

不正アクセス手法と技術的対策に関する調査

不正アクセスサーバ別詳細対策集

平成 13 年 3 月

情報処理振興事業協会

目次

1. <u>不正アクセス情報の対象とサーバ分類</u>	1
1.1. <u>各種のサーバに共通する不正アクセス対策</u>	1
1.1.1. <u>共通する対策方針</u>	1
1.1.2. <u>外部に公開する情報の制限</u>	1
1.1.3. <u>侵入への対処方法</u>	2
2. <u>メールサーバ</u>	5
2.1. <u>メールサーバの不正アクセス対策</u>	5
2.1.1. <u>サービスの限定</u>	5
2.1.2. <u>スパム対策</u>	5
2.1.3. <u>不正なメールの中継（踏み台）への対策</u>	6
2.1.4. <u>発信元の偽造への対策</u>	6
2.1.5. <u>脆弱性対策</u>	7
2.2. <u>SENDMAIL</u>	8
2.2.1. <u>旧バージョンの sendmail の問題点</u>	8
2.2.2. <u>不正アクセス対策</u>	9
2.2.3. <u>既知の不正アクセス手法</u>	9
2.2.4. <u>その他のメール転送エージェント</u>	11
2.3. <u>IMAP</u>	13
2.3.1. <u>IMAP サーバのセキュリティ対策</u>	13
2.3.2. <u>IMAP に関する不正アクセス手法</u>	13
2.4. <u>POP</u>	14
2.4.1. <u>POP サーバのセキュリティ対策</u>	14
2.4.2. <u>POP に関する不正アクセス手法</u>	14
3. <u>WEB サーバ</u>	16
3.1. <u>WEB サーバの不正アクセス対策</u>	16
3.1.1. <u>不正アクセスの傾向</u>	16
3.1.2. <u>対策の方針</u>	16
3.2. <u>情報漏洩に関する対策</u>	19

3.2.1.	Web ページからの情報の収集	19
3.2.2.	インデックス表示の設定	19
3.2.3.	重要情報（個人情報）の扱い	19
3.3.	WEB アプリケーションの脆弱性に関する対策	20
3.3.1.	既知の弱点を有する Web アプリケーションへの対策	20
3.3.2.	新たに作成された Web アプリケーションへの対策	20
3.3.3.	バッファオーバーフロー対策	21
3.3.4.	フォーマテッドストリング（シェルメタキャラクタ）対策	21
3.3.5.	機密漏洩対策	21
3.3.6.	CGI	22
3.3.7.	ASP	22
3.3.8.	php	22
3.3.9.	SSI	23
3.3.10.	フォームの返す情報の改ざんによる攻撃	23
3.3.11.	参照可能なファイルへのタグの追加	24
3.4.	サービス妨害攻撃への対策	25
3.4.1.	DoS 攻撃	25
3.4.2.	DDoS 攻撃	26
3.4.3.	サーバにおける DDoS 攻撃対策	27
3.4.4.	攻撃手法別 DDoS 攻撃対策	28
3.5.	IIS	33
3.5.1.	IIS 4.0 の設定	33
3.5.2.	IIS 5.0 の設定	33
3.5.3.	脆弱性対策	36
3.6.	APACHE	41
3.6.1.	Apache の設定	41
3.6.2.	Apache に関する脆弱性と影響を受けるバージョン	42
4.	ファイアウォール	45
4.1.	ファイアウォールの不正アクセス対策	45
4.2.	フィルタリング方式	45
4.3.	構築の基本姿勢	46
4.4.	弱点	47
4.4.1.	オープンなポートを通じた攻撃	47

4.4.2.	動的なポート番号の割り当て	47
4.4.3.	監視上の対象の制約	47
4.5.	脆弱性対策	48
5.	その他	50
5.1.	DNS	50
5.1.1.	ゾーン転送設定のミス	50
5.1.2.	BIND への攻撃	50

1. 不正アクセス情報の対象とサーバ分類

本稿は主にシステム管理者を対象に、各種のサーバについて、サーバプログラム全体の特徴、代表的・一般的な対策、個々のプロダクトに関する脆弱性対策の概要を述べる。

取り上げるサーバの種類は、攻撃者が狙うサーバプログラムの傾向、サービスが停止時の影響の大きさなどを基に以下のサーバを選択した。

- ・ メールサーバ
- ・ Web サーバ
- ・ ファイアウォール

1.1. 各種のサーバに共通する不正アクセス対策

ここでは各種のサーバに共通する不正アクセス対策について簡潔に述べる。

1.1.1. 共通する対策方針

サーバを外部に公開する場合には、たとえサーバの用途が異なったとしても、以下のようなセキュリティ対策の方針は共通するものである。

- ・ サーバの用途に不必要な機能を外部に提供しない。基本的に 1 つのホストは 1 種類のサービス（およびそれに密接な関係を持つサービス）を提供するのみに設定する。使っていないサービスは停止・削除し、ポートは閉鎖する。
- ・ 外部に過剰な情報は提供しない。Web コンテンツ、エラーメッセージ等に外部に漏らすべきではない重要な情報が不注意から含まれていることがある。
- ・ セキュリティに関する適切な設定を行い、動作に設定が反映されていることを必ず確認する。設定に関しては単一の情報源に頼らず、新たな情報の入手に努め、随時、必要に応じて修正し直す。
- ・ セキュリティホール情報の入手に努める。信頼のおける情報源からの情報を得る。必ずベンダから入手した修正プログラムを適用する。
- ・ ログを継続的に取得し、こまめに解析する。
- ・ 作業内容の記録を取る。導入したプログラム名およびバージョン、適用した修正プログラム名等についての記録を残す。対象マシン名、作業を行った日時も併記する
- ・ 導入時の作業、リモートからの更新修正作業などのためにセキュリティに穴を開けた状態を作るならば、どんなに短期間であっても、攻撃者にその綻びを気付かれて攻撃を受けることを想定しなければならない。

1.1.2. 外部に公開する情報の制限

一般に攻撃者は、マシンの特定の脆弱性を利用した侵入を試みる前に、綿密な準備を行う。攻

撃者は標的となる組織の概要を掴み、扉を叩くようにスキャンを行って情報を集め、システムに軽く接触して設定ミスや構成の弱さから防御の甘い情報の列挙を試みる。管理者が適切な防御策を施していなければ、攻撃者は以下のような情報を収集できる。

- ・ IP アドレスの範囲
- ・ DNS サーバやメールサーバのアドレス
- ・ 従業員名や電話番号
- ・ 内部ネットワークの構成
- ・ 稼動しているシステム
- ・ 動作しているサービス名、バージョン等
- ・ 開かれているポート
- ・ OS 名
- ・ ドメインリソース
- ・ 共有リソース
- ・ ユーザ名、グループ名

これらの情報を得るためのアクセスは、一般に公開された情報を取得するアクセスから、侵入を意図しなければ決して試みることがないアクセスまでを幅広く含む。脆弱性を狙った不正アクセス手法と明確に区別されるものでもない。

無作為な攻撃の標的に選ばれることを避けるために、管理者はこれらの情報の漏洩を最小限に留めるか、これらの情報を収集する試みを監視するよう適切な対策を施す必要がある。

1.1.3. 侵入への対処方法

侵入を発見した再には以下のような対処が必要となる。

- ・ 問題発生が疑われた時点から、できるだけ詳細に全てを記録する。問題の重要性によっては技術・法律の両面で細かな記録が後に役立つ。
- ・ 侵入に関する情報を過度に外部に漏洩しないようにする。連絡は、問題解決の支援者、信頼がおける連絡機関、組織のセキュリティ担当者に留める。過度の情報の漏洩により、問題の不明確化、対応策を採る前のメディアの介入、二次的な侵入被害を招く可能性がある。
- ・ 組織内のセキュリティ専門家にコンタクトを取り支援を求める。不在の場合は OS やシステムソフトウェアに関する問題対処能力のあるコンサルタントに速やかな支援を要請する。

参考として、侵入が発見時にサーバの状態を保存する方法とサーバを構築し直す方法の概要を以下に示す。作業を間違えれば復旧と侵入者の追跡が極めて困難になるため、可能な限り専門家の支援を要請すべきである。

