M-085

ユーザのネットワーク環境に対応した ホスト・ネットワーク連携型シグネチャの作成手法の提案

A Proposal of a Creation Method for the Host and Network Cooperation Type Signatures Corresponding to a User's Network Environment

田辺 光昭　　勅使河原 可海
Mitsuaki Tanabe　　Yoshimi Teshigawara

1．はじめに

近年，不正アクセスによる被害が増加傾向になりつつあり，それによりセキュリティ意識が向上しており，侵入検知システム（IDS），ファイアウォールなどのセキュリティ製品を導入するところが増えている．しかしながら，特に IDS 製品の導入により問題になっているのが，誤検知という問題である．この問題点の原因の一つとして，管理者の設定不備によるものが挙げられる．

本研究では，管理者の設定不備をなくすために，管理するネットワーク環境を仮想環境に構築し，ホストとネットワークを連携させたシグネチャを作成する手法を提案する．

2．セキュリティ技術の現状と課題

2.1　侵入検知システム（IDS）

(1)ネットワーク型侵入検知システム

一般的にネットワークに流れるデータを監視して，シグネチャと呼ばれる攻撃パターンを格納したデータベースとのマッチングを行うことで，侵入を検知する IDS である．

ネットワーク型 IDS の問題点として，シグネチャをもとに検出しているため，未知の不正通信に対して検出できないことや，攻撃されていることは検出できるが，侵入されたかどうかは管理者がログを分析しないと検出できない．また，設定によって誤検知が多くなるということなどが挙げられる．

(2)ホスト型侵入検知システム

一般的にコンピュータのログや，システム情報などを監視して，正常な行為から外れたものを異常とすることで，侵入を検知する IDS である．

ホスト型 IDS の問題点として，異常なイベントの定義が難しく，設定内容の不備によってはネットワーク型 IDS よりも誤検知率が高くなる場合があるなどが挙げられる．

2.2　ハニーポット

ハニーポットとは，プローブされ、攻撃され、侵害されることに価値を持つセキュリティリソースである [1]．ハニーポットの問題点として，複雑な構成，データ分析の難しさ，運用の難しさなどが挙げられる．

3．提案手法の目的

前章で述べたように，ネットワーク型 IDS の問題点として誤検知が多いことと，ホスト型 IDS の問題点として設定の不備によってはネットワーク型 IDS よりも誤検知が多くなることがあることを挙げた．この問題点の原因の一つとして，管理者がネットワーク環境に合わせて IDS をカスタマイズしていないことが挙げられる．

本研究では，シグネチャベースの IDS の誤検知に着目した．本稿では，仮想環境上で実環境と同等のネットワーク環境を構築し，仮想環境のホスト上で起きた事象とその時に仮想環境上のネットワークに流れていたデータを基に，攻撃手順を一連の流れとして定義したシグネチャを作成する手法を提案する．

4．提案手法

4.1　シグネチャ作成手順の概要

ユーザは，仮想環境上にユーザのネットワーク環境と同等のネットワーク環境の構築を行う．攻撃者が仮想環境上のホストに対して行った攻撃をログ情報やシステム情報，そしてネットワークに流れているデータから確認する．その後，時間を基に攻撃が行われたときに仮想環境上のネットワークに流れていたデータを取り出し，攻撃されたホスト上での事象とネットワーク上のデータとを関連付けたシグネチャを作成する．

例えば，仮想環境のホスト上で結果として www の改竄が行われたら，その時に仮想環境上のネットワークに portscan，telnet によるバージョン情報の取得，ftp にデータを転送したなどの行為と www 改竄がホスト上で起きたという事象を一連の流れとして定義してあるシグネチャを作成する．

4.2　実験環境

本研究での実験ネットワークの構成を図 1 に示す．

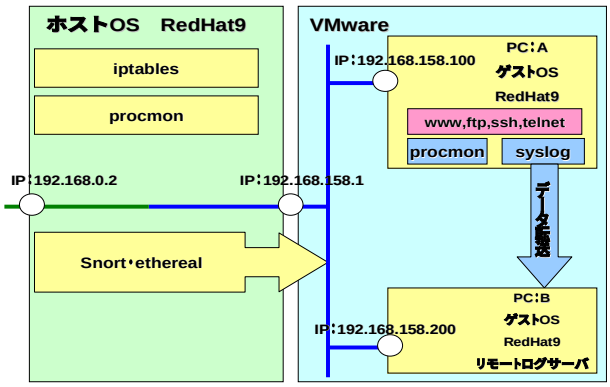

図 1　実験ネットワークの構成

本研究の実験環境では，VMware という 1 つのホスト上で，複数の仮想 OS を動かすことができるソフトウェアを用いて仮想環境を実現している．VMware をインストールする OS をホスト OS とし，VMware 上にインストー

