攻撃コード解析による不正アクセス検出手法の検討

北條　孝佳†, 佐久間　英夫, 種茂　文之‡
警察庁†　日本電信電話株式会社 NTT 情報流通プラットフォーム研究所‡

１．はじめに

近年、OS やアプリケーションに対するセキュリティホールが多数発見されており、これらを悪用してネットワークから不正アクセスを行う攻撃ツールやワームが作成され使用されることにより、多大な被害が発生している。このような不正アクセスのうちシステムを乗っ取るなど悪質なものは、バッファオーバフロー(以下「BOF」と称する)[1]と呼ばれるセキュリティホールを利用した攻撃が大半を占めているため、BOF を利用した攻撃を的確に防御できる技術が求められている。

この BOF を利用した攻撃(以下「BOF 攻撃」と称する)の多くは、脆弱なサーバ上で任意の命令実行を行うことが可能なマシン語で構成された命令列(以下「攻撃コード」と称する)がパケットに含まれている。このような不正アクセスを検出するために、IDS(Intrusion Detection System)の利用が考えられるが、既存の IDS では未知の攻撃を確実に検出できないことや誤検出が多いことなどの問題点が存在する。

そこで、我々は BOF 攻撃に対して、パケットのデータ部をマシン語の命令列として解析を行うことで、パケットが不正アクセスに関わるか否かを正確に判定する不正アクセス検出手法を提案している。本手法により、実際に既存のシグネチャマッチング型 IDS では検出できない未知の不正アクセスが検出可能である[2]。本手法では、変形された攻撃コードや未知の不正アクセスであっても攻撃コードが含まれていれば、検出が可能となっている。

本稿では、提案した不正アクセス検出手法を実装したプロトタイプの検出プログラムを用いて、公開されている攻撃ツールを利用した不正アクセスが確実に検出できることを確認し、また実トラフィックデータにおける誤検出状況について検証したので報告する。

２．BOF(バッファオーバフロー)攻撃の概要

攻撃ツールやワームの多くは、BOF を利用して任意の動作を行わせることが可能になる攻撃コードを攻撃対象となるサーバに送り込む。攻撃対象サーバ上で BOF が発生すると攻撃コードに制御が移動する。そして、攻撃コードを構成しているマシン語命令が逐次実行され、その結果、シェルの起動やファイルの改竄・削除等攻撃者が意図する行為が行われるように CPU のレジスタやメモリの値が書き換えられる。そして実行過程の最後に処理される命令(以下「攻撃決定命令」と称する)を実行することにより攻撃が完了する。

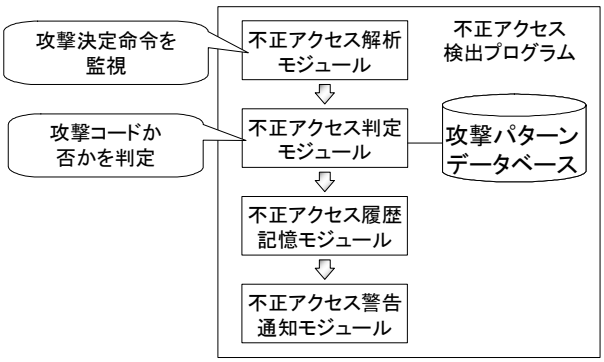

図１．攻撃コード解析検出手法の概要図

３．不正アクセス検出手法

我々が提案している不正アクセス検出手法の概要を説明する。

図１は、本手法の概要図である。不正アクセス解析モジュールは、ネットワーク上のパケットをキャプチャし、ヘッダ部を取り除いたデータ部をマシン語の命令列として擬似的に実行して解析を行う。実行される命令を常に監視し、その命令が攻撃決定命令の場合には、攻撃決定命令と CPU のレジスタの状態を不正アクセス判定モジュールに解析結果として出力する。不正アクセス判定モジュールは出力された攻撃決定命令をキーとして攻撃パターンデータベースから攻撃情報を検索し、解析結果である CPU のレジスタの状態と一致するか比較する。解析結果が攻撃情報と一致した場合には、攻撃コードが含まれる不正アクセスであると判定して、不正アクセス履歴記憶モジュールや警告通知モジュールによりパケット情報と攻撃情報をファイルに保存したり、管理者に通知したりする。

本手法は BOF 攻撃特有の攻撃コードに着目した検出手法であり、BOF 攻撃以外の不正アクセスや攻撃コードを含まない BOF 攻撃は検出することができない。また、サーバの OS やアプリケーションのバージョンに対して一意的に決まっているメモリやレジスタの状態に依存する特殊な BOF 攻撃については検出できないため、既存の IDS 等と併用・連携をする必要がある。

Discussion of Shellcode Analysis Intrusion Detection Method.
†Takayoshi HOJO, †National Police Agency
‡Hideo SAKUMA, Fumiyuki TANEMO
‡NTT Information Sharing Platform Laboratories, NTT Corporation

