

NIST Special Publication 800-42

ネットワークセキュリティテストにおけるガイドライン

米国立標準技術研究所による勧告

John Wack, Miles Tracy, Murugiah Souppaya

コンピュータセキュリティ

コンピュータセキュリティ部門
情報技術研究所
米国立標準技術研究所
Gaithersburg, MD 20899-8930

2003年12月



米国商務省 長官

Donald L. Evans

技術管理局 技術担当商務次官

Phillip J. Bond

米国立標準技術研究所 所長

Arden L. Bement, Jr.

この文書は下記団体によって翻訳監修されています

コンピュータシステム技術に関する報告書

米国立標準技術研究所 (NIST; National Institute of Standards and Technology) のITL (Information Technology Laboratory) は、国家の評価基準および標準化インフラストラクチャの技術的なリーダーシップを提供し、米国の経済および公共福祉に貢献している。ITLは、テスト、テスト技法、参照データの開発、概念実装、技術的分析の検証を行い、情報技術の開発と生産的な利用の発展に努めている。ITLの責務は、技術的、物理的、および管理上の標準とガイドラインを開発し、連邦政府のコンピュータシステム内の、取り扱いに注意を要する非機密扱い情報のセキュリティとプライバシーをコスト効率の高い方法で確保することである。NIST特別出版物800シリーズでは、コンピュータセキュリティにおけるITLの調査、ガイダンス、成果を報告し、産業界、政府機関および教育機関との共同活動についても報告する。

NIST特別出版物800-42

NIST特別出版物800-42、XXページ(2003年10月)

CODEN: XXXXX

このドキュメントでは、実験的な手順および概念を的確に記述するために、特定の商用事業体、機器、または資材に触れることがある。特定された商業事業体、装置および資料名は、米国立標準技術研究所による推奨または支持を意味するものではなく、またその事業体、資料、装置が目的に最適であることを示すものでもない。

米国政府印刷局

WASHINGTON: 2001

政府刊行物管理局、米国政府印刷局より販売

インターネット: bookstore.gpo.gov — 電話: (202) 512-1800 — Fax: (202) 512-2250

郵送: Stop SSOP, Washington, DC 20402-0001

本レポートは、原典に沿ってできるだけ忠実に翻訳するよう努めていますが、完全性、正確性を保証するものではありません。翻訳監修主体は本レポートに記載されている情報より生じる損失または損害に対して、いかなる人物あるいは団体にも責任を負うものではありません。

作成機関

米国立標準技術研究所 (NIST; National Institute of Standards and Technology) は、2002年の連邦情報セキュリティ管理法 (FISMA; Federal Information Security Management Act)、公法107-347に基づくその法的責任を推し進めるために、このドキュメントを作成した。

NISTは、すべての機関の業務および資産に適切な情報セキュリティをもたらすために、最低要件を含んだ標準およびガイドラインを作成する責任があるが、このような標準およびガイドラインは国家のセキュリティシステムには適用されない。このガイドラインは、行政管理予算局 (OMB; Office of Management and Budget) Circular A-130、第8b(3)項、『政府機関の情報システムの保護 (Securing Agency Information Systems)』の必要要件に一致しており、これはA-130の付録IV「重要部門の分析」で分析されているとおりである。補足情報は、A-130、付録IIIに記載されている。

このガイドラインは、連邦諸機関が使用する目的で用意されたものである。非政府組織が自己責任において使用することもでき、出自を明らかにすることが望ましいが、著作権の制約はない。

謝辞

本書の執筆にあたり、Timothy Grance氏、Wayne Jansen氏、Tom Karygiannis氏、Peter Mell氏、Robert Sorensen氏、Marianne Swanson氏をはじめとするNISTならびにBAHのスタッフの皆様には原稿のレビューに加え本書の質の向上に多大なるご協力をいただき、執筆者一同 (NISTのJohn Wack、Murugiah Souppaya、Booz Allen HamiltonのMiles Tracy) 心から感謝の意を表したい。

目次

1. はじめに.....	11
1.1 目的と範囲	12
1.2 定義	14
1.3 対象とする読者	15
1.4 本ドキュメントの構成.....	15
2. セキュリティテストとシステム開発のライフサイクル.....	16
2.1 システム開発のライフサイクル.....	17
2.1.1 インプリメンテーションステージ	18
2.1.2 運用ステージ.....	19
2.2 セキュリティテスト結果の記録.....	20
2.3 役割と責任	20
2.3.1 上級 IT 管理者/最高情報責任者 (CIO).....	21
2.3.2 情報システムセキュリティプログラムマネージャ (ISSM)	21
2.3.3 情報システムセキュリティ責任者 (ISSO)	21
2.3.4 システムネットワーク管理者.....	22
2.3.5 管理者と所有者	22
3. セキュリティテストの技法.....	23
3.1 テスト実行者の役割と責任.....	24
3.2 ネットワークスキャン	24
3.3 脆弱性スキャン	27
3.4 パスワードクラッキング.....	31
3.6 ファイル完全性チェッカー	35
3.7 ウイルス検知.....	36
3.8 ウォーダイアリング.....	38
3.9 無線 LAN テスト(ワードライブ).....	39
3.10 侵入テスト.....	41
3.11 テスト後の対応.....	47
3.12 情報セキュリティの一般原則.....	50
3.13 ネットワークテスト技法の比較.....	52
4. セキュリティテスト導入戦略.....	56
4.1 情報システムのセキュリティカテゴリの決定.....	56
4.2 システム別のテストコスト計算.....	58
4.3 システム別を実施する各テストの利点の明確化.....	59
4.3 システム別を実施する各テストの利点の明確化.....	59
4.4 テスト実施のためのシステムの優先順位付け.....	59

付録 A. 専門用語.....	60
付録 B. References	61
付録 C. 一般的なテストツール.....	62
C.1 ファイル完全性チェッカー	62
C.2 ネットワークスニッファ.....	63
C.3 パスワードクラッカー	64
C.4 スキャンおよび列挙ツール.....	66
C.5 脆弱性評価ツール.....	68
C.6 ウォーダイアリングツール.....	70
C.7 無線ネットワークツール.....	71
C.8 ホストベースのファイアウォール.....	72
D. 一般的なテストツールの使用例.....	73
D.1 Nmap.....	73
D.2 L0phtCrack.....	80
D.3 LANguard.....	81
D.4 Tripwire.....	83
D.5 Snort	90
D.6 Nessus	96

表

表 3.1: テスト手順の比較	54
表 3.2: 評価方法と実行頻度	56
表 C.1: ファイル完全性チェッカーツール.....	62
表 C.2: ネットワークスニッファツール	64
表 C.3: パスワードクラッキングツール	65
表 C.4: スキャンツールと列挙ツール	67
表 C.5: 脆弱性評価ツール.....	69
表 C.6: ウォーダイアリングツール	70
表 C.7: 無線ネットワークテストツール.....	71
表 C.8: ホストベースのファイアウォールツール.....	72

図

図 3.1: 4つのフェーズからなる侵入テスト	43
図 3.2: 発見フェーズにループバックされる攻撃フェーズのステップ	45

エグゼクティブサマリ

今日の複雑なコンピュータシステムの安全性を確保して運用することは、非常に困難な課題である。サービスやアプリケーションを迅速かつ安全に提供するための責務や運用上の要件は、かつてないほど重要になってきた。組織は貴重なリソースを投じて、リスク分析、証明、認定、セキュリティアーキテクチャ、ポリシー作成などセキュリティ確保に必要なさまざまな取り組みを行わなければならないが、時には統合的かつきめ細かなセキュリティテストプログラムを怠ったり、不十分なまま構築してしまいたいといった衝動に駆られる。

このガイドでは、連邦機関における効果的なセキュリティテストプログラムの必要性に焦点を当てて説明する。セキュリティテストには、いくつかの目的がある。まず第一に、いかに優れたシステムが開発されたとしても、膨大なコード、システム内部の複雑な相互作用、不特定の外部コンポーネントとの相互運用性、未知の依存関係、さらにはベンダーのコストや導入スケジュールによるプレッシャーと言った要素が絡みあい、今日のシステムは複雑化している。そのため、悪用される危険のある弱点が常に存在し、時間とともに表面に現れてくる。したがって、システム開発の最先端技術と実際のシステム運用とのギャップを埋めるには、セキュリティテストが必要になる。第二に、組織のセキュリティ運用のあり方を理解、調整、記録する上で、セキュリティテストは重要な役割を果たしている。脅威や脆弱性にさらされる変化著しい環境においては、システム開発とは別に、運用面やセキュリティ面での要件対応が求められる。大規模な攻撃に遭ってからセキュリティの状態を把握し、修復を図ろうとするのでは、コスト面でも組織としての信用面においても失うものは非常に大きく、ほとんど効果がない。第三に、セキュリティテストは、組織のセキュリティへの取り組みを改善するためにも不可欠である。計画的、系統的、総合的かつ継続的に、重要度に応じたセキュリティテストを導入することにより、システムのセキュリティ改善に向けて慎重な投資がよりしやすくなる。

セキュリティテストについての NIST の推奨事項は、以下のとおりである。

ネットワークのセキュリティテストは、システムおよびネットワークの運用管理の一環として日常的に実施する。

システムを日常的にテストして、適切なセキュリティメカニズムとポリシーに基づいてシステムが正しく設定されていることを確認する。このように定期的にテストを実施することにより、さまざまな問題を発生段階で防止する。インシデントレスポンスにかかるコストが低減されることによって、定期テストにかかるコストが相殺されると予想される。

最も重要なシステムを最初にテストする。

通常、ルーター、ファイアウォール、Web サーバー、電子メールサーバーなど、一般公開され

ているシステムを最初にテストする。そのほかの一般公開されているシステム、ファイアウォールに保護されていないシステム、さらに基幹システムなども優先的にテストする。そのほかのシステムについても、組織が定めた基準に基づいてそれぞれの重要度を評価し、その重要度に従ってテストを実施する。

テストは慎重に実施する。

ネットワークスキャン、脆弱性テスト、侵入テストなどでは、攻撃の再現が可能である。したがって、テストを実施する際には、当該関係者の同意のもとに、協調的な方法で行うことが不可欠である。

組織のニーズがセキュリティポリシーに適切に反映されていることを確認する。

テスト結果を比較する際の基準として、ポリシーを用いる。適切なポリシーが設定されていないと、テストの有用性が著しく制限される。たとえば、ファイアウォールによって特定のトラフィックの通過が許可されたことを検出しても、どのようなタイプのトラフィックやネットワークアクティビティが許可されるのかについてポリシーに規定されていなければ、こうした情報も無意味になる可能性がある。また、ポリシーが設定されている場合には、テスト結果に基づいてポリシーの改善を図ることができる。

リスクマネジメントの一環としてセキュリティテストを実施する。

テストを実施することによってこれまで認識されていなかった脆弱性や設定ミスを発見できる。その結果、たとえば脆弱なシステムへの新しいコントロールの追加や、新しい脅威環境の出現による設定変更などに伴い、現状に応じたテスト頻度の調整が必要になる場合がある。セキュリティテストを実施することにより、組織におけるセキュリティのあり方についてだけでなく、外部からの攻撃にどの程度対処できるか、また組織内の違反行為による金銭的損害や信用の失墜をどの程度回避できるかを知る上で重大な事項が明らかになる。また、テスト結果によっては、ポリシーやセキュリティアーキテクチャの見直しが必要になる場合もある。このように組織のセキュリティの状態を的確に把握することは、リスクマネジメントプログラムが適切に機能する上で極めて重要である。

十分な訓練を積んだ有能なシステム管理者やネットワーク管理者にテストを担当させる。

セキュリティテストは、必ず十分な訓練を積んだ有能な人材に担当させる必要がある。通常は、既にシステム管理を担当しているスタッフがセキュリティテストも担当することになる。システム管理の作業がますます複雑化する一方で、訓練を積んだシステム管理者の数がコンピュータシステムの増加についていけないという現状がある。優秀なシステム管理者の確保は組織のセキュリティ対策において最も重要であるため、所定のスキルを備えた人材を十分に確保して、システム管理やセキュリティテストを適切に実施できるようにする必要がある。

パッチの適用により常にシステムを最新の状態にしておく。

セキュリティテストの結果、多数のシステムにパッチを適用する必要性が生じる場合がある。パッチを適宜適用することにより、組織の脆弱性が露呈されるのを大幅に軽減できる。パッチの適用は一元化し、より多くのシステムにパッチがすばやく適用され、即座にシステムテストを実施できるようにする。

システム全体を把握する。

定期テストを行った結果、システムのセキュリティアーキテクチャの見直しが求められる場合がある。組織によっては、多くのシステムについてセキュリティ要件を特定するといった従来の方法に立ち返り、改めてセキュリティアーキテクチャの見直しを図ることが必要になる場合がある。こうした方法では、インシデントレスポンス運用時の負担は軽減されるが、セキュリティの運用面からすると効率が悪くなるのは必至である。

脆弱性テストの可能性と限界を理解する。

脆弱性テストでは、フォールスポジティブの事象が多数検出される場合や、テストツールの検知機能が不十分であるために特定の問題を検出できない場合がある。隠れた脆弱性の検知を行う際には、脆弱性テストを補完する方法として侵入テストが効果的である。ただし、侵入テストは、リソース集約型の方法であり、十分な専門知識を必要とするため、コストが高くなる場合がある。また、セキュリティテストの結果がどれほど良くても、組織のシステムは攻撃に対して常に脆弱であるということを前提に考える必要がある。

1. はじめに

インターネットの出現により組織や個人のビジネス運営に多くの変化がもたらされ、今やインターネットから享受できる利便性や通信手段なくして効率よいビジネス展開は困難であるとさえ思われる。一方、侵入者の攻撃によって発生する問題もまたインターネットの産物である。こうした攻撃は手動であれ自動であれ、多くの組織に損害や非効率性を生み出し、莫大な費用の支出をもたらしかねない。したがって、組織はインターネットを利用して業務を遂行すると同時に、インターネットのサイトを攻撃から守るための手段を見いだす必要がある。

今日のコンピュータシステムは過去のシステムに比べより強力で信頼性も高いが、管理がより困難であるという側面も否定できない。システム管理は複雑な業務であるため、システム管理者には専門的なトレーニングがますます必要とされている。さらに、トレーニングを積んだシステム管理者の数がネットワークシステムの増加に追いつかないという現状がある。その結果、組織はこれまで以上に踏み込んだ対策を講じて、システムが適切かつ安全に設定されるようにする必要がある。しかも、コスト効率のよい方法で行う必要がある。

このドキュメントでは、インターネットに接続しているシステムやネットワークの運用時におけるテストの実施について説明する。適切なセキュリティコントロールやポリシーに従ってシステムの設定を見直す必要があるかどうかを判断する上で、セキュリティテストはおそらく最も確実な方法であると思われる。また、このドキュメントで紹介するテスト方法は、ネットワーク管理者やシステム管理者をはじめとするセキュリティ担当者がシステムの安全性を確保し、できる限り攻撃からシステムを防御する際に役立てることを主眼としている。こうしたセキュリティテストは、通常のシステム管理やネットワーク管理の一環として実施すれば、インシデントの発生を防止し、未知の脆弱性を発見する上では非常にコスト効率のよい方法である。

1.1 目的と範囲

このドキュメントの目的は、ネットワークセキュリティテストのガイダンスを提供することにある。このドキュメントでは、ネットワークテストの要件を明確にし、限られたリソースの中でどのようにテスト作業の優先付けをするかについて説明するほか、ネットワークセキュリティテストの技法やツールについても説明する。¹ また、組織のネットワーク全体にわたる一貫したセキュリティテストの実施方法について説明し、重複したテスト作業を避けるためのガイダンスを提供する。さらに、組織のミッションやセキュリティ目標に応じたさまざまなネットワークセキュリティテストを紹介しながら、組織にとって実施可能な取り組み方法について説明する。

このドキュメントの主眼は、ネットワークセキュリティのテストプログラムを開始する上で必要となる技術やツールについて基本的な情報を提供することにある。ただし、すべての情報が網羅されているわけではないので、このドキュメントに記載されている参照情報に加えて、ベンダーの製品説明などその他の情報も参考にする必要がある。

¹ フリーウェア(無料ライセンス)やシェアウェア(小額ライセンス)による優れたセキュリティツールは多数あるが、こうしたツールを使用する際には注意が必要である。通常、フリーウェアやシェアウェアのツールの使用に際しては、ソースコードが専門家によってレビュー済であるかどうか、あるいはそのツールが一般に広く採用されていて、安全な既知のリポジトリからダウンロードできるかどうかを確認する必要がある。そうでない場合は、そのツールの使用は避けるべきである。付録 C では、ダウンロードして入手できるよく知られているツールをリストアップして説明する。広く使用されているフリーウェアアプリケーションをサポートする際には、社内の専門家による開発がさらに必要になる場合があるため、莫大なコストがかかる可能性がある。したがって、フリーウェアのサポートにかかるコストと市販製品のコストを比較して、最もコスト効率の良い方法を選択すべきである。

なお、このドキュメントでは、あらゆるネットワークシステムに適用可能な一般的なネットワークセキュリティテストについて説明するが、特に以下のシステムを対象としたセキュリティテストについて詳しく説明する。

- ・ ファイアウォール(内部および外部)
- ・ ルーターとスイッチ
- ・ 侵入検知システムなど関連するネットワーク境界セキュリティシステム
- ・ Web サーバー、電子メールサーバー、その他のアプリケーションサーバー
- ・ ドメインネームサービス(DNS)、ディレクトリサーバー、ファイルサーバー(CIFS/SMB、NFS、FTP など)など、その他のサーバー

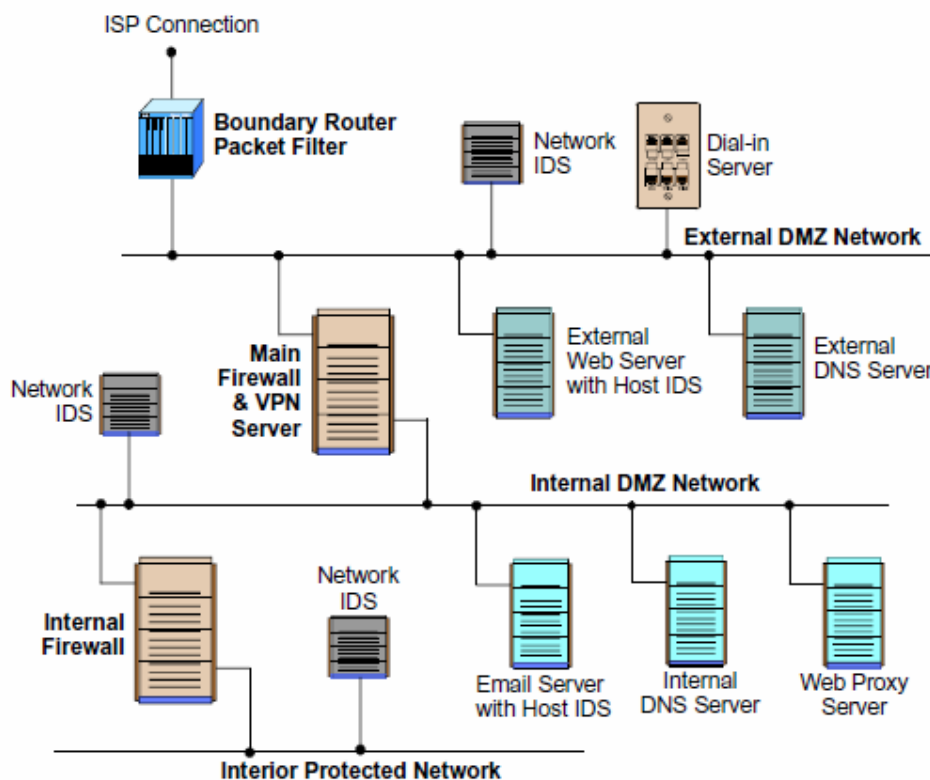


図 1.1: 初期テストの対象となる基幹システムの例

通常、これらのシステムを最初にテストしてから、デスクトップ、スタンドアロン、モバイルクライアントシステムなど組織全体で使用されているクライアントや関連システムについてテストを実施する必要がある。

このドキュメントで説明するテストは、システム開発ライフサイクルのさまざまな段階に応じて実施でき、運用環境でシステムを実行しながら日常のネットワークセキュリティテストプログラムの一環として行うことができるため非常に便利である。

1.2 定義

このドキュメントでは、システム、ネットワークセキュリティテスト、運用テスト、脆弱性といった用語が頻繁に使用されるため、それぞれの用語について以下のように定義する。

システム - 以下のいずれかのシステムを指す。

- ・ コンピュータシステム(メインフレーム、ミニコンピュータなど)
- ・ ネットワークシステム(LAN など)
- ・ ネットワークドメイン
- ・ ホスト(コンピュータシステムなど)
- ・ ネットワークノード、ルーター、スイッチ、ファイアウォール
- ・ 各コンピュータシステムのネットワークアプリケーションやコンピュータアプリケーション

ネットワークセキュリティテスト

ネットワークに関連するセキュリティコントロールを定期的にテストし、検証して、組織のネットワークと関連するシステムの完全性について情報を提供するアクティビティをいう。このドキュメントでは、セキュリティテストまたはテストという表現が頻繁に使用されるが、すべてネットワークセキュリティテストを意味する。また、テストアクティビティには、ネットワークマッピング、脆弱性スキャン、パスワードクラッキング、侵入テスト、ウォードアイリング、ウォードライビング、ファイル完全性チェック、ウイルススキャンなど第3章で説明したすべてのテストが含まれる。

運用セキュリティテスト

システムを運用環境で実行しながら、システムの運用段階で実施されるセキュリティテストを指す。

脆弱性

脆弱性の悪用につながるバグ、設定ミス、または特定の状況の組み合わせを指す。このドキュメントでは、攻撃者による直接的な悪用、分散型サービス妨害(DDOS)などの自動攻撃やコンピュータウイルスによる間接的な悪用のことを脆弱性と呼ぶ場合がある。

1.3 対象とする読者

このドキュメントは、セキュリティプログラママネージャ、技術マネージャおよび職務マネージャ、ネットワーク管理者およびシステム管理者、その他のITスタッフを対象とし、ネットワークセキュリティテストの体系的な方法について説明する。このドキュメントで説明されたテスト手順やツールを使用することにより、システム管理者は管理対象の資産の状態を把握できるようになる。また、このドキュメントは、システムが組織のセキュリティ要件に準拠しているかどうかを評価する上でも参考となり、管理者はこの情報に基づいて意思決定に必要な技術的基盤やサポートについて評価することが可能である。導入済みのセキュリティコントロールを検証、評価するためのテストプランを作成する場合も、このドキュメントを活用できる。

1.4 本ドキュメントの構成

このドキュメントは、以下のように構成されている。

- ・ 第 1 章では、概要について説明する。
- ・ 第 2 章では、テストの理論的解説に加え、システムのライフサイクルとセキュリティテストとの総体的な関係について説明する。
- ・ 第 3 章では、ネットワークセキュリティテストの最終目標と達成目標の定義、テストの重要項目の特定、テスト要件の優先順位付け、積極的または受動的テストなどについて説明する。
- ・ 第 4 章では、限られたリソースの中でどのようにセキュリティテストの優先順位を付けるかについて説明する。
- ・ 付録 A には、このドキュメントで使用する略称の一覧を示す。

- ・ 付録 B には、このドキュメントで使用する参照情報の一覧を示す。
- ・ 付録 C には、テストツールの一覧を示す。
- ・ 付録 D には、ツールの利用例を示す。

なお、項の内容によっては、Linux/Unix、Windows NT/2000/XP、TCP/IP ネットワーキングについて上級者レベルの知識があることを前提に説明されている。

2. セキュリティテストとシステム開発のライフサイクル

運用システムに対するセキュリティテストの主な理由は、潜在的な脆弱性を検出し、修正することにある。日々報告される脆弱性の件数は増加の一途をたどり、たとえば、Bugtraq² データベースに新たに報告される情報システムの脆弱性の件数は 1998 年の開始以来 5 倍以上になり、1 か月平均では当初 20 件程度であったのが、今日では 100 件を超える。多くの組織では 1 人当たりのコンピュータ台数が今後も増加するため、優秀で経験豊かなシステム管理者の需要は高まるばかりである。したがって、システムテストを定期的実施し、脆弱性や設定ミスを検出することにより、システムのセキュリティ侵害を低減することが不可欠になる。

通常、攻撃者は脆弱性を繰り返し悪用し、パッチや修正が適用されていない弱点を突いて攻撃してくる。2000 年 5 月に発行された『SANS Security Alert』には、この問題についてのレポートが掲載されている。このレポートでは、「ソフトウェアプログラムのわずかな欠陥が、大半のインターネット攻撃の原因となっている。攻撃者は余計な手間を嫌うため、ほとんどの攻撃はソフトウェアのわずかな脆弱性を突いて行われる。攻撃者は最も効果的で容易に入手できる攻撃ツールを用いて有名な欠陥を悪用するが、その際にはあくまでも欠陥が修正されていないことを前提に攻撃を仕掛けてくる」³と指摘している。

連邦機関、セキュリティソフトウェアベンダー、セキュリティコンサルティング会社、インシデントレスポンスチームを対象に行った調査では、重要なインターネットセキュリティ脆弱性のランキングで上位 20 項目について意見の一致が見られた。⁴ 『SANS Security Alert』には、これらの脆弱性がリストアップされており、それぞれの弱点克服に向けた提案が簡単に説明されている。このような環境において、ネットワーク保護に関心のあるすべての組織にとってセキュリティテストは非常に重要になってくる。