1. 侵入を受けたコンピュータを隔離する。
 - (ア) 侵入を受けたコンピュータのネットワークケーブルを抜く。
 - (イ) 利用可能であれば代用としてバックアップサーバを利用する。バックアップサーバが利用できない場合には、システムの一部停止は不可避である。
2. 侵入を受けたコンピュータのバックアップをファイルシステムの実行状態を含めて取る。
 - (ア) OS に管理されている動的なデータテーブルを全て標準ファイルにダンプする（このテーブルはあとで分析する）。現在実行中のプロセス、現在ログイン中のユーザ、現在のネットワークへの接続のリストをファイルにダンプする。
 - (イ) 異なる 2 種類のバックアッププログラムを使用し、システムのバックアップを 2 つ作成する。
3. 侵入を受けたコンピュータをシャットダウンする。
4. コンピュータを再起動する。
5. システムソフトウェアに使用されているドライブを物理的に再フォーマットする。
6. システムを再構築する。
 - (ア) OS をオリジナルメディアから再インストールする。
 - (イ) OS のパッチ（サービスパック）をベンダから入手し、最新のものまで全て適用する。同様に既知の脆弱性を修正するプログラムを全て適用する。
 - (ウ) システムをセキュリティ面で強化する。デフォルトの状態から OS に固有な再設定を行い、一般に知られる弱点を無効にする。
 - (エ) システムをリストアする。バックアップファイルを使わないこと。それらは既に侵入者によって改ざんされた可能性がある（タイプスタンプやファイルサイズは偽造可能である点に注意）。オリジナルの配布物から再構築し、固有の設定等もバックアップからのコピーを使わず、手動で再定義する。
7. コンピュータをネットワークに接続する。
 - (ア) ネットワークケーブルを接続する
 - (イ) 設定した通りの動作を行うことを確認する。
 - (ウ) 設定した通りのセキュリティが実現されていることを確認する。
8. ネットワーク上の各コンピュータを確認し、侵入されたマシンを探す。

より詳細な対処方法については以下の文書が参考になる。

- ・ RFC2196 サイトセキュリティハンドブック（日本語版）
<http://www.ipa.go.jp/security/rfc/RFC2196-00JA.html>
- ・ RFC2504 ユーザーズセキュリティハンドブック（日本語版）
<http://www.ipa.go.jp/security/rfc/RFC2504JA.html>
- ・ Steps for Recovering from a UNIX or NT System Compromise
http://www.cert.org/tech_tips/root_compromise.html

- The World Wide Web Security FAQ (日本語版)

<http://www.w3.org/Security/Faq/001031wwwsfj.ja.sjis.html#Q96>

2. メールサーバ

以下にメールサーバの全体的な対策と、各サービスプログラムに関する対策を述べる。

2.1. メールサーバの不正アクセス対策

メールサーバに対する不正アクセスには以下のような傾向が見られる。

- ・ スパムの受信
- ・ 不正中継への利用
- ・ 発信元の偽造
- ・ ホストへの侵入
- ・ サービス妨害

以下ではこれらを防ぐための対策について述べる。

2.1.1. サービスの限定

メールサーバマシンの機能は可能ならばメールサービスに限定し、他の不要なサービスは全て停止する。不要なサービスコマンドもマシンから削除する。

2.1.2. スпам対策

スパム行為には以下のような問題点がある。

- ・ サーバリソースへの負担：
ISP 等のサーバが多数のアカウントへのスパムを受信する場合はかなりの負担となる。
- ・ 無差別な送信を受ける各ユーザへの負担：
スパムは受信者にとって興味を持たない内容が殆どであり、受信にあたり金銭・時間の浪費となる。ネズミ講の類の誘いであることも多い。

スパム対策は以下のようにまとめることができる。

- ・ メールアドレスの漏洩を避ける：
多数の宛先メールアドレスを集めるための手法としては、Web ページやニュースグループに投稿された記事に記載されたメールアドレスをロボットに収集させることが多い。メーリングリスト (ML) の購読者リストも収集の対象となるため露呈しないよう注意を払う必要がある。
- ・ スпамに対するフィルタリング設定：
受信を避けるために、スパム発信源として知られているドメインを登録し、それらからのメールを一切受けつけない設定を行う。ただし、あるドメインやアドレスを完全に拒絶してしまうと利用者の不便となる可能性もある。

- ・ スпам行為への対応策（ポリシー）の策定：
内部ユーザがスパムを発信する場合も考えられる。

2.1.3. 不正なメールの中継（踏み台）への対策

スパム配信時に、送信者が自分のメールサーバを利用せず、他人のサーバの中継機能を外部から悪用することが大きな問題となっている。第三者からのメールの中継するような誤った設定が施されたサーバはスパムメールの中継に悪用される可能性が非常に高い。メールサーバを不正な中継に利用された場合、組織の信用を大きく失う。メールサーバがスパム配信元ブラックリストに登録されると、送信したメールの受信を拒否されるなど、通常の利用者のメール利用が阻害される可能性がある。

不正利用者が踏み台を用いる理由は、発信元の隠蔽と配信処理能力の借用のためである。

- ・ 発信元の隠蔽：
後述する送信元の偽造（スプーフィング）と組み合わせることで、スパム発信者の追跡を困難なものにすることができる。
- ・ 配信処理能力の借用：
膨大な量の宛先への配信を行うための不正なテクニックとして、発信者自らのサーバ資源を用いずに他人のサーバの資源を奪う形で送信する手法が用いられる。これはメールメッセージのエンベロープ内の「RCPT TO:」レコードに複数のメールアドレスを列記し、中継サーバで展開させるものである。sendmail プログラムが実行されている環境では、この展開と配送の処理を行うと処理能力の殆どを奪われるため、通常のメールの送受がほぼ不可能になる。

踏み台対策としての中継に関する設定は以下のようにまとめられる。

- ・ 自ネットワーク内からのメールは全てリレーする。
- ・ 外部からのメールは自ネットワーク内宛のメールのみ受け取り、外部からきた外部の RECIPIENT 宛のメールは配送を拒否する。

代表的なサーバプログラムである sendmail ではバージョン 8.8.x で導入された check_relay ルーチンを利用し、中継に関する設定を徹底することができる。

2.1.4. 発信元の偽造への対策

ヘッダやエンベロープの偽造により、メール発信元として第三者のアドレスを偽造するテクニックが良く知られている。電子メールでの返信を期待していないスパムメールの発信者は、発信元の特定を困難にするために偽造発信元アドレスを付けたスパムメールを配送する。

発信元としてメールアドレスを悪用された場合、以下のような問題が生じる：

- ・ エラー、返信、苦情に関する処理：

無効なアドレスへの送信は発信元にエラーメールが返信されてくるため、アドレスを騙られた側は大量のエラーメールに見舞われることになる。返信メールや苦情処理にも人的資源を割かなければならない。

- ・ 信用の喪失

スパムメールや誹謗・中傷メール等を送信したとみなされることにより、送信元を騙られた組織の社会的な信用は大きく損なわれる。

メールアドレスの偽造（スプーフィング）は技術的にはさほど困難なものではない。sendmailのようなMTA（Mail Transfer Agent）が用いるSMTPは、全て平文（7ビットASCII）でやり取りが行われ、プロトコル中での送信者のチェックも行われない。

対策としては、偽造アドレスにより発信元とみなされ外部から苦情等が来ることに備えた事前対策が必要である。外部への窓口は一元化することが望ましい。予め当該部署を設定しWebページやメールアドレス等を設置して連絡窓口を整えておく。担当部署は対処手段を決定し、組織全体への周知に努める必要がある。

アドレスを詐称されたことを示す必要に備え、自サイトから該当するメールが発信されていないことを証明するためにマシンの稼動状況等のログを保存しておく。

偽造を防ぐことはできないが、平常時よりデジタル署名をメールメッセージに適用することを心がければ、受信者側でメールの送信元を判断する助けになる。

2.1.5. 脆弱性対策

メールサーバの脆弱性を利用した不正アクセスには、以下のような傾向が見られる。

- ・ サービス妨害攻撃。脆弱性に基づくサービス妨害攻撃の手法として挙げられるものには、メールヘッダ、リクエスト等の入力に対するバッファオーバーフローが元で引き起こされるものと、サーバリソースの枯渇を狙ったものの2種類が特に多い。
- ・ 侵入（管理者アクセス権の奪取）。特にメールサーバに関してはバッファオーバーフローに関する脆弱性を利用するものが多い。
- ・ 電子メールの不正中継への利用。設定の不備（放置）を狙う場合が最も多いが、既知の脆弱性を利用して中継が行えるケースもある。
- ・ 電子メールに対するウイルス検査フィルタリング機能の無効化。

脆弱性への基本的な対策としては、問題となるプログラムを修正されたバージョンの新しいプログラムに更新することが挙げられる。

2.2. sendmail

sendmail およびその他のメール転送エージェント (MTA) に関する不正アクセス手法と対策について述べる。

sendmail は広く使われているメール転送エージェントプログラムである。高い拡張性を持ち高度な構成が可能である。sendmail はコードが 8 万行におよぶ巨大で複雑なプログラムであり、これまでに数多くのセキュリティ上の弱点が報告されている。sendmail は設定が困難なことで知られている。

2.2.1. 旧バージョンの sendmail の問題点

管理者は sendmail については、セキュリティ情報を常に収集し、必要であるなら設定の変更や最新バージョンのプログラムへの更新が求められる。

旧バージョンの sendmail は、メールの中継を認めるデフォルト設定が施されている点、プログラムにバッファオーバーフローに関する脆弱性が存在する点など、多くの問題を抱えている。これらの弱点は非常に良く知られた致命的なものであり、放置すればメールサーバへの侵入や不正な中継利用を受ける原因となる。

