6Q－03

定義ファイルを用いたセキュリティ検査システムの実装

甲斐 賢† 寺田 真敏† 岡林 稔仁‡
(株) 日立製作所 システム開発研究所†
(株) 日立電子サービス‡

1. はじめに

インターネット接続サイトのセキュリティ上の脅威が増大しつつあり、不正アクセス対策が重要な課題となっている。しかし、不正アクセスの実態は目に見えにくく、手法は多様化･高度化しているため、万全な対策を困難なものとしている。また、不正アクセス対策は危機管理モデルになぞらえ、４つのフェーズ「対策」「保証」「検知」「調査」からなるセキュリティサイクルの実現が重要であると言われている[1]。

本稿で述べるセキュリティ検査システムは「保証」フェーズに位置し、構築した情報システムの安全性確認を主目的とする。通常、「保証」フェーズを満たすために、商用ツールやフリーソフトウェア等を用いた検査が行われているが、これらは既知セキュリティホールの存在等の検査を中心に構成されており、脆弱性検査に特化している傾向がある。そのため、新規セキュリティホールの検査や脆弱性検査を含め幅広くセキュリティ検査を行うことを考えた場合、以下の課題が挙げられる。

- 検査対象システムに応じた、検査項目や設定パラメータ等のカスタマイズ性(新たな検査項目の開発、既存の検査項目のエンハンス)
- 実態が分かりにくい不正アクセス技術の本質や特徴を明らかにするノウハウ蓄積

報告者らは上記の課題に対し、テキスト型「定義ファイル」を用いたセキュリティ検査システムを開発した[2]。本稿では「定義ファイル」を用いた検査システムに関し、実装方式と検査システムの評価結果を報告する。

Implementation of Configuration File based Security Scanner
Satoshi KAI
Masato TERADA
Toshihito OKABAYASHI
Hitachi, Ltd. †
Hitachi Electronic Service, Ltd. ‡

2. 実装方式

本システムは、不正アクセスの手法の特徴部分を抽出したテキスト型「定義ファイル」を用いてセキュリティ検査を行う(図 1)。定義ファイルはスケジュール部とパラメータ部から構成される(表 1)。検査エンジンはアプリケーション層(OSI 5-7 層)とネットワーク層(OSI 3-4 層)に機能分担し開発を進めており、本稿ではネットワーク層検査システムについて報告する。ネットワーク層の定義ファイルでは、スケジュール部にパケット制御コマンド(送信/受信等)を、パラメータ部に送信データ(TCP ヘッダ/IP ヘッダ等)およびフィルタを記述する方式である。これにより、最小限の記述で検査項目の記述が可能であり、定義ファイルの記述そのものが検査内容を示すという、理解容易な記述形式となっている。

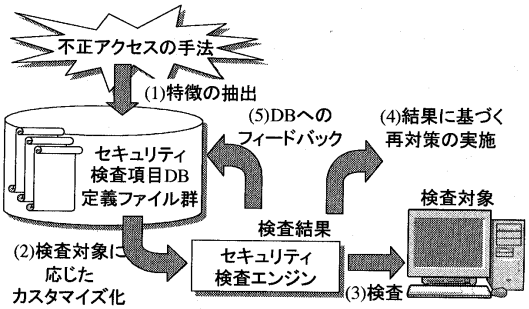

図１ 定義ファイルを用いた検査プロセス

本システムを用いた検査プロセスは、
(1)新たな検査項目の新規「定義ファイル」化
(2)登録済み「定義ファイル」に基づくカスタマイズ化
(3)「定義ファイル」を用いた検査
(4)検査結果に基づく再対策の実施
(5)検査結果および再対策のデータベース(DB)へのフィードバック
というステップ(図１)となる。

3. 「定義ファイル」型検査システムの評価

不正なパケットに関するセキュリティ検査を対象に、開発した「定義ファイル」型検査システムと、検査項目をソースプログラムで開発するシステムとを比較・評価した。その結果、「定義ファイル」ベースでは、次のような利点がある(表1)。

- 検査項目の開発ステップ数の削減

最初に不正アクセスの特徴を抽出しさえすれば、「定義ファイル」への移植は必須値のみの記述で済むため、開発ステップ数が削減できる。表1の例では約 10 分の1で済む。

- 検査内容のエンハンスの容易性

「定義ファイル」は特徴を抽出したフォーマットであるため、セキュリティ検査の作業者は検査内容を容易に理解することができ、検査のポイントとなる部分の補充・補完を定義ファイルに反映させ、再検査することが可能である。

- 不正アクセス技術のノウハウ蓄積

実態が分かりにくい不正アクセスの手法について、「定義ファイル」という(a)特徴点の抽出により容易に理解が可能な、かつ(b)実際にセキュリティ検査が実行可能なフォーマットにより、ノウハウ蓄積することが可能である。

以上のことから、本システムを利用することにより、セキュリティ検査の作業者が不正アクセスの手法を分類・類推し検査項目にフィードバックしていくことを通して、本質的な問題点を押さえた検査項目の網羅性を実現することができる。

4. おわりに

「定義ファイル」型セキュリティ検査システムの実装方式と開発システムの評価結果を報告した。開発システムによるカスタマイズ性とノウハウ蓄積の容易性は、セキュリティサイクルの「保証」フェーズにてセキュリティ検査プロセスを円滑に行う効果をもつ。さらに、不正アクセス技術のノウハウ蓄積を行うことにより、攻撃手法に対する効果的な対策を検討することを可能にし、セキュリティサイクルへの支援につなげることができる。

今後の課題は、検査内容の表現フォーマットについて、テキスト型「定義ファイル」による記述の容易性を活かした「定義ファイル」の汎用化である。

参考文献

[1] 不正侵入はこう防げ, 日経コンピュータ, No448,pp185-195,July 1998

[2] 寺田,甲斐他: 定義ファイルを用いたセキュリティ検査システムの開発, 情報処理学会 CSS'99, pp141-146, Oct 1999

表1 セキュリティ検査に関わる作業内容の比較

	ソースプログラム ベース	定義ファイル ベース
新規登録	インターネット経由等でソース入手可	特徴を抽出し、移植作業
カスタマイズ	一般的にCソース	プレーンテキスト
検査の実行	動作プラットホームへの依存度高、コンパイラ型	マルチプラットホーム対応、インタプリタ型
ＤＢへのフィードバック	プログラム自身の変更が必要	特徴点の変更のみで良い
例 (Land 攻撃)	/* Land attack */ RAW ソケットの準備 (15 ステップ) IP_HDRINCL オプションのセット(5 ステップ) IP ヘッダの作成 (30 ステップ) TCP ヘッダの作成 (30 ステップ) パケットの送信 (5 ステップ) RAW ソケットの解放 (5 ステップ) (計 90 ステップ)	[schedule] # Land attack send packet1 ← スケジュール部 [packet1] ip_p = TCP ip_src = 192.168.1.1 ip_dst = 192.168.1.1 tcp_sport = 43 tcp_dport = 43 tcp_flags = TH_SYN ← パラメータ部 (計 9 ステップ)