² <http://www.securityfocus.com/>を参照のこと。

³ SANS Institute、2000 年 5 月発行の『SANS Security Alert』の 1 ページ (<http://www.sans.org/newsletters/sac/>) を参照のこと。

⁴ Ibid の P1 を参照のこと。

2.1 システム開発のライフサイクル

システム開発の様々な段階で、システムセキュリティの評価を行うことができ、また行う必要がある。セキュリティ評価活動には、リスク評価、証明と認定(C&A)、システム監査、セキュリティテストなどがあるが、これに限定されるものではない。それぞれの活動はシステムのライフサイクルの適切な時期に実行する必要がある。こうした活動を行うことにより、システムの開発や運用が組織のセキュリティポリシーに準拠していることを確認できる。この項では、ネットワークセキュリティテストをセキュリティ評価活動として位置付けた場合、どのようにシステム開発のライフサイクルにテストを組み込むかについて説明する。

通常、システムのライフサイクル⁵には、以下の活動が行われる。

1. 開始: システムの目的、ミッション、設定などに基づいて仕様が定義される。
2. 開発および取得: 文書化された手順や要件に従って、システム開発に向けての契約が交わされ、システム構築が行われる。
3. インプリメンテーションおよびインストール: 通常ネットワーク上にシステムをインストールし、ほかのアプリケーションと統合する。
4. 運用および保守: システムのミッションとなる要件に従って、システムの運用および保守を行う。
5. 処分(廃棄): システムのライフサイクルが終了し、ネットワークから切り離してその使用を停止する。

通常、ネットワークセキュリティテストの実施は、インプリメンテーションおよび運用のそれぞれのステージでシステムの開発、インストール、統合が済んだ段階で行われる。図 2.1 には、システム開発のライフサイクルのフローダイアグラムを示す。

⁵ システムのライフサイクルには様々なモデルがあるが、セキュリティテストを実施する際の一般的な状況を説明するためにこのモデルは簡素化されている。

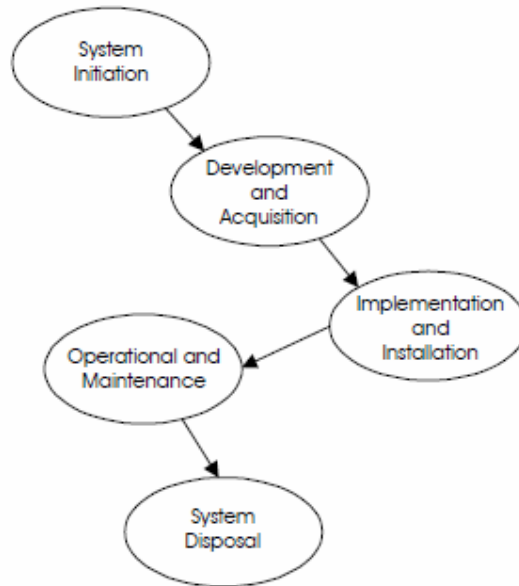


図 2.1 システム開発のライフサイクル

2.1.1 インプリメンテーションステージ

インプリメンテーションステージでは、システムの特典箇所とシステム全体についてセキュリティテストと評価を実施する必要がある。セキュリティテストと評価 (ST&E) とは、情報システムに講じた防御措置を確認または分析することで、システムが完全に統合され運用されている段階で実施される。ST&E には次のような目的がある。

- ・ セキュリティポリシー違反につながりかねない設計、インプリメントおよび運用上の欠陥を発見する。
- ・ セキュリティポリシーの強化に向けて、セキュリティメカニズムやセキュリティ保証などの特性が妥当であるかどうかを判断する。
- ・ システムの仕様と実際のインプリメントとの間でどの程度一貫性があるかを評価する。

通常、ST&E 計画の範囲には、コンピュータセキュリティ、通信セキュリティ、放射セキュリティ、物理的セキュリティ、人員セキュリティ、管理セキュリティ、運用セキュリティが含まれている。

第 3 章で説明されたすべての運用セキュリティテストをこのステージでも実施して、アクティブな

ライブネットワーク上での本格的なインプリメンテーションに先立ち、現在のシステム設定が可能な限り安全であることを確認する必要がある。運用ステージでは、運用セキュリティテストは定期的に繰り返し実行すべきである。

NIST Special Publication 800-26 の「Security Self-Assessment Guide for IT Systems」⁶には、ST&E テストの実施方法についてより詳しく説明しているため、一読しておくといよい。

2.1.2 運用ステージ

システムが運用されてからは、その運用状況を確認することが重要である。この段階では、システムが現行のセキュリティ要件に対応しているかどうかを評価し、システムの運用や使用に関わるユーザーの行動と技術管理上の機能の両方についてテストする必要がある。⁷ システムの運用状況は、第 3 章で説明されたテスト方法を用いて評価することができる。選択するテスト方法や実施頻度は、システムの重要度やテストに使用できるリソースの状況によって異なる。ただし、こうしたテストは定期的実施し、システムに大幅な変更が加えられた場合にも実施する必要がある。絶えず脅威にさらされているシステム (Web サーバーなど) や重要情報を保護するシステム (ファイアウォールなど) については、より頻繁にテストを実施すべきである。

図 2.2 からわかるように、運用ステージはさらに保守ステージに分割されている。保守ステージでは、システムアップグレード、設定変更、あるいは攻撃などの理由で、システムが一次的にオフラインにされる場合がある。

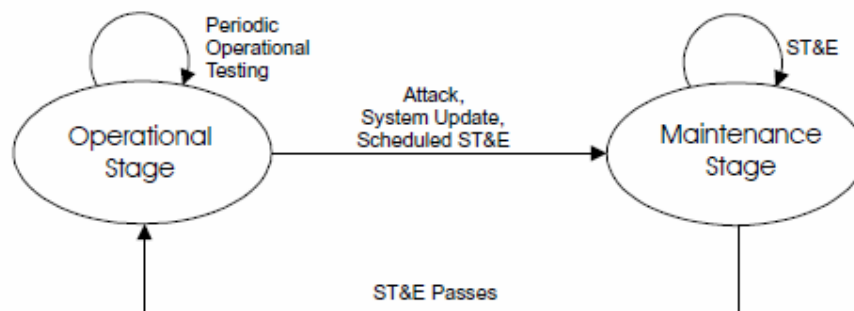


図 2.2 運用および保守ステージにおけるテスト活動

⁶ <http://csrc.nist.gov/publications/nistpubs/800-26/sp800-26.pdf> を参照のこと。

⁷ 米国標準技術局、1996年9月発行の「Generally Accepted Principles and Practices for Securing Information Technology Systems」の P24。

(<http://csrc.nist.gov/publications/nistpubs/800-14/800-14.pdf>) を参照のこと。

運用ステージでは、運用テストを定期的実施する(表 3.2 のテストスケジュールを使用可能)。保守ステージでは、インプリメンテーションステージと同様に ST&E テストの実施が必要になる場合がある。また、システムやアプリケーションの重要度に応じて、システムの運用を再開する前に ST&E テストが必要になる場合もある。たとえば、重要なサーバーやファイアウォールには完全なテストが必要でも、デスクトップシステムにはその必要がないこともある。

2.2 セキュリティテスト結果の記録

セキュリティテストは、リスク分析や緊急時対策など、システム開発のライフサイクルにおけるその他の活動について貴重な判断材料となる。したがって、セキュリティテストの結果を記録して、こうした情報をその他の IT 分野やセキュリティ関連分野の担当者が利用できるようにすべきである。テスト結果は具体的に、次のように使用できる。

- ・ 是正措置の参考情報として使用
- ・ 検出された脆弱性の対処方法の特定
- ・ 組織のセキュリティ要件への準拠度を把握する際のベンチマークとして使用
- ・ システムセキュリティ要件の実行状況の評価
- ・ システムセキュリティの改善に対する費用便益分析
- ・ リスク評価、評価認定(C&A)、パフォーマンス改善など、システムのライフサイクルにおけるその他の活動の促進

2.3 役割と責任

セキュリティテストは、システム開発ライフサイクルの様々な段階で実施でき、それぞれの状況を把握する上で重要な情報を提供するため、セキュリティテストの実施や結果には多くの IT 担当者やシステムセキュリティ担当者が関心を寄せている。したがって、この項では、セキュリティテストにおけるセキュリティ担当者の役割と責任について説明する。ただし、必要とされる役割は組織ごとに異なる場合があるため、ここで説明する役割がどの組織にも当てはまるとは限らない。

2.3.1 上級 IT 管理者/最高情報責任者 (CIO)

上級 IT 管理者と CIO は、組織のセキュリティ体制が適切であることを確認し、上級 IT 管理者については組織全体の情報システムの保護に向けて指示や助言も与える。また、上級 IT 管理者や CIO は、セキュリティテストに関する以下の活動に対して責任を負う。

- ・ 組織の情報セキュリティに関するポリシー、基準、実施手順の構築および維持に向けて調整を行う。
- ・ 組織全体にわたる一貫したセキュリティ評価プロセスの構築と準拠を徹底する。
- ・ 開発プロセスに参加し、セキュリティテストにおける意思決定やテスト対象システムの優先順位付けを行う。

2.3.2 情報システムセキュリティプログラムマネージャ (ISSM)

情報システムセキュリティプログラムマネージャ (ISSM) は、組織のセキュリティポリシーに定められた基準、ルール、規定の導入状況や準拠状況を監視する。また、セキュリティテストに関する以下の活動に対して責任を負う。

- ・ 標準の運用手順(セキュリティポリシー)を構築し、導入する。
- ・ セキュリティポリシー、セキュリティ基準およびセキュリティ要件への準拠を徹底する。
- ・ 基幹システムを特定し、個々のシステムのセキュリティポリシー要件に応じて定期テストをスケジュールする。

2.3.3 情報システムセキュリティ責任者 (ISSO)

情報システムセキュリティ責任者 (ISSO) は、各組織内における情報セキュリティのあらゆる点について監督する責任がある。また、組織の情報セキュリティ業務が、組織全体または部門ごとに設定されたポリシー、基準および実施手順に準拠するよう徹底させる。また、ISSO はセキュリティテストに関する以下の活動に対して責任を負う。

- ・ 担当分野におけるセキュリティ基準と実施手順を構築する。

- ・ セキュリティツールやセキュリティメカニズムの構築と導入に向けた調整を行う。
- ・ メインフレーム、分散型システム、マイクロコンピュータ、ダイヤルアクセスポートなど(これに限定されない)、組織が管理するすべてのシステムの設定プロファイルを管理する。
- ・ テストの実施によりシステム運用面での完全性を維持し、IT 担当者が基幹システムについてスケジュール通りのテストを実施することを確認する。

2.3.4 システムネットワーク管理者

システムおよびネットワーク管理者は、担当するシステムごとにセキュリティ要件への対応を図り、日常業務の一環として行う必要がある。セキュリティの問題については、外部から解決策が提供される場合もあれば(ベンダーまたはセキュリティインシデントレスポンスチームからセキュリティパッチや対応処置などが提供される場合)、組織内で解決される場合もある(セキュリティ担当部門など)。また、管理者はセキュリティテストに関する以下の活動に対して責任を負う。

- ・ システムの完全性、保護レベル、セキュリティ関連のイベントを監視する。
- ・ 担当する情報システムリソースにおいてセキュリティ異常が検出された場合に、対策を講じる。
- ・ 必要に応じてセキュリティテストを実施する。
- ・ 導入されたセキュリティ対策を評価、確認する。

2.3.5 管理者と所有者

システムの管理者と所有者は、システム資産が規定のセキュリティ要件すべてに準拠していることを確認する必要がある。また、テスト結果や提案事項が適切に反映されていることを確認する責任がある。

3. セキュリティテストの技法

セキュリティテストには様々な技法がある。次のセクションでは、それぞれの技法について説明し、さらにその長所と短所について説明する。各テスト技法の特徴については、表3.1と表3.2に要約されている。テストの開始から実行に至るまで大部分を手動で行うテスト技法もあれば、自動化によりあまり人間が介在せずに行われるテスト技法もある。どのようなテスト方法を採用する場合でも、セキュリティテストを設定し実行する担当者はセキュリティやネットワークについて十分な知識が必要であり、ネットワークセキュリティ、ファイアウォール、侵入検知システム、オペレーティングシステム、プログラミング、ネットワークプロトコル(TCP/IP など)について高い専門知識が求められる。

この項では、次のテスト技法について説明する。

- ・ ネットワークスキャン
- ・ 脆弱性スキャン
- ・ パスワードクラッキング
- ・ ログレビュー
- ・ 完全性チェッカー
- ・ ウイルス検知
- ・ ウォーダイアリング
- ・ ウォードライビング(802.11 または無線 LAN テスト)
- ・ 侵入テスト

ネットワークセキュリティの状況を大局的に把握するために、複数のテストを組み合わせる場合がよくある。たとえば、侵入テストではたいていの場合ネットワークスキャンと脆弱性スキャンの両方が実行され、侵入の標的とされる可能性のある脆弱なホストやサービスを特定する。また、脆弱性スキャナではパスワードクラッキングが実行される場合もある。上記のテストを個別に実行したのでは、ネットワークやセキュリティの全体像は把握できない。この項の終わりの表 3.1

に、それぞれのテスト技法の長所と短所を示す。

テストの実行後には、テスト結果の記録、システム所有者へのテスト結果の報告、脆弱性対策やパッチ適用の徹底など、所定の手順に従って対応する必要がある。第 3.11 項では、テストの実施後に行うべき事項について説明する。

3.1 テスト実行者の役割と責任

この項で説明するテストは、テストの一環としてネットワークスキャンを担当するネットワーク管理者など特定の個人だけが実行できる。また、テストの規模によっては、CIO の許可が必要になる場合もある。テストを実施する組織の慣例として、ネットワークマッピングが行われることをほかのセキュリティ担当者や管理者、ユーザーに通知する。こうしたセキュリティテストの多くは攻撃を再現しその痕跡を残すことになるため、関係者にはあらかじめ通知をして、混乱や不要なコストが生じないようにする必要がある。法執行機関への通知がセキュリティポリシーに規定されている場合などは、現地の警察当局に通知しておいた方が賢明である。

3.2 ネットワークスキャン

ネットワークスキャンではポートスキャンも実行し、組織のネットワークに接続される可能性があるすべてのホストを検知する。また、ファイル転送プロトコル(FTP)およびハイパーテキスト転送プロトコル(HTTP)など、これらのホストで運用されるネットワークサービスや、WU-FTPD、IIS、HTTP サービス用 Apache など、特定のサービスを実行するアプリケーションもスキャンされる。スキャンの結果は、ポートスキャンツールでスキャンされたアドレススペース内で稼動しているアクティブなホストやサービス、プリンタ、スイッチ、ルーターをすべて一覧にして表示される。つまり特定のネットワークアドレスを持つデバイスや、別のデバイスへのアクセスが可能なデバイスはすべてスキャンされる。

Nmap⁸などのポートスキャナ(付録 B 参照)では、TCP/IP ICMP の ECHO パケットと ICMP ECHO_REPLY パケットを使用して、ユーザーが指定したアドレス範囲内にあるアクティブなホストが最初に認識される。アクティブなホストが検知されると、今度はオープン TCP/UDP ポート⁹についてスキャンされ、特定のホスト上で稼動しているネットワークサービスが検知される。様々なスキャナがあるがスキャン方法はそれぞれ異なり、長所と短所がある。個々のスキャナの特徴については該当のドキュメントで説明する(付録 D 参照)。たとえば、ファイアウォール経由のスキャンに適しているスキャナもあれば、ファイアウォール内部のスキャンに適しているスキャナもある。TCP/IP プロトコルについて十分理解していない場合は、付録 B で参考資料を確認できる。

基本的にどのようなスキャナもアクティブになっているホストと開いているポートの検出を行うが、中にはスキャンされたホストについて詳細情報を提供するスキャナもある。開いているポートをスキャンした際に収集される情報から、標的にされているオペレーティングシステムを特定できることがよくある。これをオペレーティングシステムのフィンガープリントと言う。たとえば、TCP ポート 135 と 139 が開いているホストの場合、Windows NT または 2000 がホストである可能性が高い。また、TCP パケットのシーケンス番号や、TTL(Time To Live)フィールドなど ICMP パケットへの応答なども、オペレーティングシステムを特定するための手がかりとなる。ただし、オペレーティングシステムのフィンガープリントが絶対確実な方法とは言えない。ファイアウォールでは特定のポートやトラフィックがフィルタリング(ブロック)されるため、システム管理者はシステムの応答方法を変えることにより本当のオペレーティングシステムを偽装することが可能である。

また、特定のポートで実行されているアプリケーションを識別する際に効果的なスキャナもある。たとえば、TCP ポート 80 がホスト上で開いていることが検知されると、ほとんどの場合、そのホストは Web サーバーを実行している。ただし、脆弱性を検知する上で重要なのは、インストールされている Web サーバー製品の種類である。たとえば、Microsoft IIS サーバーの脆弱性は Apache web サーバーの場合とは大きく異なる。クライアント(この場合は Web ブラウザ)が接続した際に、リモートポートで傍受しリモートホストから送信されるバナー情報をキャプチャすることによってアプリケーションを識別できる。通常、バナー情報はエンドユーザー(Web サーバー/ブラウザ)には見えないが、この情報が送信されると、アプリケーションの種類やバージョンだけでなくオペレーティングシステムの種類やバージョンなど豊富な情報が表示される。ただし、セキュリティ意識の高い管理者によってバナー情報が変更される可能性があるため、この方法も絶対確実とは言えない。バナー情報をキャプチャすることを、バナーグラブと呼ぶ場合もある。

⁸ 詳細情報と無料ダウンロードについては、<http://www.insecure.org> を参照のこと。

⁹ TCP/IP 用語では、アプリケーションがトランスポート層(TCP/UDP)から情報を受信する場所をポートと言う。たとえば、TCP ポート 80 で受信されたすべてのデータは Web サーバーアプリケーションに転送される。IP アドレスによって特定のホストが識別されると、そのホスト上で実行されているサービス(HTTP、FTP、SMTP など)がポートから識別できる。

ポートスキャンでは、アクティブになっているホスト、サービス、アプリケーション、オペレーティングシステムが検知されるが、トロイの木馬のポート以外の脆弱性は検知されない。脆弱性の検知が可能なのは、マッピングやスキャンの結果を解釈した人間が手動で行う場合に限られる。このようにすることで、セキュリティ担当者は脆弱なサービスを識別し、トロイの木馬が存在するかどうかを確認できる。スキャン手順自体は自動化が進んでいるが、スキャンされたデータの解釈は自動で行うことはできない。

ネットワークスキャンには次の目的がある。

- ・ 未承認のホストが組織のネットワークに接続していなかどうかを確認する。
- ・ 脆弱なサービスを検知する。
- ・ 組織のセキュリティポリシーに違反するサービスを検知する。
- ・ 侵入テストに備える。
- ・ 侵入検知システム (IDS) の設定を支援する。
- ・ フォレンジック捜査の痕跡を収集する。

スキャンの結果を解釈するには、人間による比較的高度な技術が必要になる。また、スキャンを実行することにより、帯域が消費されネットワーク応答時間が遅くなるため、ネットワークの運用が妨害される可能性もある。一方、ネットワークスキャンでは、組織が IP アドレススペースのコントロールを維持し、承認されたネットワークサービスだけが実行されるようにホストを設定することができる。ネットワーク運用に対する妨害を最小限に抑えるためにも、スキャンソフトを選択する際には十分な注意が必要である(付録 C 参照)。また、ネットワークスキャンを勤務時間後に実施して、運用への影響を小さくすることも可能であるが、その場合には、いくつかのシステムが起動していない可能性があるため注意が必要である。

ネットワークスキャンの結果は記録し、検出された欠陥は修正する必要がある。また、ネットワークスキャンの結果によっては、次の是正措置が必要になる場合がある。

- ・ 未承認のホストを調査し、接続を切断する。
- ・ 不要なサービスや脆弱なサービスを無効または廃止する。

- ・ 脆弱なサービスへのアクセスについては、そのサービスへのアクセスが必要な特定のホスト(ホストレベルのファイアウォールや TCP ラッパーなど)に限定するように、脆弱なホストの設定を変更する。
- ・ エンタープライズファイアウォールの設定を変更し、既知の脆弱なサービスに対する外部アクセスを制限する。

3.3 脆弱性スキャン

脆弱性スキャナは、ポートスキャンの概念をもう一段階進めたものである。ポートスキャナと同様、脆弱性スキャナではホストや開いているポートが検出されるが、同時に関連する脆弱性についての情報も提供される。これは、人間がスキャン結果を分析する方法とは異なる。ほとんどの脆弱性スキャナでは、検出された脆弱性を軽減するための情報が提供される。

また、脆弱性スキャナにはプロアクティブなツールが備わっているため、システムおよびネットワーク管理者は攻撃者より先に脆弱性を発見できる。脆弱性スキャナは、表面的脆弱性に対する組織の脅威を診断するには比較的迅速かつ簡単な方法である。¹⁰

¹⁰ 表面的脆弱性とは、ほかの脆弱性からは孤立して存在する場合の弱点を言う。脆弱性のリスクレベルを特定する上で困難なのは、脆弱性がそれぞれ孤立して存在することはほとんどない点にある。たとえば、特定のネットワーク上にリスクの低い脆弱性がいくつか存在する場合、そうした脆弱性が組み合わさるとリスクの高い脆弱性になる。一般に、脆弱性スキャンでは複数の脆弱性が組み合わさった場合のリスクについては認識されず、個々の脆弱性が低リスクとして診断されるため、ネットワーク管理者に対してセキュリティ対策が十分であるという誤った認識を与えてしまう。複数の脆弱性が組み合わさった場合のリスクを調査するには、侵入テストが信頼性の高い方法である。

脆弱性スキャナは、スキャンされたホストについて脆弱性を検出する。また、旧バージョンのソフトウェアを検出して適用可能なパッチやシステムアップグレードを特定したり、組織のセキュリティポリシーへの準拠度や違反について評価したりする。その際に、脆弱性スキャナでは、オペレーティングシステムやホスト上で稼働中の主なソフトウェアアプリケーションを特定し、既知の危険との照合を行う。スキャナは膨大な脆弱性データベースを使用して、一般的なオペレーティングシステムやアプリケーションに関連する欠陥を識別する。¹¹

また、ほとんどの場合、検出された脆弱性を軽減するための重要な情報やガイダンスが提供される。さらに、脆弱性スキャナは自動的に是正措置を実行し、特定の脆弱性の修正を行う。その場合、脆弱性スキャナのオペレータには脆弱なホストへのルートアクセス権や管理者アクセス権が付与されていることが前提になる。

一方、脆弱性スキャナには重大な弱点がいくつかある。一般に、脆弱性スキャナは表面的脆弱性だけを検出し、スキャン対象のネットワーク全体のリスクレベルには対応できない。また、スキャンの手順自体はかなり自動化されているものの、フォールスポジティブエラー率(脆弱性がない場合に脆弱性ありの報告がされる)が高くなる可能性がある。したがって、ネットワークシステムやオペレーティングシステムのセキュリティや管理について専門知識を持った担当者がスキャン結果を分析しなければならない。

脆弱性スキャナがホストの脆弱性を確実に検出するには、ポートスキャナより多くの情報を必要とするため、スキャンによって生じるネットワークトラフィックもはるかに多くなる傾向がある。これにより、スキャン対象のホストやネットワークに対して、あるいはスキャンを行うトラフィックが通過するネットワークセグメントに悪影響を及ぼす場合がある。また、ほとんどの脆弱性スキャナでは、サービス拒否(DoS)攻撃に対するテストも行われるが、テスト実行者の経験が浅い場合には、スキャン対象のホストに対して大きな悪影響を及ぼしかねない。

脆弱性スキャナにはあらゆる既知の脆弱性を適宜反映させることができないためか、難解な脆弱性よりもよく知られている脆弱性の検出に適している。これは、メーカーがスキャナの処理速度を高速に保とうとする事情とも関係している(つまり、脆弱性の検出件数が多くなればテスト件数も増えるため、スキャン処理全体の速度低下につながる)。

¹¹ NISTでは、脆弱性情報や関連するパッチ情報のデータベースを <http://icat.nist.gov> で管理している。また、このデータベースの脆弱性名称には、ほかのデータベースやベンダーでも採用している CVE(Common Vulnerabilities and Exposures) による統一名称を使用している。脆弱性スキャンには次の機能が備わっている。

- ・ ネットワーク上のアクティブなホストの検出
- ・ ホスト上のアクティブなサービス(ポート)や脆弱なサービス(ポート)の検出
- ・ アプリケーションやバナーグラブの検出
- ・ オペレーティングシステムの検出
- ・ 識別されたオペレーティングシステムやアプリケーションに関連する脆弱性の検出
- ・ 設定ミスの検出
- ・ ホストアプリケーションの用途とセキュリティポリシーへの準拠の評価
- ・ 侵入テストの基盤設定

脆弱性スキャナには、ネットワークベーススキャナとホストベーススキャナの 2 種類ある。ネットワークベーススキャナは、主に組織のネットワークのマッピング、開いているポートと関連する脆弱性の検出のために使用される。ほとんどの場合、ネットワークベーススキャナは対象システムのエクスポートポートの制約を受けない。また、ネットワーク上の 1 台のシステムにインストールするだけで、多数のホストを迅速に識別し、検査することができる。一方、ホストベーススキャナはテスト対象のホストごとにインストールする必要があり、主に特定のホストオペレーティングシステムとホストアプリケーションの設定ミスや脆弱性を検出するために使用される。ホストベーススキャナはネットワークベーススキャナより詳しい脆弱性検知が可能のため、通常、ホスト(ローカル)へのアクセス権だけでなく、ルートアカウントまたは管理者アカウントが必要になる。また、ホストベーススキャナの中には、設定ミスを修正する機能を備えているものもある。