以下に各バージョンの sendmail の持つ問題とその対策をまとめる。

- ・ 5.x で表されるバージョンの sendmail (R5 sendmail) には多くの致命的なセキュリティホールが存在することが知られている。旧バージョンであれば攻撃者にサーバ情報を収集された場合に、極めて魅力的な攻撃対象と見なされ得る。また、旧バージョンのプログラムは適切なサポートを受けらず、脆弱性を修正することが全くできない場合もある。このような理由から、8.x.x で表されるバージョンの sendmail (R8 sendmail) の中で入手可能な最新のバージョンへ可能な限り早く変更する必要がある。
- ・ sendmail-8.8.x 以降にはスパム対策および踏み台対策のための設定内容をチェックするツール check_relay が付属する。
- ・ 8.8.8 以前の sendmail は第三者によるメールの中継を許可する設定がデフォルトで取られているため、それらのバージョンの sendmail が導入されている場合はスパム対策のための設定と動作確認が必須である。
- ・ sendmail-8.9.0 以降ではデフォルト時の sendmail.cf ファイルに第三者からのメールの中継を行わないよう設定されている。
- ・ 8.9.0 以降のバージョンの sendmail についてもいくつかの脆弱性が指摘されている。Bugtraq (<http://www.securityfocus.com>) 等で脆弱性に関する最新情報を集める必要がある。

2.2.2. 不正アクセス対策

sendmail の利用に際しては、入手可能な最新バージョンのプログラムを導入して適切な設定を慎重に施す必要がある。sendmail は歴史のある有名なプログラムであるが、導入後に脆弱性が報告される可能性は未だにある。脆弱性情報に注意を払う必要が特にあるプログラムの 1 つである。可能ならば最新のプログラムを導入すべきである。

sendmail に関する不正アクセス対策項目を以下に示す。

- ・ メール受信に利用していない不要な sendmail サービスの停止、プログラムの削除
- ・ 最新バージョンの sendmail の導入
- ・ スпамメールの不正中継配信への対策
- ・ サービス妨害攻撃への対策
- ・ サーバのアカウントに関する情報列挙への対策

sendmail の設定に関しては、定義ファイル sendmail.cf は難解な記述方法で書かれている。設定変更時にこのファイルを直接編集することは極めて困難なので、条件を判りやすく記述したファイルから sendmail.cf を自動的に作成するツール CF を用いる。

2.2.3. 既知の不正アクセス手法

重大な危険を及ぼす sendmail の脆弱性の例を以下に挙げる。

- ・ VRFY コマンドと EXPN コマンドを利用したユーザアカウント情報の列挙・特定 (R5 sendmail)。R8 sendmail ではこの要求を無効にする記述を mail.cf に組み込むことが可能。
- ・ debug コマンドの脆弱性 (バージョン 5.58)。debug コマンドが有効に設定されているため、リモートからの攻撃者が任意のシェルコマンドを実行可能。この脆弱性はインターネットワームに利用された。CA88-01、CA93-14、BID:1、CVE-1999-0095
- ・ mail from および rcpt to の脆弱性 (バージョン 5.58、5.59 などの 8.6.10 未満のバージョン)。SMTP を介してリモートからの攻撃者がルート権限を取得可能。不正な mail from および無効な rcpt to アドレスを指定することで、他のプログラムにメールの内容がリダイレクトされる。CVE-1999-0203
- ・ ident 関数の脆弱性 (バージョン 8.6.9)。IDENT 関数の脆弱性により、リモートからの攻撃者はルートアクセス権を取得可能。BID:2311

- ・ MIME エンコーディング処理におけるバッファオーバーフロー脆弱性（バージョン 8.8.0、8.8.1 および 8.8.3、8.8.4）。リモートからの攻撃者がルート権限を不正取得可能な脆弱性のひとつはバージョン 8.8.0 および 8.8.1 に存在する。またこれと直接の関係は無いもののやはり MIME エンコーディングに関する脆弱性でルート権限を不正取得可能なものが 8.8.3 および 8.8.4 に存在する。CVE-1999-0206、CVE-1999-0047
- ・ mail.local の脆弱性（バージョン 8.9.3）。sendmail に含まれる mail.local プログラムが終端である「.\n」文字列の適切な確認を行わない。このため「.\n」が最後につけられた 2047 文字に及ぶ長い文字列が送られた場合、メッセージの終端をごまかすことが可能である。偽のメッセージ終端記号の後に続く任意のテキストは mail.local によって LMTP コマンドとして扱われる。偽のメッセージ等を sendmail によるフィルタリングやロギングを受けずに任意のメールボックスに送信できる。リモートからの攻撃者はローカルメール配信に対するサービス妨害や、メールボックスの破壊が可能。CVE-2000-0319

2.2.4. その他のメール転送エージェント

sendmail 以外のメールサーバプログラムを実行している場合についても、初期設定ミスなどの脆弱性を攻撃されて不正中継に悪用されることや、バッファオーバーフロー攻撃により侵入を受ける可能性があり、適切な対策が必要となる。

以下にその他のメール転送エージェントに関する不正アクセス手法の例をあげる。

NetWin DMail における ETRN リクエストのバッファオーバーフロー脆弱性 (CVE-2000-0490)

- ・ 手法：Dmail デモンにおける脆弱性。260 文字以上の長い ETRN リクエストを渡すことでバッファオーバーフローを起こすことが可能。
- ・ 影響：リモートからの攻撃者はサーバをクラッシュさせることによるサービス妨害、およびルート権限での任意の命令の実行が可能。
- ・ 予防対策：DMail 2.7r または、2.8k にアップグレードする。

Netwin DMailWeb and CWMail Server のメール中継に関する脆弱性 (CVE-2000-0610)

- ・ 手法：改行コードを含むユーザ名を用いることで、認証機構をバイパスし、登録されたユーザでなくてもログインが可能である。
- ・ 影響：リモートからの攻撃者はサーバのメール機能を利用可能な不正なメールの中継にサーバを悪用できる。
- ・ 予防対策：バージョン 2.6j 以降へのアップグレードを行う。

Lotus Domino Server における ESMTP バッファオーバーフローに関する脆弱性 (CVE-2000-0452)

- ・ 手法：ESMTP サービスの rcpt to、saml from、soml from などの FROM コマンドを扱うコードで、バッファオーバーフローに関するチェックが適切に行われていない。サーバが上記コマンドを 4KB 以上の引数とともに受け取った場合、システムクラッシュが発生する。
- ・ 影響：リモートからの攻撃者はサービス妨害攻撃が可能。機能の復旧には再起動が必要となる。
- ・ 予防対策：Lotus Domino Version 5.0.5 へのアップデートを行う。

Lotus Domino Server の ESMTP に関する脆弱性 (CVE-2000-0452)

- ・ 手法：ESMTP サービスの rcpt to、saml from、soml from などの FROM コマンドを扱うコードで、バッファオーバーフローに関するチェックが適切に行われていない。サーバが上記コマンドを 4KB 以上の引数とともに受け取った場合、システムクラッシュが発生する。

- ・ 影響：リモートからの攻撃者はサービス妨害攻撃が可能。機能の復旧には再起動が必要となる。
- ・ 予防対策：バージョン 5.0.5 へのアップデート。

MsgCore/NT におけるサービス不能脆弱性 (CVE-2000-0075)

- ・ 手法：smtp クライアント(あるいは手動入力で送信するユーザ)によって単一の接続において複数回の HELO/ MAIL FROM/ RCPT TO / DATA シークエンスが送られると、これらの値を蓄えるために割り当てられたメモリが解放されない。このため送信対象とされたマシンのメモリが枯渇し、機能が停止する。
- ・ 影響：リモートからの攻撃者はホストを再起動が必要な状態にフリーズさせ、サービスの妨害を行うことが可能。
- ・ 予防対策：この問題は MsgCore 2.x では修正されている

2.3. IMAP

IMAP (Internet Message Access Protocol) はクライアントからホストのメールを読み出すためのプロトコルである。IMAP サーバはメッセージをサーバで集中的に管理するため、POP サーバに比べ柔軟な処理が可能である。

IMAP を利用する上で、各ユーザはサーバ上にメールボックスを置くことができる。メールサーバからメッセージを選択的にダウンロードすることや、複数のクライアントマシンから同一のメールボックスにアクセスすることが可能である。

2.3.1. IMAP サーバのセキュリティ対策

当然のことながら IMAP を利用しないならサーバからサービスを削除し、ポートを閉じておくべきである。以下に IMAP に関するセキュリティ対策をまとめて示す。

- ・ 脆弱性対策：
IMAP プログラムに関してはいくつかの脆弱性が報告されている。最新版のインストールが必要である。
- ・ アクセス制御の徹底：
クライアントの IP アドレスやドメイン名によって IMAP サーバへのアクセスを制限する。
- ・ 認証の強化・暗号化：
IMAP サーバへの接続の認証に使われるユーザ名とパスワード名は平文でネットワークを流れるために盗聴に合う可能性がある。認証に用いられるパスワードを暗号化する方法 (CRAM-MD5) を採用することで認証機構を強化できる。しかし、この場合はメールメッセージ本文の暗号化は行われない。
SSH を利用したポートフォワーディングにより通信内容を保護することも可能である。この場合は認証機構だけでなくメッセージ本文も暗号化される。

