直面した企業は SolarWinds が初めてではありません。また、ニュースの見出しではストーリーの全貌は語られません。

実際に侵害が始まったのは攻撃が開始されるよりもかなり前、悪意のあるコードが Orion の DNA に仕込まれたときからでした。そのコードは CI/CD ビルドの一部となり、まるで時限爆弾のようにソフトウェアの内部に隠されていたのです。ハッカーたちはそのまま辛抱強く待ちました。それは非常に長い導火線を持つ時限爆弾でした。彼らは、待つことによって最大の損害を与えられることを知っていたのです。その後何カ月にもわたり、Orion は世界中に拡散され、政府機関、大企業、他のソフトウェア会社にもインストールされていきました。

そして 2020 年 3 月を過ぎたある日、時限爆弾は爆発しました。侵入に気付いたときにはもう手遅れでした。マルウェアが実行されたからというだけでなく、Orion が至る所で導入されていたからです。1 つの侵害箇所、1 台のサーバーを隔離すれば済むという問題ではありませんでした。パッチが必要なのは 1 カ所だけではありませんでした。攻撃の規模はあまりに大きく、感染したシステム、企業、組織の総数は今も不明で、今後もわからないままでしょう。





1 https://www.reuters.com/article/us-cyber-solarwinds-microsoft/solarwinds-hack-was-largest-and-most-sophisticated-attack-ever-microsoft-president-idUSKBN2AF03R

SolarWinds前とSolarWinds後

SolarWinds への攻撃で驚かされるのはその規模の大きさです。感染した Orion ソフトウェアは 33,000 以上の顧客に販売されました。「Sunburst（雲間から突然現れる強烈な太陽の光）」という、いかにもな名前を持つ悪意のあるコードは、18,000 もの組織に配布されました。このトロイの木馬は、何も知らない何万社もの企業や政府機関のネットワークのファイアウォールの内部に、何カ月もの間潜んでいました。

その後、フォレンジック調査によって米国フォーチュン 500 企業のうち 425 社、米国のトップ 10 電気通信会社すべて、米国のトップ会計事務所のうち 5 社、さらには Microsoft 社までもが Sunburst の侵入を受けていたことが判明しました。ホワイトハウス、米軍の 5 軍すべて、財務省、国防総省、国務省、司法省、国土安全保障省、国家安全保障局、国家核安全保障局が、Sunburst のせいで外国の諜報機関からのアクセスを許してしまいました。米国以外では、NATO、英国政府、欧州議会などが侵入を受けました。署名付きで安全であるはずのソフトウェアによって運ばれた Sunburst は、公共機関から民間企業、大企業から中小企業に至るまで各ゲートを掻い潜りました。

ほんの 10 年前、サイバーセキュリティの専門家はサプライチェーン攻撃が起こる可能性は極めて低いと述べていました。それはサイバー犯罪者が付け込む種類の脆弱性の特性に合致していなかったためです。1 人の人間あるいは 1 つの会社がビルドのチェーン上のすべてのリンクを保護することは不可能であるため、サプライチェーンを保護するための取り組みは無駄なタスクでしかありませんでした。この考えがあまりに広く浸透していたため、SAFECode の Stacy Simpson 氏は業界のリーダーと共同で DevOps 専門家に警告を喚起するレポートを発行し、そのような見方を捨てて、CI/CD サプライチェーンの侵害ポイントをクローズする、コード署名のベストプラクティスの導入を呼びかけました[2]。

しかし、ほとんどの人はそれでも、サプライチェーン攻撃を実行することも、あるいは予防することも不可能だと思い込んでいました。SolarWinds の侵害があって、世界は学習しました。

予防できたはずの大惨事に関するフォレンジック調査

SolarWinds 攻撃につながった脆弱性のタイプは決して目新しいものではありません。感染したコードは、セキュリティを重視する専門家の間で CI/CD パイプラインの草創期から問題視されていたものでした。しかし、SolarWinds の侵害前は、これほど大規模で世間の注目を浴びる大惨事が起きたことはなく、しかも完全に予防できるはずのものだったのです。

