Project Blackfin

インテリジェントエージェントを使用した
侵害検知の自動化

PROJECT BLACKFIN

F-Secure

目次

概要 3

はじめに 4

異常検知の方法論 - サイバーセキュリティにおける考察 7

プロセスの動作 – そのプログラムは正常に動作しているのか？ 7
リソース使用量 - オブジェクトは通常通り使用されているか？ 7
ユーザの行動 - ユーザの行動は正常か？ 8
データ表現とマシンラーニングアプローチに関して 8
異常検知モデルの組み合わせによる特定の現象の検知 9

設計するうえで考慮すべきこと 17

拡大した侵害の検知のメリット 19

今後の展開について 20

向上した脅威インテリジェンス収集 20
対応アクションの自動化 20
新たに出現する特性 21

まとめ 22

エフセキュアについて 23

概要

本ペーパーでは、ネットワーク上の攻撃者が実行したアクションを正確に追跡するために設計された分散型異常検知のアプローチについて説明します。図で示めされたアプローチでは、エンドポイントとネットワークの両方、および集中バックエンドで複数のマシンラーニングモデルを実行し、これらのモデル間で学習状態の通信と複製を行います。本ペーパーで説明するリサーチは「Project Blackfin」と呼ばれるものであり、複数年にわたる調査プロジェクトの第 1 弾として F-Secure によって実施されました。

本ペーパーに記載されているメソッドの一部は、エフセキュア独自の動的侵害検知ソリューションに既に実装されています。このプロジェクトの短期的な目標には、以下の項目が含まれています。

(i) 敵対行為を検知するための、より汎用的な新しい手法の開発
(ii) ネットワーク上の複数のエンドポイントに対する攻撃者の行動を追跡できるメカニズムの開発
(iii) 脅威インテリジェンス収集機能の向上および自動化
(iv) 自動応答アクションを実装および改善する方法の理解
(v) 各エンドポイントでコンテキストリスク分析を実行できるメカニズムの実装

はじめに

現代の侵害検知戦略は、一般に、エンドポイントおよびネットワークトラフィックデータのストリームの収集と集約、発生時のデータの処理と分析、およびその後の分析と監査のために中央の場所にデータを保存することに依存しています。発生データは、手書きのルールとマシンラーニングモデルを含むアルゴリズムによって処理されます。新しい戦術、技術、および手順（TTP）とその侵害指標（IoC）が発見されると検知ロジックが更新され、新たに発見された攻撃ベクトルを見落としていないことを確認するために、過去の履歴データとの照合を行うことがあります。

後述するその性質により、収集されたデータの集約に依存する集中検知アプローチでは、複数の異なるソース間でイベントを相関させたり、新しく発生したイベントを履歴データを迅速に相関させたりする能力が制限されてしまっています（特に、これらの重要なイベント間に大きな時間差がある場合）。また、攻撃者の行動の兆候を検知するために必要な大量のデータをコスト効率よく処理するのは至難の業です。これらのソリューションは、主に攻撃者の行動の個々のインスタンスを迅速に識別するように設計されており、多くの場合、ネットワーク上の複数のエンドポイントで攻撃者の過去のアクションを追跡するうえでの効果が低下します。侵害の兆候が検知された時、多くのケースでは手動の検証手順が必要とされ、その後「ヒト」で構成されたチームがその対応にあたります。応答アクションには、侵害されたシステムの分離、フォレンジックの実行による攻撃者のアクションの判定、攻撃者のネットワークへのアクセスを排除するためのシステムのクリーニングまたは再イメージングが含まれます。

攻撃者がエンドポイントの制御、偵察の実行、他のエンドポイントへのラテラルムーブメント、永続化メカニズムの確立、データ漏洩、および単方向または双方向のコマンドアンドコントロールチャネルの確立に使用する手法には、様々なメカニズムが用いられています。多くの場合、複数のエンドポイントからの様々なタイプのシステム情報の調査が必要です。攻撃者が実行したアクションの検知に関連する情報には、プロセスの作成、ネットワーク接続、ログオンイベント、名前付きパイプの作成、モジュールの読み込み、ファイルアクセス、レジストリアクセス、システムログエントリなどがあります。敵対的なアクションの痕跡のほとんどは、オペレーティングシステムまたはネットワーク上で自然に発生するアクティビティと混在してしまうため、攻撃者がおこなう可能性のある全てのアクションを検知するために多くの異なるメカニズムが必要となります。さらにそれらのアクションは、システム、一連のシステム、またはネットワーク上の数千の無害なイベントと正確に区別する必要があります。

通常、攻撃者が実行するアクションによって生成されるイベントはごく少数となります。例えば、ターゲットエンドポイントでのプロセスの作成、ソースとターゲット間のネットワーク接続、そしてターゲットエンドポイントでの名前付きパイプの作成のみが、攻撃者によるラテラルムーブメントの痕跡となります。システム管理者が組織内のマシンでリモート管理タスクを実行すると、

非常によく似たイベントが生成されます。攻撃者のアクションをシステム管理者のアクションと区別する方法はいくつかあります。ソース IP アドレスがシステム管理者のマシンのいずれにも属していないこと、リモート実行が組織の管理ツールキットの一部ではない、またはターゲットマシンで実行されるコマンドが管理者によって通常発行されるコマンドではなかったこと。全ての場合において、十分に自信を持って敵の活動を見つけるには、事前に知識を持ち、正規のシステム管理者の IP アドレスのリストや既知の管理者ツールのリストなどの全てが文書化されているとは限らないターゲット環境を熟知している必要があります。