2.3.2. IMAP に関する不正アクセス手法

IMAP の脆弱性に関する不正アクセス手法について、以下に例を挙げる。

SuSE IMAP サーバにおける認証機構の脆弱性 (CVE-2000-0233)

- ・ 手法：IMAP の認証機構を迂回可能。詳細不明。
- ・ 影響：リモートからの攻撃者はサーバの imap 管理者のアクセス権を獲得可能。
- ・ 予防対策：パッチの適用

2.4. POP

POP (Post Office Protocol) はホストで受信されたメールメッセージをクライアントに受信するためのプロトコルである。

クライアントが相手から直接インターネットメールを受け取ることはできないので、メールサーバにいったんメールを受信し、メールサーバ上で動作している POP サービスを通して配信を受ける。sendmail に代表される MTA が動くメールサーバの多くでは、クライアントの PC にメールを配信するために POP サーバを動作させて用いている。

POP の現在のバージョンは 3 である (POP3 と呼ばれる)。

2.4.1. POP サーバのセキュリティ対策

以下に POP サーバのセキュリティ対策を示す。

- ・ 脆弱性対策：
既知の脆弱性が修正された最新版のプログラムをインストールする。
- ・ アクセス制御の徹底：
クライアントの IP アドレスやドメイン名によって POP サーバへのアクセスを制限する。
- ・ 認証の強化・暗号化：
POP サーバへの接続の認証に使われるユーザ名とパスワード名は平文でネットワークを流れるために盗聴に合う可能性がある。認証に用いられるパスワードを暗号化する方法 (APOP 認証) を採用することで認証機構を強化できる。しかしながら、APOP 認証を用いてもメッセージ本文は暗号化されない点には注意が必要である。
SSH を利用したポートフォワーディングにより通信内容を保護することも可能である。この場合は認証機構だけでなくメッセージ本文も暗号化される。

2.4.2. POP に関する不正アクセス手法

POP サーバプログラムの脆弱性について、以下にいくつかの例を挙げる。

vpopmail における不正なフォーマット文字列のユーザ入力に関する脆弱性 (CVE-2000-0583)

- ・ 手法: vpopmail の一部であり認証を行う vchkpw プログラムの脆弱性。vsprintf() 呼出へのフォーマット文字列が適切にチェックされず、ユーザの入力するデータを不正に用いることができる。任意のフォーマット指示を含む USER あるいは PASS コマンドの入力によりスタック値を上書き可能である。
- ・ 影響: vpopmail サーバプロセスの持つアクセス権で任意のコマンドを実行可能。
- ・ 予防対策: バージョン 4.8 へアップグレードする。

Netwin DMailWeb および CWMail Server におけるサービス妨害攻撃に対する脆弱性 (CVE-2000-0611)

- ・ 手法：ユーザは特定の POP3 サーバにログインすることができる。多量のアカウトを作成することで、メールサーバによって使われる SMTP サービスに処理を殺到させることが可能。
- ・ 影響：リモートからの攻撃者は、SMTP に対するサービス妨害攻撃が可能。
- ・ 予防対策：バージョン 2.6g 以降の DMailWeb にアップグレードする。あるいは DmailWeb のコンフィグレーションファイルに `- force_primary = true - valid_pop = {安全な POP サーバのリスト}` を加え、ログイン時に POP サーバへの認証を要求するよう設定する。

MDaemon 2.8.5.0 POP サーバにおける UIDL コマンドによるサービス妨害攻撃に対する脆弱性 (CVE-2000-0501)

- ・ 手法：POP サーバの脆弱性。pass コマンドの後に UIDL コマンドを入力しレスポンスが返ってくる前に即座にサーバから抜けると、競合状態が起こり、サーバがクラッシュする。
- ・ 影響：リモートからの攻撃者によりサーバがクラッシュする。復旧にはアプリケーションの再起動が必要。
- ・ 予防対策：バージョン 2.8.6.0 以降へのアップデート

Qualcomm Qpopper における fgets に関する脆弱性 (CVE-2000-0320)

- ・ 手法：qpopper はメッセージテキストの終端である「`¥n`」文字列の適切な確認を行わない。ユーザのメールボックスからのデータの読み込みに用いられる fgets() あるいはこれに類似する mfgets()は、1024 バイトの固定長バッファにデータを読み込み、「`¥n`」文字か 1023 バイトを読み込んだときに文字列を返す。「`¥n`」が最後につけられた長さ 1023 文字におよぶ文字列が送られた場合、「`¥n`」以後の文字列は新たなメッセージとして扱われる。
- ・ 影響：リモートからの攻撃者はサービス妨害や、メールボックスの破壊、ウイルスチェックを迂回してのメール送付が可能。
- ・ 予防対策：パッチの適用

3. WEB サーバ

以下では、まず一般に Web サーバに必要とされる対策について述べ、次に情報漏洩対策、コンテンツへの攻撃に関する対策、サービス妨害対策、最後に Web サーバを構成する個々のプログラムに関する不正アクセス手法について記述する。

3.1. Web サーバの不正アクセス対策

Web サーバには、データベースとのインタフェース、プロキシー機能、認証機能、パフォーマンス最適化のための機能、API や CGI スクリプトによる拡張性などが与えられている。サーバの実行ファイルのサイズはかなり大きく、処理も複雑であるため、開発時のチェックが十分行われていない製品に未発見のバグが存在する可能性が高い。サーバ管理者から見れば、Web サーバの設定項目は数多く、技術的な詳細の理解と把握が困難である。

インターネットに Web ページ・コンテンツを公開する場合だけでなく、イントラネット、エクストラネットでサーバを利用する場合でも対策は必要となる。

他のサービスをリモートから管理するためのツールに Web サーバ機能が与えられている場合には、それらのサーバに対して Web サーバと同様の注意を払う必要がある。

3.1.1. 不正アクセスの傾向

Web サーバに対する不正アクセスには、以下のような傾向が見られる。

- ・ 侵入：アクセス権の奪取や Web ページの改竄など。CGI プログラムやサーバソフトウェアのセキュリティホールを利用した侵入手法が知られている。
- ・ 機密の漏洩：侵入により重要な情報が攻撃者に奪われる場合、セキュリティホールを用いてサーバ上の情報を参照される場合、管理者の設定ミスや管理の不徹底により重要な情報が公開された状態になる場合などがある。
- ・ サービス妨害：サービス妨害攻撃手法によるサーバの停止あるいは破壊。

Web サーバに関して報告される不正アクセス手法の多くは、Web サーバ自体 (httpd) の脆弱性ではなく、その機能を拡張する要素 (CGI 等) に存在する脆弱性に基づくものである。特に古いバージョンのサーバにデフォルトでインストールされているテストスクリプトやサンプルスクリプトが抱える脆弱性や、商用/非商用の著名なスクリプトに存在する脆弱性が、攻撃の糸口にされやすい。

3.1.2. 対策の方針

(1) 運用ポリシーの明確化

サイトの管理方法、コンテンツ内容 (訪問者に認める権限の範囲、利用可能なサーバサイドアプリケーションの種類)、コンテンツの作成方法、監視とバックアップの体制、緊急時の対応方

法などを明確にしておく。

公開するディレクトリには非公開ファイルを一切置かない。顧客情報等の機密性を要する情報は別のマシン上に置き、安全な手法で必要な情報だけを間接的に参照するような構成を取る。これらの方針を当初から定めて徹底することで機密漏洩を防ぐ効果が期待できる。

(2) コンテンツの復旧方法の確立

可能ならばサーバとは別のマシン上で構成を含めた全てのコンテンツを作成し、これをバックアップしたものをサイトで公開する。攻撃者が改竄を行っても確実にコンテンツを復元できる。小規模なサイトであれば大量のアクセスに高速に読出し可能な記憶装置で対応する必要が無いため、書き込みのみが可能なメディアにコンテンツをバックアップして公開する手法が取れる。このような手法を取ればリモートからの改竄は不可能になる。

(3) 導入時の対策

最新バージョンの Web サーバを利用し、修正プログラムを適用する。修正プログラムおよびセキュリティ関連情報の入手方法を確立しておく

デフォルト設定には不備が見られることが多いため、必要であれば適切な設定に改める。

(4) サービスの限定

Web サーバマシンの機能は Web サービスに限定し、他の不要なサービスは全て停止する。

- ・ 不要なサービスコマンドはマシンから削除する。
- ・ Anonymous FTP サーバと httpd を混在させない。
- ・ コンテンツから呼び出さないシェル・インタプリタは削除する。
- ・ サーバ上にコンパイラは置かない