2 http://safecode.org/publication/SAFECode_Software_Integrity_Controls0610.pdf

多くの企業と同様、SolarWinds は Orion を正規の署名付きソフトウェアパッケージとしてリリースしました。ソフトウェアに署名することにより、SolarWinds はビルドおよびリリース後に改ざんされたり、マルウェアに感染したりした場合は通知されることを顧客に約束していました。署名付きソフトウェアが侵害を許したことで、デジタル世界は一旦立ち止まってサプライチェーンプロセス全体を再検討せざるを得なくなりました。署名前にコードにマルウェアが仕込まれ、サプライチェーンに入り込む可能性があるとすれば、一体どのように顧客やエンドユーザーを守ればよいのでしょうか。

コード署名そのものが実証済みの確実なテクノロジーであることには変わりありません。しかし、SolarWinds の侵害により、コード署名だけではサプライチェーンを守るのに十分でないことが明らかになりました。コード署名に加えて、ソースコードのスキャン、サードパーティライブラリのスキャン、職員の適切な身元調査、安全なビルド環境がなければ、ソフトウェアは攻撃に晒される可能性があります。それは家にセキュリティシステムを設置しているにもかかわらず、毎回 1 時間しか稼働させないようなものです。

SolarWinds 後、世界はサプライチェーン攻撃の新たなベクター（運び屋）について学び、ソフトウェアの署名は DevOps サイクルの最後のステップかもしれないが、それだけでは不十分であることを痛感しました。プロセス全体を保護し、コントロールし、検証する必要があります。完全なセキュリティが確保されなければ、DevOps のほころびが埋まることは決してありません。

署名はビルドの最終ステップではなく、すべてのステップで行うべきもの

ソフトウェアに署名すると、署名した瞬間から改ざんに対して保護されるようになります。それはただの暗号化ではありません。コードに署名することで、その開発元の身元が検証され、ダウンロード後にコードの完全性がチェックされ、ソフトウェアが改変されていないことをユーザーが確認できるようになります。署名付きソフトウェアは、ソフトウェアが信頼できることを示します。そのため、マルウェアに感染されていないことを検証可能な方法で DevOps チームがソフトウェアリリースを提供することが大前提となります。署名付きソフトウェアにマルウェアが含まれていると、公共またはプライベートのソフトウェア配布ネットワークの至る所に存在するコード署名チェックをすべてすり抜けてしまいます。つまり、侵害されたコードが含まれていても、署名付きソフトウェアであるがために、サプライチェーンの他の部分およびユーザーアクセスポイントのセキュリティゲートウェイに、問題を探さなくてよいと指示することになります。

CI/CD プロセスにおいてソフトウェアの完全性を保護することが、完全な信頼性を提供する唯一の方法です。チームの開発サイクルの最後にソフトウェアに署名するだけでは不十分です。サプライチェーン内の他のチームが開発したコード（オープンソースコードやサードパーティライブラリコードを含む）は、スキャンしてマルウェアやその他の異常がないかチェックする必要があります。ソフトウェアのビルドにはクリーンなコードのみを使用し、適切な方法で発行、管理、保管されている正当な署名鍵を使って署名する必要があります。署名鍵へのアクセスは、署名鍵の使用権限を持つユーザーまたは自動ビルドサーバーのみに制限します。

それは非常に面倒な作業に思えるかもしれません。そういうわけで、開発者はよく脆弱なセキュリティ対策や抜け道を使って開発を進めようとします。スピードと効率が DevOps の肝であることは誰もが知っています。納期遅延につながる煩雑な手順は、CI/CD プロセスを停滞させかねません。早くないソリューション、むしろ遅延や中断を発生させるようなソリューションは、そもそもソリューションと呼べません。

多くの場合、署名プロセスは自動化できます。人が介入する必要を減らすことで、セキュリティ体制の完全性について妥協することなく、開発者はコードの開発に専念できます。

ベストプラクティスと安全でないプラクティス

ソフトウェアの署名自体は確かに重要ですが、署名ポリシーおよび手順の管理の重要性は見過ごされがちです。そしてそこに付け入る隙が生まれます。ソフトウェアのセキュリティには 2 つの選択肢があります。ベストプラクティスを選択するか、そうせずにサプライチェーンを攻撃に晒すかです。