攻撃者がシステムでアクションを実行すると、1 つまたは複数のシステムで単一または非常に少数のイベントが生成され、ネットワーク上の隣接するエンドポイントから他の何千ものイベントとともに発生します。そのため、ネットワーク上の複数のエンドポイントからの発生イベントを集約および分析するように設計されたシステムでは、そのアクションを検知するために、非常に具体的なルール、または複数のエンドポイントからのデータの複雑な相互相関が必要となります。したがって、発生イベントのストリームを分析することを前提に構築されたシステムは、適切にアラートを出すことに失敗ケースが多く見受けられます。イベントが一般的すぎて見逃すことになったり、逆に注意しすぎて管理者をが大量の誤検知に翻弄されることがあります。このようなシステムからアクション可能な情報を取得するには、誤検知を抑制するように設計されたルールの策定とメンテナンスが必要です。これは非常に時間と手間のかかる作業です。

ネットワーク上の複数のマシンで、数日、数週間、さらには数ヶ月にわたって、攻撃者のわずかなアクションを自動検知するように設計することは、非常に複雑な作業です。侵入が検知されると、経験豊富なフォレンジック専門家のチームが対応にあたりますが、侵入者がとったアクションや侵入中にどのシステムにアクセスしたかのタイムラインを作成するにはかなりの時間がかかるという事実を認識することが大切です。自動化されたシステムが攻撃者の過去のアクションをリスト化するには、数千の個別のエンドポイントから収集された特定の量の履歴データに対して複雑なロジックを実行する必要があります。

異常検知メソッドは、環境内の動作のベースラインを確立し、そのベースライン内に収まらないイベントが発生したときに警告を発します。例えば、企業のネットワーク上のトラフィックに適用される異常検知を使用して、どの IP アドレスが互いに直接通信するかをベースラインで理解し、通常は直接通信しない 2 つの IP アドレスが突然通信を開始した場合にアラートを出すことができます。

本ペーパーでは、ネットワークを介して攻撃者が実行するアクションを正確に追跡するための分散型異常検知のアプローチについて詳しく説明します。ここで説明するアプローチでは、エンドポイントとネットワークの両方、および集中バックエンドで複数のモデルを実行し、通信そしてこれらのモデル間で学習した状態の複製を含みます。この方法により、以下が可能になります。

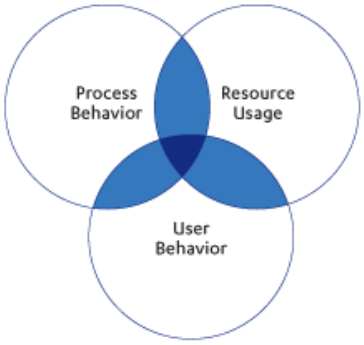
- 脅威をより適切かつ効率的に検知し、より迅速で正確な対応アクションにつなげる。
- エンドポイントレベルでの脅威検知方法と、ネットワーク上のシステム全体で関連する異常の理解を共有するメカニズム。
- 検知機能を改善するための組織の状態、ローカリティ、またはサブネット (単一のマシンではない) の積極的な利用。
- ラテラルムーブメントのアクションの発見の自動化に関する手法。
- 攻撃者がネットワーク上のどこでどのように操作しているかを理解するための自動化についての手法。

本ペーパーの目的は、このプロジェクトの一部として実施されたリサーチについて説明し、説明された方法論を強化するために使用できる追加のメカニズムを提案し、説明されたアプローチに伴う潜在的な利点と落とし穴を挙げていくことです。エフセキュアが実施している「Project Blackfin」と呼ばれる、複数年にわたるこのリサーチプロジェクトは、社内のエンジニア、リサーチャー、データサイエンティスト、そして外部のアカデミックパートナーとの協業により推進されています。説明したメカニズムの一部は、エフセキュアの製品に既に実装されています。

異常検知の方法論 - サイバーセキュリティにおける考察

異常な動作を検知するための汎用的なアプローチの 1 つが、一般的な動作のベースラインの作成です。これは、対象のオブジェクト、システム、またはユーザを一定期間プロファイリングすることによって行われ、低レベルのオペレーティングシステムイベントから、ユーザが示す行動パターン、ネットワーク上のオブジェクト間の相互作用まで、組織内の様々なデータソースのプロファイルを作成できます。ユースケースに応じて、プロファイリングはエンドポイント、サーバ、ネットワーク、または複数のエンドポイントからのデータの集約されたストリーム全体で実行されます。

攻撃者を特定するためには、攻撃者の行動を可視化する可能性のあるデータソースを理解する必要があります。これにアプローチする 1 つの方法は、問題空間を 3 つのカテゴリに分けることです。



プロセスの動作 – そのプログラムは正常に動作しているのか？

例に見られるパターンにあるような、実行可能プロセスの作成、プロセス間の親子関係（コンピュータ上のどの実行可能プロセスが他のどのプロセスを起動したか）、そしてモジュールロードやシステムリソースアクセスなどのプロセス自体によって実行されるアクションは、攻撃者が使用する多くの戦術（未知の脆弱性の悪用、実行可能ファイルのトロイの木馬化など）を発見するうえで役立ちます。

リソース使用量 - オブジェクトは通常通り使用されているか？

異常なリソース使用パターン（CPU 負荷、メモリ使用量、ディスクのリード／ライト、ネットワークのリード／ライト）は、ランサムウェア、クリプトマイナー、データ持ち出し技術、または侵入者によって開始されたその他のカスタムタスクなど、悪意のある行動を発見する際に役立ちます。