Web サービスについても不要な機能や危険な機能は全て停止する。

- ・ 利用する CGI プログラムについてはその設定と脆弱性をチェックする
- ・ SSI を無効化する。あるいは exec の include を行う機能を無効化する

(5) 実行権限等の設定

さらに厳しいセキュリティを達成するためには、いくつかの手法がある。

- ・ chroot システムコールを使用し、サーバの起動ディレクトリを見かけ上のルートディレクトリにすることができる。この処置により起動ディレクトリの配下のディレクトリ以外へのアクセスを禁じることができる。
- ・ root 権限でのサーバの起動、停止、再起動を取りやめ、Web サーバ実行の専用アカウントを作成する。

(6) ログイン制限

Web サーバ管理者以外によるシステムへのログインを制限する。サーバ上でのコンテンツ編集作業は行わず、他のマシンで作成したデータを転写して用いる。

(7) 認証・暗号化

情報の発信先を限定するためにユーザ認証によるアクセス制御を利用することが可能である。サーバにアカウントを作成するものではないため、セキュリティ上の問題となる可能性は小さい。

(8) アクセスログからの侵入検知

不正なアクセスを受けた際やエラーが発生した場合に、唯一の情報となるのがログである。特に、攻撃者の特定やサーバに存在する弱点の特定を行う場合は、役立てることが可能な証拠は唯一ログのみとなる。通常サーバに共通する syslog のログは、安全性を高めるためにはネットワーク内部の別のマシンに転送して記録を取ることが望ましい。この他に Web サーバのログとしてはサーバ訪問者のログ (access_log)、エラーログ (error_log) がある。これらのログからセキュリティに関連する重要性の高い部分を抜き出して解析を行うツール (ログ解析ツール) を利用する。

3.2. 情報漏洩に関する対策

Web サーバは外部に情報を提供し、訪問者から情報を収集する性質を持つ。このため、適切な管理が行われていない Web サーバからは本来外部に公開してはならない情報が漏洩することが問題となる。

3.2.1. Web ページからの情報の収集

内部へのアクセスの足がかりとなるような情報がコンテンツのソースに含まれることがある。ソース作成者の所属、連絡先（電子メールアドレス、電話番号）、コメント、ローカルなディレクトリ構造、スクリプトのソース、などの情報を得ることが可能である。

このために攻撃者はオフラインブラウジングツールを利用する。これは自動的にサイト内の公開されたファイルを取得し、サイトのミラーイメージをローカルなマシン上に構築するものである。攻撃者は自らのローカルなマシン上にサイトをコピーし、時間をかけた弱点の解析や検索ツール等による情報の洗い出しを行う。

対策としては、コンテンツに極力そのような情報を含めないようチェックを徹底することが挙げられる。また、攻撃者がこのような情報を得るために一括収集を行うことに備え、機械的なダウンロードをログから監視する。単一の発信元から小刻みかつ集中的に送られてくる GET 要求で判別が可能である。CGI スクリプトには自身へのアクセスを監視し、自動化プログラムへの接続を拒絶するものや、接続に対して無意味なデータを返すものも存在する。

3.2.2. インデックス表示の設定

サイト訪問者に対するディレクトリー一覧の表示を許可すると、検索エンジンのロボットや攻撃者に公表を意図していないファイルの存在の有無を知られる原因ともなる。

公表すべきでない情報の扱い方に関する対策としては、機密性を要するファイルは一切サーバ上に置かないことがあげられる。

システム管理上サーバに置かれるログ等を保護するためには、以下の対策が有効である：

- ・ ディレクトリー一覧表示を禁止する
- ・ 念のために各ディレクトリには index.html ファイルを作成して配置する
- ・ 機密性を要するファイルには攻撃者による想定が困難なファイル名をつける

3.2.3. 重要情報（個人情報）の扱い

Web サイトにおいて訪問者に対して行ったアンケートの結果や、電子商取引サイトの購買記録、入社説明会申し込みのための個人の履歴情報などは、非常に慎重な扱いが要求される。これらは、サーバのセキュリティが破られた際の影響を考え、別のサーバ上に記録することが望ましい。Web サーバではフォーム等に入力された情報をデータベースへ受け渡すための処理のみを行い、サーバ上にデータファイルを作成しない。

3.3. Web アプリケーションの脆弱性に関する対策

CGI スクリプト等のプログラムをサーバ上で実行することで、Web サイト上で多様な機能を提供することができる。反面、不用意に作られたスクリプトは、Web サーバのセキュリティを大きく損なう。このようなプログラムについては、サーバ上で実際に動作させる前に、潜在的な脆弱性を持たないことを厳密にチェックする必要がある。

以下に Web アプリケーションに必要な対策について述べる。対策は主に既知の弱点を有する Web アプリケーションの利用に関する対策と、新たに作成する Web アプリケーションにおける作成時の対策に分けられる。

3.3.1. 既知の弱点を有する Web アプリケーションへの対策

古いバージョンの Web サーバには、放置すれば致命的な脆弱性を有するテストスクリプトが、導入時にデフォルトでインストールされている。

これらの危険なスクリプトが放置されているかどうかを自動検出するツールが数多く存在する。ツールの中にはあるアドレス範囲内の複数サイトをスキャンし脆弱性をチェックするものや、既知の多数の脆弱性を同時にチェックするものもある。

以下のような対策があげられる。

- ・ 自動検出ツールをサイトに適用して脆弱性を発見し、修正する。
- ・ 該当するディレクトリを確認して、不要なスクリプトは全て消去する。
- ・ 脆弱性を有する CGI プログラム (testcgi や phf など) に対して攻撃者がスキャンを試みたかどうかはログで確認する。

3.3.2. 新たに作成された Web アプリケーションへの対策

新規作成された Web アプリケーションは、十分なチェックを行わなければ、プログラム作成者が想定しなかった入力や、本来とは異なる入力元からの入力を受けつけてしまう可能性を持つ。また、きちんと制限しなければ本来意図した以外の機能を持ち得る。出力内容は公開可能な範囲で、渡す必要がある情報に限定しなければ機密が漏れるもととなる。

Web アプリケーション作成時にセキュリティに関して注意すべき事柄を述べる。

- ・ ユーザや他のプログラムからの入力については、可能であれば入力元の認証を行う。
- ・ 入力内容の仮定や盲目的な信頼を避ける。フォームに書かれた固定値等についても改竄されたリクエストを受け取る可能性がある。入力可能な範囲を限定する。入力されるサイズ、入力される値の範囲をチェックする。特殊文字列、スクリプトやシェルコマンド文字列は受けつけない。
- ・ 処理手順のチェック、処理内容のチェックを行う。可能ならばコマンド呼出を伴う関数は用いない。用いる場合は呼出内で有効に機能するような不正な文字列を渡さないようにする。

- ・ 機能、結果の出力、エラーの出力は最小限のものにする。例えば、アクセス禁止、ファイルの非在等の異なる場合によって異なるエラーを出すと、ファイルの存在が露呈するものとなる。全て同一のエラーを出す、あるいは全く黙殺することがより望ましい。
- ・ 処理中に予期せぬ事態が発生した場合は、警告を発して処理を中止し、ログに記録する。

以下に、特に入力に関する対策としてバッファオーバーフロー対策とフォーマットドストリング対策、出力に関する対策として機密漏洩対策をあげ説明する。

3.3.3. バッファオーバーフロー対策

C のようなコンパイルされる言語でプログラムを書く際には、ユーザ入力情報のサイズを仮定したコーディングは避ける。仮定したサイズより大きい入力を受けつけたときにバッファオーバーフローを起こす原因となる。入力の終端まで文字列をコピーする関数 (strcpy() や strcat() など) は使用を避け、代用としてより安全な関数 (strncpy() や strncat() など) を使うか、バッファオーバーフロー対策済みのライブラリの関数を用いる。

3.3.4. フォーマットドストリング (シェルメタキャラクタ) 対策

リモートユーザの入力データを確認せずにシェルコマンドに渡すことは避ける。

入力されたデータをそのままコマンドの引数として用いる場合には、攻撃者が選んだコマンドを実行されてしまう危険性がある。

対策としては、入力バッファの内容を処理に渡すコマンド呼出を可能な限り避ける。コマンド呼出が避けられない場合は、与えるパラメータを厳密に検査するよう作り込む。可能であれば入力を許可する文字列だけを通す手法 (ポジティブチェック) を取る。特殊な記号文字を指定してそれらを削除する手法 (ネガティブチェック) は高い自由度が残されるが、プログラム作成者の想定外の入力によりチェックが迂回される可能性も残る。

3.3.5. 機密漏洩対策

サイトとサーバホストに関する情報をアプリケーションに渡すことはなるべく避ける。不用意にシステム情報を参照する関数を呼び出して表示させるべきではない。特にエラー時の出力からシステムに関する重要な情報が露呈するケースが多く知られている。