オペレーティングシステムや主要なアプリケーションについてセキュリティパッチやソフトウェアバージョンが最新であるかどうかを確認するためにも、脆弱性スキャンを実行する必要がある。脆弱性スキャンは、スキャン結果の分析に人間の介入をかなり要求するため、人海戦術的な要素がある。また、帯域を消費し、応答時間が遅くなるため、ネットワーク運用が混乱する場合がある。とはいえ、攻撃者に脆弱性を発見され悪用される前に脆弱性を軽減するには、脆弱性スキャンは極めて重要である。脆弱性スキャンは少なくとも 3 か月か半年に 1 度は実施すべきである。ファイアウォール、公開 Web サーバー、その他の境界の侵入ポイントにある基幹システムについては、ほぼ継続的にスキャンする必要がある。また、1 台の脆弱性スキャナですべての脆弱性を検出することは不可能であるため、複数のスキャナを導入すべきである。一般的には市販のスキャナとファ

イアウォールスキャナ¹²を併用する方法が採用されている。

脆弱性スキャンの結果は記録し、検出された欠陥は修正する必要がある。脆弱性スキャンの結果によっては、次の是正措置が必要になる場合がある。

- ・ 脆弱なシステムのアップグレードまたはパッチ適用により、識別された脆弱性を必要に応じて緩和する。
- ・ システムにパッチが即座に適用されない場合(オペレーティングシステムのアップグレードによってアプリケーションが停止してしまう場合等)、脆弱性緩和策(技術または手順面)を講じ、システムが危険にさらされる可能性を最小限にする。
- ・ 設定管理プログラムを改善して、システムが定期的にアップグレードされるようにする。
- ・ 脆弱性の警告やメーリングリストを監視する担当者を割り当てて、組織の環境に妥当かどうかを判断し、適切なシステム変更を実施できるようにする。
- ・ 組織のセキュリティポリシー、アーキテクチャ、あるいは文書の内容を変更し、セキュリティ業務の中でシステムの更新やアップグレードが適宜行われるようにする。

ネットワークベースあるいはホストベースの脆弱性スキャナには、無料、有料の両方がある。入手可能な脆弱性スキャンツールについては、付録 C を参照のこと。

¹² こうした方法が一般的なウイルス対策として採用されている。デスクトップと電子メールサーバにそれぞれ異なる製品を使用することで、相互に不足部分を補うことができる。

3.4 パスワードクラッキング

パスワードクラッキングプログラムを実行することにより、攻撃されやすいパスワードを特定できる。また、ユーザーのパスワードが強力であることを検証することもできる。通常、パスワードはハッシュと呼ばれる暗号として保管され送信される。ユーザーがコンピュータやシステムにログオンしてパスワードを入力すると、ハッシュが作成され保管されているハッシュと比較される。作成されたハッシュと保管されているハッシュが一致すると、ユーザーが認証される。

侵入テストや実際の攻撃では、パスワードクラッキングによってキャプチャされたパスワードハッシュが使用される。パスワードハッシュは、ネットワークを通過して送信するときに傍受されたり(ネットワークスニッファを使用)、攻撃されたシステムから検索することができる。パスワードハッシュをシステムから検索する場合は、そのシステムの管理者アクセス権あるいはルートアクセス権が必要になる。

ハッシュが入手されると、一致するハッシュが見つかるまで、自動パスワードクラッカーによってハッシュが次々に作成される。ハッシュを最も速く作成する方法として辞書攻撃があり、辞書やテキストファイルに登録されたすべての単語が攻撃に使用される。主要言語、マイナーな言語、人名、有名なテレビ番組などあらゆる辞書がインターネットから入手できるため、一般的でない言葉であっても、辞書に登録されている単語である限り攻撃には弱い。

もう一つのクラッキング方法としてハイブリッド攻撃がある。この攻撃は辞書攻撃をベースにしているが、辞書に登録されている単語に数字や記号を追加するといった方法である。ハイブリッド攻撃は、使用するパスワードクラッカーによって様々な方法で実行される。よくある方法としては、文字を記号や数字に置き換える方法がある(p@ssword、h4ckme など)。また、単語の先頭や末尾に記号や数字を追加するといった方法もある(password99、password\$%など)。

パスワードクラッキングの中で最も強力なのが、総当り(ブルートフォース)攻撃と呼ばれる方法である。総当り攻撃は時間がかかる場合もあるが、たいていの場合パスワードポリシーでパスワード変更にかかる時間よりはるかに短時間で実行できる。したがって、総当り攻撃で見つけられたパスワードは攻撃に非常に弱いことになる。総当り攻撃では、ランダムにパスワードが作成され、それに対応するハッシュが作成される。しかしながら、非常に多くの組み合わせが考えられるため、1つのパスワードのクラッキングに何か月もかかることもある。理論的には、十分な時間と処理能力があれば、総当り攻撃によってすべてのパスワードはクラッキング可能である。通常、侵入テストや実際の攻撃では、複数のコンピュータが使用されるため、パスワードクラッキングのタスクを分散できる。また、複数のプロセッサを使用することで、強力なパスワードのクラッキングに要する時間を大幅に短縮できる。

強力とされる Linux や Unix のパスワードは文字長が長く(最低 11 文字以上)、複雑になっている(大文字小文字、特殊文字、数字を取り混ぜている)。一方、Windows パスワードを強力にするには、少し複雑な話になる。Windows2000 以前の Windows バージョンには LanMan パスワードハッシュが用いられており、これには弱点がいくつかある。まず第 1 に、LanMan では大文字小文字の区別がされず、英字はすべて大文字に変換される。そのため、パスワードクラッカーが試行する文字の組み合わせが少なくなる。第 2 に、LanMan のすべてのパスワードは 7 文字から成る 2 つのハッシュとして保存される。したがって、パスワードがちょうど 14 文字の場合には、パスワードを 2 つに分割してそれぞれ 7 文字のハッシュが作成される。一方、パスワードが 14 文字未満の場合は、文字を追加して 14 文字のパスワードにする。LanMan パスワードのハッシュはこのように 2 分割されるため、パスワードクラッキングに対する抵抗力が低くなる。¹³ Windows 用パスワードクラッカーである L0pht Crack の使用例については付録 D を参照のこと。

¹³ パスワードクラッキングを数学的に見ると、7 文字から成る 2 組のハッシュの方が 14 文字のハッシュが 1 組ある場合よりはるかに簡単にクラックできる。パスワードを適切な長さにするには、Windows 2000 以前のバージョンではパスワード長を 7 文字か 14 文字のどちらかにし、大文字小文字、数字、特殊文字を取り混ぜるようにする。特にパスワードに敏感なアカウントの場合には、拡張文字である ASCII 文字が用いられる(標準のキーボードには拡張文字に対応するキーは存在しない)。こうした拡張文字を入力するには、Alt キーを押しながら、テンキーから数字を入力する(たとえば、Alt + 0174 = ®)。拡張文字と対応するキーボード入力については、Windows Character Map アプリケーションを使用して確認できる。

システムに対して毎月一度あるいは継続的にパスワードクラッカーを実行し、組織全体で適切なパスワード構成が使用されるように徹底することが必要である。また、クラックされかねないパスワードが異常に多いと判明した場合は、次のような対策を講じることができる。¹⁴

- ・ ポリシーに従って選択したパスワードがクラックされた場合には、ポリシーを変更して、パスワードのクラック率を低くする必要がある。こうしたポリシー変更によりユーザーがパスワードを覚えにくくなり、その結果メモを取らざるを得なくなった場合は、現在のパスワード認証方法を別の方法に変更することも考慮する必要がある。
- ・ クラックされたパスワードがポリシーに従って選択されていない場合には、クラックに弱いパスワードを選択した場合の影響についてユーザーを教育する必要がある。同じユーザーが繰り返しポリシー違反を犯す場合は、管理者はさらに進んだ対策を講じて(トレーニングの追加、適切なパスワードの選択を強化するパスワード管理ソフトウェアの導入、アクセス拒否など)、ユーザーがポリシーに準拠できるようにする必要がある。ほとんどのサーバープラットフォームでは、管理者が最小パスワード長と文字の組み合わせを設定できる。

パスワードフィルタをサポートできるシステムでは、強力なパスワードが強制的に使用されるようにフィルタを設定することで、パスワードクラッキングの実行頻度を減らし、クラッキングの実行を不要にすることも可能である。どのように強力なパスワードでも、ネットワークを介して自由に送信されることが多い。したがって、組織が率先してより強力な認証方法の採用に取り組むべきである。

¹⁴ 多くのシステム、特にインターネットにさらされているシステムでは、1つのパスワードがクラックされただけでも受け入れがたいとすべきである。攻撃者はシステムに一度アクセスできれば極めて優位な立場になるため、多くの場合ゲストアカウントのパスワードが1つクラックされただけでもシステム全体を十分に攻撃できる(さらに悪い場合には、パスワードもないアカウントがある)。また、管理者レベルのパスワードやルートレベルのパスワードが危険にさらされた場合も、受け入れがたい状況である。

3.5 ログのレビュー

様々なシステムログを確認することにより、IDS ログ、サーバーログ、その他、システムやネットワーク上の監査データを集めたログなど、組織のセキュリティポリシーからはずれたアクティビティを検出できる。ログのレビューや解析は、従来のテストには含まれていないが、稼働中のシステムのアクティビティを動的に把握でき、そうしたアクティビティをセキュリティポリシーの目的や内容と比較することができる。基本的には、監査ログを確認すれば、ポリシーに従ってシステムが稼働しているかどうかを検証できる。

たとえば、IDSセンサーがファイアウォールの後ろ(そのテリトリー内)に設置されている場合、そのセンサーのログを確認すれば、ファイアウォールによってネットワークの通過を許可されたサービスリクエストや通信内容を調査できる。未承認のアクティビティがファイアウォールを越えて侵入したことがセンサーで検知された場合は、ファイアウォールの設定はもはや安全ではなく、ネットワークにバックドアが存在することを意味する。

Snort は豊富なサポート機能を備えた無料の IDS センサーである。また、ネットワーク侵入検知システムとして、リアルタイムトラフィック分析と IP ネットワーク上でのパケットロギングができる。Snort はプロトコル分析、コンテンツの検索やマッチングを実行でき、バッファオーバーフロー、ステルスポートスキャン、CGI(共通ゲートウェイインターフェース)攻撃、SMB(システムメッセージブロック)偵察、OS フィンガープリントなど様々な攻撃や偵察行為を検知できる。また、Snort には、回収すべきトラフィックや通過すべきトラフィックを記録するための柔軟なルール言語や、モジュール式プラグインアーキテクチャを採用した検出エンジンが使用されている。Snort のリアルタイム警告機能には、syslog 用警告メカニズム、ユーザー指定ファイル、Unix ソケットが備わっており、Samba の smbclient によって Windows クライアントに WinPopup メッセージを表示することもできる。Snort には主として三通りの用途がある。つまり、tcpdump のようなパケットスニッファとして、パケットロガー(ネットワークトラフィックのデバッグなどに便利)として、あるいは完全なネットワーク侵入検知システムとして使用できる。

手動による監査ログのレビューは極めて厄介であり、時間がかかる作業である。監査ツールの自動化により、レビューにかかる時間を大幅に短縮でき、一連のアクティビティに対するログの内容をまとめたレポート(事前定義またはカスタマイズ)を作成できる。ログに適用されたフィルタが不要なものだけを除外し、それ以外のものはすべて通過させるように設定することが重要である。

主要なサーバーやファイアウォールに対しては、ログのレビューは毎日行わないとしても頻繁に実行する必要がある。またこの場合にも、ログ確認ツールを使用することは、システム管理者が問題や疑わしいアクティビティを特定する上で非常に便利である。必要なセキュリティ設定の導入

状況をテストする場合は、1か月に一度行えば十分である。ただし、主要なシステムアップグレードの結果、レビューが必要になる場合は例外である。システムの設定がポリシーに従っていない場合は、次のような対策を講じることができる。

- ・ 脆弱なサーバーは不要であれば削除する。
- ・ 必要に応じてシステムを再設定し、攻撃のチャンスを抑える。
- ・ ファイアウォールポリシーを変更して、脆弱なシステムやサービスへのアクセスを制限する。
- ・ ファイアウォールポリシーを変更して、攻撃のソースとなる IP サブネットからのアクセスを制限する。

3.6 ファイル完全性チェッカー

ファイル完全性チェッカーは、保護対象のファイルごとにチェックサムを計算して保存し、ファイルチェックサムのデータベースを構築する。また、システム管理者がファイルの変更、特に未承認の変更を識別する際のツールとなる。保存されたチェックサムを定期的に計算し直して、現在の値と保存されている値をテストし、ファイルの変更を検出する必要がある。通常、ファイル完全性チェッカーの機能は、市販のホストベース侵入検知システムに含まれている。

完全性チェッカーは、人間によるやり取りをあまり必要としないため便利なツールであるが、効果的に実行されるように注意して使用する必要がある。ファイル完全性チェッカーでは、最初の参照データベースを作成するために安全なシステムが必要である。セキュリティが既に侵害されたシステムでは暗号ハッシュが作成される場合があるため、このイベントによってテスト実行者にセキュリティについて誤った認識を与えかねない。攻撃者がシステムのセキュリティを侵害したり、データベースを変更して攻撃の痕跡を隠蔽したりできないように、参照データベースはオフラインで保存する必要がある。また、ファイル完全性チェッカーによってもフォールスポジティブの警告が発行される可能性もある。ファイルの更新やシステムパッチの導入が実施されるたびにファイルが変更されるため、チェックサムデータベースの更新が必要になる。その結果、データベースを最新の状態に保つのが難しくなる場合がある。しかしながら、完全性チェッカーは、たとえ一度しか実行されないとしても(システムのインストール時)、セキュリティが侵害された疑いがある場合に変更されたファイルを特定するには便利なアクティビティであることに変わりはない。近頃は、一般的に使われている 32 ビット巡回冗長検査(CRC)チェックサムによって検知されないような方法で、攻撃者はファイルを変更できるようになった。したがって、チェックサムデータベースに保存されているデータの完全性を確保するためにも、SHA-1 などの強力なチェックサムの使用を推奨する。

完全性チェッカーは、セキュリティ侵害の影響を受けかねないシステムファイルに対し毎日実行すべきである。また、見込まれる被害の範囲を特定する際にセキュリティ侵害の疑いが認められた場合にも、完全性チェッカーを実行すべきである。システムファイルに対する未承認の変更が完全性チェッカーによって検知された場合は、セキュリティの問題とみなして、組織のインシデントレスポンス、レポートポリシー、対処手順に従って調査を行う必要がある。ファイル完全性チェックのためのフリーウェアである LANguard の使用例については、付録 C を参照。

3.7 ウイルス検知

インターネットに接続していたり、リムーバブルメディア(フロッピーディスク、CD-ROM など)、シェアウェアあるいはフリーウェアのソフトウェアを使用している組織はすべて、コンピュータウイルス、トロイの木馬、ワーム¹⁵への感染の危険性がある。ウイルス、トロイの木馬、ワームによる影響の程度は、コンピュータ画面にポップアップメッセージが表示されるといった無害のものからハードドライブ上のすべてのファイルを削除するといった破壊的なものにまで及ぶ。また、悪意あるコードの場合には、機密情報の漏えいや破壊の危険性もある。

主なウイルス対策プログラムとしては、ネットワークインフラストラクチャにインストールするプログラムと、エンドユーザーのコンピュータにインストールするプログラムの 2 種類がある。どちらの方法にも長所短所はあるが、高いセキュリティレベルが求められる場合には、通常両方のプログラムをインストールする必要がある。

通常は、ネットワークインフラストラクチャにインストールされているウイルス検出ソフトをメールサーバーにインストールしたり、組織のネットワーク境界に設置したファイアウォールと組み合わせられて使われる。サーバー型ウイルス検出プログラムは、ウイルスがネットワークに侵入する前や、ユーザーが電子メールをダウンロードする前にウイルスを検出できる。また、すべてのウイルス検出プログラムは頻繁に更新して有効状態にしておく必要があるという点においても、サーバー型ウイルス検出プログラムには利点がある。サーバー型プログラムはクライアントホストに比べて数が限定されるため、更新がはるかに容易である。

¹⁵ これらはすべて悪意あるコードの例で、総称してウイルスと呼ばれる場合があるが、それぞれの感染や蔓延の形態はかなり異なる。

もう一方のウイルス検出ソフトはエンドユーザーのコンピュータにインストールされる。このソフトは電子メール、フロッピー、ハードディスク、ドキュメントなどから悪意あるコードを検出するが、その対象になるのはローカルホストだけである。また、場合によっては、Web サイトから悪意あるコードを検出することもある。この種のウイルス検出プログラムはネットワークパフォーマンスへの影響が少ないが、ウイルス定義の更新をエンドユーザーに依存しているため、必ずしも信頼できる方法とは言えない。現在では、ほとんどのウイルス対策ソフトには、ウイルス定義リストを自動更新する機能が備わっている。

たとえどのようなウイルス検出プログラムを使用しても、あらゆるウイルスを識別できるようにウイルス識別データベース(時にはウイルス署名データベースとも呼ばれる)が最新の状態に保たれていなければ、完全なウイルス対策にはなり得ない。ウイルス識別データベースが最新の状態でなければ、新しいウイルスが検出されることはない。ウイルス対策ソフトはウイルスを検出する際、ファイルの内容を既知のコンピュータウイルス定義と比較して、感染したファイルを特定し、可能な場合には感染ファイルの隔離や修復を行うが、それが不可能な場合には感染ファイルを削除する。より高度なプログラムでは、現在使用しているウイルス検出データベースでは識別されない可能性がある新しいウイルスや変異したウイルスを検出するために、ウイルスに似たアクティビティについても検出される。この方法は完全ではなく、フォールスポジティブが検出される可能性もあるが、ウイルス対策を保護層を一段増やすことができる。

ワームやトロイの木馬といったウイルスやその他の悪意あるコードは、コンピュータシステムに多大な損害をもたらす可能性がある。ウイルス検出ソフトで最も重要なのは、ウイルス定義ファイルのアップデートを定期的かつ頻繁に提供し、大規模なウイルスがインターネット上で蔓延している場合にはオンデマンドアップデートを提供することにある。データベースが頻繁に更新されていれば、より多くのウイルスがウイルス対策ソフトによって検出される。こうした措置が事前に講じられていれば、大規模なウイルス感染の危険性も最小限にとどめられる。次に、ウイルス対策ソフトを使用する上での留意事項を示す。

- ・ ウイルス定義ファイルは少なくとも週に一度は更新し、大規模な新型ウイルスが発生した場合には必ず更新する。
- ・ ウイルス対策ソフトはバックグラウンドで絶えず実行し、可能であれば経験則に基づいてウイルスが検出されるように設定する。
- ・ ウイルス定義ファイルの更新後に、完全なシステムスキャンを実行する。

3.8 ウォーダイアリング

適切に設定されたネットワークでも、未承認のモデムは脆弱性として見落とされる場合がよくある。こうした未承認のモデムは、ほとんどすべてのセキュリティ対策をすり抜けるための手段となる。攻撃者やネットワーク管理者が利用可能なモデムのありかを突き止めるために電話番号を次々にダイヤルできるソフトウェアパッケージが存在する(付録 C 照)。これをウォーダイアリングと言う。4 台のモデムに接続されているコンピュータであれば、数日で 10,000 件の電話番号にダイヤルできる。ウォーダイアラーの中には、モデムを突き止めると、限定的に自動ハッキングを試みる者もいる。モデムを使って発見された番号についてはすべて公開される。

ウォーダイアリングは少なくとも年に一度は実施する必要がある、従業員や企業の電話システムへの支障をなるべく少なくするように勤務時間後に行う(ただし、勤務時間後にモデムがオフになると検出ができなくなるため、その辺のバランスを考えて調整する)。その場合、組織の電話番号はすべて確認する必要があるが、大量の電話がかかってくると悪影響がある番号については除外する(24 時間体制のオペレーションセンター、緊急電話番号など)。ほとんどのウォーダイアリングソフトでは、テスト実行者が電話番号リストから特定の番号を除外できるようになっている。

未承認のモデムが検出された場合には、調査を行い、必要に応じて切り離す。構内交換機(PBX)の管理者に問い合わせれば、該当の番号が割り当てられているユーザーを特定できるはずである。ただし、モデムを外すことができない場合は、そのモデムへのインバウンドコールがブロックされるように PBX を設定する必要がある。また、インバウンドコールが必要な場合には、強力な認証方法を導入すべきである。

インターネット経由の攻撃が多くの注目を浴びているが、多くの攻撃は未承認のモデムによって仕掛けられている。ほとんどのラップトップにはモデムが付いているため、ラップトップの普及に伴いこの問題が深刻化してきている。承認されたモデム経由で番号を1つ見つけ出せば、攻撃者は境界のセキュリティをすり抜けて、検知されずにネットワークに直接アクセスできる可能性もある。

3.9 無線 LAN テスト(ウォードライブ)

無線技術はネットワークの中でも急速に成長しつつある分野である。最も有名な無線 LAN プロトコルは 802.11b であるが、802.11b に採用されている現在の WEP(Wireless Equivalent Privacy) には深刻な欠陥がある。その上、802.11b が実装されたほとんどの装置ではデフォルト設定が安全でないため、さらにリスクが助長されている。無線 LAN は、攻撃者がファイアウォールや IDS をすり抜けるための手段となるため、未承認のモデムに代わるネットワーク¹⁶へのバックドアとして急速に広まりつつある(ファイアウォールの外部に配置されていない場合)。

攻撃者や悪意ある侵入者は、無線ネットワークカードを備えたラップトップを携帯して、事務所の駐車場や近隣地域を定期的に巡回し、オープンになっているアクセスポイントへの接続を試みようとする(これをウォードライビングと言う)。発見された無線ネットワークの場所については、Web サイトで公開されている(<http://www.netstumbler.com> 参照)。多くの無線装置の通信可能範囲は現在 300~600 フィートであるが、メーカーが新製品を発表するのに伴い通信可能範囲はさらに広がっている。攻撃者は、カードの受信範囲を広げるために、より大きなアンテナを無線ネットワークカードに付けている場合も少なくない。

残念ながら、802.11b ネットワークプロトコルにはセキュリティ上の脆弱性が多数あり、次のような攻撃を受けやすくなっている。

- ・ 挿入攻撃
- ・ 無線トラフィックの傍受と監視
- ・ サービス拒否
- ・ Client to Client 攻撃

¹⁶ NIST Special Publication 800-48 の『Wireless Network Security: 802.11, Bluetooth, and Handheld Devices』には、無線セキュリティについて詳しく説明されている
http://csrc.nist.gov/publications/nistpubs/800-48/NIST_SP_800-48.pdf を参照のこと。

アクセスポイントがデフォルトのまま最も安全性の低い状態で設定されていると、無線ネットワークにおけるセキュリティのリスクはさらに高まる。こうした設定ではインストールは簡単であるが、無線ネットワークをインストールするネットワーク管理者やユーザーがセキュリティ上の責任を負わざるを得なくなる。

多くの組織にとって、無線ネットワークのメリットがリスクを上回っているようである。比較的安全な無線ネットワークを運用するには、無線ポリシーを作成して、すべての従業員と提携業者にポリシーの浸透を図る必要がある。その低コストと使いやすさを考えると、未承認や設定が不適切な無線 LAN がないかどうかネットワークを定期的にテストしたり、サイトを定期的にスキャンして近隣の無線 LAN から危険な兆候を受信していないかどうかを確認する必要がある。無線ネットワークカードとテストツール(付録 C 参照)を備えた携帯用コンピュータを 1 台以上用意すれば、こうした無線 LAN の検出も実行しやすくなる。無線ネットワークのテスト頻度は、次の条件によって異なる。

- ・ テストする場所の物理的要因(たとえば、一般のアクセスから数千フィート離れた安全な施設に所在する建物の場合には、往来の激しいビジネス街にある事務所に比べてテスト頻度は少なくなる)
- ・ 組織が直面する脅威レベル
- ・ ネットワークリソースに対する組織の管理形態(たとえば、ネットワークを一元管理している組織の場合は、分散型ネットワークをサポートしている組織よりテスト頻度は少なくなる)
- ・ WPA(Wi-Fi Protected Access)や RSN(Robust Security Network)などのより強力なネットワークセキュリティ技法の採用
- ・ 組織のネットワークにおけるデータの機密性

一般的な基準として、リスクや脅威が高い組織では少なくとも月に一度は未承認や不適切な設定の無線 LAN についてテストを行う必要がある。また、不定期な監査も行うべきである。

3.10 侵入テスト

侵入テストとは、システムの設計や導入方法について知識のあるテスト実行者がシステムのセキュリティ機能の回避を試みるセキュリティテストである。侵入テストの目的は、攻撃者が通常使用するツールや技法を用いてシステムへのアクセス方法を特定することにある。したがって、侵入テストを行う前に十分に検討、計画し、関係者などに通知する必要がある。

