5F-01

REST API を用いたリアルタイム検知手法の提案

松田 亘† 藤本 万里子† 満永 拓邦†
東京大学 †

1. はじめに

サイバー攻撃の手法は巧妙化しており，国内外で被害が増加傾向にある．
攻撃の痕跡は各機器のログに残ることがあるため，ログをリアルタイムに近いタイミングで収集・分析し，セキュリティ担当者に通知を行うことで，被害を最小化することができる．
ただし，商用環境では大量のログデータをリアルタイムに処理するにあたって，性能劣化を抑止する必要がある．
本研究では，Web ベースの API である REST API を用いて，性能劣化を抑えつつ，リアルタイムに近いタイミングで攻撃検知や防御を行う仕組みを提案する．

2. 国内で観測された攻撃例

2.1 Struts 2 の脆弱性を狙った攻撃

Struts 2[1]はオープンソースの Java の Web アプリケーションフレームワークで，日本でも広く利用されている．2016 年以降から 2017 年 11 月 27 日時点までで，Struts 2 には 20 件以上の脆弱性が見つかっており，国内でも被害が報告されている[2]．これらは脆弱性が公開されると数時間で攻撃コードが公開され，攻撃が開始されることがある．

2.2 Active Directory の脆弱性を狙った攻撃

Active Directory(以下，AD)は Microsoft 社が提供するディレクトリサービスで，ドメインと呼ばれる管理単位を持ち，ドメインに属するアカウントやコンピュータ，ファイルなどのリソースを集中的に管理することができる．組織に侵入した攻撃者は効率的に組織を侵害するために， AD を狙う傾向にある[3]．

3. 関連研究

Zirije Hasani らはログ分析ツール群である Elastic Stack[4]を用いて，ログファイルに記録された SQL クエリの統計情報を 30 秒ごとに可視化している[5]．しかし，異常を検知した際に自動的に通知するような機能を実装しておらず，運用者が可視化された画面を目視で確認する必要がある．今後，Elastic Stack のツールの一つである Logstash などの機能を用いて，リアルタイムに通知する機能などを拡充させることを課題としている．

4. 提案手法

本研究では， REST API を用いて，リアルタイムに近いタイミングで攻撃検知およびアラート通知を行う仕組みを提案する．

4.1 REST API

REST API は HTTP/HTTPS を使用して利用できる API で，REST1と呼ばれる設計原則に従って設計される．各 API は一意な URI によって利用可能であり，統一されたインタフェースを持つ．攻撃検知のプログラムを REST API で実装することのメリットは以下である．
・Web ベースの統一されたインタフェースを持つため，既存のシステムへの組み込みが比較的容易である
・常駐プログラムとして動作するため，共通的に必要な前処理などを予めメモリにロードしておくことが可能で，性能劣化を抑えることができる．

4.2 Struts2 サーバに対する攻撃防御

2.1 で紹介した Struts2 の脆弱性を狙う攻撃について，サーブレットフィルタ2と REST API を組み合わせて防御を行う手法を提案する．Struts2 に対する攻撃は，攻撃コードを含むリクエストを Struts2 で実装された Web サーバ(以下，Struts2 サーバ)に送信する方法が典型的であるが，サーブレットフィルタを使用することで，リクエストに含まれる文字列を検知プログラムで分析し，攻撃と判断した場合は通信を遮断することが可能になる(図 1)．そのためには，リクエストを受信する度に，検知プログラムを実行する必要がある．本研究では検知プログラムの実装方法として，REST API を用いる手法と，スタンドアロンのプログラムとして実装し，プロセス呼び出しによって実行する手法を検討する．検知プログラムを使用しない時の Struts2 サーバのレスポンス時間をベースとして，各手法の性能を比較する．検証にあたって，POST リクエストで約 200 文字のデータを 1000 回送信し，レスポンス時間の平均を算出する．

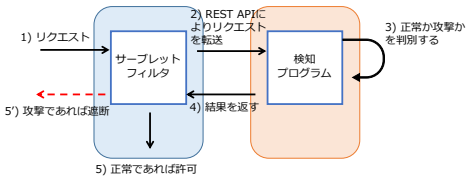

図 1. サーブレットフィルタと REST API を用いた Struts 2 の攻撃防御の概念図

†1WATARU MATSUDA, The University of Tokyo
†2 MARIKO FUJIMOTO, The University of Tokyo
†3TAKUHO MITSUNAGA, The University of Tokyo

1 https://searchmicroservices.techtarget.com/definition/REST-representational-state-transfer
2 Java サーブレットの実行前にフィルタリング処理などを行うための Java EE の標準機能

