

"Detect and Prevent WebShell Malware"の概要

2020/04/26 脆弱性対応研究会

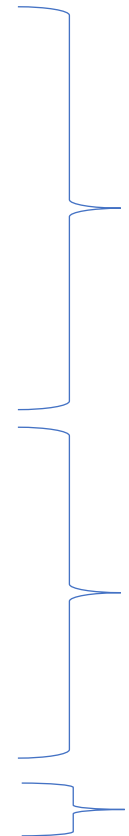
hogebuga

本資料について

- NSA(National Security Agency)及びAustralian Signals Directorateが公開した「Detect and Prevent Web Shell Malware」の概略理解のための資料です。
 - リンク
 - <https://www.nsa.gov/News-Features/News-Stories/Article-View/Article/2159419/detect-prevent-cyber-attackers-from-exploiting-web-servers-via-web-shell-malware/>
 - <https://media.defense.gov/2020/Apr/22/2002285959/-1/-1/0/DETECT%20AND%20PREVENT%20WEB%20SHELL%20MALWARE.PDF>
 - 一部翻訳版を、脆弱性対応研究会のgithubに配置した
 - https://github.com/hogehuga/vulnRespStudyGroup/blob/master/report/detect_and_prevent_WebShell_malware/translate-detect_and_prevent_WebShell_malware.md
 - この資料を基に記載します。

全体の構成

"Detect and Prevent WebShell malware"は以下の構成になっている

- Summary
 - Mitigating Actions(DETECTION)
 - "Known-Good" Comparison
 - Web Traffic Anomaly Detection
 - Signature-Based Detection
 - Unexpected Network Flows
 - Endpoint Detection and Response(EDR)Capabilities
 - Other Anomalous network Traffic Indicators
 - Mitigating Actions(PREVENTION)
 - Web Application Permissions
 - File Integrity Monitoring
 - Intrusion Prevention
 - Network Segregation
 - Harden web Service
 - Mitigating Action(RESPONSE and RECOVERY)
- 
- 緩和策（検知）
✓ 検出方法についてのガイド
- 緩和策（予防策）
✓ WebShellを設置されない
ようにするためのガイド
- 緩和策（対応と回復）
✓ 詳細な記載なし？

Summary

WebShellマルウェアを利用した攻撃が増える

- WebShellマルウェアとは
 - 被害者のWebサーバ上に配置されるソフトウェア
 - これにHTTPやHTTPSでアクセスすることで、任意のコマンドを実行できる
 - また、攻撃者のコマンドをほかのシステムに転送する中継ノードとして機能する
 - 精機のアクセスに偽装して、永続的にアクセスを提供する
- インターネットに接続したシステムだけが被害を受けるわけではない
 - 攻撃者は、内部のシステムやネットワークデバイス管理インタフェースなどの「直接インターネットに公開していない」Webサーバに展開されることがよくある
 - なぜなら、内部のWebアプリケーションは、パッチ管理が遅れていたり、セキュリティ要件が許容範囲が広い（厳密ではない）から

緩和策(検知)

WebShellは、変更が容易、暗号化/エンコード/難読化が採用されているために検出が困難。

WebShellを検知するには、以下のようなものがある。

- 「既知の正常な状態」との比較
- Webトラフィックの異常検出
- シグネチャベースの検出
- 予期せぬネットワークフロー
- EDR機能
- その他の、異常なネットワークトラフィック指標

検知(1)

- 「既知の正常な状態」との比較
 - Webアプリケーションが変更されたことを検知することで、発見が可能
 - 但し、タイムスタンプでの検出は偽装されて検知ができない場合があるが、最初のトリアージとしては選択可能な手段
- Webトラフィックの異常検出
 - WebアクセスのIPアドレス、UserAgent、URI、リファラなどが手掛かりとなる
 - ログをスクリプトに食わせて解析する、SIEMを用いる、などの方法が選択可能
- シグネチャベースの検出
 - 変更や難読化のため、あまり有用ではない可能性がある

検知(2)

- 予期せぬネットワークフロー
 - 未使用のポートの利用などのネットワークアクティビティを検査することで、検出が可能
- EDR機能
 - Webサーバ上でのipconfig実行など、異常な動作をログとして検出することで検出が可能
- その他の異常なネットワークトラフィック指標
 - 以上に大きなデータ通信、通常とは異なる時間でのアクセス、地理的にバラバラなリクエスト、などを検出する

緩和策(予防)

インターネットに接続したサーバと内部サーバについて、多層防御により予防することができる

- Webアプリケーションの更新
- Webアプリケーションのアクセス許可
- ファイルの完全性監視
- 侵入防止
- ネットワークの分離
- Webサーバの強化(hardening)

予防：

- Webアプリケーションの更新
 - パッチリリースから24時間以内に攻撃が始まることもあるため、自動更新や頻繁な手動対応が必要
- Webアプリケーションのアクセス許可
 - Webサービスは、最小減のアクセス権とすべき
 - WEbアクセス可能なディレクトリに直接書き込んだり、コード変更が可能な権限を持たないように、設定をするべき
- ファイルの完全性監視
 - 上記のようなアクセス権監視ができない場合は、特定のファイルのみ許可する、などを行うソフトウェアを利用する

予防：

- 侵入防止(IPS)
 - IPSの防御も有効だが、シグネチャベースの検出であるため完全とは言えない
 - 多層防御の一環として使用し、単一のポリシーですべてのサーバに適用するのではなく、ターゲットにあった固有のルールを適用する必要がある
- ネットワーク分離
 - 攻撃が内部へ伝搬するのを阻害することが可能
 - Zero Trustアーキテクチャとするのが良い
- Webサーバの強化
 - 未使用ポートやサービスへのアクセスブロックや、定期的な脆弱性スキャン、ホストベースのセキュリティシステムの高度な機能を有効にするなどが有効

緩和策(RESPONSE/RECOVERY)

WebShellが発見された場合、どこまで侵入されたか、の情報が重要。

影響を受けたものを全て確認し、攻撃者を配乗しないまま対応をすれば、再度別のチャンネルを介してアクセスされてしまう可能性が高い。