ユーザの行動 - ユーザの行動は正常か？

特定のユーザのコンテキストで、(どこから、どこへを含む) ログオンイベントとログオフイベント、アプリケーションの起動、実行時間、アクセスしたネットワークの場所、アクセスしたファイル (ファイルサーバ上のファイルを含む) をプロファイリングすることにより、侵入者の存在、データの流出、または実際にユーザが悪意を持って振る舞うことを示すパターンなどの異常な動作の発見につながります。

上に挙げた 3 つの項目は、異常な振る舞いを見つけ出すための多くの方法の一部に過ぎないことに留意してください。

データ表現とマシンラーニングアプローチに関して

使用可能な各データソースは、様々な方法でまとめられ実行される可能性があります。例えば、実行可能プロセスに関連するデータは、シーケンス、ノードエッジグラフ、タイムライン、様々な集計、または単純なカウンターとしても表すことができます。また、選択したデータ表現に応じて、各データソースの分析またはモデリングに使用できる多くの方法があり、それらの方法は様々な方法で組み合わせることもできます。エンドポイントマシンで実行またはトレーニングされるように設計されたモデルを検討する場合、リソースの使用が懸念事項となります。より多くのリソースを消費する手法は、エンドポイントよりもバックエンドシステムでの使用により適している場合があります。

AI とマシンラーニングの最新技術を検討する場合、ディープラーニングのアプローチは、その人気に比例して最も多く用いられています。ディープニューラルネットワークは、多くの困難なタスクにおいて比類のない成果をもたらしており、今後もそうあり続けることが予想されます。エフセキュアでもディープラーニングを使用しており、ディープラーニングに反対するつもりはまったくありませんが、効果的なマシンラーニングベースのソリューションを実装するための唯一の実行可能なアプローチとは言えないと考えています。最新のレビューでは、動的攻撃の検知にはディープラーニング以外のアプローチの方がより効果的であることが多いことが示されています。ただし、最近のいくつかの NLP (自然言語処理) およびオブジェクト認識方法は、この領域において有効であるとされるケースもあります。私たちは、多くのアプローチを試し、私たちが解決しようとしている問題にとって最適な方法を選択しました。

一部の領域では、分類精度の最大化が主な目的です。これらの領域では概ね、事後分布の最も正確なポイント推定値を生成するアプローチが望まれます。画像認識などのアプリケーションがこのカテゴリーに分類されます。アカデミックデータセットで実行される従来のテストは、人間の

精度に到達しそれを超えることに重点を置いていました。ディープニューラルネットワークなどの複雑な分類器は、多数のサンプルによってトレーニングされた場合、非常に柔軟な決定境界が作成されるため、こうした問題に対しての非常に優れたソリューションとなります。

他の領域においては、モデルが下した決定の結果は平等と呼ぶにはほど遠いものです。場合によっては、失敗は非常に費用がかかるものとなるか、致命的になることさえあります。医療やサイバーセキュリティなどの分野では、どの選択肢が最適かを単に選択するだけでは十分ではありません。正しい意思決定を行うためには、意思決定に対する確信を理解することが必要です。意思決定の不確実性を考慮し、事後分布の推定値を提供し、モデルに事前知識をエンコードできるモデリングアプローチにより、最終決定に関連するリスクをより明確に理解できます。このようなモデリングアプローチは常に「平均的」なものであり最適とは限りませんが、領域によっては重要な意思決定とその結果の理解の両方に最適となる場合があります。

動的な攻撃の検知に使用するように設計されたモデルは、絶えず変化する環境（ソフトウェアやハードウェアもかなり定期的に変更される場合があります）において、適切に機能する必要があります。また、変更可能な環境でタスクを実行できるモデルは、長期間にわたって正確に実行される可能性があります。攻撃に対するモデルの堅牢性は、敵対者がいる場合にも重要です。そのため、特に連合学習または分散学習のアプローチも使用されている場合、より単純で堅牢なモデルは、データやモデルポイズニングなどの一般的な敵対攻撃に対抗するのにより適しています。より単純なモデルは、意思決定の説明可能性など、他の利点ももたらす可能性があります。これにより、誤検知と検知漏れの識別が大幅に容易になります。複雑なモデルの個々のデータポイントの影響を理解することは困難な作業ですが、より複雑なモデルでしか解決できない問題もあるため、解釈が難しいモデルです。多くの場合トレードオフですが、意識的に行われる必要があります。何事にも、特効薬となるものは存在しないのです。

異常検知モデルの組み合わせによる特定の現象の検知

前述の各カテゴリに関連付けられた複数の異なるモデルのアウトプットを組み合わせることにより、システムで発生していることをコンテキスト面から理解し、ダウンストリームロジックが特定のイベントまたはアイテムが異常であるかどうか、そしてアラートを出す必要の有無をより正確に予測することができます。このアプローチにより、特定のロジックを検知システム自体に組み込むことなく、攻撃者のアクション（またはアクションのシーケンス）を検知するための一般的な方法論が可能になります。これらの設計パラメーター内に構築されたシステムは、次のような質問に正確に回答できるはずです。