侵入テストは組織の情報セキュリティ計画にとって重要な活動であるが、多くのテスト要員が必要になり、またテスト対象システムへのリスクを最小限に抑えるための高度な専門知識が求められる。影響が最小限に抑えられたとしても、ネットワークスキャンや脆弱性スキャンによって組織のネットワークの応答時間が遅くなることがある。さらには、侵入テストの過程でシステムが実際に破壊されたり、運用不能に陥ったりする危険性がある。強いて言えば、侵入者も同様にシステムを運用不能にできるという現実を知ることができるという利点はある。侵入テストの担当者が経験豊かであれば、こうしたリスクは軽減できるかもしれないが、完全になくなることはあり得ない。

侵入テストは法規制や組織のポリシーに反するツールや技術を使用して、攻撃を模擬実験することにあるため、侵入テストを実施する前に正式な許可を得ておくことが不可欠である。これを活動規則とも言い、次の項目が含まれている。

- ・ テスト対象の IP アドレスとその範囲
- ・ テスト対象外ホスト(テストの対象とされないホスト、システム、サブネットなど)
- ・ 使用可能なテスト技法(ソーシャルエンジニアリング、DoS など)とツール(パスワードクラッカー、ネットワークスニッファなど)のリスト
- ・ テストの実施時間帯(営業時間、勤務時間外など)
- ・ テスト期間の指定
- ・ 侵入テストを実施するコンピュータの IP アドレス(侵入テストによる合法的攻撃と実際の悪意ある攻撃とを管理者が区別できるようにする)
- ・ 侵入テスト実行チーム、テスト対象システム、テスト対象ネットワークの連絡先
- ・ テストで発生する誤警報による法執行機関の出動を防止する措置

- ・ 侵入テストチームが収集した情報の処理

侵入テストは公開、非公開のどちらでも実行できる。侵入テストをどちらの方法で行うかによって、ブルーチームまたはレッドチームと呼ばれる。ブルーチームの場合は、組織の IT スタッフに通知しその同意の下で侵入テストが行われる。一方、レッドチームの場合は、組織の IT スタッフには通知せずに、上層部にだけ通知し許可を得てテストが行われる。組織によっては、信頼できる第三者にレッドチームの実施を依頼することがある。その場合、攻撃が実際に仕掛けられていることが確認されれば、実際の攻撃に対する場合と同じ対策が講じられる(つまり、IT スタッフが目にするアクティビティは演習ではなく、実際の攻撃として認識される)。この信頼できる第三者からは、テスト実行者、管理者、IT スタッフ、セキュリティスタッフに対応するエージェントが派遣され、活動の仲立ちをし、コミュニケーションが円滑に行われるようにする。この種のテストでは、ネットワークセキュリティだけでなく、IT スタッフによるセキュリティインシデントへの対応、組織のセキュリティポリシーの知識や実行についてもテストできるという利点がある。レッドチームは、警告して行う場合と警告なしで行う場合がある。

一方、ブルーチームによる侵入テストは、レッドチームに比べてコストがかからず、頻繁に実施できる。レッドチームの場合は、秘密に実行する要件が伴うため、より時間とコストがかかる。レッドチームを秘密の環境で行うには、スキャンなどの行為をゆっくり実行し、テスト対象の侵入検知システムやファイアウォールの検知能力を超えないように振舞う必要がある。しかしながら、レッドチームの場合は、システム管理者がテストについてあまり意識していないため、日常のセキュリティ状況を把握する上ではよい方法である。

侵入テストでは内部攻撃と外部攻撃に対するシミュレーションを行うことができる。内部テストと外部テストの両方をする場合、通常は外部テストが最初に実施される。外部侵入テストでは、外部ソースから内部ネットワークに許可されるトラフィックの量と種類がファイアウォールによって制限される。どのプロトコルが通過を許可されるかによって異なるが、FTP、HTTP、SMTP、POP など一般的なアプリケーションプロトコルが最初の攻撃の標的とされる。

実際の外部攻撃をシミュレーションする場合には、テスト実行者には攻撃対象の IP アドレスとその範囲についてだけ知らされ、それ以外は攻撃対象の環境について実際の情報は提供されない。したがって、テスト実行者自身がこうした情報をひそかに収集しなければならず、公開されている Web ページやニュースグループなどのサイトを参考に情報収集を行う。次に、ポートスキャナや脆弱性スキャナを使用して攻撃対象となるホストを特定する。外部攻撃ではファイアウォールから侵入する可能性が高いため、内部攻撃の場合よりはるかに少ない情報を収集すればよいことになる。外部から侵入できそうなネットワーク上のホストをいくつか特定できたら、そのうちの 1 つのホストに不正アクセスを試みる。アクセスに成功したら、このホストを踏み台にして、通常は外部からの

アクセスができないほかのホストにも不正アクセスを仕掛ける。これからわかるように、侵入テストでは最小限のアクセスを利用してさらにアクセスを広げるという反復処理が行われる。

内部侵入テストは外部侵入テストと似ているが、攻撃対象が(ファイアウォールを超えて)内部ネットワークになり、ネットワークへのアクセス権がある程度付与される(通常はユーザーとして、場合によってはさらに上位レベルのアクセス権が付与される)。侵入テストの実行者は、与えられたアクセス権をさらに利用して、ネットワークへさらに深くアクセスしようと試みる。テスト実行者はこのようにして、特定の権限を持つ者だけがアクセスできるネットワーク情報を得ることができる。通常は、一般の従業員と同等のアクセス権を得られるが、テストの目的によってはシステム管理者またはネットワーク管理者レベルのアクセス権まで得られる。

侵入テストは、4つのフェーズで構成される(図 3.1 参照)。

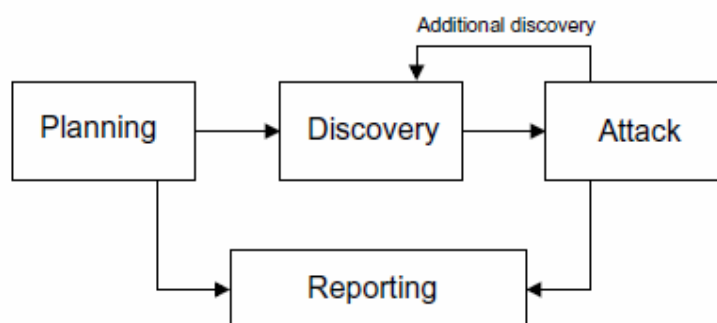


図 3.1: 4つのフェーズからなる侵入テスト

計画フェーズでは、ルールが確認され、経営管理者の最終承認が得られ、テストの目標が設定される。このフェーズでは、侵入テストを成功させるための下準備が行われる。計画フェーズではテストはまだ実施されない。

発見フェーズではテストが実際に開始される。このフェーズでは、第 3.2 項で説明したネットワークスキャン(ポートスキャン)が実行され、攻撃対象が特定される。ポートスキャンに加え次のような技法も用いて、攻撃対象となるネットワークについての情報が収集される。

- ・ DNS への問合せ
- ・ InterNIC(whois)照会
- ・ 攻撃対象組織の Web サーバー検索による情報収集

- ・ 攻撃対象組織の LDAP サーバー検索による情報収集
- ・ パケットキャプチャ(通常は内部テストの場合のみ)
- ・ NetBIOS 列挙(通常は内部テストの場合のみ)
- ・ NIS(通常は内部テストの場合のみ)
- ・ バナーGrab

発見フェーズの 2 番目のステップとして、脆弱性分析がある。このステップでは、スキャンされたホストのサービス、アプリケーション、オペレーティングシステムが脆弱性データベースと比較される(脆弱性スキャナの場合は、この処理は自動的に行われる)。通常は、テスト実行者が自分のデータベースやパブリックデータベースに基づいて脆弱性を手動で識別する。¹⁷ 手動による識別方法は、新しい脆弱性やあまり知られていない脆弱性の識別する上でより効果的であるが、自動スキャナよりはるかに時間がかかる。

攻撃の実行は、侵入テストの最も重要な部分である。ここでは、前のフェーズで発見した脆弱性をエクスプロイトし、実際に検証を行う。攻撃が成功すると、脆弱性が検証され、セキュリティ危機を軽減するための防衛措置が特定される。攻撃の実行時に行われるエクスプロイト¹⁸では、攻撃者であれば得られる最大レベルのアクセスは許可されないことがよくある。その代わりに、エクスプロイトを通してテストチームは攻撃対象のネットワークとその潜在的な脆弱性についてより多くのことを知ることができる場合があり、またそのネットワークのセキュリティ状態の変更につながる場合がある。どちらの場合でも、さらに分析とテストを行い、ネットワークの実際のリスクレベルを把握する必要がある。この点については、「図 3.2 発見フェーズへループバックされる攻撃フェーズのステップ」のフィードバックループに示されている。

¹⁷ 有名な脆弱性データベースとして、<http://icat.nist.gov/icat.cfm> や <http://cve.mitre.org/> <http://www.securityfocus.com/>などがある。

¹⁸ エクスプロイトとは、脆弱性を悪用して攻撃方法として組み立てた手順、つまり脆弱性を突いたプログラムやスクリプトのことである。フリーウェアに対する注意事項は、エクスプロイトプログラム(スクリプト)にも当てはまる。www.securityfocus.com など多くの脆弱性データベースでは、ほとんどの検出済みの脆弱性についてエクスプロイトの手順やコードが明らかにされている。実際には、エクスプロイトプログラム(スクリプト)は特定の脆弱性を悪用するための専用ツールである。

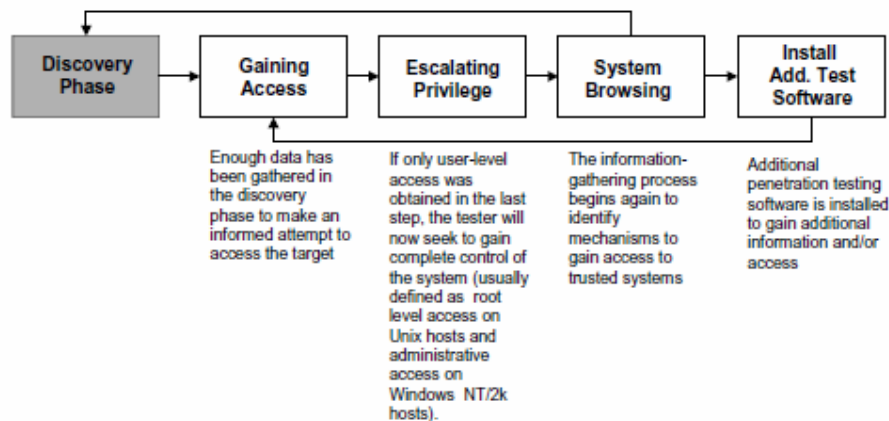


図 3.2: 発見フェーズにループバックされる攻撃フェーズのステップ

脆弱性スキャナでは脆弱性が存在する可能性だけが確認されるが、侵入テストの攻撃フェーズでは脆弱性を悪用して、脆弱性が存在することを実証する。侵入テストや悪意ある攻撃で悪用されるほとんどの脆弱性は、次のいずれかに分類される。

- ・ カーネルの欠陥

カーネルコードはオペレーティングシステムの中心部分を指す。カーネルコードによってシステムの全般的なセキュリティモデルが実行されるため、カーネルにセキュリティ上の欠陥が発生するとシステム全体が危険にさらされる。

- ・ バッファオーバーフロー

入力された文字列の長さがプログラムによってチェックされなかった場合にバッファオーバーフローが発生する。これは通常、プログラミングの甘さに起因している。バッファオーバーフローが発生すると、任意のコードがシステムに導入され、実行プログラムの権限を利用して実行される。このコードは、Unix システムのルートとして、また Windows システムの SYSTEM(管理者に相当)として実行されることが多い。

- ・ シンボリックリンク

シンボリックリンク (symlink) とは、ファイルのすり替えを指す。通常、ファイルに付与されたアクセス許可がプログラムによって変更される。アクセス許可を設定したプログラムが実行されると、ユーザーが戦略的に symlink を作成してプログラムを欺き、基幹システムのファイルを修正したりリストしたりする場合がある。

- ・ **ファイルディスクリプタ攻撃**

ファイルディスクリプタとは、システムがファイルを管理する場合にファイル名の代わりに使用するプラスの整数のこと。特定のファイルディスクリプタには暗黙の用途がある。権限のあるプログラムによって不適切なファイルディスクリプタが割り当てられると、そのファイルはセキュリティ侵害の危険にさらされる。

- ・ **レースコンディション**

プログラムやプロセスが特権モードを放棄する前に特権モードに入ったような場合にレースコンディションが発生する。プログラムやプロセスが特権モードにあるときに、そのプログラムやプロセスを利用する攻撃のタイミングをユーザーが決めることができる。特権モードにあるプログラムやプロセスのセキュリティ侵害に成功すれば、攻撃者はレースに勝ったことになる。通常、レースコンディションではシグナルハンドリングとコアファイルマニピュレーションが行われる。

- ・ **ファイル/ディレクトリへのアクセス許可**

ファイル/ディレクトリへのアクセス許可では、ファイルやディレクトリに対するユーザーやプロセスのアクセス許可を管理する。適切なアクセス許可はシステムのセキュリティにとって重要である。アクセス許可が不適切な場合には、様々な攻撃が仕掛けられ、パスワードファイルの読取りや書込み、信頼できるリモートホストリストへのホストの追加などが行われる。

- ・ **トロイの木馬**

トロイの木馬プログラムには BackOrifice、NetBus、SubSeven などがあり、カスタムビルドすることもできる。アクセスが確立されると、カーネルルートキットを使用していつでもシステムにバックドアを設置できる。

- ・ **ソーシャルエンジニアリング**

ソーシャルエンジニアリングとは情報システムへのアクセスや情報システムに関する情報を入手するために説得術や偽装行為を使用する方法を言う。一般的には、人間同士の会話ややり取りによって行われる。通常使用する手段は電話であるが、電子メールや直接会って行われる場合もある。ソーシャルエンジニアリングでは、2 種類のアプローチが展開される。最初のアプローチでは、侵入テストの実行者がユーザーが困った状況にあるようなふりをして、組織のヘルプデスクに電話をかけ、対象のネットワークやホストの情報を聞き出し、ログイン ID や信用情報を入力してパスワードをリセットさせるといった方法が使われる。2 番目のアプローチでは、ヘルプデスクのふりをして、ユーザーに電話をかけ、ユーザーの ID とパスワードを聞き出すという方法がとられる。この方法はかなり効果的であるといえる。

報告フェーズは、侵入テストのほかの3つの各フェーズと同時に発生する(図3.1参照)。計画フェーズでは、作業ルール、テスト計画、許可書などが作成される。発見フェーズと攻撃フェーズでは、通常ログを保管して、必要に応じてシステム管理者や経営陣などに定期的にレポートを提出する。一般に、テストの終了時に、テスト全体の報告書を作成し、検出された脆弱性、リスク評価、検出された弱点の軽減に向けたガイダンスについて説明する。

侵入テストは、組織のネットワークの脆弱性の程度、ネットワークが標的にされた場合の被害の程度を把握するうえで重要である。侵入テストはコストがかかり潜在的な影響が伴うため、1年に1回実施すれば十分であると思われる。侵入テストの結果は真摯に受け止め、検出された脆弱性を軽減するための措置を講じる必要がある。テスト結果は用意ができ次第、組織の管理職に提出すべきである。

是正措置として、検出され悪用された脆弱性の解決、組織のセキュリティポリシーの変更、セキュリティの改善手順の作成などを実施するほか、従業員のセキュリティ意識を養うトレーニングを実施して、不適切なシステム設定やセキュリティがもたらす影響について確実に理解をうながす。また、テスト実行者の人数を減らして定期的にテストを実施することにより、組織のセキュリティポリシーが順守され、必要なセキュリティレベルが維持されていることを確認する必要がある。次の侵入テストを実施するまでの間にほかのテスト(ネットワークスキャン、脆弱性スキャンなど)を実施し、発見された欠陥を是正しておけば、次回の侵入テストにも実際の攻撃に対しても十分に準備が整った状態で臨むことができる。

3.11 テスト後の対応

ほとんどの組織の場合、テストを実行することにより迅速な対応が必要な問題が明確になるようである。こうした問題への対処や緩和の仕方は、テストの過程において最も重要なステップである。問題に対する最も一般的な根本的原因と対処方法について次に説明する。

組織のセキュリティポリシーの欠如(または徹底不足): おそらく、システムの不適切なセキュリティの最大の原因は、組織のセキュリティポリシーが欠如していることにある。セキュリティポリシーは整合性を確保する上で重要である。整合性は、予測可能な行為につながるため、適切なセキュリティ状態を保つには重要な要素である。整合性があるからこそ安全な設定を容易に維持することができ、セキュリティ上の問題も容易に特定できる(予測可能な行為に反するものが問題とみなされるため、わかりやすい)。組織ごとにセキュリティポリシーを設定し、ユーザーや管理者に対してポリシーを周知する必要がある。一般に、セキュリティポリシーには次の項目が含まれる。

- ・ 組織の基準規則(通常、特定の技術、パラメータ、手順について一定の用途の指定)
- ・ プライバシー(電子メール、Web 使用の監視の有無)
- ・ 利用規定(組織のコンピュータやネットワークリソースについての利用)
- ・ 役割と責任(ユーザー、管理者、経営者)
- ・ 説明責任(監査、インシデントハンドリング)
- ・ 認証(パスワード、バイオメトリクスなど)
- ・ リソースの可用性(冗長性、回復、バックアップ)
- ・ コンプライアンス(違反、影響、罰則)

設定ミス

安全なまたは推奨された方法でシステムが設定されていない場合に発生する。設定ミスを改善したり最小限に抑えるには様々な対策が考えられる。

- ・ 基幹システムや基幹ネットワークに対する設定管理プロセス(設定管理委員会)を導入する。設定管理プロセスを通して、システムやネットワークに対する変更を管理し、組織のポリシーへの準拠を徹底する。ただし、設定管理プロセスによって、アップデートやセキュリティパッチ¹⁹を適用するタイミングが遅れないようにする必要がある。
- ・ 入手可能な設定項目チェックリストを作成、または使用する。チェックリストを用意して設定項目をリストアップすることにより、より安全なシステムやネットワークを提供できるようにする。チェックリストは、連邦機関、ベンダー、個人など様々なソースから入手できる。

¹⁹ NIST Special Publication 800-40 の『Procedures for Handling Security Patches』を参照のこと (<http://csrc.nist.gov/publications/nistpubs/800-40/sp800-40.pdf>)。

ソフトウェアの信頼性

多くの攻撃では、コンピュータやネットワークに使用されるソフトウェアコードのエラー（バグ）が悪用される。ソフトウェアエラーによる問題を最小限に抑えるには様々な方法がある。社内でコードを開発する場合、コードの開発やテストは正しい手順に従って行い、適切な品質管理を確実に実施する必要がある。外部のベンダーから購入したコードの品質に関しては、十分な管理を行うことは多少難しくなる。したがって、こうしたリスクを軽減するには、ベンダーが提供するアップデートやパッチを定期的にチェックし、タイミングよく適用する必要がある。市販のソフトウェアを購入する場合は、脆弱性データベース (<http://icat.nist.gov> など) をチェックし、ベンダーのソフトウェアの過去の実績を調査すべきである（ただし、過去の実績が将来の実績を必ずしも適切に反映するとは限らない）。

パッチ適用の不履行

ソフトウェアは非常に複雑になってきているため、ソフトウェアエラー（バグ）はもはや避けることはできない。エラーが発見されると、通常ソフトウェアベンダーはパッチを発行してエラーの修正や緩和を図り、悪意あるエンティティが悪用できないようにしている。残念ながら多くの管理者は、時間、リソースまたは知識が十分でないために、タイミングよくパッチを適用できないことがある。適切なパッチを適用しシステムを最新の状態に保っていれば、ほとんどのネットワーク侵入は回避できたとする推測からしても、パッチがタイミングよく適用されていないことがわかる。

また、テスト結果によっても、組織のネットワークやセキュリティアーキテクチャに大幅な変更が必要とされる場合がある。たとえば、侵入テストの結果、ファイアウォールの数を増やすなどして防御の壁を厚くする必要が提起されることがある。あるいは、多数のシステムにパッチを適用し最新の状態に保つには、集中管理が必要になり、更には、コンピュータシステムの調達や設定スキームの中央集中化が必要になるといったことも考えられる。したがって、組織の運用環境の全体像を把握し、運用環境をどのように変更すればテストを容易に実施でき脆弱性の危険性を軽減できるかといったことを検討するうえでも、テスト結果を多いに役立てることができる。

3.12 情報セキュリティの一般原則

セキュリティ上の問題に対応する場合、次のような情報セキュリティの一般原則に留意すべきである。^{20, 21}

- ・ **単純性**

セキュリティメカニズム(情報システム全般)はできる限り単純にすべきである。多くのセキュリティ問題は、複雑性が原因になっている。

- ・ **フェイルセーフ**

システム障害時にも、安全性を確保できるようにする。障害が発生した場合にも、セキュリティが行使されるようにする必要がある。セキュリティを失うより、機能を失ったほうがよいと考えるべきである。

- ・ **完全な仲介**

直接情報にアクセスするのではなく、アクセスポリシーを実施する仲介を使用すべきである。一般的な例として、ファイルシステム許可、Web プロキシ、メールゲートウェイなどがある。

- ・ **オープンな設計**

システムセキュリティはシステムの導入やコンポーネントを秘密にせずに実現すべきである。隠すことによるセキュリティは機能しない。

- ・ **権限の分離**

機能はできる限り分離し、精度を追求すべきである。この考え方は、システムとオペレータやユーザーにも当てはまることである。システムオペレータやユーザーの場合、役割をできる限り分散するようにすべきである。たとえば、リソースが許す限り、システム管理者とセキュリティ管理者の役割は区別する必要がある。

²⁰ 2001年11月発行、Curtin, Matt 共著の『Developing Trust: Online Privacy and Security』、Saltzer, Schroeder 共著の『The Protection of Information in Computer Systems, Volume 63』P1278～1308を参照のこと。

²¹ 概説については、NIST Special Publication 800-14『Generally Accepted Principles and Practices for Securing Information Technology Systems』(<http://csrc.nist.gov/publications/nistpubs/800-14/800-14.pdf>)、あるいはNIST Special Publication 800-27『Engineering Principles for Information Technology Security (A Baseline for Achieving Security)』(<http://csrc.nist.gov/publications/nistpubs/800-27/sp800-27.pdf>)を参照のこと。

- ・ **意識改革**

ユーザーはセキュリティの必要性について理解することが求められる。こうした考え方は、トレーニングや教育により養うことができる。また、日常的業務で使い勝手のよさを感じられるようなセキュリティメカニズムを導入する必要がある。ユーザーがセキュリティメカニズムを厄介であると感じると、なるべく使わずに済むような方法を考えようとする。たとえば、ランダムパスワードが非常に強力であっても覚えにくい場合には、パスワードをメモしたり、ポリシーに従わないでも済む方法を見つけようとすることがある。

- ・ **多層防御**

セキュリティメカニズムを単独で導入しても十分でないことを理解すべきである。個々のセキュリティメカニズムが侵害されてもホストやネットワークの侵害にはつながらないように、セキュリティメカニズム(防御)の層を厚くする必要がある。情報システムのセキュリティに特効薬はない。

- ・ **侵害の記録**

システムやネットワークが危険にさらされた場合には、その記録やログを作成しておく必要がある。こうした情報はネットワークやホストのその後のセキュリティ確保に役立てることができ、攻撃者が使用した攻撃やエクスプロイトの方法を特定できる。また、将来のホストやネットワークの安全性の改善に役立てることもできる。さらに、攻撃者を特定し、法的手段に訴える場合にも有効な情報となりうる。

NISTには数多くのドキュメントがあるため、システムを再構成する際に参考になる。特に、次のドキュメントは大いに活用できると思われる。

- ・ **SP 800-48, Wireless Network Security: 802.11, Bluetooth, and Handheld Devices,**

http://csrc.nist.gov/publications/nistpubs/800-48/NIST_SP_800-48.pdf

- ・ **SP 800-46, Security for Telecommuting and Broadband Communications,**

http://csrc.nist.gov/publications/nistpubs/800-48/NIST_SP_800-46.pdf

- ・ **SP 800-45, Guidelines on Electronic Mail Security,**

<http://csrc.nist.gov/publications/nistpubs/800-45/sp800-45.pdf>

- ・ **SP 800-44, Guidelines on Securing Public Web Servers,**

<http://csrc.nist.gov/publications/nistpubs/800-44/sp800-44.pdf>

- ・ **SP 800-43, Systems Administration Guidance for Windows 2000 Professional,**

http://csrc.nist.gov/itsec/guidance_W2Kpro.html

・ **SP 800-41, Guidelines on Firewalls and Firewall Policy,**
<http://csrc.nist.gov/publications/nistpubs/800-41/sp800-41.pdf>

・ **SP 800-40, Procedures for Handling Security Patches,**
<http://csrc.nist.gov/publications/nistpubs/800-40/sp800-40.pdf>.