４．提案手法の実装プログラム

我々は、本検出手法を実現するために CPU シミュレータを作成して擬似的に解析を行う方式と実 CPU で処理を行う方式を提案している[2]。不正アクセスの検出を対象としているサーバの CPU は Intel 製 32 ビットであり、OS は Linux である。

これに加えて、パケットを命令列として実行する際に無限ループの発生を監視する無限ループ監視プログラムの実装を行った。この監視プログラムは命令実行中に発生するループをチェックし、それが無限ループと判断できる場合、実行を停止し次の処理に進む。また、無限ループと判断されなかった場合も、予め設定した上限回数を超えたループの場合には、対象のパケットの情報を警告ファイルに出力する。無限ループ監視プログラムを実装することにより、警告ファイルに出力されるパケット数は実装しない場合と比較して 50%減少した。

５．攻撃ツールによる検出結果

インターネット上に公開されている攻撃ツール[3]を 10 種類取得し、本検出プログラムが検出可能か検証を行った。

攻撃対象サーバの脆弱性に依存しないよう、攻撃ツールに含まれる対象サーバの情報取得機能は取り除いた。さらに、本検出プログラムはフラグメント化されたパケットに攻撃コードが分割された場合には対応していないため、フラグメントが発生する場合にはいずれかのパケット内に攻撃コードが配置されるように改良した。

本手法と比較するため、NGSec 社の攻撃コード検出ツール(NIDSFS)[4]と Snort[5]でも同様の検証を行った。Snort のバージョンは 2.1.0、シグネチャは 2003 年 12 月 17 日のものを使用した。

検出結果を表１に示す。本手法による検出率は 100%であった。これより、公開されている攻撃ツールを使用した不正アクセスは他の IDS よりも非常に高い精度で検出できることが分かる。

NIDSFS は NOP(何も行わず次の命令を行うマシン語命令でバッファを埋める目的のために利用)や擬似 NOP(命令を実行するが、エラー発生せず NOP と同様の目的で利用)が連続で閾値以上にならないと検出しないため、50%の検出率であった。一方、Snort は攻撃ツールに対するシグネチャが存在して検出する場合と攻撃パケットの中に含まれる攻撃コードの一部を発見するシグネチャにより検出する場合と 2 種類存在した。攻撃ツールの大半はこの攻撃コードを発見するシグネチャにも当てはまらなかったため、検出ができなかった。

６．実トラフィックデータを用いた評価

実際に流れている通常のトラフィックデータ((1)FTP 通信 6,807 パケット、(2)TCP の 25、80、443 番ポートの通信 139,581 パケット)を用いて、

表１．攻撃検出結果

攻撃名	本手法	NIDSFS	Snort
rsync-2.5.1	○	○	○
proftp-1.2.7	○	○	○
wsmp-3.0.0	○	×	×
SDI-halflife-3.1	○	○	○
Light Httpd-0.1	○	×	○
Mdbms-0.99b	○	○	×
WSMP-0.0.6	○	×	×
zkfingerd-r3.0.9	○	×	×
nfs-xlog-2.95.4	○	○	×
sendmail-8.11.6	○	×	×

本手法を実装したプログラムの誤検出(False Positive)を調査した。NIDSFS と Snort でも同様の調査を行った。

(1)のデータはどの IDS でも誤検出がなく、(2)のデータは NIDSFS だけが誤検出をし、Snort は BOF 攻撃に関するシグネチャに関しては誤検出がなかった。NIDSFS で誤検出されたパケットは NOP 及び擬似 NOP が連続で閾値以上になっていた画像データ等のバイナリデータで、同様のデータが頻繁にやり取りされていれば全て誤検出になってしまう。また、本手法において 4 章で説明した無限ループ発生の警告ファイルには(1)の場合で約 13%、(2)の場合では約 2%のパケットが出力されているが、今回はこの警告ファイルの詳細は未解析である。

７．まとめと課題

今回、実トラフィックデータと攻撃検証ツールを用いて誤検出及び検出率の検証を行った。本手法を用いた検出プログラムは、公開されている攻撃ツールによる不正アクセスのうち、本手法が対象にしている BOF 攻撃に関して非常に高い精度で検出が可能であるということ、また実トラフィックデータによる誤検出が起こらないことが分かった。

今後は、警告ファイルに出力されるパケット数の減少方法や CPU シミュレータ方式の対応命令拡張、検出処理を行った場合の負荷等の検討、攻撃パターンデータベースの追加などに取り組んでいく予定である。

参考文献

[1] 三輪信雄，新井悠：ネットワーク攻撃詳解，ソフト リサーチ センター(株)

[2] 北條孝佳，佐久間英夫，種茂文之：“シェルコード解析による不正アクセス検出手法”，情処 CSEC，pp1-6，2003 年 12 月

[3] SecuriTeam，http://www.securiteam.com/

[4] NGSec，NIDSFS(NIDSfindshellcode)，http://www.ngsec.com/ngresearch/ngtools/

[5] Snort，http://www.snort.org/