- そのプログラムは正常に動作しているか？
- そのオブジェクトは正常に使用されているか？
- ユーザの行動は正常か？

上記の項目のうち 2 つ以上を組み合わせることで、複数の観点から対処できる異常を検知する手法となります。ここで、これらの概念を実際の例を用いて説明します。

例: ネットワーク上の通常は見受けられない潜在的に有害な行動の発見

セキュリティリサーチャーの直観としては、cmd.exe を起動する winword.exe のインスタンスは、システム上の悪意のあるアクティビティのシグナルである可能性が高いことを示しています。システム上のプロセス作成イベントに関する統計情報を収集する単純なモデルを利用することにより、悪意のあるアクティビティを示す実行可能ファイル間の親子関係を一般的な方法で識別できるはずです。ここでは、疑わしいものかどうかを確認するために、プロセス作成イベントをスコアリングする方法を示します。プロセス作成イベントは、次の場合に異常と判断されます。

子プロセスはよく見受けられるものである - 子プロセスは通常 (任意の名前の親プロセスによって) 生成される　かつ　親プロセスは共通である - 親プロセスは通常 (任意の名前の) 新しいプロセスを開く　かつ　親プロセスとその子プロセスの組み合わせはまれである – 特定の名前の親プロセスによって、この特定の名前の親プロセスの子プロセスが開かれることはほとんどありません。

最初の例に対して、このロジックをいくつかの単純な正規化で適用すると、winword.exe と cmd.exe の両方が一般的なものですが、winword.exe と cmd.exe のペアはまれであるため、異常なアクティビティの強いシグナルとなります。そして、このプロセスの組合せはスピアフィッシング攻撃でよく見られるものです。Metasploit によって生成されたペイロードも同様に機能します。この方法論は、特に未知の悪意のあるサンプル、特に標的型攻撃用に意図的に作成されたサンプルを迅速に識別するための一般的なメカニズムを表しています。例えば、winword.exe がネットワーク上のコンピューターで cmd.exe を起動する場合、このアクションの原因となったドキュメントに関心があります。この手法は、この分野ですでに実証済みです。

異常なプロセス作成イベントを検知するために説明した方法論は、ネットワーク上の複数のエンドポイントに学習を分散するためにフェデレーションラーニングメカニズムを簡単に利用できるシステムの良い例です。フェデレーションラーニングは、本ペーパーで説明している他のいくつかの方法にも適用できます。

フェデレーションラーニング

フェデレーションラーニングは、多数のクライアントに分散されたトレーニングデータを使用して高品質の集中型モデルをトレーニングすることを目標とするマシンラーニング設定です。スマートフォンの予測テキスト入力は、フェデレーションラーニングシステムの例のひとつです。予測テキスト入力は、スマートフォンのキーボードに入力するときに単語の補完を提案し、提案はスマートフォン自体で実行されるマシンラーニングモデルによって提供されます。スマートフォンのユーザが提案された単語を選択するか、提案を無視して手動で単語を完成すると、スマートフォンのモデルはそのユーザの好みを学習します。時間の経過とともに、ローカルモデルは、ユーザが望む単語を提案することでより向上します。ネットワーク上の各スマートフォンは、ローカルモデルを中央サーバに定期的に送信し、中央サーバでは、送信されたモデルのパラメーターを使用して中央モデルが更新されます。次に、この中央モデルはネットワーク上の全てのスマートフォンに定期的に展開され、ローカルモデルの更新または拡張に使用されます。時間の経過とともに、中央モデルは何百万人ものユーザから学習し、更新メカニズムにより、各ローカルデバイスで実行されるテキスト予測モデルを改善します。新しいスマートフォンは、そのローカルモデルがユーザの行動から学習するまで、有用な汎用ベースラインとして機能するトレーニング済みモデルを受け取ります。

例: ランサムウェアのような振る舞いの検知

ランサムウェアは、多くの場合、ファイルシステムを横断し、特定の条件（ドキュメント、画像、ビデオなどのファイルタイプ）に一致するファイルを暗号化するマルウェアの種類です。ランサムウェアプロセスは、ファイルシステム内のファイルを順番に列挙および変更する一意であり、通常は信頼できないプロセスであるため、異常な振る舞いを示します。ファイルシステムベースの異常検知ロジックの観点からは、ランサムウェアプロセスによって示される動作パターンは他のほとんどのプロセスと比較して非常に不規則であり、特にプロセス自体に関する関連メタデータ（実行可能ファイルの署名、構造、システムに到着した日時、ダウンロード元などに関する情報など）と組み合わせた場合、直ちにアラートが必要になる可能性があります。一部のソフトウェア自動更新プログラムが、新しいファイルを書き込むのではなく、ディスク上のファイルを変更するというのは事実です。ただし、このようなアップデーターはかなり定期的に実行され、毎回同じファイルのセットで変更を実行し、実際に信頼される可能性があります。そのため、アクションはモデルのベースラインの一部として記録される可能性が高いため、アラートは生成されません。

例: ラテラルムーブメントの兆候の検知

攻撃者によるラテラルムーブメント行動の検知は、既知のセキュリティ違反検知方法では解決することが難しいタスクでした。ユーザログオンがシステムの所有者または敵のどちらに属しているかを判断する問題について考えてみましょう。全てのユーザはユニークであり、出社時間、休憩時間、休暇取得の時期、職種や職務、閲覧する Web サイトなど、全てが異なります。各ユーザに意味のあるベースラインを作成する唯一の真の方法（誰かがマシンに侵入したのであって、単なるマシン上での動作ではないことを正確に検知できるようにするため）は、各ユーザを個別にプロファイルすることです。通常、組織レベルまたは役割レベルでユーザの行動をプロファイリングしようとすると、過度に一般化され、各ユーザの行動の意味を理解できないモデルになります。このような汎用モデルは、一般的な営業時間、週末、祝日など、ユーザ間の共通の共有動作を学習するのに役立ち、標準のソフトウェア使用状況とネットワークトポロジに関する情報も学習できます。