3.13 ネットワークテスト技法の比較

表 3.1 と表 3.2 は、前述のテスト技法の比較したものである。

テストの種類	長所	短所
ネットワークスキャン	<ul style="list-style-type: none"> ・ 高速(脆弱性スキャナ、侵入テストと比較した場合) ・ ネットワーク上のホストの数によるが、ホストのスキャンに効果的 ・ 多数の優れたフリーウェアツールが入手可能 ・ 高度自動化(スキャンコンポーネント) ・ 低コスト 	<ul style="list-style-type: none"> ・ 既知の脆弱性が直接検知されない(トロイの木馬に使用されるポート 31337、ポート 12345 などは検知可能) ・ 最終テストとしてではなく、侵入テストの準備テストとして使用 ・ 欠陥の分析に高度な専門知識が必要
脆弱性スキャン	<ul style="list-style-type: none"> ・ スキャン対象のホストの数によってはかなりの高速処理が可能 ・ フリーウェアツールが入手可能 ・ 高度自動化(スキャン) ・ 既知の脆弱性の検出 ・ 検出された脆弱性の軽減策についての提案提供 ・ 高コスト(市販のスキャナ)から低コスト(フリーウェアスキャナ) ・ 定期的な実行が容易 	<ul style="list-style-type: none"> ・ フォールスポジティブの発生率が高い。 ・ 特定のホストを対象に大量のトラフィックが作成される(ホストのクラッシュや一時的にサービス拒否が発生)。 ・ ステルスでの実行が不可能(IDS、ファイアウォール、エンドユーザーにまで容易に見つかる。ただし、スタッフの対応や警告メカニズムのテストには効果がある) ・ 初心者が使用した場合には危険が伴う(特に、DoS 攻撃の場合)。 ・ 最新の脆弱性を見逃すことが多い ・ 表面的脆弱性のみを検出

侵入テスト	<ul style="list-style-type: none"> ・ 攻撃者が使用する攻撃方法とツールによってネットワークのテストを実施 ・ 脆弱性の検証が可能 表面低脆弱性以外の脆弱性も検知し、アクセスを拡大する脆弱性の悪用方法を繰り返し実証可能 ・ 脆弱性が単なる理論上の概念でないことを実証 ・ セキュリティ問題の対応に必要な現実的方法と証拠を提供 ・ ソーシャルエンジニアリングによるネットワークセキュリティの手順と人的要素のテストが可能 	<ul style="list-style-type: none"> ・ 高度な専門知識が必要 ・ テスト実行者が多数必要 ・ 処理が遅く、ホストのクラックには数時間か数日かかる場合がある ・ 処理時間が長いため、中規模または大規模ネットワーク上のすべてのホストを個々にテストするのは不可能 ・ 経験者が浅いテスト実行者が実行すると危険が伴う ・ 特定のツールや技法はエージェントの規定により使用が禁止または制限される(ネットワークスニッファ、パスワードクラッカーなど) ・ 高コスト ・ 組織に支障をきたす可能性がある
パスワードクラッキング	<ul style="list-style-type: none"> ・ 脆弱なパスワードの検出が速い ・ パスワードの長所と短所の明確化 ・ 導入が簡単 ・ 低コスト 	<ul style="list-style-type: none"> ・ 悪用の危険性がある ・ 組織によっては使用が制限される
ログレビュー	<ul style="list-style-type: none"> ・ 貴重な情報の提供 ・ 履歴情報を提供する唯一のデータソース 	<ul style="list-style-type: none"> ・ 手作業による確認のため手間がかかる ・ 自動ツールが不完全なため、重要情報までフィルタリングされる
ファイル完全性チェッカー	<ul style="list-style-type: none"> ・ ホストのセキュリティ侵害を判断する上で信頼性が高い ・ 高度自動化 ・ 低コスト 	<ul style="list-style-type: none"> ・ インストール前のセキュリティ侵害の検出は不可能 ・ システム更新時にチェックサムの更新が必要 ・ 攻撃者により変更された場合の防衛策がないため、チェックサムの保護が必要(読取専用 CD-Rom など)
ウイルス検知	<ul style="list-style-type: none"> ・ ウイルスの防止と削除に優れている ・ 低/中コスト 	<ul style="list-style-type: none"> ・ 絶えず更新が必要 ・ フォールスポジティブの問題 ・ 新規脆弱性への対応とウイルスの高速再現に限界がある

ウォードライアリング	・ 未承認のモデムの検出に効果的	・ 特に公衆交換回線網を使用した場合の法律上の問題 ・ 処理速度が遅い
ウォードライピング	・ 未承認の無線アクセスポートの検出に効果的	・ ほかの組織のシグナルを解釈した場合の法的問題の発生 ・ コンピュータ、無線ネットワーク、ラジオ工学の高度な専門知識が必要

表 3.1: テスト手順の比較

表 3.2 は、テストカテゴリの評価要因の一覧を表示したものである。カテゴリ 1 のシステムは、組織に対してセキュリティを提供し、そのほか重要機能も提供する基幹システムである。これらのシステムには次のようなシステムが含まれる。

- ・ ファイアウォール、ルーター、侵入検知システムなどの境界防衛システム
- ・ Web サーバーや電子メールサーバーなどのパブリックアクセスシステム
- ・ 侵入者が攻撃対象とする可能性がある DNS、ディレクトリサーバー、その他の内部システム

カテゴリ 2 のシステムはその他のすべてのシステムで、ファイアウォールなどによって防衛されるシステムが含まれるが、定期的なテストが必要である。

テストの種類	カテゴリ 1 の実頻度	カテゴリ 2 の実頻度	利点
ネットワークスキャン	継続的または3か月に1回	半年に1回	<ul style="list-style-type: none"> ・ ネットワークストラクチャを列挙し、一連のアクティブなホストと対応するソフトウェアを特定する ・ ネットワークに接続された未承認のホストを検出する ・ 開いているポートを検出する ・ 未承認のサービスを検出する
脆弱性スキャン	3 か月または 2 か月に 1 回(リ	半年に 1 回	<ul style="list-style-type: none"> ・ ネットワークストラクチャを列挙し、一連のアクティブなホストと対応するソフトウェアを特定する

	スクの高いシステムの場合は頻度を増やす)、あるいは脆弱性データベースの更新時。		<ul style="list-style-type: none"> 対象となるコンピュータセットを特定し、脆弱性分析を中心に実行する ターゲットセットの潜在的脆弱性を検出する オペレーティングシステムと主要なアプリケーションのセキュリティパッチとソフトウェアバージョンが最新であることを確認する
侵入テスト	1年に1回	1年に1回	<ul style="list-style-type: none"> 組織のネットワークがどの程度侵入に弱いかわたどの程度の被害が発生する可能性があるかを確認する セキュリティインシデントへの対応、組織のセキュリティポリシーとシステムセキュリティ要件の知識と実施について IT スタッフをテストする
パスワードクラッキング	継続的または有効期限ポリシーと同じ頻度	有効期限ポリシーと同じ頻度	<ul style="list-style-type: none"> クラックがより困難なパスワードの作成にポリシーが有効かどうかを検証する 組織のセキュリティポリシーに準拠したパスワードをユーザーが選択することを検証する
ログレビュー	毎日(ファイアウォールなどの基幹システムの場合)	1週間に1回	<ul style="list-style-type: none"> システムの運用がポリシーに準拠していることを検証する
ファイル完全性チェッカー	1か月に1回(疑わしいインシデントがある場合)	1か月に1回	<ul style="list-style-type: none"> 未承認のファイル変更を検出する
ウイルス検知	1週間に1回または必要に応じて	1週間に1回または必要に応じて	<ul style="list-style-type: none"> ウイルスがシステムにインストールされる前に検出する
ウォーダイ	1年に1回	1年に1回	<ul style="list-style-type: none"> 未承認のモデムを検出し、保護対象のネットワ

アリング			ークへの未承認のアクセスを防止する
ウォードラ イピング	継続的ま たは1週間 に1回	半年に1 回	・未承認の無線アクセスポイントを検出し、保護 対象のネットワークへの未承認のアクセスを防 止する

表 3.2: 評価方法と実行頻度

4. セキュリティテスト導入戦略

セキュリティテストの目的は、組織全体に最大の利益をもたらすことにある。運用および保守フェーズにおいてテストの種類と頻度を決定する際には(最小規模のテストと総合テストの両方について)、セキュリティカテゴリ、テストのコスト、組織のシステム全体に対する利点などの項目に基づいてテストの優先付けが必要になる。実施するテストの種類を決定する場合、インプリメンテーションフェーズでは単一のシステムだけを考慮すればよいが、運用および保守フェーズではさらに複雑になる。テストの効果を最大限に引き出すには、優先順位付けの段階でシステムの相互接続性を考慮する必要がある。最高情報責任者(CIO)やIT 上級管理者は、優先順位付けのプロセスに参加して、組織全体の展望を反映させる必要がある。この項では、実際に応用できる優先順位付けプロセスについて説明する。

4.1 情報システムのセキュリティカテゴリの決定

FIPS Publication 199 の²² Standards for Security Categorization of Federal Information and Information Systems (Pre-publication final)₂ (2003年12月発行)には、組織における情報システムのセキュリティカテゴリを決定するためのガイドラインが説明されている。FIPS Publication 199 のセキュリティカテゴリの定義は、任務の遂行、資産の保護、法的責任の履行、日常業務の維持、さらに個人の保護する上で必要な情報システムが脅かされた場合に予想される組織への潜在的影響に基づいて設定される。情報システムの運用によって生じる組織へのリスクを評価するには、脆弱性や脅威に関する情報と共にセキュリティカテゴリを活用すべきである。²² FIPS Publication 199 では、セキュリティ違反(機密性、完全性または可用性の損失)が万一発生した場合に組織や個人に及ぼす潜在的な影響を3つのレベルに分類して定義している。

²² NIST Special Publication 800-30 の²² Risk Management Guide₂ では、リスク評価を実施する際のガイダンスについて説明されている。

<http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf> を参照のこと。

次のような場合は、潜在的影響は小さいとされる。

機密性、完全性または可用性の損失により、組織の運営やその資産あるいは個人²³ に対して限定的な悪影響を及ぼす可能性がある場合。限定的な悪影響とは、例えば機密性、完全性または可用性の損失により、

- (i) 組織の任務遂行能力低下の程度やその期間が組織の主たる任務を遂行できる範囲内にとどまってはいるが、業務の効率が明らかに低下した場合
 - (ii) 組織の資産に軽微な損害を与えた場合
 - (iii) 軽微な財務上の損失を与えた場合
 - (iv) 個人に対して軽微な被害を与えた場合
- などを指す。

次のような場合には、潜在的影響は中程度とされる。

機密性、完全性または可用性の損失により、組織の運営やその資産あるいは個人に対して深刻な悪影響を及ぼす可能性がある場合。深刻な悪影響とは、例えば機密性、完全性または可用性の損失により、

- (i) 組織の任務遂行能力低下の程度やその期間が組織の主たる任務を遂行できる範囲内にとどまってはいるが、業務の効率が著しく低下した場合
 - (ii) 組織の資産に重大な損害を与えた場合
 - (iii) 重大な財務上の損失を与えた場合
 - (iv) 生命の損失あるいは生命を脅かすような重大な傷害に発展しないまでも、個人に対して重大な被害を与えた場合
- などを指す。

²³ 個人に対する悪影響には、個人が法的手段に訴えることができるプライバシーの侵害なども含まれる。

また、次のような場合には、潜在的な影響が大きいと判断される。

機密性、完全性または可用性の損失により、組織の運営やその資産あるいは個人に対して致命的または壊滅的な悪影響を及ぼす可能性がある場合。致命的または壊滅的な影響とは、例えば機密性、完全性または可用性の損失により、

(i) 組織の任務遂行能力低下の程度やその期間が組織の主たる任務を遂行できない程度に及んだ場合

(ii) 組織の資産に広範囲にわたる損害を与えた場合

(iii) 広範囲にわたる財務上の損失を与えた場合

(iv) 生命の損失あるいは生命を脅かす重度の傷害につながる致命的または壊滅的な被害を個人に対して与えた場合

などを指す。

4.2 システム別のテストコスト計算

システムのセキュリティカテゴリを決定する際には、それぞれのテストの実施にかかるコストを計算する必要がある。コストは、次のような様々な要因によって決定される。

- ・ LAN、WAN、単一データベース、主要アプリケーションなど、テストの対象となるシステムの規模
- ・ テスト対象システムの複雑性
異種混合オペレーティングシステム環境を持つ大規模組織のネットワークをテストする場合はよりコストがかかる。
- ・ テストに必要な人的介入のレベル
- ・ テストの実施において選択するテストサンプルの有効性とサンプルのサイズ
ネットワークホストのサンプルにネットワークスキャンを実施するのは無意味と思われるが、侵入テストを実施する場合にはホストをサンプルとして選択するのは妥当である。

また、システム別に、実施するそれぞれのテストの種類についてコストを数値化する必要がある。

4.3 システム別の実施する各テストの利点の明確化

組織におけるテストの重要度以上にテストのコストがかからないようにするには、テストの実施によって得られる利点を明確にし、できる限り具体的な数値で表す必要がある。テスト全般の利点は、攻撃者に悪用される前に脆弱性を検出することにあるが、そのほかにも利点がある。テストの利点を明確化する際に考慮すべき要因として、次のような点が挙げられる。

- ・ テストの実施前にはわからなかったシステムやネットワークについての知識を取得する価値：知識が深まることにより組織の資産管理が容易になる。
- ・ テストで検知された欠陥を是正することによる、侵入や事業妨害の発生可能性の大幅な低下
悪用される可能性のある脆弱性の件数を減らすことができる。

4.4 テスト実施のためのシステムの優先順位付け

ステップ1～3の結果を評価して、セキュリティテストを実施するためにシステムの優先順位を付ける必要がある。この分析では、セキュリティカテゴリ、テストのコスト、テストの利点に基づいてシステムをランク付けし、リストにする。また、リストでは、対象となるシステムごとに個々のテストの種類に応じて必要なリソース(コスト)を明確にする。最低限必要なリソースを決定するには、まず最初に最も影響が大きいシステムについて最低限必要となるテストを特定する。また、セキュリティテストに利用可能なリソースを明確にし、必要なリソースと比較する。必要なリソースと利用可能なリソースとの間にギャップがあるため、優先順位リストで影響度が高いと判断された最も影響が大きいシステムについて最低限必要なテストを実施できない場合は、リソースを追加して必要なセキュリティテストを実施できるようにする必要がある。テストのコストを計算することにより、追加のリソースが必要な理由を数値的に裏づけできる。大部分の基幹システムについて資金が明確にされたら、優先順位の低いシステムについても頻度を少なくして降順にテストを実施することも可能である。最終的には、影響度の高いシステムからなる優先リストを作成し、適切なテスト技法と頻度に基づいてテストが行われるようにする。

付録 A. 専門用語

- CGI 共通ゲートウェイインターフェース(Common Gateway Interface)
- CIO 最高情報責任者(Chief Information Officer)
- CRC 巡回冗長検査(Cyclic Redundancy Check)
- DNS ドメインネームシステム(Domain Name System)
- DoS サービス拒否(Denial of Service)
- GUI グラフィカルユーザーインターフェース(Graphical User Interface)
- HTTP ハイパーテキスト転送プロトコル(Hypertext Transfer Protocol)
- ICMP インターネット制御メッセージプロトコル(Internet Control Message Protocol)
- IDS 侵入検知システム(Intrusion Detection System)
- IIS Microsoft インターネット情報(Web)サーバー(Microsoft's Internet Information(Web) Server)
- InterNIC インターネットネットワーク情報センタ(Internet Network Information Center)
- ISSM 情報システムセキュリティマネージャ(Information Systems Security Manager)
- ISSM 情報システムセキュリティ責任者(Information Systems Security Officer)
- IT 情報技術(Information Technology)
- LAN ローカルエリアネットワーク(Local Area Network)
- LDAP ライトウェイトディレクトリアクセスプロトコル(Lightweight Directory Access Protocol)
- NETBIOS ネットワーク基本入力出力システム(Network Basic Input/Output System)
- NIS ネットワーク情報システム(Network Information System)
- OS オペレーティングシステム(Operating System)
- PBX 構内交換機(Private Branch Exchange)
- POP 郵便局プロトコル(Post Office Protocol)
- POTS 一般電話回線(Plain Old Telephone System)
- RFC リクエストフォーコメント(Request For Comments)
- SANS システム管理、ネットワーキング、セキュリティ(System Administration, Networking, and Security)
- SMB 簡易メッセージブロック(Simple Message Block)
- SMTP 簡易メール転送プロトコル(Simple Mail Transfer Protocol)
- SNMP 簡易ネットワーク管理プロトコル(Simple Network Management Protocol)
- ST&E (Security Testing and Evaluation)
- TCP/IP 伝送制御プロトコル/インターネットプロトコル(Transmission Control Protocol/Internet Protocol)
- UDP ユーザーデータグラムプロトコル(User Datagram Protocol)
- WAN 広域ネットワーク(Wide Area Network)
- WU-FTPD ワシントン大学提供の FTP サーバー(Washington University FTP server)

付録 B. References

- D. Brent Chapman and Elizabeth D. Zwicky, Building Internet Firewalls, 1995.
- Federal CIO Council, How To Eliminate The Ten Most Critical Internet Security Threats, November 2000.
- Hatch, Brian, et al., Hacking Exposed: Linux, 2001.
- Information Security Institute (ISI) Swiss Army Knife Reference, Resource for Security & Audit Professionals, Michael Ira Sobol (MIS) Training Institute.
- MIS Training Institute, TCP/IP Network Security and Vulnerability Testing.
- NIST, ITL Bulletin, Advising Users On Information Technology, May 1999.
- NIST, ITL Bulletin, Computer Attacks: What They Are and How to Defend Against Them, May 1999.
- NIST, FIPS Pub 199 (Pre-publication Final), Standards for Security Categorization of Federal Information and Information Systems, December, 2003.
- NIST, SP 800-14, Generally Accepted Principles and Practices for Securing Information Technology Systems, September 1996
- NIST, SP 800-41, Guideline on Firewalls and Firewall Policy, January 2002.
- NIST, SP 800-61 (Draft), Computer Security Incident Handling Guide, September, 2003.
- National Information System Security Glossary, NSTISSI No. 4009, January 1999.
- Office of Management and Budget, Circular A-130, February 1996.
- Scambay, Joel, et al., Hacking Exposed: Second Edition, 2001.
- System Administration, Networking, and Security (SANS) Institute, SANS Security Alert, May 2000.
- SANS Institute, SANS Snap: Computer and Hacker Exploits – Step by Step.
- SANS Institute, SANS Snap: Intrusion Detection – The Big Picture.
- MIS Training Institute, Staying Ahead of the Hackers: Network Vulnerability Testing.
- Stevens, W. Richard, TCP/IP Illustrated, Volume 1: The Protocols, 1994.

付録 C. 一般的なテストツール

C.1 ファイル完全性チェッカー

ツール	機能	Web サイト	Linux/ Unix	Win 32	コス ト
Aide	Unix, Linux	http://www.cs.tut.fi/~rammer/aide.html	■		無料
説明	AIDE (Advanced Intrusion Detection Environment) は Tripwire の無料代替品である。ファイルの整合性をチェックし、Unix 系および Linux 系の各種プラットフォームをサポートする。				
LANGuard	Windows 2000/NT	http://www.gfi.com/languard/		■	無料
説明	LANGuard ファイル完全性チェッカーは、Windows 2000/NT のファイルの変更、追加、削除の有無をチェックして侵入検知を行うユーティリティである。				
Tripwire	Windows, Unix, Linux、ルーター	http://www.tripwiresecurity.com/	■	■	Unix の場 合は 無料
説明	Tripwire はファイルの改ざんを監視し、ファイルの完全性を検証して、ネットワークホスト上のデータに違反があった場合は管理者に通知する。				

表 C.1: ファイル完全性チェッカーツール

C.2 ネットワークスニッファ

ツール	機能	Web サイト	Linux/ Unix	Win32	コスト
Dsniff	Unix スニッ ファ	http://www.monkey.org/~dugsong/dsniff/	■		無料
説明	Dsniff はネットワーク監査および侵入テストを行うためのツール群である。Dsniff、filesnarf、mailsnarf、msgsnarf、urlsnarf、webspy が、対象となるデータ(パスワード、電子メール、ファイルなど)のネットワークを受動的に監視する。Arpspoof、dnsspoof、macof は、攻撃者が通常利用できないネットワークトラフィック(レイヤー2 スイッチなど)の傍受を容易にする。Sshmitm と webmitm は PKI の弱い結合を利用して、リダイレクトされた SSH と HTTPS に対してアクティブな monkey-in-the-middle 攻撃を行う。				
Ethereal	GUI を備え た Unix/Windo ws スニッ ファ	http://www.ethereal.com/	■	■	無料
説明	Ethereal は Unix および Windows 用の無料のネットワークプロトコルアナライザである。Ethereal を使用して、ライブネットワークからのデータやディスク上のキャプチャファイルからのデータを解析できる。また、パケットごとにサマリや詳細情報を確認しながら、キャプチャデータを対話的にブラウズできる。Ethereal には、豊富な表示フィルタ言語、再構築された TCP セッションストリームの表示機能、802.11 パケットの解析機能などの強力な機能が用意されている。				
Sniffit	Unix スニッ ファ	http://reptile.rug.ac.be/~coder/sniffit/sniffit.html http://www.symbolic.it/Prodotti/sniffit.html (Windows 用)	■	■	無料
説明	Linux、Unix、Windows の各種バージョンに対応できる汎用スニッファとして機能するフリーウェア				
Snort	Unix スニッ ファ/IDS	http://www.snort.org	■	■	無料
説明	Linux、Unix、Windows の各種バージョンに対応でき、軽量 IDS および汎用スニッファとして機能するフリーウェア。				
TCPDump	Unix スニッ ファ	http://www-nrg.ee.lbl.gov/	■		無料
説明	Linux と Unix の各種バージョンに対応できる汎用スニッファとして機能するフリーウェア				

WinDump	Windows スニッファ	http://netgroup-serv.polito.it/windump/		■	無料
説明	TCPDump をベースとした Windows 版汎用スニッファとして機能するフリーウェア				

表 C.2: ネットワークスニッファツール

C.3 パスワードクラッカー

ツール	機能	Web サイト	Linux/ Unix	Win32	コスト
Crack 5	Unix パスワードクラッカー	http://www.crypticide.org/users/alecm/	■		無料
説明	Crack は Unix などのパスワードファイルの脆弱性を迅速に検知するためのパスワード推測プログラムで、パスワードファイルのコンテンツのスキャンや弱いログインパスワードを誤って選択したユーザーの検索が行われる。				
IMP 2.0	Novell Netware パスワードクラッカー	http://www.wastelands.gen.nz		■	無料
説明	Imp は GUI (Win95/NT) を備えた NetWare パスワードクラッキングユーティリティである。NDS ファイルやバインダリファイルからアカウント情報が直接ロードされるため、ユーザーは様々な攻撃方法を用いてアカウントパスワードの解析を試みることができる。				
John the Ripper	Windows および Unix パスワードクラッカー	http://www.openwall.com/john/	■	■	無料
説明	John the Ripper は高速パスワードクラッカーで、現在では Unix 版、DOS 版、Win32 版、BeOS 版がある。このツールの主な目的は弱い Unix パスワードを検出することにあるが、そのほかにも様々なハッシュタイプがサポートされている。				
L0pht Crack	Windows パスワードクラッカー	http://www.securityfocus.com/tools/1005		■	有料
説明	Windows NT、2000、XP 用のパスワードクラッキングユーティリティ。				

Nwpcrack	Novell Netware パスワードクラッカー	http://ftp.cerias.purdue.edu/pub/tools/novell/		■	無料
説明	Novell Netware 用のパスワードクラッキングユーティリティ				

表 C.3: パスワードクラッキングツール

C.4 スキャンおよび列挙ツール

ツール	機能	Web サイト	Linux/ Unix	Win32	コスト
DUMPSec	Windows 数値化ツール	http://www.systemtools.com		■	無料
説明	DumpSec は Microsoft Windows 用のセキュリティ監査プログラムである。ファイルシステム、レジストリ、プリンタのアクセス許可 (DACL) と監査設定 (SACL) をダンプし、簡潔で判読しやすいリストボックス形式を共有することで、システムのセキュリティホールを簡単にわかるようにする。また、DumpSec ではユーザー、グループ、レプリケーション情報もダンプされる。				
Firewalk	ファイアウォールフィルタルールマッパー	http://www.packetfactory.net/firewalk/	■		無料
説明	Firewalking は、IP パケット応答を分析するトレースルートのような技法を使って、ゲートウェイ ACL フィルタを特定し、ネットワークをマップするツールである。また、パケット転送デバイスに設定されているフィルタルールを特定する技法が用いられている。				
Fscan	ポートスキャン	http://www.foundstone.com/		■	無料
説明	FScan はコマンドラインによるポートスキャンで、TCP ポートと UDP ポートの両方がスキャンされる。				
LANguard Network Scanner	ポートスキャナ、OS 検出	http://www.gfi.com/languard/lanscan.htm		■	無料
説明	LANguard Network Scanner はネットワークセキュリティを監査するための無料のセキュリティポートスキャナである。ネットワーク全体をスキャンし、ホスト名、共有、ログオンユーザー名など個々のコンピュータの NetBIOS 情報を提供する。また OS 検出、パスワード強化テスト、レジストリ問題の検出などが行われ、レポートが HTML 形式で出力される。				
NDS Snoop	Novell 数値化ツール	http://www.novell.com/coolsolutions/		■	無料
説明	様々な NDS のオブジェクトや値が列挙される。				