コード署名のベストプラクティスを導入すれば、署名はサプライチェーンの弱点にはなりません。むしろ、コード署名を正しく導入すれば、マルウェアの挿入防止にも役立ちます。

1. 署名鍵と秘密鍵を保管し、使用を管理する

秘密鍵は複製されたり、多くの場所に保管される可能性があります。USB トークンなどの物理鍵は紛失したり、施錠されていない場所に保管されたりすることがあります。不満をもった従業員や不注意な従業員によって鍵の場所を見失ったり、盗まれたりすることもあります。

秘密鍵を完全に保護するには 2 つの方法があります。適切に管理された HSM（ハードウェアセキュリティモジュール）に鍵を保管するか、マネージド PKI サービスに鍵の管理を任せるかです。

2. 多要素認証と、FIPS 準拠の保管機能を導入する

パスワードは忘れることもあれば侵害されることもあります。鍵は共有されたり、紛失したり、盗まれることがあります。同様に、物理 USB トークンは不適切な場所に置かれたり、誤使用されることがあります。

人為的ミスや悪意から守るため、強力なアクセスコントロールを導入する必要があります。多要素認証（MFA）では、ユーザーのアイデンティティを示す 2 種類の形式の資格情報を要求します。HSM には強力な認証機能と耐物理タンパー性によって鍵を保護し、認証済みのユーザーのみがアクセスできるようにします。

3. トレーサビリティの高い運用を行う

エンドツーエンドの監査とユーザの把握ができなければ、鍵の不正使用や署名の異常を追跡することは困難あるいは不可能です。

署名の前に、鍵とユーザーの両方の認証が必要で、しかもこの認証および署名は CI/CD プロセスの各ステップで完了する必要があります。ユーザの紐づけを適切に行い、誰が何に署名したかを把握できるようにするには、署名を追跡して監査可能にしなければなりません。署名鍵が認証済みのユーザーによって、不正な時間に使用された場合、ソフトウェアが次のプロセスに移る前に IT セキュリティチームが問題を検出できます。

4. アカウントの権限管理を行う

権限管理を行わずに鍵を使用する場合、鍵と署名の使用を保護し監視することは困難あるいは不可能です。

鍵は権限と役割に基づいて割り当て、製品ごと、プロジェクトごと、またはチームごとに分ける必要があります。管理者が、秘密鍵と証明書を生成して鍵をチームに分配できるようにする必要があります。ユーザーは、署名関連ワークフローに割り当てられたユーザーに制限する必要があります。ビルドサーバーなどの非人間コンポーネントには、自動化を可能にするための鍵を割り当てる必要があります。

5. 組織全体で定めた権限管理を行う

組織のポリシーは、個人の役割や権限と同じくらい重要です。全体的なポリシーと制御構造がなければ、個々のセキュリティプロセス間の整合性が取れず、脆弱性が生まれる可能性があります。

組織全体から製品開発までの全面的な内部セキュリティポリシーを作成してこれを遵守します。これには、暗号化アルゴリズム、鍵のサイズ、証明書の有効期限、証明書のタイプ、承認ワークフローが含まれます。

6. 鍵の交換をする

同一の鍵を使用して複数箇所のコードに署名した場合、1 カ所から侵害されただけで全体がリスクに晒されます。

複数の鍵を迅速、簡単に発行でき、鍵をローテーションできる機能により、1 カ所の侵害が広範囲の攻撃になるのを防ぐことができます。

7. 特定の種類の鍵を特定のプロジェクトに割り当てるポリシーを制定する

さまざまな種類の鍵を綿密に制御しなければ、組織は鍵のアクセスと使用について可視性を失い、不正または不適切な鍵署名アクティビティの可能性を広げてしまいます。

特定のユーザーとチームに制限された、特定の製品またはプロジェクトに対して特定の署名鍵を再利用するようにします。可能であれば、オンザフライの秘密鍵と証明書を使用して、各リリースが一意の鍵と証明書で署名されるようにします。