各ユーザを個別にプロファイルすることが、攻撃者を発見する可能性を高めるうえで役立ちます。ただし、特定のユーザのアクティビティのベースラインだけでは、ログオンイベントが侵入者に属しているかどうかを判断するには不十分な場合があります。しかし、複数の特殊なモデルの出力を組み合わせるアプローチを検討するのであれば、そのような決定を行うことは非常に容易になります。では、以下の項目を組み合わせて使用するとどうなるかを考えてみましょう。

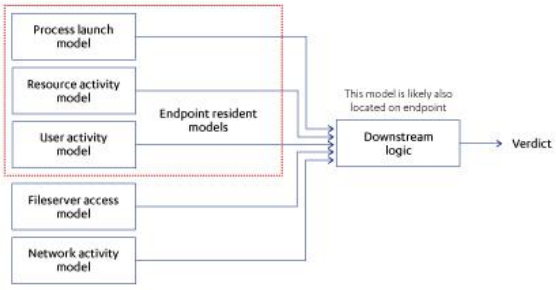
- ユーザのベースラインプロセス起動動作パターンを追跡するモデル - ユーザカウントがプロセスを起動および終了する時、それが何のプロセスであるか、そしてユーザが起動したどのプロセスが他のプロセスを起動するのか。
- ユーザのシステムのベースラインリソースアクティビティパターン（CPU 負荷、メモリ使用量、ディスクアクティビティ、ネットワークアクティビティなど）を追跡するモデル。
- ユーザのログオンおよびログオフパターンをキャプチャするモデル。
- サーバ上のファイルへのアクセスを含む、ユーザのファイルアクセスパターンを追跡するモデル。
- 組織のネットワーク全体のネットワークトラフィックパターンを追跡するモデル（ネットワークセンサーモデルまたはバックエンドモデル）。

上記のモデルのアウトプットを組み合わせることにより、ダウンストリームロジックは、攻撃者によって生成されたパターンと、影響を受けるシステムのユーザによって通常生成されたパターンをより簡単に区別することができます。攻撃者がそのユーザのシステムに侵入した場合、次のような事実の組み合わせにより検知をすることができます。

- 攻撃者が、ユーザがこれまでほとんどまたは全く起動していないプロセスを起動した。
- 攻撃者が、ユーザが通常アイドル状態である時間帯に、エンドポイントでアクティビティを実行した。
- 攻撃者が、ユーザがこれまで一度もまたはほとんどアクセスしたことのないファイル（例えばファイルサーバ）にアクセスした。
- 攻撃者が、ユーザがこれまでに行ったことがない、またはめったに行わない他のエンドポイントへのネットワーク接続を行った。

次に、説明した検知プロセスの 1 つの可能な例を示します。

集約モデルによって学習されたベースラインによっては、これまでに説明されたアプローチは、インサイダーによる脅威など、悪意のあるユーザ行動の検知するのに役立つことがあります。例えば、社に不満を持っている従業員が数時間または数日間かけて内部サーバからローカル PC、クラウドサービス、または USB ドライブに大量の機密ファイルのコピーを開始しようとした場合、本ペーパーで述べている方法論を使用してアラートを生成できます。ユーザ行動のモデリングは、EU の GDPR において「潜在的にプライバシーを危険にさらす」活動と見なされ、ユーザ行動の異常を検知するように設計された機能は、関連するプライバシー要件に準拠する必要があります。



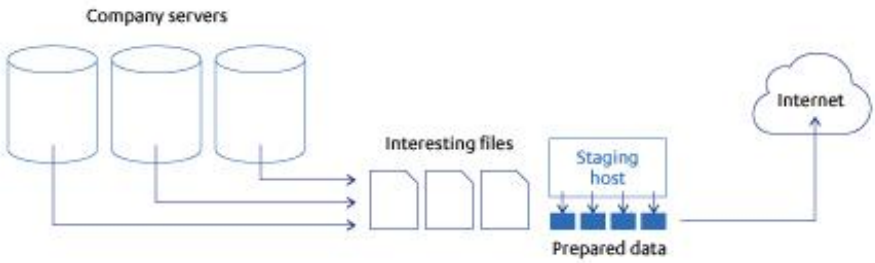
例: 機密情報への敵対的アクセスの検知

攻撃者は一般的に、組織のネットワーク内の 1 つまたは複数の主要なシステムにアクセスすると、機密データを検索して盗み出します。攻撃者はその実行のために、ファイルサーバ上の必要なファイルまたはディレクトリにアクセスする権限を持つアカウントを使用して、ネットワーク上の 1 つまたは複数のエンドポイントにログインする可能性があります。その後、ファイルサーバ上で、閲覧、検索、コピーなどの操作を実行します。企業秘密、ソースコード、顧客情報、財務記録、給与情報、戦略計画など、最終的に持ち出されるデータは、スパイ活動、恐喝、または金銭的利益に使用される可能性があります。