Nmap	ポートスキャナ、OS 検出	http://www.insecure.org/Nmap/	■	■	無料
説明	Nmap (Network Mapper) はネットワーク調査またはセキュリティ監査用のオープンソースのユーティリティである。当初大規模ネットワークを迅速にスキャンするユーティリティとして設計されたが、単一のホストについてもスキャンできる。Nmap では未加工の IP パケットを使用することにより、ネットワークで使用されるホストの種類、提供されるサービス(ポート)、実行するオペレーティングシステムとそのバージョン、使用するパケットフィルタとファイアウォールの種類などが特定される。				
Solarwinds	ネットワーク列挙	http://www.solarwinds.net/		■	有料
説明	ネットワークツール、管理ツール、検知ツールが組み合わされている。				
SuperScan	ポートスキャナ、OS 検出、バナー列挙	http://www.foundstone.com/		■	無料
説明	GUIを備えたポートマッパーである。大規模ネットワークを迅速にスキャンし、ネットワーク上で利用されるホストの種類、提供されるサービスとそのサービスのバージョン、オペレーティングシステムの種類とバージョンを特定する。また、DNS のリバースルックアップが実行される。				

表 C.4: スキャンツールと列挙ツール

C.5 脆弱性評価ツール

ツール	機能	Web サイト	Linux/ Unix	Win32	コスト
CyberCop Scanner	脆弱性スキャナ	http://www.pgp.com/products/	■	■	有料
説明	CyberCop Scanner はネットワークベースの脆弱性スキャンツールで、ネットワークホスト上のセキュリティホールを検出する。				
ISS Internet Scanner	脆弱性スキャナ	http://www.iss.net/		■	有料
説明	ISS Internet Scanner はネットワークベースの脆弱性スキャンツールで、ネットワークホスト上のセキュリティホールを検出する。				
Nessus	脆弱性スキャナ	http://www.nessus.org/	■	■ (クライアントのみ)	無料
説明	無料のネットワークベースの脆弱性スキャンツールで、ネットワークホスト上のセキュリティホールを検出する。				
SecureScanNX	脆弱性スキャナ	http://www.vigilante.com/securescan/		■	有料
説明	SecureScan NX は企業向けのネットワークセキュリティ評価ツールで、組織のネットワークやファイアウォールをプロアクティブに調査して脆弱性を評価し是正措置を提案する。				
SAINT	脆弱性スキャナ	http://www.wwdsi.com/saint/	■		有料
説明	SAINT は SATAN のアップグレードバージョンで、コンピュータネットワークのセキュリティを評価するためのツールである。				
SARA	脆弱性スキャナ	http://www-arc.com/sara/	■		無料
説明	Sara は無料のネットワークベースの脆弱性スキャンツールで、ネットワークホスト上のセキュリティホールを識別する。				

SATAN	脆弱性スキャナ	http://www.fish.com/satan/	■		無料
説明	SATAN はシステム管理者の支援ツールである。一般的なネットワーク関連のセキュリティ問題を識別し、実際に悪用できるかどうかは確認せずに問題について報告する。				

表 C.5: 脆弱性評価ツール

C.6 ウォーダイアリングツール

ツール	機能	Web サイト	Linux/ Unix	Win32	コスト
PhoneSweep	ウォーダイアラ	http://www.sandstorm.net/		■	有料
説明	市販のウォーダイアリングプログラムで、複数のモデムをサポートし自動侵入を試みることができる。				
Telesweep	ウォーダイアラ	http://www.securelogix.com/telesweepsecure/		■	有料
説明	市販のウォーダイアリングアプリケーションで、複数のモデムをサポートし自動侵入を試みることができる。				
THC	ウォーダイアラ	http://www.thc.org/releases.php		■	無料
説明	無料の DOS ベースのウォーダイアリングプログラムです。				
ToneLoc	ウォーダイアラ	http://www.securityfocusonline.com/tools/category/26		■	無料
説明	無料の DOS ベースのウォーダイアリングプログラムです。				

表 C.6: ウォーダイアリングツール

C.7 無線ネットワークツール

ツール	機能	Web サイト	Linux/ Unix	Win32	コスト
Aerosol	無線スニッ ファ	http://www.sec33.com/sniph/aerosol.php		■	無料
説明	Aerosol は無料の無線 LAN スニッファツールで、WEP 暗号鍵の解読もできる。また、受動的に送信データを監視し、十分なパケットが収集されると暗号鍵を計算する。				
AirSnort	無線スニッ ファ	http://airsnort.shmoo.com/	■		無料
説明	AirSnort は無料の無線 LAN スニッファツールで、暗号鍵を回復させる。また、受動的に送信データを監視し、十分なパケットが収集されると暗号鍵を計算する。				
Kismet	無線スニッ ファ	http://www.kismetwireless.net/	■		無料
説明	Kismet は 802.11b 無線ネットワークスニッファである。Linux でサポートされているほとんどすべての無線カードを使用してスニッフィングを行うことができる。				
Netstumbler	無線スニッ ファ	http://www.netstumbler.com		■	無料
説明	Netstumbler は 802.11b ツールで、利用されているネットワークを盗聴し、そのアクセスポイントのデータを記録する。また、パケット PC のバージョンも提供される。				
Sniffer Wireless	無線スニッ ファ	http://www.sniffer.com/		■	有料
説明	Sniffer Wireless は市販の無線 LAN スニッファで、ネットワーク監視、キャプチャリング、復号化、フィルタリングなどの機能が用意されている。				
WEPCrack	WEP 暗号 化クラッカー	http://sourceforge.net/projects/wepcrack/	■		無料
説明	WEPCrack は、RC4 鍵スケジューリングの最近検出された弱点を利用して、802.11 WEP 暗号鍵をクラックするツールである。				
WaveStumbler	無線ネット ワークマッ パー	http://www.cqure.net/tools08.html	■		無料
説明	WaveStumbler は無料で利用できるコンソールベースの Linux 向け 802.11 ネットワークマッパーである。チャンネル、WEP、ESSID、MAC など無線ネットワークの基本特性がレポートされる。				

表 C.7: 無線ネットワークテストツール

C.8 ホストベースのファイアウォール

ツール	Web サイト	Linux/Unix	Win32	MacOS	コスト
BlackIce	http://www.networkice.com/		■		有料
McAfee Personal Firewall	http://www.mcafee.com/		■		有料
NeoWatch Personal Firewall	http://www.neoworx.com/		■		有料
Net Barrier	http://www.intego.com/netbarrier/			■	有料
Norton Personal Firewall	http://www.symantec.com/		■		有料
PC Viper	http://www.pcviper.com/		■		有料
Securepoint	http://www.securepoint.cc/		■		無料
SINUS	http://www.ifi.unizh.ch/ikm/SINUS/	■			無料
SmoothWall	http://www.smoothwall.org/	■			無料
Sygate Personal Firewall	http://www.sygate.com/		■		無料 ²⁴
T.Rex	http://www.opensourcefirewall.com/	■			無料
Tiny Firewall	http://www.tinysoftware.com/		■		無料 ²⁵
Winproxy	http://www.winproxy.com/		■		有料
ZoneAlarm	http://www.zonelabs.com/		■		有料または無料

表 C.8: ホストベースのファイアウォールツール

注: Windows XP には、ホストベースのファイアウォールが備わっているため、ポートブロックを実行できる。

²⁴ 個人向けのみ無料

²⁵ バージョン 3 以前の個人向けのみ無料、事業者向けは有料

D. 一般的なテストツールの使用例

D.1 Nmap

アクティブなホストやそれに対応するサービス(ポートを開く設定など)を確認するポートスキャナとして、Nmap が一般的に使用されている(Web サイトの付録 C を参照のこと)。Nmap は、ポートの開閉を調べるための様々なポートスキャンを実行できる。また、生の IP パケットを利用して、ネットワークで利用可能なホスト、開いているサービスやポート、ホストを実行しているオペレーティングシステムの種類とバージョン、パケットフィルタの種類、設置されているファイアウォールなどの情報を確認できる。

Nmap によってサポートされる最も基本的なポートスキャンとして、`-sT` オプションフラグ(Nmap では大文字小文字を区別)を用いた TCP 接続スキャンがある。ホストオペレーティングシステムにより提供される `connect()` システムコールを利用して、リモートホスト上の任意のポートまたはすべてのポート(ユーザー選択による)への接続を開こうとする。ポートがリスニング状態にあるか開いていれば、`connect()` に成功するが、`connect()` に失敗した場合はポートがリスニング状態でなく閉じていることになる。この種のスキャンを実行するには、特別な権限は必要とされない。

TCP `connect()` スキャンのように容易に検出されないスキャン手法として、TCP SYN スキャンがある。この手法は、Nmap が TCP 接続を完了しないことから、SYN ステルススキャンあるいはハーフオープンスキャンとも呼ばれる(図 D.1 参照)。このスキャンは `-sS` フラグを使用して実行される。Unix または Linux のホストで Nmap を実行する場合、この種のスキャンに必要なカスタム SYN パケットを作成するにはルート権限が必要になる。

SYN ステルススキャンでは、最初に SYN パケットが送信され、あたかも Nmap を実行しているコンピュータが純粋に TCP 接続を開始するかのようにふるまう。次に、Nmap を実行しているホストが応答を待つ。SYN|ACK 応答があれば、ポートがリッスン状態であるかまたは開いていることがわかる。一方、RST 応答が返された場合は、ポートが非リッスン状態であるかまたは閉じていることがわかる。SYN|ACK 応答が返されると、RST が即座に送信され接続がキャンセルされる。この最後のアクションは、SYN フラッド攻撃といった DoS 攻撃を引き起こさせないようにするために必要である。応答待ち状態の接続はすべてバッファーストアされるため、こうした操作が可能になる。ところが RST が送信されないと、ターゲットホストのバッファ一杯になってしまう場合がある。このような状態になると、RST が返されるかあるいは応答待ち状態のリクエストがタイムアウトになるまで完全なリクエストが処理されなくなり、その結果 Dos 攻撃のターゲットにされてしまう。

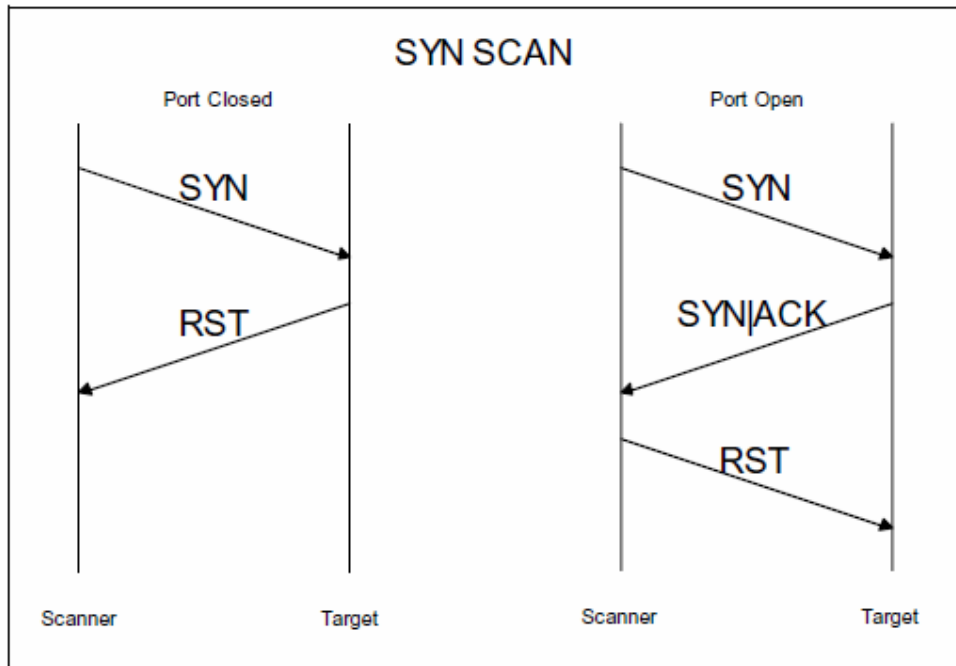


図 D.1: SYN ステルススキャン

SYN スキャンではまだ発見される可能性があり不十分であるとされる場合は、ステルス FIN、クリスマスツリー、Null などのスキャンモードを使用できる。これらのスキャンには、それぞれ -sF フラグ、-sX フラグ、-sN フラグが使用されている。FIN スキャンでは空の FIN スキャンがプローブとして使用される(図 D.2 参照)。また、クリスマスツリースキャンでは FIN、URG、PUSH の各フラグがオンになるが(図 D.3 参照)、Null スキャンではこれらのフラグがすべてオフにされる(図 D.4 参照)。これらのどのスキャン手法においても、RST の応答があった場合はポートが非リッスン状態または閉じていることを表し、RST の応答がない場合はポートがリッスン状態または開いていることを表す。この応答は RFC 793(リクエストフォーコメント)に基づいているが、Microsoft ではこうした機能は導入していないため、ステルス FIN、クリスマスツリー、Null などのスキャンを Windows ホストで実行しても適切に機能しない。これらのスキャンに必要なカスタムパケットを作成するには、ルート権限が必要になる。

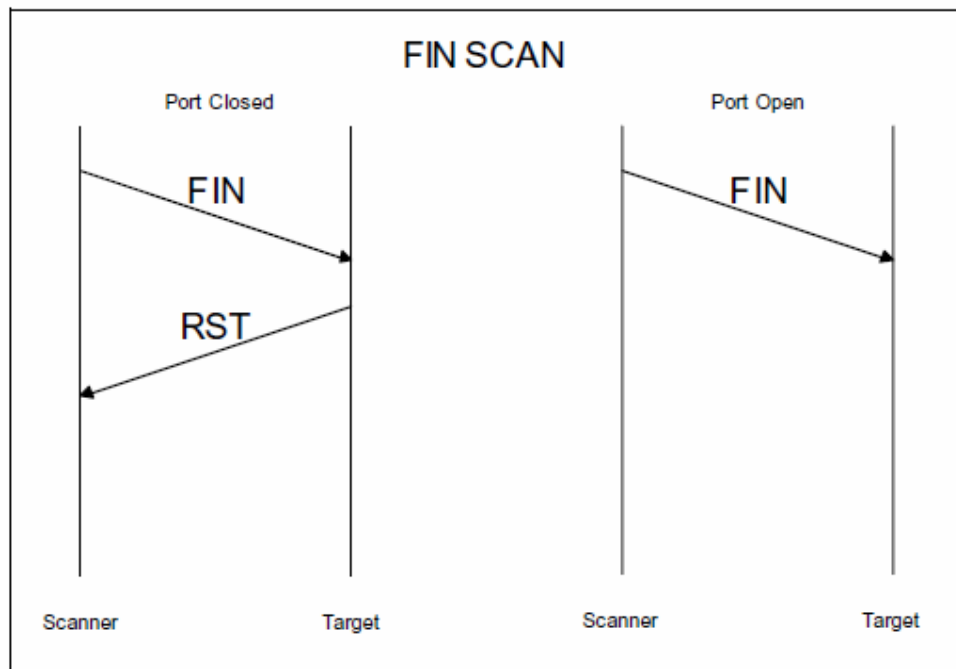


図 D.2: FIN スキャン

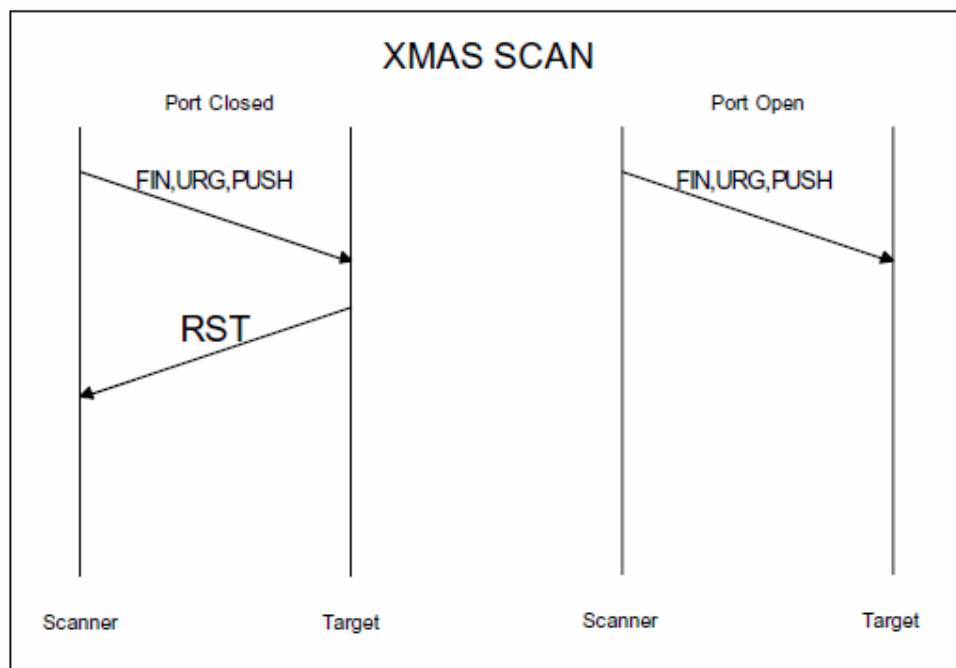


図 D.3: XMAS スキャン

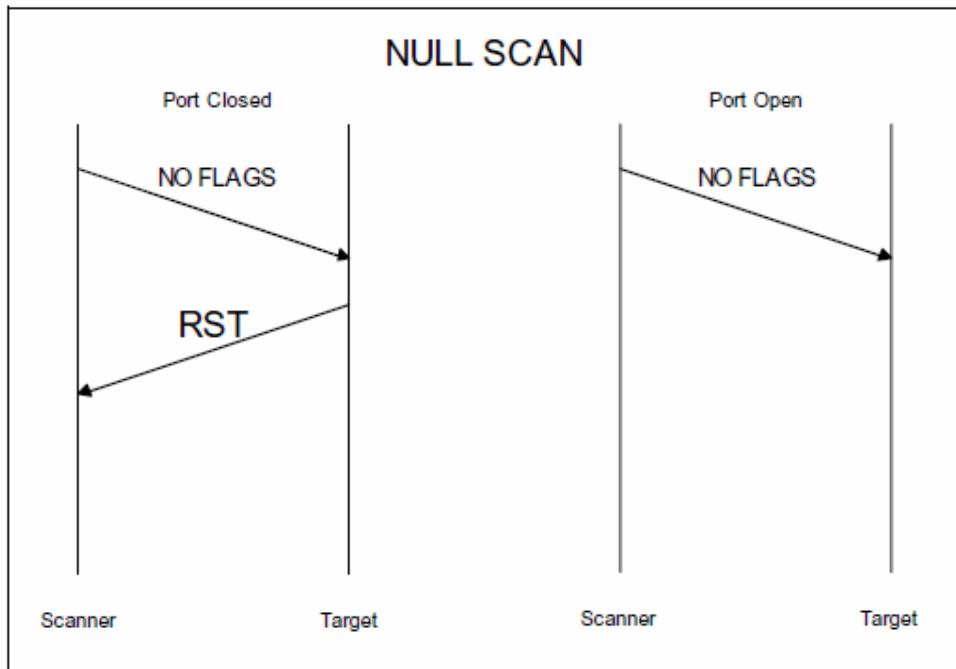


図 D.4: Null スキャン

通常、スキャンを実行する前に、ping によってネットワーク上のホストの起動状態が確認される。ICMP エコーのリクエストや応答を許可しないネットワーク環境では、-P0 フラグを使用することにより、スキャンの実行前にホストを確認しないようにできる。この方法は、通常、ファイアウォールを介してスキャンする場合に効果的である。

また、Nmap では、そのほかにも TCP ping、接続リクエスト ping、ICMP ping (ICMP エコーリクエスト) といった ping リクエストを使用することもできる。TCP ping は、-PT フラグを立てて、TCP ACK パケットを標的ネットワーク全体に送信して、応答が返されるのを待つ。これに対し、ネットワーク上で起動しているホストは RST 応答を返すことになる。接続リクエスト ping は、-PS フラグを立てて、接続リクエストまたは SYN パケットを標的ネットワークに送信する。これに対し、ネットワーク上で起動しているホストは RST 応答を返すことになる。ICMP ping の場合は、-PI フラグを立てて、ICMP エコーリクエストパケットをネットワークに送り、ICMP エコー応答を待って、ネットワーク上で起動状態にあるホストを確認する。デフォルトの ping タイプでは、-PB フラグを立てて、TCP ping と ICMP ping を同時に使用する。このようにすることで、どちらか 1 つの ping をフィルタリングするファイアウォールの裏で運用されているホストを発見できる。

さらに Nmap には、スキャンされたホストが実行しているオペレーティングシステムとそのバージョンをリモートでフィンガープリンティングするという機能がある。この場合、Nmap はホストの

TCP/IP スタックに問い合わせを行うが、オペレーティングシステムやそのバージョンごとに異なる応答が返されることがわかっているため、応答によってオペレーティングシステムが特定される。この機能は、-O フラグを立てれば実行できる。

Nmap を実行するためのコマンドラインフォーマットは次のようになる。

```
nmap [Scan Type(s)] [Options] <host or net #1 ... [#N]>
```

図 D.5 は、クラス C ネットワークの SYN スキャンの設定例を示している。

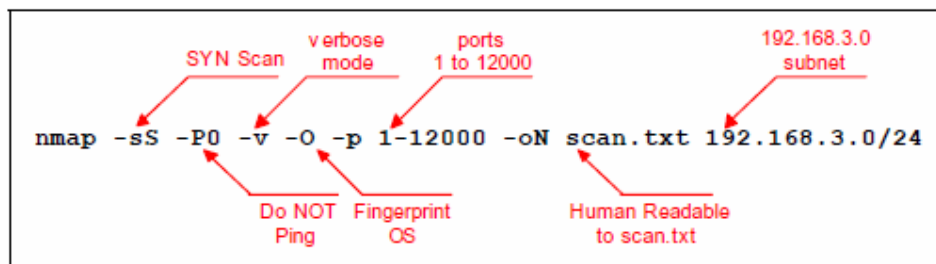


図 D.5: Nmap 設定例

このスキャンでは、クラス C サブネット 192.168.3.0 上のホストを最初に特定せずに、ステルススキャンを実行できる。さらに、ポート 1 ~ 12,000 までをチェックして開いているサービスを確認できる。ポートのマッピング後に、Nmap がオペレーティングシステムとそのバージョンについてフィンガープリンティングを試行する。このスキャンからの出力はすべて verbose モードで行われ(スキャンされたホストについて詳細情報を提供)、人間が理解できる形式(バイナリー形式ではなくテキスト形式)で保存され、file scan.txt に出力される。図 D.6 は、スキャン対象ホストについての Nmap のアウトプットを示している(この例のように、テキストファイルには確認されたアクティブなホストごとにその結果が表示される)。

```
Host hp5.doahq.gov (192.168.3.10) appears to be up ... good.
Initiating SYN half-open stealth scan against hp5.doahq.gov
(192.168.3.10)
Interesting ports on hp5.doahq.gov (192.168.3.10):
(The 1516 ports scanned but not shown below are in state: closed)
Port      State  Service
21/tcp    open   ftp
23/tcp    open   telnet
80/tcp    open   http
280/tcp   open   http-ngat
515/tcp   open   printer
631/tcp   open   unknown
9100/tcp  open   unknown
TCP Sequence Prediction: Class=trivial time dependency
                        Difficulty=1 (Trivial joke)
Sequence numbers: 6E1BC7 6E1BC8 6E1BC8 6E1BC9 6E1BCA 6E1BCB
Remote OS guesses: HP Print Server, HP LaserJet Printer
```

図 D.6: Nmap の出力例

D.1.1 Nmap コマンドサマリ

使用法:

nmap [スキヤンの種類] [オプション] <ホストまたはネットリスト>

スキヤンの種類:

-sT TCP connect()スキヤン: TCP スキヤンの中では最もよく使用されている。完全なTCPハンドシェークを実行するため、最も検出されやすい。

-sS TCP SYN スキヤン: 完全な TCP 接続が確立されないため、ハーフオープンスキヤンと呼ばれる。このスキヤンでは TCP ハンドシェークが完全に実行されず、検出されにくいいため、ステルススキヤンとも呼ばれる。このスキヤンを実行するには、ルート権限が必要になる。

-sP Ping スキヤン: ICMP ping を使用してアクティブなホストを検出する。ポートスキヤンは実行されない。

-sF ステルス FIN スキヤン: 空の FIN パケットがプローブとして使用される。

-sU UDP スキヤン: ホスト上で UDP ポートが開いていることを確認する場合に、このスキヤンが使用される。ホストによっては ICMP のエラーメッセージレートが制限されるため、UDP スキヤンに時間がかかる場合がある。

オプション:

-P0 このオプションを使用可能にすると、スキャンの実行前に Nmap によってホストのネットワーク接続が確認されない。ICMP エコーリクエストを遮断するファイアウォールを介してスキャンが実行される場合に、このオプションは効果的である。

-PT TCP ping を使用してアクティブなホストを特定できる。プローブパケットの宛先ポートを設定するには、**-PT <ポート番号>**とする。このオプションはアクティブホストを特定するという点では**-sP**と類似しているが、ICMP に依存しないという点で異なる。このため、ICMP エコーリクエストを遮断するファイアウォールを介してスキャンが行われる場合には効果的である。

-F このオプションでは高速なスキャンモードを実行できる。高速スキャンを実行可能にすると、Nmap に付属するサーバーファイルにリストされているポートのみがスキャンされる。

-O このオプションでは、TCP/IP フィンガープリンティングによってリモートホストの識別が実行される。

-h nmap このオプションでは、使用オプションのクイックレファレンス画面が表示される。

-n/-R このオプションは、DNS 解決を常に実行しないか(**-n**)、あるいは常に実行するか(**-R**)を nmap に指示する。

-v このオプションを実行すると、verbose モードが起動する。verbose モードでは詳細情報が表示されるが、さらに詳細な情報が必要な場合はオプションを 2 回指定する(**-v -v**)。