URL や Cookie には重要な情報を入れてはいけない。これらは基本的にネットワーク上で公開されている情報と考える。個人情報、パスワード、暗号鍵などを持たせてはいけない。

機密情報を含むファイルは、リモートユーザからのアクセスに制限を課し、推定が困難なファイル名をつける。デフォルトのファイル名 (data.csv など) や機密情報に関わるファイル名をつけてはいけない。

3.3.6. CGI

(Common Gateway Interface) 良く知られた脆弱性を持つ古いサンプルスクリプト (test-cgi、php、Irix 付属の webdist.cgi 等) にある脆弱性については、削除で対応できる。これらのスクリプトや古い教科書を参考に作成した CGI スクリプトは、バッファオーバーフローやシェルメタキャラクタの実行に関する脆弱性を持つ可能性がある。

安全な CGI スクリプトの作成方法については以下のサイトが良い参考になる。

- ・ CERT 勧告
<http://www.cert.org/advisories/CA-1997-25.html>
- ・ W3C による WWW セキュリティ FAQ
<http://www.w3.org/Security/Faq/www-security-faq.html> (原文)
<http://www.w3.org/Security/Faq/001031wwwsfj.ja.sjis.html> (日本語版)

3.3.7. ASP

ASP (Active Server Pages) は Microsoft により提唱され推進されている。コードは VBScript で書かれ、ブラウザでの HTML の表示機能やバックエンドのデータベースへのアクセスに優れる特徴がある。

IIS3.0 および 4.0 における ASP の実装には良く知られたソースコード露呈に関する脆弱性がある。これらは最新のバージョンの IIS では修正されている。

また IIS では適切な設定が施されていないと ASP 実行時のエラーからディレクトリ構造やソースコードが露呈する危険性がある。

Linux 上で動作する ChiliSoft ASP forLinux 3.0/3.5/3.5.2 についても複数の脆弱性が報告されている。修正プログラムを適用する必要がある。

- ・ Bugtraq
<http://www.securityfocus.com/bid/978>
<http://www.securityfocus.com/bid/2454>
<http://www.securityfocus.com/bid/2407>
<http://www.securityfocus.com/bid/2409>
<http://www.securityfocus.com/bid/2410>
<http://www.securityfocus.com/bid/2376>
<http://www.securityfocus.com/bid/2334>

3.3.8. php

PHP () は、HTML ファイル内に記述するタイプのスクリプト言語であり、パーサ php.cgi を通常の CGI として使用することや、モジュールを Apache サーバに組み込むことができる。組み込みにした場合には処理速度の高速化、サーバ負荷の低減をはかることが可能とされる。

CERT 及び W3C からは、複数のバージョンの php.cgi について、決して cgi-bin ディレクト

りに配置しないよう嚴重な勧告が行われている。パーサを実行可能な形でインターネット上に公開すると、Web サーバのホストマシン上で任意のシェルコマンドの実行が可能になる。

PHP に関しては以下のドキュメントが参考になる。

- ・ CERT 勧告
<http://www.cert.org/advisories/CA-1996-11.html>
- ・ W3C による WWW セキュリティ FAQ
<http://www.w3.org/Security/Faq/www-security-faq.html> (英語: 原文)
<http://www.w3.org/Security/Faq/001031wwwsfj.ja.sjis.html> (日本語版)
- ・ PHP マニュアル
<http://www.php.net/manual/en/> (英語: 原文)
<http://www.php.net/manual/ja/> (日本語)

3.3.9. SSI

SSI (Server Side Include) はクライアントに渡す HTML 形式のドキュメントに他のファイルの情報やプログラムの実行結果を挿入することを可能にする技術である。SSI では `exec include` という手法でコマンドや CGI をサーバの権限で実行して結果をドキュメントに反映できる。便利な機能のように思えるが、適切なアクセス制御と処理時の厳密なフィルタリングを行わなければ危険である。攻撃者は SSI が解釈を実行する HTML ドキュメントに SSI コードを挿入し、任意のコマンドを実行可能である。 `exec cmd` 命令によるシステムコマンドの命令や `mail` 命令による機密情報の外部への漏洩が主に狙われる。対策を以下に挙げる。

- ・ SSI を用いる必要が無いならサーバからその機能を取り除く。
- ・ SSI を用いる場合は、厳密なチェックを行ったプログラムによる利用に限定する。利用可能な拡張子を「.shtml」に限定する。実行可能なディレクトリをひとつにまとめ、集中管理する。
- ・ 入力に基づく処理を行う全ての部分について、サーバの解析に渡す前に不正な SSI が含まれる行を削除するようなフィルタを通す。

3.3.10. フォームの返す情報の改ざんによる攻撃

フォームの `hidden` 属性の誤った使い方によりシステムが脆弱になることが知られている。頻繁に変更されるデータについて、サーバ側のプログラムの更新を避け、フォームが `hidden` 属性で返す値を書き変えて置き、ユーザの入力と共にサーバに受け取って処理することは危険である。以下に例をいくつか挙げる。

例 1: 商品の価格を `hidden` 属性で持たせてしまうミス:

```
<INPUT TYPE="hidden" NAME="価格" VALUE="2000">  
<INPUT TYPE="hidden" NAME="商品番号" VALUE="173">
```

上のようなタグを HTML の中に書いた場合に、もし、受け取ったフォームから商品番号 173 番の商品の価格 (VALUE の値) を取り出して何らチェックにかけずに処理に渡してしまうようなコードをサーバ側で実行すれば、攻撃者はフォームを改ざんしたページを用いて好きなように商品の値段を変更して注文できてしまう。

例 2: 管理者のメールアドレスを hidden 属性で持たせる場合:

```
<INPUT TYPE="hidden" NAME="質問連絡先" VALUE="help@target_com.com">
```

上のようなタグを HTML の中に書いた場合、質問連絡先アドレスがそのまま sendmail に渡されるような処理が行われている可能性がある。メールアドレスが渡される際のシェルメタキャラクタに関するチェックが甘かった場合には、サーバ上で任意のコマンドを実行される危険性がある。また、メールアドレスが正しい宛先であるかは確認不可能である。フォームへの入力以外の機密情報が送信メールに含まれる場合は情報の漏洩に繋がる。

この種の欠陥は非常に良く見られる。対策として、改ざんを受けた場合に問題が起きる情報は hidden 属性タグに入れて扱わないことが必要である。一貫性を必要とするデータはクライアント側に渡さずサーバマシン (よりセキュアに構築するなら別のホストマシンのデータベース) に全て置き、参照は HTML のフォームからではなく、サーバでの実行ファイルから行う。

3.3.11. 参照可能なファイルへのタグの追加

サーバやクライアントに解析されるファイルにユーザが書き込み可能な機能は潜在的な弱点となる。掲示板プログラムなど対話的な情報共有において動的に生成されるページデータには、悪意あるコードが書き込まれる可能性があるため注意が必要である。以下にいくつかの例をあげる。

- ・ 攻撃者はユーザ名とパスワード入力を求める JavaScript コードを書き込み、クライアントのブラウザやメーラーにウィンドウを開かせ、訪問者を騙し入力させることが可能である。
- ・ SSI の exec 機能 (前述) が有効なサーバにおいて、攻撃者は不正な SSI 行の書き込み、サーバ上で任意の命令を実行することが可能である。

対策としては、サーバ側では、ユーザより入力された内容を厳密にチェックし、サーバ側の機能を制限すること、共有された情報の内容をこまめに管理することが挙げられる。クライアント側では実行機能を制限し、警告を発するような設定を施し、確認を怠らないことが挙げられる。

これらの対策を取っても、サーバやクライアントプログラムの脆弱性を突く攻撃と組合せた周到な攻撃が計画される可能性がある。脆弱性と修正プログラムに関する情報に注意を払う必要がある。

3.4. サービス妨害攻撃への対策

サービス妨害（DoS：Denial of Service）攻撃は主に Web サーバに対して行われる。以下にその概念、既知の主要な攻撃手法、対策方法について述べる。

3.4.1. DoS 攻撃

コンピュータやネットワークにより提供されるサービスを妨害する攻撃全般を指す。

最も一般的なサービス妨害攻撃は、コンピュータやネットワークの帯域幅や接続性を狙う。帯域幅攻撃は、ネットワークに大量のデータを送り込み、利用可能なネットワーク資源を全て消費し、正規ユーザのリクエスト処理を困難なものにする。接続性攻撃は、ネットワークに大量の接続リクエストを送りこみ、利用可能な OS 資源を全て消費し、コンピュータによる正規ユーザの要求処理を困難なものにする。

特に、インターネットサーバにより提供されるサービス（WWW、FTP、DNS、SMTP によるメール等）を標的として妨害する攻撃が、一般に入手可能なツールを利用して行われている。このようなサービス妨害攻撃には、大きく 2 つの種類がある：