ファイルサーバで実行される異常検知モデルを使用して、ファイルアクセスのベースラインの動作（どのユーザがどのファイルにアクセスするか、通常どのファイルにアクセスするか、どの操作を実行するか）を学習することができます。こうしたモデルは、特に本ペーパーで述べている他のタイプのモデルと組み合わせると、ユーザが以前にアクセスしたことのない多くのファイルにアクセスした場合、ユーザが通常とは異なるエンドポイントからファイルにアクセスする場合、またはファイルサーバで再帰的なファイルアクセス操作が実行される場合など、攻撃者の兆候を検知できるシステムの構築に使用できます。どのファイルにアクセスしたか、ファイルを更新したか、どこにファイルをコピーしたか、どのエンドポイントからファイルにアクセスしたか、どのユーザアカウントが侵害されたのかなど、こうしたシステムから収集された情報は、侵入者が実行したアクションを特定するうえで非常に貴重です。

例: Web ベースの抽出メカニズムの検知

データの持ち出し（侵害されたネットワークからデータをコピー）のための様々な手法が存在し、それらのほとんどは被害者の組織のネットワーク上の一般的なトラフィックと融合するように設計されています。これらの手法には、Dropbox、YouTube、Pastebin、Github などの一般的な Web サービス、電子メール（Gmail などの無料の Web メールサービスで作成されたアカウントであることが多い）、カスタム抽出ツール、およびプロトコルの悪用（dnscat など）の利用が含まれます。豊富な経験を持つ攻撃者は、被害者のネットワークから収集した情報に基づいて、ステルスな抽出手法を選択する可能性があります。流出データは、多くの場合、被害者のネットワーク上のどこかで一時的に保存され、データ損失防止ソリューションによる検知を回避するために、流出前にパッケージ化（圧縮、暗号化、小さなチャンクに分割）されます。



個々のエンドポイントでのプロセスの動作を学習するために設計された異常検知モデルを使用して、いくつかのタイプの流出技術を発見することができます。例えば、エンドポイントの Web ブラウザープロセスが突然多くのファイルにアクセスする場合、データが Web サービスにアップロードされていることを示している可能性があります。このようなシステムのアウトプットは、ネットワーク異常検知モデルの出力と相関する場合、どのファイルがどこに流出したのかを特定するために使用できます。特定のデータ流出アクションは、予期しない大規模なデータ移動を警告

するように設計されたファイルサーバベースのモデルによっても検知できます。最後に、データを盗み出すための準備に使用される手順は、ほとんどのユーザのシステムで異常に映ります。これは通常、専用のツールとコマンドライン操作の使用を伴うためです。

例: サプライチェーン攻撃に関連する特性の検知

サプライチェーン攻撃は、攻撃者がある会社 (A 社) をまず侵害して、その会社が取引している他の会社または組織 (B 社) にアクセスする時に発生します。攻撃者は、標的である B 社の防御が堅牢で直接侵害するのが困難な場合にサプライチェーン攻撃を使用することがあり、これには様々な手法が使用されます。最近の有名なサプライチェーン攻撃には、ソフトウェアベンダーの侵害があります。これは、攻撃者がソフトウェアベンダーのソフトウェアに悪意のあるコードを送り込み、エンドユーザたちが、トロイの木馬化されたそのソフトウェアを新しいバージョンに更新する際に害をもたらすというものでした。ソフトウェアを常に最新の状態にしておくという行為が悪用されたケースです。

ShadowHammer、ShadowPad、そして CCleaner など、近年注目を浴びたサプライチェーン攻撃は、Authenticode で署名された Windows バイナリへの暗黙的な信頼を悪用することにより、攻撃者がシステムを簡単に侵害する方法です。従来のセキュリティ制御は、パフォーマンスと誤検知防止の名の下に署名済みのバイナリをホワイトリストに登録することが多いため、この種のシナリオに対抗することに苦労しています。

トロイの木馬化したソフトウェアを利用するサプライチェーン攻撃は、通常、署名手順の前に悪意のあるコードをバイナリに密かに埋め込みます。これを行うために、以下のメカニズム (単数または複数) を使用します。

- 攻撃者が、バイナリのソースコードの一部に悪意のある変更を加える。
- 攻撃者が、署名ステップの前に、悪意のある機能をコンパイル済みのバイナリに「パッチ」する (つまり、ディスク上のバイナリのバイトを変更する)。
- 攻撃者が、バイナリのビルドプロセス中に悪意のある機能が導入されるように、被害者の組織のビルド環境を変更する。

特定の条件下でバイナリ中の悪意のあるペイロードが実行されるケースがあります。これには、以下のようないくつかの潜在的な理由があります。

- ペイロードが特定の被害者または特定のシステムでのみ実行される。例としては、マルウェアが ATM マシンに組み込まれたオペレーティングシステムでのみ実行されることや、マルウェ

アはウクライナにあるマシンでのみ実行される、など。
- 自動分析プロセスを回避したり、リバースエンジニアリングを抑制したりする。例としては、実行可能ファイルが仮想マシンで実行されていないこと、またはインストルメンテーションの下で実行されている、ことを検証する、または、ペイロードが起動する前に自動分析システムが分析を終了することを期待しつつペイロードが実行前にしばらく待機すること。
- システムに前提条件が存在すること、つまりシステムに必要なライブラリと機能が含まれていることを確認する (例えば、マルウェアは Windows 7 以降のシステムでのみ動作する)。

悪意のあるペイロードは、システムモジュールによって提供される機能を利用する場合が多いです。これらのモジュールがハイジャックされたバイナリの初期化中にロードされたことが保証されていないため、ペイロードに含まれるコードはこれらの機能を利用可能にするために遅延ロード操作を実行できます。実行可能ファイルの異なるバージョン間のモジュールロードタイミングパターンの変更が検知された場合、この情報を他のメタデータと組み合わせて使用して、アプリケーションがトロイの木馬化されているかどうかを判断できます。