-oN <logfile name> 指定したファイルにスキャンの結果を通常のテキスト形式で出力する。

-oM <logfile name> 指定したファイルにスキャンの結果をマシン語で出力する。

--resume <logfile name> キャンセルされたネットワークスキャンを再開する。ログファイル名は、中断されたスキャンのログ(テキスト形式またはマシン語)を指定する必要がある。最後のスキャンが正常に実行され、結果がログファイルに記録されると、Nmap がコンピュータを起動する。

Nmap 使用例

nmap -v target.example.com

target.example.com マシンのすべての予約 TCP ポート(1 ~ 1024)をスキャンする。**-v** オプションによって、verbose モードが起動する。

nmap -sS -O 192.168.10.1/24

192.168.10.x クラス C のネットワーク上でアクティブになっているすべてのホストについてステルス SYN スキャンを実行する。このサンプルでは、スキャンされたホストで稼動しているオペレーティングシステムを特定する。このスキャンでは SYN スキャンと OS 検出オプションが実行されるため、ルート権限が必要になる。

**nmap .n .v .sS .O .oN nmap-sS-O_172.30.100.20-31_230.N .oM nmap
sS-O_172.30.100.20-31_230.M 172.30.100.20-31,230**

172.30.100.20 ~ 172.30.100.31 までと 172.30.11.230 のアドレスに対してステルス SYN スキャンが実行される。Verbose、-v、TCP/IP -O フィンガープリンティングの各モードがアクティブになる。-n オプションでは DNS 解決が試行されないように Nmap が設定される。ファイル名に nmap-sS-O_172.30.100.20-31_230.N を指定すると、人間が理解できるテキスト形式 (-oN) で出力され、ファイル名に nmap-sS-O_172.30.100.20-31_230.N を指定すると、マシン語 (-oM) で出力される。

D.2 L0phtCrack

最もよく使われるパスワードクラッカーの1つに、Windows NT と 2000 用の L0phtCrack がある。そのほかのプラットフォーム用のパスワードクラッカーについては付録 C にリストアップされている。L0phtcrack には、ハッシュを取得するために、ネットワークを移動しながらパスワードをキャプチャしたり、Windows レジストリからパスワードをコピーしたり、Windows 緊急修復ディスクからパスワードを検索したりする機能がある。

ハッシュが取得されると、まず最初に L0phtCrack は辞書攻撃を仕掛ける。L0phtCrack によって使用された辞書がユーザーによって選択されるか、同梱されている辞書が使用される場合がある(ただし、インターネットではより総合的な辞書を手に入れる)。L0phtCrack によってリスト内の個々の単語のハッシュが取得され、そのハッシュはクラック対象のハッシュと比較される。比較されたハッシュが一致した場合は、L0phtCrack によってパスワードが発見されたことになる。

After L0phtCrack による辞書攻撃が完了すると、ハイブリッド攻撃によって再度単語リストを検索する。最後に L0phtcrack は総当たり攻撃を仕掛けて、あらゆる文字の組み合わせを試しながら、残りのハッシュをクラックする。総当たり攻撃で L0phtCrack によって使われる文字列をユーザーがコントロールすることもできる。この文字列が長ければ長いほど、クラックに時間がかかる。図 D.7 は、L0pht Crack のスクリーンショットを示している。

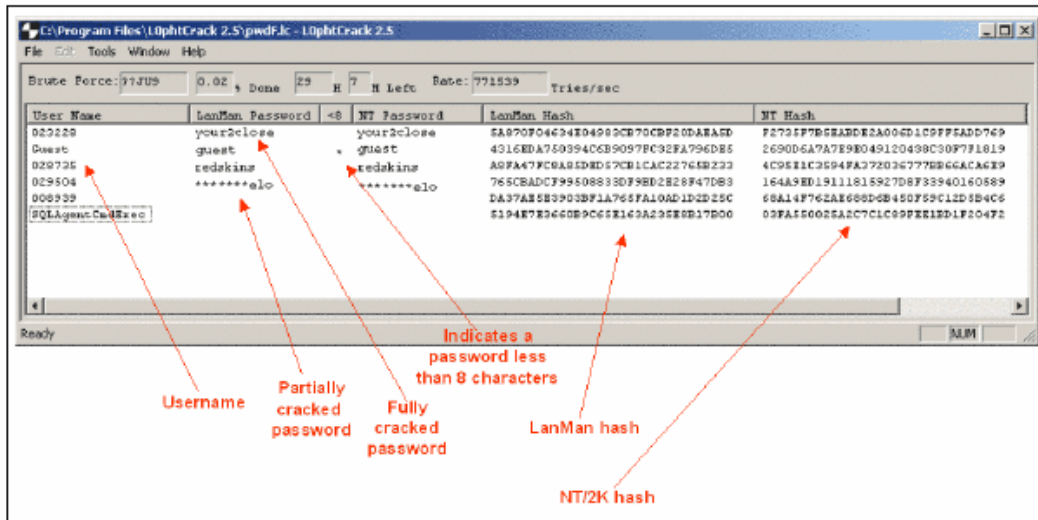


図 D.7: L0phtCrack による総当たり攻撃

D.3 LANGuard

ファイル完全性チェッカーについては、フリーウェア、シェアウェア、市販のものなど様々なチェッカーが出回っている。中でもよく使われているフリーウェアのチェッカーとして、Windows NT/2000用のLANGuard File Integrity Checkerがある。このチェッカーはコマンドラインまたはGUIのどちらからでも設定できる。このチェッカーでは、ファイルに対して警告を発する際に、電子メールによる警告が送信されるように設定することもできる。

LANGuard をスタートメニューから設定する場合は、[LANGuard File Integrity Checker configuration]を選択する(図 D.8を参照)。

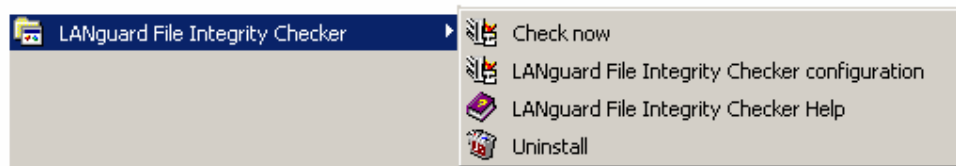


図 D.8: LANGuard File Integrity Checker Configuration の起動

これにより、LANGuard 設定ウィンドウが開く。このウィンドウから、LANGuard が監視するファイル、フォルダ(ディレクトリ)、ドライブを選択できる。また、オペレーティングシステムのルートディレクトリやサブディレクトリ(winnt)、ウイルス対策プログラムのディレクトリ、重要なブートファイルなども選択できる。新しいウイルス対策署名をダウンロードした場合などファイルを更新するたびに、チェックサムも更新することも重要である。さらに、電子メールを更新した場合は、SMTP サーバー

の IP アドレスや受信側の電子メールアドレスの設定も必要になる。詳細については、図 D.9 を参照。

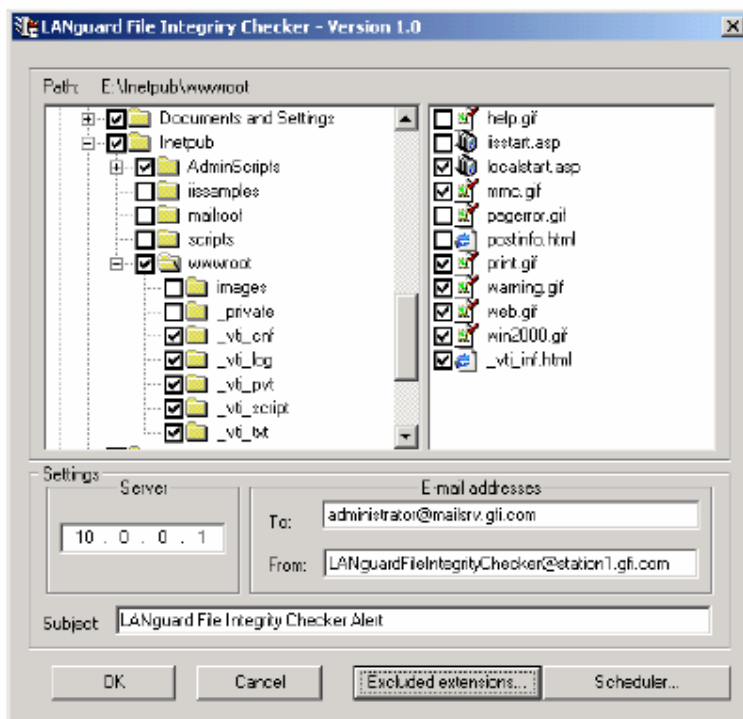


図 D.9: LANguard File Integrity Checker Configuration の設定

設定が完了すると、比較が実行されるたびに電子メールが指定したアドレスに送信され、ファイルの変更や追加が検出される。このチェックは手動で実行したり、定期的に自動実行されるようにスケジュールしたりできる。図 D.10 は、LANguard の電子メールのサンプルを示している。

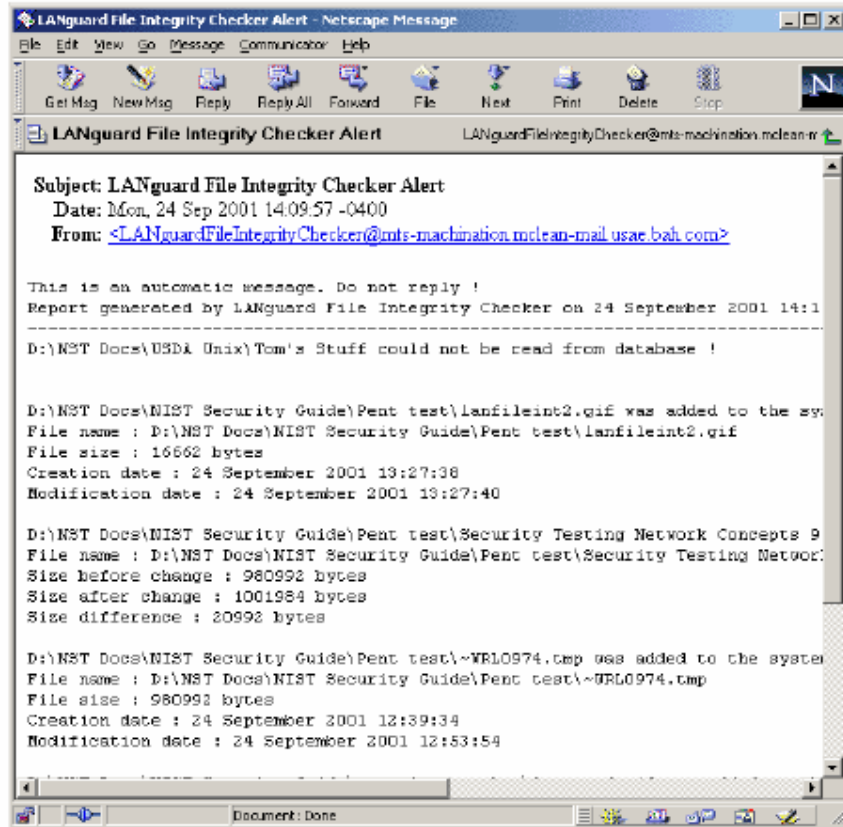


図 D.10: LANguard File Integrity Checker の通知メール

D.4 Tripwire

Tripwire は、UNIX と Windows のオペレーティングシステム用のファイルシステム整合性チェックプログラムである。Tripwire を使用する前に、検証するディレクトリとファイル、またそれぞれの属性を指定する設定ファイルを作成する必要がある。次に、Tripwire を実行して、設定ファイルで指定したファイルとディレクトリに対応する暗号チェックサムのデータベースを作成する。

Tripwire プログラム、設定ファイル、初期化されたデータベースを改ざんから保護するには、ディスクや CD-ROM など物理的に書き込み禁止を設定できる媒体にコピーした方がよい。こうした読み込み専用の媒体に保存されたデータが信頼できる参照プログラム、設定、データベースとなるため、システム上のディレクトリやファイルの整合性テストに安心して使用できる。さらに、個々のディレクトリやファイルの内容を表す暗号チェックサムに加え、Tripwire データベースにも次に示す項目の検証に使用できる情報が含まれている。

- ・ アクセス権限やファイルモード設定 (有効な実行設定など)
- ・ ファイルシステムの Inode 番号
- ・ リンクの数
- ・ 所有者のユーザーID
- ・ アクセス権が付与されたユーザーグループのグループ ID
- ・ 項目のサイズ
- ・ 項目への前回のアクセス日時、項目に対する前回の変更、項目の inode に対応する作成日時

ほとんどのシステムについて、すべての基幹オペレーティングシステムのディレクトリとファイル、さらに管理者が重要であると判断したディレクトリとファイルの完全性を検証できるように、管理者が Tripwire を設定する必要がある。また、これらのディレクトリやファイルに関連する実行可能プログラム、デーモン、スクリプト、ライブラリ、設定ファイルについても特別な注意が必要になる。デフォルトの Tripwire 設定ファイルはほとんどのオペレーティングシステムに有効であるが、個々の要件が反映されるようにこの設定ファイルを十分にレビューし編集すべきである。検証するファイルとディレクトリの属性を選択する際に、管理者はファイルやディレクトリがシステムでどのように設定されているかを考慮する必要がある。たとえば、イベントによってレコードの書込みが発生するとログファイルも変更されるため、ログファイルのサイズが変更されていないことを確認するのは通常適切ではない。ただし、システムバイナリのサイズの変更やログファイルのアクセス権の変更を監視することは必要である。

インストールプログラムでは、Unix システムや Linux システムに対して Tripwire をインストールする場合の手順はステップごとに示されている。一方、Tripwire のインストールプログラムを使用すれば、Windows でのインストールは比較的容易に行えるが、Windows 用のシステムについてはインストール手順が示されていない。したがって、管理者は Tripwire をダウンロードしてインストールする前に、Tripwire をインストールするホストには次のソフトウェアアクセスがインストールされていることを確認しておく必要がある。

- ・ MD5 暗号チェックサムプログラム
- ・ ダウンロードファイルを解凍する GZIP

- ・ ソフトウェア配布の真偽を検証する PGP

- ・ AC コンパイラ

Tripwire をダウンロードまたは購入するには、<http://www.tripwiresecurity.com/>にアクセスする。Tripwire をダウンロードしたら、MD5 チェックサムを検証する。

D.4.1 UNIX への Tripwire のインストール

Tripwire ディストリビューションをインストールするは、十分なスペースがある格納先を選択する。また、インストールの前に、Tripwire ユーザーマニュアルを参照する。このマニュアルには、トラブルシューティングについての提案や今回の導入の対象外の事項についても詳細な情報が記載されている。

Tripwire をダウンロードしたら、次のコマンドで解凍する。

```
$ gunzip Tripwire-1_3_1-1_tar.gz
```

Tripwire の配布版を展開するには、次のシステム tar コマンドを実行する。

```
$ tar xvf Tripwire-1.3.1-1_tar
```

このコマンドによって tw_ASR_1.3.1_src というサブディレクトリが作成される。このサブディレクトリ内ですべての操作を実行する。

作成されたディレクトリには複数のファイルが存在する。そのうちの1つが Tripwire README ファイルである。このファイルには、設定や操作について様々な方法や設定に関する情報が記載されている。また、Ported というファイルには、Tripwire が移植されたプラットフォームとオペレーティングシステムがリストされている。リストから該当するオペレーティングシステムを探して、そのシステム設定を確認する。この設定情報は、Tripwire を正しく構築する上で必要である。管理者は README ファイルと Ported ファイルを確認することにより、特定のシステムへの Tripwire の設定方法について理解を深めることができる。

管理者は Ported ファイルのシステム設定に基づいて Makefile を変更し、Tripwire が特定のオペレーティングシステムに適切に設定されるように調整する必要がある。また、./include/config.h ファイルを編集し、個々のシステムに合わせて調整することも必要である。Tripwire 設定ファイルのパスとファイル名は ./include/config.h に指定されている。管理者は、Tripwire を設定しそのファイルを格納する場所を決定しなければならない。デフォルトを無許可の変更から適切に保護するために

も、格納場所は読み込み専用または取り外し可能な媒体にすべきである。

次に、管理者は Tripwire 設定ファイルの初期バージョンを作成する。配布版の ./config ディレクトリには、複数のオペレーティングシステム向けに様々なテンプレートが格納されている。管理者は、Tripwire 設定ファイルの格納先として指定されたディレクトリへ該当の OS に対するデフォルトファイルをコピーする必要がある。

```
# cp config/<appropriate OS config>/etc/tw.config
```

次に、管理者は /etc/tw.config を編集して(マニュアル参照)、ローカルシステムのバイナリファイル、そのほかの重要なファイル、またモニターの対象にしたいファイルを設定する。設定が完了したら、make コマンドを使用して、Tripwire をコンパイルし実行可能プログラムにする。

\$ make

make install コマンドを使用してインストールを開始し、Tripwire バイナリーと man ページを正しいシステムディレクトリに格納する。この操作を行うには、ルートアカウントを使用しなければならない。また、管理者が必要なすべてのファイルを次のようなディレクトリに手動で格納することもできる。

```
# cp man/siggen.8 /usr/local/man/man8/  
# cp man/tripwire.8 /usr/local/man/man8/  
# cp man/siggen.8 /usr/local/man/man8/  
# cp src/tripwire /usr/local/bin/  
# cp src/siggen /usr/local/bin/
```

Tripwire 配布版には、ビルドプロセスをチェックするスクリプト起動のテストスイートが備わっている。このテストスイートを実行するには、次のコマンドを入力する。

\$ make test

これにより、./test ディレクトリの Tripwire データベースのコピーに照らして Tripwire のビルドをテストするスクリプトが開始する。すべて問題なく進めば、テストの出力がスクリプトが提供する予想値と一致する。テストスイートの詳細については、Tripwire ユーザーマニュアルを参照すること。

インストールと設定の詳細については、Tripwire ユーザーマニュアル、README ファイル、man の該当ページを参照する。

D.4.2 Tripwire の使用に向けた準備

Tripwire がコンパイルされテストされたら、次に対応が必要な事項がいくつかある。すべてのファイルに対して同じレベルの保護が必要であるとは限らない。Tripwire には暗号化署名アルゴリズムがいくつか用意されている。実行速度が速いアルゴリズムもあれば、より安全性が高いアルゴリズムもある。(個々のアルゴリズムについては、Tripwire ユーザーマニュアルを参照のこと。) 管理者は、セキュリティとパフォーマンスがバランスよく反映されるように設定ファイルを調整する必要がある。Tripwire のデフォルト設定では、MD5 と Snefru の 2 つのアルゴリズムを使用して暗号化チェックサムが計算される。ほとんどのファイルやディレクトリの場合、MD5 だけで十分である。

Tripwire は次のいずれかのモードで実行できる。

1. データベース作成
2. 完全性チェック
3. インタラクティブアップデートモード
4. データベースアップデート

データベース作成と完全性チェックについては、次の項で説明する。

(1) Tripwire データベースの作成

完全性チェックでは、前回作成されたデータベースが存在し、このデータベースに対して比較を実行することが必要になる。このデータベースは tw.config ファイルによって作成される。tw.config ファイルが設定できたら、用意したフロッピーディスク(またはその他の適切な媒体)を挿入して、次のコマンドを入力する。

```
# mount -n /dev/disk /floppy
# tripwire -initialize
```

最初のコマンドではフロッピーがマウントされ、2 番目のコマンドでは、tw.db_<ローカルシステムのホスト名>というファイルが/floppy/databases/ディレクトリ内に作成される。このファイルは信頼できるコピーとして、ホストについてファイルシステム整合性チェックを実行する際に Tripwire によ

で参照される。管理者は自動あるいはインタラクティブによるアップデートモードを選択して、システムに変更が加えられるたびにその変更が Tripwire に反映されるようにデータベースを管理できる。

データベースが作成できたら、データベースと同じディスクに Tripwire プログラムのコピーとその設定ファイルを保管して、Tripwire ソフトウェアと重要なファイルが保護されるようにする。ルート権限を持つユーザーだけがファイルを読むことができるように、ディスクに書き込まれたファイルの所有者とアクセス権の設定によってアクセスを制限する。すべてのファイルが書き込み禁止のフロッピーディスクに保管されるため、ディスクのバージョンと信頼できる参照用コピーを比較することによって変更が容易に確認できる。このステップが完了したら、フロッピーディスクを取り外す。このステップを実行するには、次のコマンドを使用する。

```
# cp /etc/tw.config /floppy
# cp /usr/local/bin/tripwire /floppy
# umount /floppy
```

ディスクの書き込み禁止タブをロックして、書き込みできないようにする。これで、この書き込み禁止ディスクが信頼できる参照元となる。このフロッピーディスクを安全な場所に保管する。このディスクをコピーして作業用コピーを作成し、オリジナルコピーが日常の業務では使用されないようにする。

(2) 整合性チェック

信頼できる参照データが保存されている読み取り専用のフロッピーディスクを保管場所から持ってくる。書き込み禁止が設定されていることを確認し、フロッピーディスクを次のようにマウントする。

```
# mount -n /dev/disk /floppy
# echo 'test' > /floppy/test
/floppy/test: cannot create
```

最後のコマンドの後にファイルテストが存在する場合、フロッピーディスクは書き込み禁止にはならない。

個々のディレクトリとファイルを信頼できる参照データと比較する。内容やその他の属性が変更されたファイルが特定される。書き込み禁止フロッピーから直接 Tripwire を実行し、使用する設定

(-c オプション)とデータベース(-d オプション)ファイルを次のように特定する。

```
# cd /floppy
```

```
# ./tripwire -c ./tw.config -d ./databases/tw.db_<the local system's host name>
```

特定された変更の中から予期されなかった変更について調べる。Tripwire では次の変更が識別される。

- ・ 変更されたファイルまたはディレクトリ
- ・ 紛失したファイルまたはディレクトリ
- ・ 新しいファイルまたはディレクトリ

変更が許可されたアクティビティによるものでない場合は、インシデントレスポンスの手続きを即座に実行する。まず、インシデントの発生を社内のセキュリティ連絡先に報告し、必要に応じて追加のデータとして Tripwire レポートを提供する。

信頼できる参照データを安全な場所に戻す。Tripwire によって報告された変更がすべて予想されるものであれば、組織で定めた手続きに則って Tripwire データベースの信頼できる参照コピーを安全に更新する。

Tripwire のスキャンおよび更新プロセスが完了したら、参照データのフロッピーを取り外し、安全な場所に戻す。

```
# umount /floppy
```

D.5 Snort

Snort は、その開発者である Martin Roesch がライトウェイトネットワーク検知ツールと呼んでいるように、小規模の TCP/IP ネットワークを監視するためのネットワーク検知システムである。Snort は様々な疑わしいトラフィックや攻撃の試みを検知する。また、Snort は GNU (General Public License) のもとにどのような環境においても無償で利用できる。ルールベースのロギングを使用して、コンテンツのパターンマッチングを実行することにより、様々な攻撃やプローブが検出される。ルールセットには簡単な言語を使用しているため、新しい攻撃やエクスプロイトが発生すると新しいルールが迅速に処理される。

D.5.1 Snort プラグイン

レポート通知、ログファイルの監視、Snort によって作成される警告を簡素化するために、Snort と併せて実行されるプログラムが多数追加されている。解析用のフロントエンドとして設計されたプログラムの一部を次に紹介する。

- ACID (Analysis Console for Intrusion Databases) は、PHP ベースの解析エンジンで、Snort によって作成されたインシデントのデータベースを検索し処理できる。ACID の機能には、クエリビルダ検索インターフェース、レイヤ 3 とレイヤ 4 のパケット情報を表示するパケットビュー、警告管理システム、チャート統計ジェネレータなどがある。
- ARIS Extractor は Snort ログの解析に使用され、SecurityFocus の ARIS データベースに解析用の出力データを送る前に実行される。ARIS データベースは、ネットワーク管理者が疑わしいネットワークトラフィックや侵入の試みを匿名で提出できるサービスで、あらゆる貢献者からのデータを使用して詳細な解析やトラッキングを実行できる。
- Snort Report は Snort 用の PHP ベースフロントエンドで、MySQL や POSTGRESQL のデータベースから分かりやすいリアルタイムのレポートを作成する。
- SnortSnarf は Perl プログラムで、Snort の警告ファイルを HTML 形式に変換する。このプログラムはスケジュールに従って実行でき、作成した HTML レポートを診断や問題の追跡に使用できる。

D.5.2 Snort のインストール

Snort は次のような様々なプラットフォーム向けのパッケージとしてリリースされコンパイル

されている。

- ・ Linux
- ・ OpenBSD
- ・ FreeBSD
- ・ NetBSD
- ・ Solaris
- ・ SunOS 4.1.X
- ・ HP-UX
- ・ AIX
- ・ IRIX
- ・ Tru64
- ・ MacOS X Server
- ・ Win9x/NT/2000

コンピュータに Snort をインストールする最初のステップでは、必要なファイルをすべてダウンロードする。使用するプラグインやアドオンプログラムに応じて、ダウンロードするファイルも異なる場合がある。主に次のようなファイルが必要になる。

1. Snort
2. Snort ルール
3. WinPcap (Windows 用)

SnortとSnortルールは、公式のSnort web ページ(<http://www.snort.org>)からダウンロードできる。WinPcap は、Windows 用の無償で利用できるパケットキャプチャアーキテクチャである(<http://netgroup-serv.polito.it/winpcap/install/default.htm>)。パケットフィルタは、ネットワークカードから生のデータをキャプチャし、送信できる機能を追加するデバイスドライバである。WinPcapを使用することにより、キャプチャされたパケットをフィルタリングしバッファにストアできる。ネットワークから生データパケットをフィルタリングしてストアする機能より、WinPcap は Windows プラットフォームにおけるSnort NIDSの重要なコンポーネントとなっている。