8. 未署名のソフトウェアを決して自社の環境外に出さない

署名ツールに署名用のファイルをすべて転送することは、時間がかかり、多くのリソースを消費するだけではなく、転送中の傍受や改ざんの可能性を開くことにもなります。

そのすべての懸念はハッシュ署名によって解決できます。ハッシュのみをクラウドにアップロードすることで、ローカルでの署名とおなじくらい高速になります。コード自体は内部サーバーに留まるため、知的財産が盗まれたり改変されることはありません。

9. 定期的にコードやライブラリを全てスキャンする

単純なステップに思えるかもしれませんが、署名は信頼の印です。ユーザーは署名付きコードを安全なコードとみなします。開発者は「信頼できるコード」にウィルスやマルウェアが紛れ込んだ状態で顧客に送信してしまうことを避けなければなりません。

コンパイル済みのコードをスキャンするだけではなく、ソースコードとサードパーティライブラリコードもスキャンします。 悪意のあるコードが挿入されていないかチェックして、ソフトウェアがすべてクリーンであることを確かめてから署名してください。

10. 署名を再現可能にし、署名対象を検証できるようにする

ビルドをベースラインと比較できないと、バイナリが同一で、一定数のユーザーによって再現可能かどうかを確認することはできません。確認できないとマルウェアが未検出のまま仕込まれるリスクが増大します。

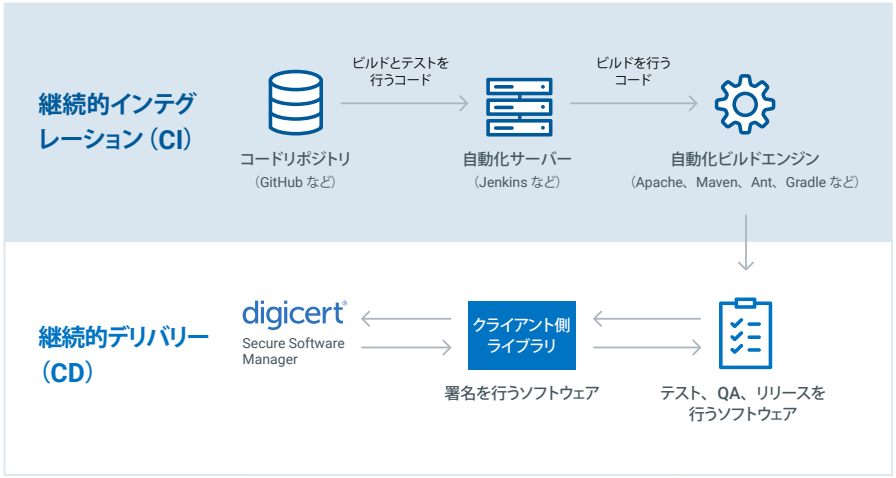
署名用に送信したバイナリのハッシュを、署名前のベースラインと比較します。この比較は、本番ビルドによって、テストおよび QA サイクルで期待される結果と同じ結果が生成されていることを検証するために必要です。