より一般的には、典型的な動作のベースラインを生成するために、個々の実行可能ファイルのアクションを経時的にプロファイルできます。これらのアクションには、ファイルアクセス、レジストリアクセス、モジュールのロード、プロセスとスレッドの作成、ネットワークアクセスなどが含まれます。観測された新しいバイナリの動作がプロファイルされた動作と大きく異なる場合は、トロイの木馬化を示している可能性があります。

バックエンドで、または最新のマルウェア対策プログラムによってローカルで実行される実行可能ファイルの静的分析は、脅威の発見に役立つことがあります。静的な機能や、ファイルシステム内の場所、バイナリのサイズ、ヘッダー、文字列、関連するハッシュ、インポートなどのメタデータを使用して、新しいバージョンのバイナリと以前のバージョンを比較し、振る舞いベースのインジケータと併用することで、悪意によって変更を加えられたソフトウェアを高い確率で識別し、誤報のリスクを最小限に抑えることができます。

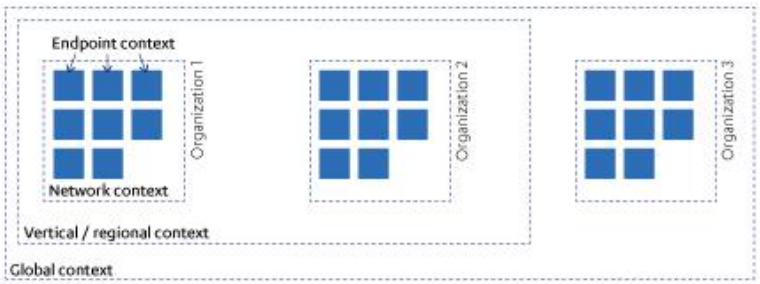
設計するうえで考慮すべきこと

本ペーパーで取り上げている例は、Project Blackfin の一部として研究されているほんの一部の検知メカニズムです。私たちのリサーチが進むにつれて、記述されたモデル間の相互作用、およびそのような相互作用がもたらす価値についての理解がより深まることが期待されています。その理解は、将来の設計決定にたどり着くうえで非常に役立つものです。

正確なベースラインを学習するには、モデルごとに異なるトレーニング期間が必要であり、一部のモデルは他のモデルよりも一般的です。例えば、異常な実行可能動作を検知するように設計されたモデルは、全てのエンドポイントシステムで同様の表現を学習します。そのため、これらのタイプの汎用モデルを事前にトレーニングし、新しいエンドポイントに実装して、直ちに効果を発揮することができます。ただし、ユーザの行動をベースラインとするように設計されたモデルでは、各個人の行動を学習する時間が必要です。こうしたケースでは、汎用的で、事前トレーニング済みのユーザの行動モデルは、勤務時間や祝日など、組織内のユーザ間で共通の行動を既に学習しているため、新しいエンドポイントに実装できます。これらの事前トレーニングモデルは、モデルが時間の経過とともにシステムの所有者の特定の動作を学習する開始点を提供します。

本ペーパー中の例のいくつかは、複数のモデルのアウトプットが後続のモデルまたはアルゴリズムへのインプットとして使用されるメカニズムを示しています。例えば、説明したラテラルムーブメント検知のメカニズムでは、エンドポイントでローカルに実行されているモデルからのアウトプットと、ネットワークおよびバックエンドで実行されているモデルを組み合わせる必要があります。同様のメカニズムが他のタイプのアクションの検知に役立つと期待されます。エンドポイントレベル、ネットワークレベル、そしてグローバルレベルの観測の組み合わせにより、様々な種類の攻撃を一般的な方法で検知できるシステムを簡単に作成できます。

連合学習などのメカニズムにより、1 つのシステムで学習した表現をネットワーク上の他のモデルと共有することができます。学習済みの表現は、様々な度合いで、同じコンテキスト内の他のユーザの役に立ちます。コンテキストは、1 つ以上の同様の属性でグループ化されたユーザまたはエンドポイントのセットであり、次の図で示されているように、コンテキストには、従業員の役割、チーム、組織、業界、または全員 (グローバルコンテキスト) を含むことができます。



実際には、各個人をよりよく理解するために、複数の重複するコンテキストを作成すると便利な場合があります。コンテキストは以下のようなカテゴリに対して作成できます。

- 金融の専門家 (システムで使用される特定のソフトウェア、または他のユーザがアクセスしない特定のネットワークリソースへのアクセスを行う者、という点において)。
- ソフトウェア開発者 (マシンで実行されている開発ソフトウェアと、システム上で新しいユニークのバイナリが検知されることが多い、という点において)。
- 夜間シフトで勤務するユーザ (ローテーション勤務を導入している企業において)。
- アドミン (管理タスクを実行するためにネットワーク上の他のマシンにアクセスする者、という点において)。

各ユーザのコンテキストは、ローカルモデルの拡張方法と、自分のモデルを共有するユーザを決定します。このように、脅威と異常は、各エンドポイントでのこれらのモデルの組み合わせに基づいて識別され、コンテキストによって決定されます。最終的には、異常の実際の意味と関連性は、異常が観察されるコンテキストから生じます。関連する脅威を検知できるようにするには、関連するコンテキストを選択できる必要があります。さらに、脅威が検知されると、その脅威が発見されたコンテキストを使用して、脅威が他にどこに存在するかを判断します。