創価大学大学院工学研究科
Graduate School of Engineering , Soka University
fmsnabe@rio.odn.ne.jp

ルする OS をゲスト OS とする．本研究では，ホスト OS，ゲスト OS ともに RedHat9 を使用している．

ホスト OS 上では，iptables，snort，ethereal，procmon がインストールされている．また，snort を管理しやすくするために，ACID 等もインストールされている．VMware 上には 2 つのゲスト OS，PC:A と PC:B が存在し，PC:A は実際にサービスを提供するサーバとして稼動している．提供しているサービスは，www，ftp，ssh，telnet の 4 つである．PC:B は，リモートログサーバとして稼動している．これは，PC:A に侵入された時，ログの改竄等を行われても PC:A のログの正当性を保証するために一定時間毎に PC:B のリモートログサーバに PC:A のログを転送するようになっている．

iptables では，NAT の設定を行い，PC:A と外部の通信を可能にしている．また，ホスト OS，PC:B のリモートログサーバに侵入されないように，外部から通信が出来ないようにしている．snort は，仮想環境上のネットワークを監視するために導入している．ethereal は，仮想環境上のネットワークに流れているデータを取得するために導入している．procmon は，ネットワークを監視している場合，ssh などを使ったバックドアを仕掛けられた際に，暗号化された通信により攻撃者の行動が不明になってしまうので，システム側でコマンド履歴を自動保存してくれるため導入している．

4.3 IDMEF

本研究では，独自のシグネチャを作成するため，アラートで報告されるものは共通のフォーマットを使用する必要があると考えられるため，IDMEF(Intrusion Detection Message Exchange Format)をベースにして検討する[2].

IDMEF とは，IDS が疑わしいと考えられる事象に対して自動的にアラートを報告するために使用することができる標準のデータ・フォーマットである．この標準のフォーマットの開発は，商用製品，オープンソースおよび研究システム間の相互運用を可能にし，ユーザはそれらの長所と短所に応じた最適なシステムを実装することができるようになる．portscan を IDMEF で表したものを図 2 に示す．

図 2　Simple Port Scanning

5．提案手法の利点

5.1 システムの利点

セキュリティ技術の問題点で挙げたように IDS では誤検知があるが，提案手法では 4.1 の作成手順の概要で述べたように，ホスト上で起きた事象とその時ネットワークに流れていたデータを関連付けた一連の流れでシグネチャを作成している．そのため，ネットワーク型 IDS やホスト型 IDS のような誤検知を減らすことができる．

また，ユーザのネットワーク環境に対応させたものを仮想環境上で構築することにより，そのネットワーク環境に対応したシグネチャを作成することができる．今回の実験では 4.2 に示したようなネットワーク環境を構築したが，mail サーバを追加するなどユーザのネットワーク環境に対応させることができる．

実稼動しているサーバでは，攻撃を受けたり侵入されるようなことが起きるとシステムを停止することになる．本手法では，仮想環境を用いることで実際に攻撃を受けたとしても，復旧作業が容易に行える．仮想環境はハニーポットとしての役割も果たしていると言える．

5.2 提案シグネチャの利点

ネットワークにプローブデータや攻撃データが流れた時に従来の IDS では多くのアラートを出力していたが，提案するシグネチャでは，ホスト上での事象とその時ネットワークに流れていたデータを関係付けて一連の流れとしてシグネチャを作成しているので，多くのアラートを出力することがなくなる．portscan を例に取ると，従来の IDS であれば portscan を定義したシグネチャのアラートが多く出力されてしまうが，提案したシグネチャでは portscan だけを検出してアラートを挙げるのではなく，portscan からホスト上で起こった事象までの一連の流れを検出してアラートを挙げるようになる．また，攻撃手法やワーム等の亜種に対して検知できる効果が期待できると考えられる．

6．まとめと今後の課題

本研究では，IDS の問題点である誤検知を解決するために，仮想環境上に実環境と同等のネットワーク環境を構築し，仮想環境のホスト上で起きた事象とその時に仮想環境上のネットワークに流れていたデータの情報を基に，攻撃手順を一連の流れとして定義したシグネチャを作成する手法を提案した．

今後の課題としては，仮想環境上で実環境と同等のサービスを提供しているため，侵入されたときに踏み台にされないように気を付けなければならない．また，データ分析からシグネチャ作成まで全てが手動であるため，自動化できるところを検討する必要がある．また，攻撃者が時間的に間隔をあけた攻撃やプローブを行った場合，攻撃元が単体ではなく複数にわたる場合，そしてアプリケーションの脆弱性を攻撃された場合にそれらを検知できるような手法を検討する必要がある．

参考文献

[1]Honeypots&Honeynets
http://www.securitymap.jp/sp/docs/honeypot-honeynet.pdf
[2]IDMEF
http://www.ietf.org/internet-drafts/draft-ietf-idwg-idmef- xml-11.txt