- ・ インターネットプロトコルの特性を悪用し、ネットワークに接続されたコンピュータに過剰な負荷をかけ、サービスの提供を困難あるいはほぼ不可能にする攻撃
- ・ サーバアプリケーションに存在する既存の脆弱性を突き、サービスやホストの停止を導いてサービス提供するを不能にする攻撃

具体的には以下に示すような様々な手法がある：

(1) SYN flood :

- ・ 解説：プロトコルスタックレベルの攻撃手法。攻撃者は送信元を偽造した SYN パケット（接続要求）を連続して送信し、標的となったサーバは届くことのない SYN-ACK パケットを一定時間待つ。このハーフオープン接続はいずれタイムアウトになるが、一定数以上の接続要求を連続して送り続けることで、サーバは正当な接続要求を受け付けなくなりサービス不能状態に陥る。
- ・ 対策：ベンダの提供する修正を適用して SYN 攻撃を検出/回避することが可能である。Linux カーネル 2.0.30 以降では SYN Cookie という暗号化チャレンジレスポンスプロトコルを採用し、リソースを消費しないような対策を採っている。

(2) Ping of Death

- ・ 解説：プロトコルスタックレベルの古典的攻撃手法。プロトコルスタックの実装バグへの攻撃。規定のサイズ（65536 バイト未満）を超える大きさの ping パケットを送り、対応できないコンピュータやルータをダウンさせる。
- ・ 対策：ほぼ全ての OS の現バージョンにおいて対策が取られている。1996 年以前の

OS は影響を受ける可能性があるので、修正プログラムを適用するか、バージョンアップが必要。

(3) e-mail bombing

- ・ 解説：メールボム。アプリケーションレベルの攻撃手法。処理可能な範囲を超えたメールを送りつけて、メールサーバの資源を消費させる。

(4) WinNuke

- ・ 解説：プロトコル実装上の脆弱性を突いた古典的なサービス妨害攻撃手法。標的の Win9x クライアント（あるいは Win NT SP2 以前）の NETBIOS（TCP ポート 139 番）に OOB/URG（Out of Bounds/）のデータを送信し、マシンをダウンさせる。旧バージョンの Windows では OOB データに対する適切な処理を行うような実装が行われていなかったため脆弱性となっていた。
- ・ 対策：新たなバージョンの Window では修正が施されている。古いバージョンの Windows には修正プログラムを適用するか、バージョンを更新する。

3.4.2. DDoS 攻撃

分散型サービス妨害（DDoS：Distributed Denial of Service）攻撃とは、標的コンピュータに対し多数のコンピュータから組織的なサービス妨害攻撃を仕掛ける行為を指す。DDoS 攻撃には、ネットワーク接続されたコンピュータに過剰負荷をかけてサービス提供を阻むタイプのサービス妨害攻撃手法が応用される。攻撃者は数百台、数千台のコンピュータを踏み台に用いて、その資源を無断借用して標的に攻撃を仕掛ける。クライアント/サーバテクノロジーを応用した組織的な攻撃により、1 つあるいは複数の標的コンピュータに対する負荷を増幅し、サービス不能状態に陥れる。

Web サイトに対して行われる分散型サービス妨害攻撃は、対象の Web サイトに何十万件ものリクエストが同時に送信されるものである。攻撃が実行されている間は、一般のサイト訪問者が正規のページリクエストを行っても、要求はすべて失敗するか、ページのダウンロードに大変な時間を要するようになる。

DDoS 攻撃の場合は、実際に攻撃を行うサーバが攻撃の発信元であるとは限らない。DDoS 攻撃に用いられるマシンには、攻撃実行用プログラム（エージェントプログラム）を仕掛けられた複数のサーバやクライアントマシン（「歩兵」や「ゾンビ」などと呼ばれる）と、ネットワークを利用して遠隔から同時攻撃の指令を出す管理プログラム（マスタープログラム）を仕掛けられたサーバがある。

エージェントプログラムはトロイの木馬型プログラムである。サーバだけではなく、インターネットクライアント（PC）にコンピュータウイルスが感染する経路を通してエージェントプロ

グラムが仕掛けられ、攻撃の踏み台にされることもある。常時接続された PC はサーバ機に比べて管理が甘いことが多く、攻撃者にとって便利な存在である。

マスタープログラムは、侵入や盗難によって得た不正なアカウントを使用して、コンピュータ上にインストールされる。マスタープログラムは、指定された時間にインターネット上の不特定のコンピュータにインストールされたエージェントプログラムと通信する。マスタープログラムは、数百、数千のエージェントプログラムに攻撃を開始させることが可能である。

この複雑な構成により、複数の第三者の管理するサーバが攻撃に加担させられる。発信源の追跡は、多段の構成の末端から侵入を受けたサーバを逆に辿って記録を調べる必要があり、攻撃に偽造 IP アドレスが用いられるため、困難である。

攻撃者は、踏み台として非営利のコンピュータを最も多く利用する。一般的に非営利のコンピュータへの侵入ほど容易であることが知られている。大学のシステムは格好の踏み台となっている。大学はしばしば人員不足であり、そのシステムは、学生が学習に利用できるようにセキュリティレベルが最小に設定されている。

DDoS 攻撃の範囲は地域的に限定されない。世界中のすべてのインターネットサーバが攻撃の踏み台に使用される可能性がある。

3.4.3. サーバにおける DDoS 攻撃対策

分散型サービス妨害攻撃に対する素早く簡単な防御方法は存在しない。単純かつ最善の解決法は、コンピュータへの侵入や、攻撃の踏み台への悪用を防ぐことである。サーバ管理者、PC 利用者は、トロイの木馬プログラムを仕掛けられて攻撃に加担することが無いようにするために、日頃から対策を取る必要がある。

ここではサーバが DDoS 攻撃に加担することを防止するための対策について述べる。

(1) システムスキャンによる発見

システムスキャンを行う。自分のインターネット・コンピュータが知らないうちに DDoS 攻撃の踏み台として使用されていないことを確認する。

サーバのファイルシステム上に、既知の DDoS ツール（エージェントプログラムやマスタープログラム）が存在する可能性がある場合は、ファイルシステムスキャンツールを入手して、DDoS ツールの存在を割り出す。できれば複数の異なるツールによる検出が望ましい。セキュリティツールベンダから入手した複数の検出ツールを適用してその結果を比較する。コンピュータウイルスと同様に、新たな DDoS 攻撃手法が開発されたり、既存の DDoS 攻撃ツールが改良されたりすると、これまでの検出ツールは古くなり効果が無くなる。最新の DDoS 攻撃手法に対処するためには最新のツールを用いる必要がある。

既知のエージェントプログラム（トロイの木馬プログラム）については、PC 上での検出に、市販のコンピュータウイルス対策ソフトウェア等のスキャンも有効である。

(2) エージェント発見時の対策

(ア) 侵害されたサーバの状態の保存

サーバの状態を完全に保存し、後の解析のための資料を残す。

(イ) サーバの再構築

サーバにエージェントプログラムが発見された場合は、そのサーバの管理権を攻撃者に完全に掌握されている可能性が高い。システムのオリジナルメディアからの再インストールを含む再構築を検討する必要がある。

(ウ) セキュリティホールの修正

セキュリティホールや設定の不備によりトラフィック量を増大させる攻撃が知られている。管理者は該当するセキュリティホールには修正プログラムやバージョンアップで対処し、設定に不備が無いことを確認する。

(3) サーバのセキュリティの強化

(ア) システムファイルの暗号化チェックサムを作成

自らのサイトのコンピュータ(PC、サーバ)に、意識的にインストールしたソフトウェア以外のソフトウェアがインストールされていないことを検証する手段を講じ、無自覚の内にエージェントをインストールされないようにする。日頃から本来のシステム管理状態の把握に努め、本来のシステム状態からの差異を識別するためのツールを用いることが有効である。

(イ) 境界フィルタリングの実装

境界フィルタリングを実装する。

3.4.4. 攻撃手法別 DDoS 攻撃対策

以下に DDoS 攻撃手法を挙げ、その対策手法を述べる。

(1) smurf

ICMP flood 攻撃とも呼ばれる。IP サブネット・ブロードキャスト複製機能を利用する単純だが効果的な DDoS 攻撃手法。smurf のモデルは攻撃ツール、増幅用ネットワーク、標的の三者からなる。一般に攻撃ツールは侵入されたマシンにインストールされる。増幅用ネットワークとしては管理の甘いルータが選ばれる。

攻撃ツールは標的のソース IP アドレスを持つ ping (ICMP echo リクエスト) パケットを偽造し、増幅用ネットワークの IP ブロードキャストアドレスに向けて連続的に送信する。増幅用ネットワークの全てのマシンがなりすまされた標的のアドレスに応答を返す。偽造パケットを送られた増幅用ネットワークは意図せずに攻撃に加担してしまうことになる。