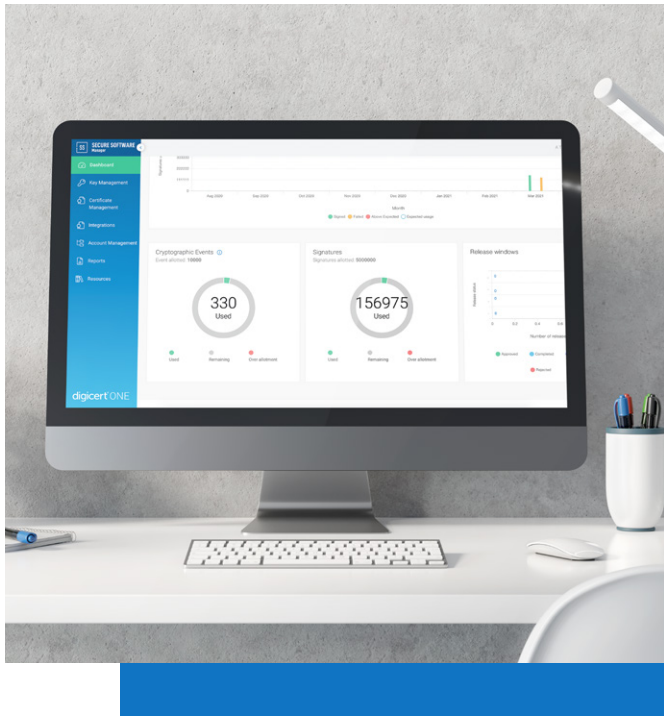
継続的署名がスピードへの対応を促進する

DigiCert Secure Software Manager は、スピード重視の DevOps を促進するために設計されています。Secure Software Manager は自動化プロセスを活用することで、余計な作業も遅延も発生せず、エンドツーエンドのセキュリティを含む最高レベルの信頼を提供する継続的な署名を可能にします。

コード、ソフトウェア、バイナリに対し、迅速かつ簡単にさまざまなスケールで署名できます。侵害されたコードをリリースする前にマルウェアの侵入を検出するために、ベースラインパラメータを自動化する再現可能なビルドプロセスを作成します。鍵の管理を保護し、権限ベースのアクセス制御を行い、レポートおよび追跡可能を使用して可視性と説明可能性を実現します。DigiCert Secure Software Manager は、CI/CD プロセスの各段階でコード署名ベストプラクティスを実装、維持管理するためのワンストップソリューションです。

DigiCert® Secure Software Manager を使用した CI/CD プロセスによる自動ソフトウェア署名





対岸の火事とせず、今すぐ教訓を生かしましょう

Stacy Simpson 氏が CI/CD サプライチェーンの保護を呼びかける SAFECode レポートを発行したとき、彼女自身もそれが潜在的な攻撃の域を超えてデジタル史上最大とも言える極めてリアルな攻撃になるとは思ってもみなかったことでしょう。2010 年当時、サプライチェーン攻撃の侵害が起こる可能性は極めて低いと考えられていたのです。その 10 年後に起きた、SolarWinds の侵害は破壊的な不正と窃盗の代表的事例とみなされるようになりました。

皮肉なことに、CI/CD サプライチェーンの脅威を解決する方法は 2010 年にすでにわかっており、SAFECode のレポートに記載されていました。DevOps の各チームや単一の組織ではチェーン全体をコントロールすることは不可能であるため、サプライチェーンサイバー攻撃に対処する唯一の方法は、開発中にコードの完全性を継続的にモニタリングするベストプラクティスを導入することです。ビルドプロセスに関わる全員がコードをスキャンし、署名すれば、脆弱なリンクに対してサプライチェーン全体が保護されます。

2010 年当時は、そんなことはとても実現できないと思われていたかもしれません。セキュリティ対策によって CI/CD プロセスに遅延や中断が生じるようでは、対策自体が侵害のリスクと同じくらいの悪影響を DevOps に及ぼしてしまう可能性があります。

しかし今、コード署名は自動化できます。自動化するだけでも DevOps で必要なツールが提供されるので、CI/CD 全体における攻撃対象が大幅に制限されます。自動化を導入することで、DevOps チームは継続的な署名を実装し、最小限の人的介入によってコード、ソフトウェア、アプリをエンドツーエンドの完全性と暗号化システムで保護できるようになります。

SolarWinds 侵害の成功を見て、その他のサイバー犯罪者が既に独自のサプライチェーン攻撃を開始している可能性は大いにあります。その脅威から身を守るための手段の 1 つは、業界を挙げてコード署名のベストプラクティスに向けて動くことです。サプライチェーンの全員がコードを監視し保護していれば、各々の開発のつなぎ目はパイプラインの次のポイントへの引き渡しの中で保護されます。

つまり、SolarWinds はサイバー犯罪史の驚きの逸話というだけではありません。SolarWinds の件はコードおよびソフトウェア開発の歴史における重要な分岐点となりました。DevOps の世界は、何の警告も受けていなかったとは言えません。潜在的なリスクがあることは知られていました。この現実的な大惨事は、今や明白な事実です。プロセスを遅延も中断もさせることなく脅威から容易に保護するためのツールは存在します。今こそ業界全体でこのようなツールを活用し、コード署名のベストプラクティスを導入し、DevOps サプライチェーンを最初からリリースまで保護できるようにするときです。