D.5.3 Snort ルール

Snort には snort.conf がデフォルトのルールセットとして用意されている。このファイルは出発点としてはよいが、個々のニーズや要件に合わせてファイルの変更を希望する管理者も少なくないであろう。Snort ルールはシンプルであるが、検知機能としては効果的である。表 D.1 に示すように、Snort ルールセットには次のような種類のルールを含めることができる。

ルールの種類	機能
プロトコル	TCP、UDP、ICMP、IPで、Snort が疑わしい行為を検出する際に解析するプロトコル
IP アドレス	マッチングに使用するソース IP アドレスと宛先 IP アドレス
方向	トラフィックの方向
対象のポート	ポートトラフィックをオンにしてマッチングできるようにする。
オプションフィールド	パケットに対するマッチングルールに使用する追加オプション

表 D.1: Snort ルールの種類

Snort ルールのオプションフィールドによって、ルールに含めるオプションがさらに広がる。ルールのオプションは論理 AND によって処理される。つまり、ルールのすべてのオプションが真になっていないと、Snort ではルールアクションが実行されない。表 D.2 に、これらのオプションの一部をリストアップする。

オプションフィールド	機能
コンテンツ	特定のパターンのペイロードを検索する。
フラグ	指定した設定について TCP フラグをテストする。
ttl	IP ヘッダの TTL フィールドをチェックする。

itype	ICMP タイプフィールドのマッチングをする。
icode	ICMP コードフィールドのマッチングをする。
minfrag	IP フラグメントサイズの閾値を設定する
id	特定の値について IP ヘッダーをテストする。
ack	特定の TCP ヘッダの識別番号を検索する。
seq	特定の TCP ヘッダのシーケンス番号を検索する。
logto	特定のファイル名にルールをマッチングさせるパケットのログを記録する。
dsiz	パケットペイロードのサイズをマッチングさせる。
offset	コンテンツ検索を開始するパケットペイロードに offset を設定する。
depth	検索の開始地点を起点にしてバイト数を設定する。
msg	パケットによるイベント作成時に送信されるメッセージを設定する。

表 D.2: Snort ルールオプション

パケットが指定されたルールパターンと一致した場合に Snort が使用できる 5 つの基本アクションがある(表 D.3 参照のこと)。

ルールアクション	機能
Pass	パケットを無視して、通過させる。
Log	ランタイムに指定されたロギングルーチンに完全なパケットを書き込む。
Alert	選択した警告方法を使用してイベント通知内が作成され、パケットのログが記録される。
Activate	警告してから、別のダイナミックルールをオンにする。
Dynamic	Activate ルールがアクティブになるまで使用されず、ログルールとして機能する。

表 D.3: Snort ルールアクション

次に、<http://www.snort.org/docs/lisapaper.txt> に示されている Snort ルールの例をいくつか紹介する。

log tcp any any -> 10.1.1.0/24 79

このルールでは、10.1.1 のクラス C ネットワークアドレススペースに向かうポート 79(フィンガー) についてすべてのインバウンドトラフィックが記録される。

オプションフィールドの使用例を次に示す。

```
alert tcp any any -> 10.1.1.0/24 80 (content: "/cgi-bin/phf"; msg: "PHF probe!");
```

上記のルールによって、ローカルネットワークの Web サーバーにおける PHF サービスへのアクセスの試みが削除される。このようなパケットがネットワーク上で削除されると、イベント通知警告が作成され、ランタイムに選択されたロギングメカニズムを使用してパケット全体のログが記録される。

そのほかの Snort ルール例については、上記に示したドキュメントまたは Snort のデフォルトルールセットである.conf ファイルに記載されている。

D.5.4 Snort の使用法

Snort には次のような 3 つの主要モードがある。

1. スニッファモード
2. パケットロガーモード
3. ネットワーク侵入検知モード

Snort スニッファモードは、傍受された TCP/UDP/ICMP/IP ヘッダの一部またはすべてを画面に表示するための基本的な方法である。このモードは tcpdump への出力に非常によく似ている。表 D.4 には、スニッファモードで使用する簡単なフラグを示す。

フラグ	機能
-v	IP、TCP、UDP、ICMP のそれぞれのヘッダを出力する。
-d	IP、TCP、UDP、ICMP のトラフィックのパケットデータを出力する。
-e	IP、TCP、UDP、ICMP のトラフィックのデータリンクレイヤヘッダを出力する。

表 D.4: Snort スニッファモードフラグ

フラグは組み合わせて、結果を累積できる。パケットをディスクに記録するには、パケットロガー

モードを使用する。表 D-5 には、パケットロガーモードで使用する簡単なフラグを示す。

フラグ	機能
-l <log dir>	スニッファモードによって指定されたパケットが<log dir>に保存される。
-h <home network>	ホームネットワークに対するログを記録するには、ホームになるネットワークを指定する。
-b	バイナリモードまたは tcpdump 形式でログを記録する。
-r <log file>	読取専用モードで、さらにスクリーニングを実行するためにログファイルを再生する。

表 D.5: Snort ロガーモードフラグ

ネットワーク侵入検知モードは様々な方法で設定できる。ロギング方法に加え、複数の警告出力モードがある。デフォルトの警告方法では、完全な警告が発行され、デコードされた ASCII 形式でログが記録される。

表 D.6 には、警告出力モードがまとめられている。

Alert	説明
-A fast	タイムスタンプ、警告メッセージ、ソース IP、宛先 IP、ポートが表示される簡単なフォーマット
-A full	デフォルトモードで、警告メッセージとパケットヘッダ全体がプリントされる。
-A unsock	UNIX ソケットに警告を送り、ほかのプラットフォームがリスンできるようにする。
-A none	警告をオフにする。

表 D.6: Snort IDS モードフラグ

デフォルトのデコードされた ASCII フォーマットを使用して、バイナリファイルにパケットのログを記録したり、ログをまったく記録しないこともできる。パケットのロギングを無効にするには、.N コマンドラインのスイッチを使用する。

コマンドラインフラグや設定については、『Snort documentation』でさらに詳しく説明されている。表 D.7 には、Snort に関する情報と使用できるツールについて説明されているサイトのリンクを示す。

サイト/ツール/情報	Web サイト
ACID	http://www.cert.org/kb/acid/
ARIS	http://aris.securityfocus.com
Incident.org Plugin	http://www.incident.org/snortdb/
Snort	http://www.snort.org
Snort Documentation	http://www.snort.org/documentation.html
Snort User Manual	http://www.snort.org/docs/writing_rules
Snort Downloads	http://www.snort.org/downloads.html
Snort Report	http://www.circuitsmaximus.com
Snorticus Shell Scripts	http://snorticus.baysoft.net/
SnortSnarf	http://www.silicondefense.com/software/snortsnarf/
Whitehats.com	http://www.whitehats.com
WinPcap	http://netgroup-serv.polito.it/winpcap

表 D.7: Snort Web リソース

D.6 Nessus

Nessus は、Renaud Deraison によってリリースされた高速のモジュール式脆弱性スキャナである。この無償のクライアントサーバツールでは、リモートでネットワーク監査を実行し、プラグイン形式でインターネットセキュリティコミュニティによって日々更新されるデータベースに照らして既知の脆弱性を列挙しテストする。一般的なプラグインやセキュリティテストがバックドア、サービス拒否、ファイル、Windows などを対象に行われる。ユーザーは、NASL (Nessus Attack Scripting Language) スクリプトを使って新しいセキュリティテストを書くことによって、テストスイートを拡張できる。Nessus は、すべてのテストが実行されるホストにインストールされたサーバコンポーネントと、スキャンをコントロールするために別のシステムに配備されたクライアントソフトウェアで構成されている。スキャンの結果は完全にエクスポート可能なレポート形式で出力され、検出された脆弱性、リスクレベル、エクスプロイトへの対策などが表示される。

D.6.1 Nessus プラグイン

Nessus は、次のプラグインファミリーに分類される様々なセキュリティテストを実行できるようにデフォルト設定されている。

- ・ バックドア
- ・ CGI 悪用
- ・ CISCO
- ・ デフォルトの Unix アカウント
- ・ サービス拒否
- ・ フィンガーの不正使用
- ・ ファイアウォール
- ・ FTP
- ・ シェルのリモート取得
- ・ ルートのリモート取得
- ・ 一般
- ・ その他
- ・ ネットウェア
- ・ ポートスキャン
- ・ リモートファイルアクセス
- ・ RPC

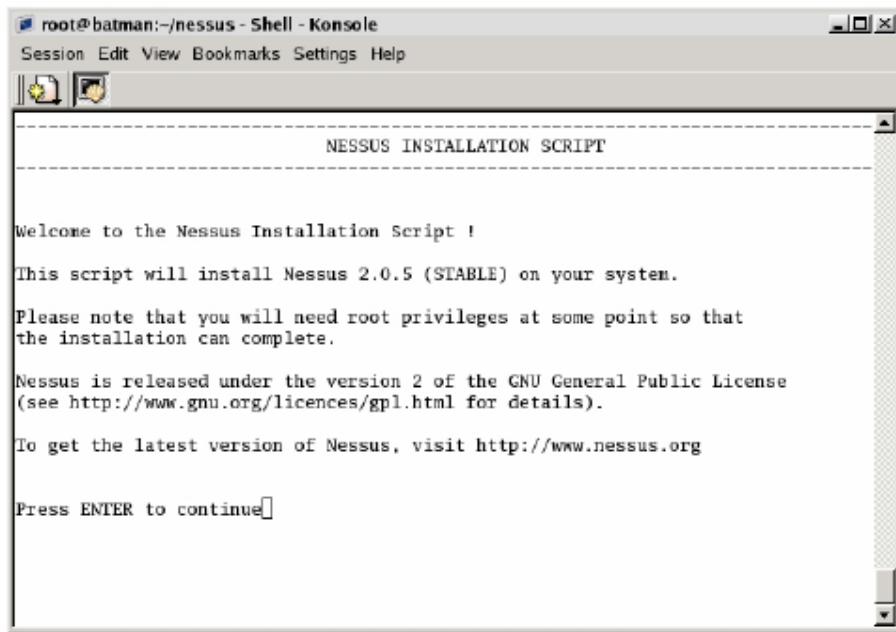
- ・ 設定
- ・ SMTP 問題
- ・ SNMP
- ・ 未テスト
- ・ 無用なサービス
- ・ Windows
- ・ Windows: ユーザー管理

セキュリティチェックの完全なリストについては、<http://cgi.nessus.org/plugins/dump.php3> を参照のこと。

D.6.2 Nessus のインストールと使用法

Nessus のサーバーコンポーネントは、Solaris、FreeBSD、GNU/Linux などの POSIX システムで稼動する。Nessus のクライアントソフトウェアは、多くのオープンソースのプログラムで使われている Widgets セットである GTK と共に実行される。また、Windows プラットフォーム用に特に設計されたクライアントプログラムもある。インストールパッケージは公式の Nessus Web ページ (<http://www.nessus.org/download.html>) からダウンロードできる。

1. `nessus-installer.sh` スクリプトをダウンロードし、`sh nessus-installer.sh` を実行してスタンドアロンパッケージをインストールする。



```
root@batman:~/nessus - Shell - Konsole
Session Edit View Bookmarks Settings Help

-----
                    NESSUS INSTALLATION SCRIPT
-----

Welcome to the Nessus Installation Script !

This script will install Nessus 2.0.5 (STABLE) on your system.

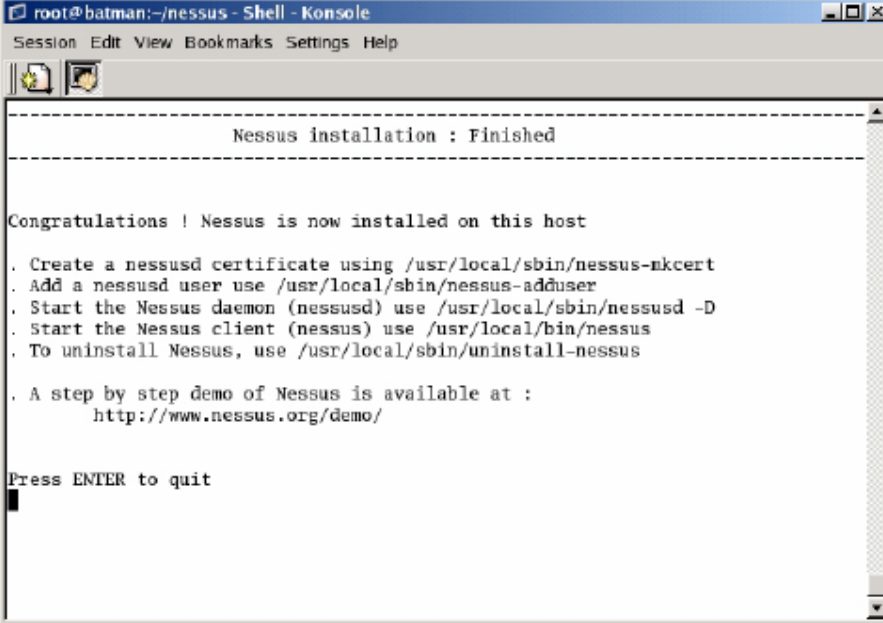
Please note that you will need root privileges at some point so that
the installation can complete.

Nessus is released under the version 2 of the GNU General Public License
(see http://www.gnu.org/licences/gpl.html for details).

To get the latest version of Nessus, visit http://www.nessus.org

Press ENTER to continue
```

2. 質問にいくつか答えると、Nessus がコンパイルされ、システムにインストールされる。次の図は、プログラムが正常にインストールされたことを表すメッセージと、Nessus の各種コマンドを表示した画面を示している。



```
root@batman:~/nessus - Shell - Konsole
Session Edit View Bookmarks Settings Help

-----
Nessus installation : Finished
-----

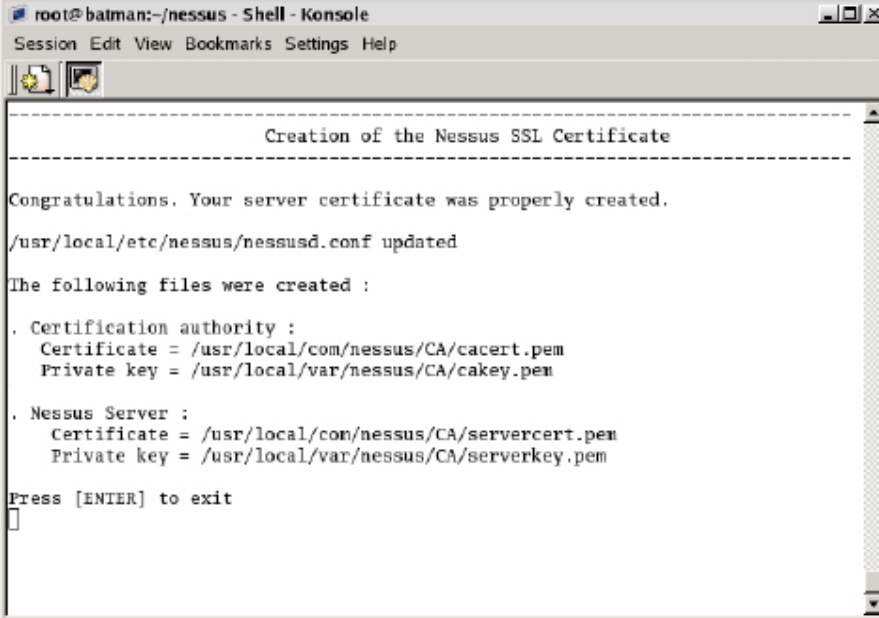
Congratulations ! Nessus is now installed on this host

. Create a nessusd certificate using /usr/local/sbin/nessus-mkcert
. Add a nessusd user use /usr/local/sbin/nessus-adduser
. Start the Nessus daemon (nessusd) use /usr/local/sbin/nessusd -D
. Start the Nessus client (nessus) use /usr/local/bin/nessus
. To uninstall Nessus, use /usr/local/sbin/uninstall-nessus

. A step by step demo of Nessus is available at :
  http://www.nessus.org/demo/

Press ENTER to quit
█
```

3. `/usr/local/sbin/nessus-mkcert` コマンドを実行すると、nessusd の証明書が作成される。次の図は、証明書が正常に作成された場合の画面表示を示している。



```
root@batman:~/nessus - Shell - Konsole
Session Edit View Bookmarks Settings Help

-----
Creation of the Nessus SSL Certificate
-----

Congratulations. Your server certificate was properly created.
/usr/local/etc/nessus/nessusd.conf updated

The following files were created :

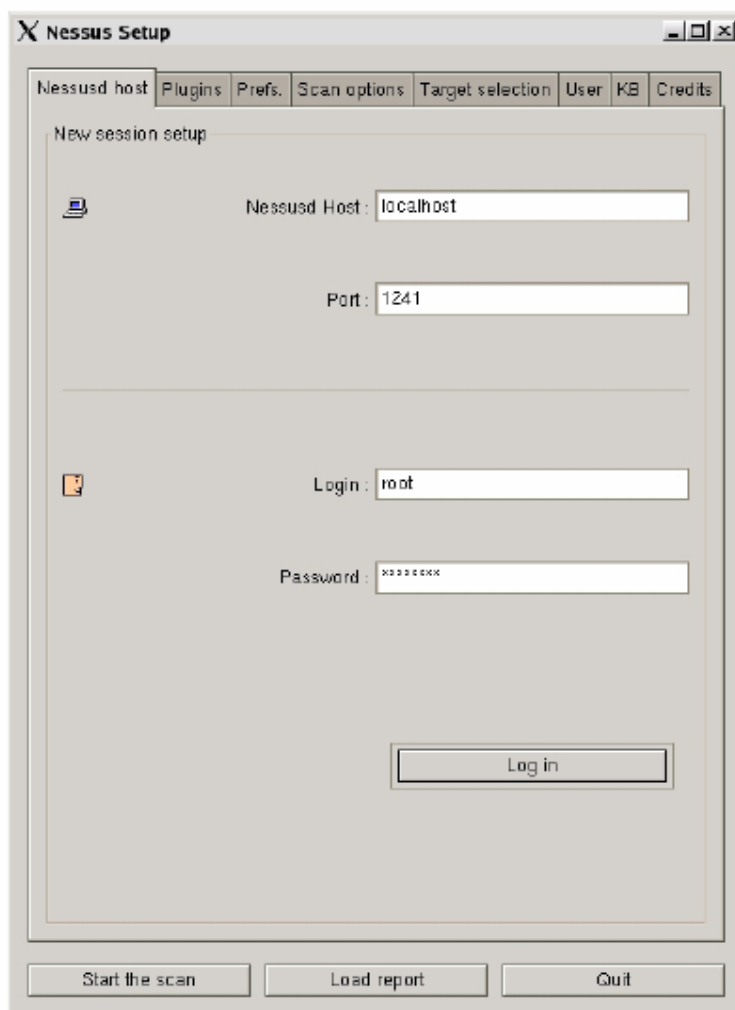
. Certification authority :
  Certificate = /usr/local/com/nessus/CA/cacert.pem
  Private key = /usr/local/var/nessus/CA/cakey.pem

. Nessus Server :
  Certificate = /usr/local/com/nessus/CA/servercert.pem
  Private key = /usr/local/var/nessus/CA/serverkey.pem

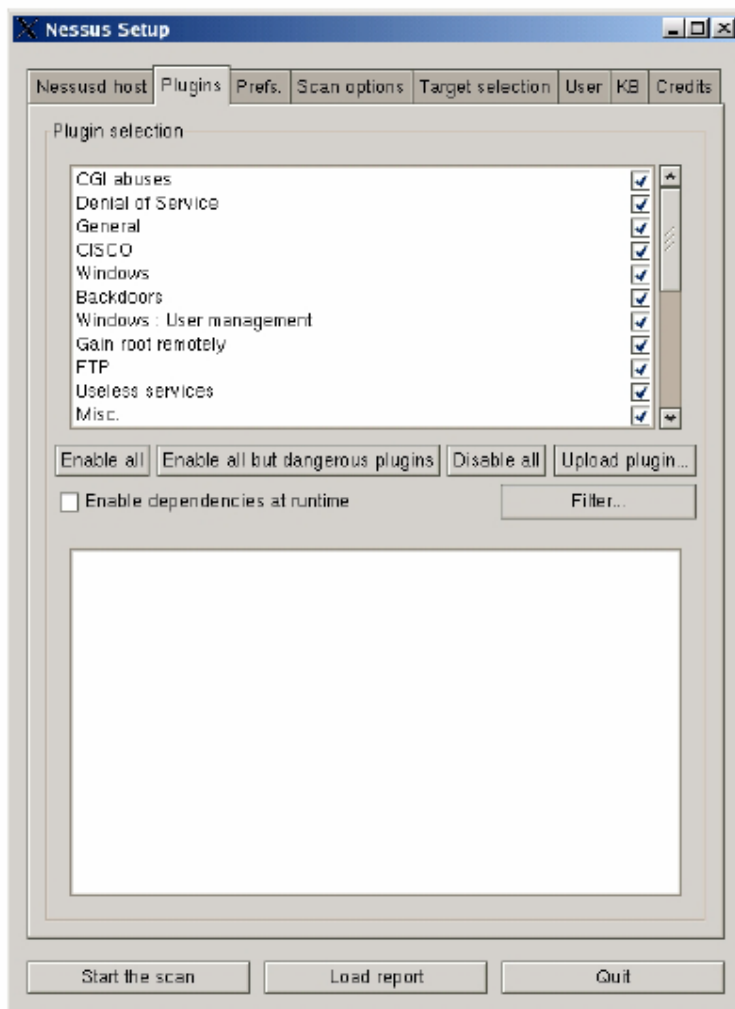
Press [ENTER] to exit
█
```

4. `/usr/local/sbin/nessus-adduser` コマンドを実行すると、スキャンの実行に使用する Nessus のユーザーアカウントが作成される。

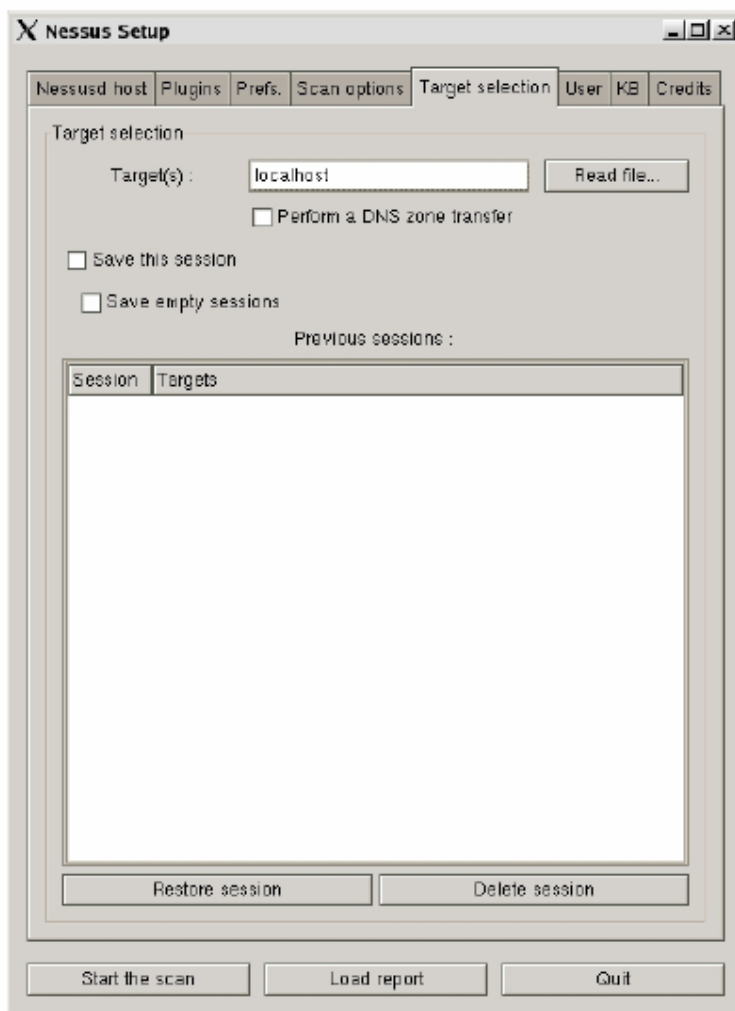
5. スクリプトを自動更新するには、`/usr/local/sbin/nessus-update-plugins` コマンドを使用する。このように設定することで、Nessus サイトから最新のセキュリティチェックをダウンロードできる。
6. `/usr/local/sbin/nessusd .D` コマンドを実行して、Nessus デーモンを起動する。
7. `/usr/local/bin/nessus` コマンドを実行して、脆弱性監査の設定と実行に使用できる Nessus クライアント(nessus)を起動する。
8. プログラムを実行できるように、ユーザー名とパスワードを入力する。



9. ホストのスキャンに使用するセキュリティチェックが入っている別のプラグインを選択する。注: Nessus には、脆弱なターゲットシステムをクラッシュさせることができる様々なサービス拒否テストが用意されている。



10. ターゲットのホストまたはシステムを選択して、ステータスを開始する。



11. スキャンが完了すると、開いているポート、検出されたサービス、セキュリティのインパクトと深刻度、推奨される解決策などがレポートに出力される。このレポートは HTML、XML など様々な形式で保存できる。