4.3 Active Directory 管理サーバに対する攻撃検知

2.2 で紹介した AD に対する攻撃を，Elastic Stack を用いてリアルタイムに検知する手法を提案する．ドメインコントローラ1に Winlogbeat と呼ばれるログ転送エージェントをインストールし，ドメインコントローラのイベントログを Logstash にリアルタイムに転送する(図 2)．Logstash は，ログの整形などを行い，Elasticsearch と呼ばれる検索エンジンにログを転送する機能を持つ．Logstash は任意のプログラムを実行することも可能なため，提案手法では Logstash から AD の攻撃を検知するプログラムを起動し，攻撃と判断した場合はログに攻撃を示すフラグを付与する．4.2 と同様に，検知プログラムを使用しない場合と REST API による検知プロセス呼び出す場合，検知プログラムをプロセスとして起動する場合の処理時間の差を比較する．ドメインコントローラ上で Windows コマンド(ipconfig) を 1000 回実行し，Logstash がログを受信してから，検知プログラムの判定結果を受け取るまでの時間の平均を算出する．

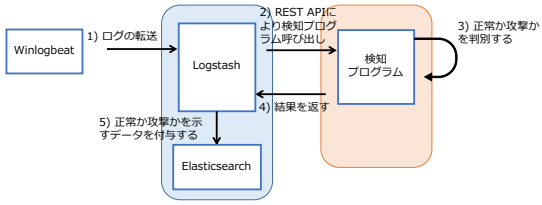

図 2. Elastic Stack と REST API を用いた AD の攻撃検知の概念図

5. 検証結果

5.1 Struts2 に対する攻撃防御の応答時間

4.2 で述べた手法の性能測定結果を表 1 に示す．

表 1　サーブレットフィルタと連携した時の検証結果
Table 1　Test results with servlet filter

手法	応答時間(秒)
検知プログラムなし	0.07
REST API による検知プログラム呼び出し	0.26
プロセス起動による検知プログラム呼び出し	32.44

検知プログラムを REST API によって実装した場合，多少の応答時間の劣化が発生しているものの，利用者にとって負担にならないレベルの劣化であると考えられる．

5.2 Active Directory に対する攻撃検知の応答時間

4.3 で述べた手法の性能測定結果を表 2 に示す．

検証のための Windows コマンドは，バッチプログラムによって 1000 回連続で実行し，全コマンド実行が完了するまでに約 40 秒要した．検知プログラムをプロセス起動によって実行した場合，検知プログラムの実行完了

1 AD 環境を集中管理するサーバ

表 2　Elastic Stack と連携した時の検証結果
Table 2　Test results with Elastic Stack

手法	応答時間(秒)
検知プログラムなし	0.004
REST API による検知プログラム呼び出し	0.900
プロセス起動による検知プログラム呼び出し	45.749

に時間を要し，結果として，Elasticsearch へ全ログが転送されるまでに 200 秒以上のタイムラグが発生した(図 3)．これは，ログが転送される度に検知プログラムを外部プロセスとして起動し，検知に必要なデータなどをロードするのに時間を要したためであると考えられる．一方，検知プログラムを REST API で実装した場合は，検知プログラムを使用しない場合とほぼ同等の応答時間を担保できることがわかった．これは，API 呼び出しに要する時間を短縮できること，検知に必要なデータを予めメモリにロードできることが起因していると考える．

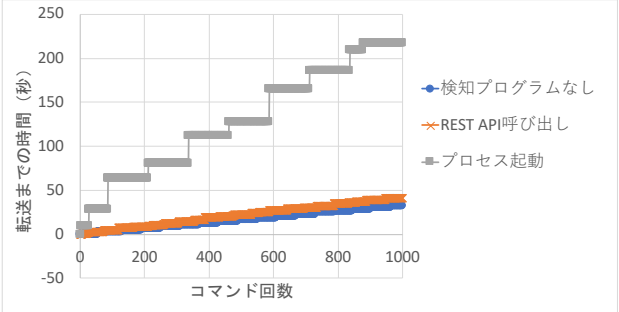

図 3. 各手法において Elasticsearch へログを転送するまでに要する時間

6. まとめ

攻撃検知において，検知プログラムを REST API として実装することで，既存システムに容易に組み込むことが可能で，かつ性能劣化を抑止できることが分かった．結果として，攻撃の被害抑止に活用できると考える．

参考文献

[1] The Apache Software Foundation. "Apache Struts". https://struts.apache.org/
[2] The Apache Software Foundation. "Apache Struts Security Bulletins". https://cwiki.apache.org/confluence/display/WW/Security+Bulletins
[3] Shingo Abe. “Detecting Lateral Movement in APTs -Analysis Approach on Windows Event Logs “. https://www.first.org/resources/papers/conf2016/FIRST-2016-105.pdf
[4] Elasticsearch. “Elastic Stack Features”. https://www.elastic.co/products/stack
[5] Zirije Hasani. “Real Time Analytic of SQL Queries Based on Log Analytic” http://proceedings.ictinnovations.org/attachment/paper/377/real-time-analytic-of-sql-queries-based-on-log-analytic.pdf