拡大した侵害の検知のメリット

各エンドポイントで実行されているモデルは、収集された全てのデータをバックエンドにストリーミングする必要なしに決定を下すことができます。これにより、データの送信、処理、およびストレージのコストを削減でき、最終的にはユーザのソリューションのコスト効率が向上します。

また、ローカルで実行されるモデルは、脅威をより迅速にキャッチし、ネットワーク上の他のシステムと連携して異常を検知します (例えば、事実や学習の共有、情報の要求など)。

考慮すべきもう 1 つの重要事項は、ユーザのプライバシー規制です。これは、顧客の環境を離れることができるデータ、特に攻撃検知ソリューションが持つユーザの行動データに対するアクセスレベルに制限を設けるものです。これには、ドキュメントの内容、ログイン資格情報を含む URL などが含まれます。ただし、ローカルモデルはこのデータを使用することができます。以前はバックエンド中心の検知ソリューションに対して制限が設けられていたデータソースの可用性は、潜在的に新しい検知方法や能力にオープンとなる可能性があります。

今後の展開について

近い将来、標的型攻撃を特定して攻撃から防御するするために必要なプロセスには、ソフトウェアベースの技術と人間の専門知識の組み合わせが引き続き含まれるであろうことは素晴らしいことです。したがって、Project Blackfin の短期的な目標は以下のものとなります。

- 敵対行為を検知するための、より一般的な新しい手法の開発。
- ネットワーク上の複数のエンドポイントにわたる攻撃者のアクションを追跡できるメカニズムの作成。
- 脅威インテリジェンス収集機能のさらなる改善および自動化。
- 自動応答アクションを実装および改善する方法の理解。
- 各エンドポイントでコンテキストリスク分析を実行できるメカニズムの実装（各エンドポイントのリスクは、近接する他のエンドポイントで観察されるアクションによって決定されます)。

向上した脅威インテリジェンス収集

異常な動作を検知するように設計された汎用メカニズムは、必然的に新しい悪意のあるサンプルと動作を発見します。これらの発見は、脅威インテリジェンスにおける重要な出来事を表しており、その結果はリサーチャーに送信され、同じネットワークまたは他のネットワーク内で同様の脅威を識別するために使用することができます。

対応アクションの自動化

クライアント側のロジックを実装して、フォレンジックデータの収集や、検知された攻撃者からのアクションに応じてエンドポイントをネットワークから隔離するなどのアクションを実行できます。エンドポイント間の通信により、複数のマシンにまたがるフォレンシックタイムラインの収集、または攻撃に応じた複数のエンドポイントまたはネットワークセグメントの分離などの自動化された対応が可能になります。

新たに出現する特性

エンドポイント間の動的通信と、本ペーパーで説明されているコンテキストの方法でのメタデータとモデルパラメータの共有の組み合わせにより、検知機能をさらに向上させるための、まだ私たちの想像のおよばないメカニズムが明らかになってくるでしょう。さらに、本ペーパーで説明されているシステムで過去に収集されたデータを適用する実験を行うことで、一部の参加モデル内でこれまで予測されていなかった緊急行動を発見したいと考えています。発見できた場合、これらのプロパティは今後のさらなる研究の方向性の基礎の形成に貢献する可能性があります。

まとめ

エフセキュアは独自のテクノロジースタックでの実験、プロトタイピング、そして新しい方法論の実装により、将来の侵害検知、フォレンジック分析能力、そしてサイバーセキュリティソリューションにおける対応能力を飛躍的に向上させる可能性のある、エンドポイント、サーバ、およびネットワークで実行されるモデル間の豊富な相互作用を特定しました。本ペーパーでは、この分野での初期の発見のいくつかを取り上げました。取り上げた手法の一部は、既にエフセキュアのソリューションに実装されています。本ペーパーで説明されているメカニズムは、現在理解されているよりも広く問題空間に適用できる可能性があります。この分野での継続的な研究には、単一の中央モデルからより多くのロジックを移動し、ネットワーク上のエンドポイントで実行され、エンドポイントを囲む自律的で適応性のあるマシンラーニングエージェントに移行することが含まれます。これを実行することにより、自然な群知能の相互作用に基づいて、インテリジェントエージェントが共通の目標を達成するために通信および協力する集団知能技術を作り出すために、緊急行動を利用できるようになると期待しています。この長期間のリサーチは「Project Blackfin」において継続され、定期的に新しい結果として発表される予定です。

エフセキュアについて

エフセキュアほど現実世界のサイバー脅威についての知見を持つ企業は市場に存在しません。数百名にのぼる業界で最も優れたセキュリティコンサルタント、何百万台ものデバイスに搭載された数多くの受賞歴を誇るソフトウェア、進化し続ける革新的な人工知能、そして「検知と対応」。これらの橋渡しをするのがエフセキュアです。当社は、大手銀行機関、航空会社、そして世界中の多くのエンタープライズから、「世界で最も強力な脅威に打ち勝つ」という私たちのコミットメントに対する信頼を勝ち取っています。グローバルなトップクラスのチャネルパートナー、200 社以上のサービスプロバイダーにより構成されるネットワークと共にエンタープライズクラスのサイバーセキュリティを提供すること、それがエフセキュアの使命です。
エフセキュアは本社をフィンランド・ヘルシンキに、日本法人であるエフセキュア株式会社を東京都港区に置いています。また、NASDAQ ヘルシンキに上場しています。

詳細は以下のページをご覧ください

(英語) https://www.f-secure.com/en/welcome
(日本語) https://www.f-secure.com/ja_JP/

また、Twitter @FSECUREBLOG でも情報の配信をおこなっています。