(2) fraggle

UDP flood 攻撃とも呼ばれる。攻撃者は送信元 IP アドレスを偽造した UDP パケットを、第三者である増幅用ネットワークのブロードキャストアドレスのポート 7 (echo) に送りつける。増幅用ネットワークが UDP サービスのエコーを有効にしていれば標的アドレスへ大量のパケットを返信する。ICMP の代わりに UDP を用いた smurf 攻撃と言える。

踏み台とならないための対策：

- ・ 境界ルータのディレクテッドブロードキャスト機能を無効にする。さらに OS で可能ならば、ブロードキャスト ICMP echo パケットを廃棄するよう構成する。

(3) trinoo/Trin00

エージェントプログラムとマスタープログラムからなる DDoS 攻撃ツール。単数あるいは複数の標的に対して UDP パケットでネットワークを氾濫させる UDP flood 攻撃手法を複数のコンピュータを利用して組織的に実行するものである。

trinoo のエージェントは標的のコンピュータに大量の UDP パケットを配信する。標的にされたコンピュータは、届いた UDP パケット 1 つ 1 つに対し、ICMP port unreachable メッセージを生成するためネットワーク帯域幅を消費し、サービス不能状態となる。

trinoo のマスタープログラムは複数のエージェントの IP アドレスを管理し、攻撃者による遠隔操作を可能にする。マスターとエージェントの間の通信は UDP パケットが用いられる。

初期の trinoo は、Solaris、Linux のような UNIX 環境を対象にプログラムされたものであったが、その後 Windows マシン向けにコンパイルされたエージェント WinTrin00 が作られている。trinoo が登場した 1999 年末当時の UNIX 環境への侵入は RPC サービスに存在する脆弱性をついた攻撃により行われていた。

標的となった場合の対策：

- ・ trinoo による攻撃を受けるとシステムは UDP パケットで溢れる。同じソース IP アドレスで、同じ宛先 IP アドレス、同じソースポートだが、異なる宛先ポートの、複数の UDP パケットを探すことで trinoo による攻撃を見極められる。

踏み台にならないための対策：

- ・ trinoo を検出し全滅させるための自動化プログラムは以下で入手できる。

<http://www.fbi.gov/nipc/trinoo.htm>

- ・ エージェントとマスターの間の通信は全て UDP (タイプ 17) で行われる。また、マスターへの攻撃者の接続はポート 27655 への TCP (タイプ 6) telnet 接続が多用される。これらの特徴から侵入検知システムやログ解析を通じて通信を検出可能。この他にも、多くの痕跡からマスタープログラムとエージェントプログラムの存在

を確認することが可能である。

(4) TFN

TFN (Tribal Flood Network) は trinoo と同様のマスター/エージェントを用いた DDoS 攻撃ツールである。TFN は UDP flood 攻撃、ICMP flood 攻撃、ICMP ブロードキャスト攻撃、古典的な SYN flood 攻撃など、複数の種類の攻撃が実行できる。また、TFN はソース IP を偽造したパケットを生成する能力を持つ。

標的となった場合の対策：

- ・ SYN flood 攻撃に関しては DoS 攻撃の場合と同様の対策が可能である。
- ・ UDP flood 攻撃の特徴は trinoo による攻撃と同様である。
- ・ ICMP ブロードキャスト攻撃に関しては smurf 攻撃の場合と同様の対策が可能。
- ・ ICMP flood 攻撃については、ICMP echo パケットおよび ICMP echo-reply パケットを一切通さないようルータを設定すれば防御が可能となる。しかしルータを通して ping などの他のインターネットプログラムを使用できなくなる。

踏み台にならないための対策：

- ・ 以下に示すような TFN のシグニチャに基づいて検出を行うシステムスキャンツールを各セキュリティベンダが開発している。クライアントにおいては最新の定義ファイルを持つウイルススキャンツールが有効である。
- ・ TFN マスタープログラムとエージェントプログラム間の通信は、ICMP echo-reply パケットを利用する。実際の指示はバイナリ形式で 16 ビットの ID フィールドに組み込まれる。この特徴に基づく ICMP に対するフィルタが適用できる。
- ・ TFN マスタープログラムは、エージェントプログラムの位置を含む IP アドレスリストを読み込んで動作する。このアドレスのリストは、Blowfish 暗号で暗号化されていることがある。暗号化されていない場合は平文で書かれたリストの情報を元にエージェントを容易に識別可能である。
- ・ 標準的な TFN エージェントプログラムはファイル名「td」で、マスタープログラムは「tfn」という名前で、システム上に存在することが知られている。
- ・ TFN エージェントは、その動作にあたり ICMP echo-reply パケットの発信元を確認しない。このため内部ネットワーク内に TFN が存在するかどうか疑わしい場合は適切な ICMP パケットを作成して送信すれば、隠されているプロセスを明らかにすることが可能である。

(5) TFN2K

TFN2K は TFN の「改良」バージョンの攻撃ツールである。TFN2K は以下の点で TFN と異なる。

- ・ マスターとエージェント間の通信に TCP、UDP、ICMP などの複数のプロトコルから 1 つを利用する。このためプロトコルのフィルタリングが困難になっている。
- ・ システムをクラッシュさせるパケットや、不安定にするパケットを送信する能力を持つ
- ・ 内部ネットワークのマシンから送信されたように見える IP アドレスを持つパケットを偽造できる。このため出口フィルタリングが無効化される。

踏み台にならないための対策：

- ・ 境界ルータで出口フィルタリングを設定する。
- ・ 上流プロバイダに入口フィルタリングを配置するよう依頼する。

(6) Stacheldraht

Stacheldraht (「有刺鉄線」というドイツ語) もまたプログラムが何千ものエージェントプログラムと通信する TFN や trinoo 同様のクライアント/サーバモデルに基いている。Stacheldraht には、攻撃者とマスタープログラム間の通信の暗号化機能、rcp (リモートコピー) の活用によるエージェントプログラムの自動更新機能が追加されている。

Stacheldraht による DoS 攻撃は、複数の攻撃パターンがあり、またソース IP アドレススプーフィングが実行可能なため、防御は極めて難しくなる。Stacheldraht による攻撃には、UDP flood、TCP SYN flood、ICMP echo request flood、ICMP directed broadcast などがある。

標的となった場合の対策：

- ・ Stacheldraht マスタープログラムとエージェントプログラム間の通信は、主として ICMP echo と echo-reply を用いて行われる。内部ネットワーク上の ICMP echo と echo-reply パケットの送受を拒むようルータを構成することで、Stacheldraht エージェントからの攻撃を防御可能である。しかしこれを行うと ping などのインターネットプログラムが使えなくなる。

踏み台にならないための対策：

- ・ エージェントプログラムは、有効なマスタープログラムの IP アドレスを含むリストを読み込む。リストは Blowfish 暗号により暗号化されている。エージェントはリスト上の各マスタープログラムと通信を試み、これに成功するとインストールされているシステムでパケットソースアドレスのスプーフィングの可否を判断するテストを行う。これら 2 つの動作は、侵入検出システム、またはシグニチャから sniffer で検出可能である。
- ・ エージェントは、各マスターに、ID フィールドに値 666、データフィールドに文字列 skillz を含む ICMP echo-reply パケットを送信する。マスターがパケットを受信すると、値 667 を含む ID フィールドと、文字列 ficken を含むデータフィールドを

返信する。エージェントとマスターは、定期的にこれらのパケットを交換して通信する。これらのパケットを監視すれば Stacheldraht を検出可能である。

- エージェントが前述したスプーフィングの可否のテストを試みる時には、偽の発信元アドレス「3.3.3.3」を使用する。マスターはこの偽造パケットを受信すると、なりすましがうまくいっていることを知らせるために ICMP パケットデータフィールド中に文字列「spoofoorks」を入れてリプライする。Stacheldraht は、これらの値を監視すれば検出可能である。
- エージェントは、ICMP echo-reply パケットの発信元を確認しない。このため、適切な ICMP パケットを送りつけてやれば Stacheldraht プロセスを暴き出せる。
- David Dittrich 氏によって書かれたエージェント検出プログラム (C 言語)
http://staff.washington.edu/dittrich/misc/ddos_scan.tar

3.5. IIS

3.5.1. IIS 4.0 の設定

IIS 4.0 (Windows NT4.0 Server) について、その設定上の注意を述べる。

- ・ 必要最低限のインターネットサービスをインストールする
- ・ 適切な認証メソッドを設定する
- ・ 適切な仮想ディレクトリ権限/Web アプリケーション領域を設定する
- ・ 実行可能コンテンツの信頼性を確認する
- ・ IP アドレス/DNS アドレスの制約事項を設定する
- ・ 新しいルート証明を IIS に移行する
- ・ SSL (Secure Sockets Layer) を設定する
- ・ ログの記録を有効にする
- ・ Index Server がドキュメンテーションのみにインデックスを作成するようにする
- ・ Microsoft Certificate Server の Web サーバ登録用 ASP ページをロックする
- ・ IISADMPWD 仮想ディレクトリを削除する
- ・ RDS サポートを無効にする
- ・ すべてのサンプルアプリケーションを無効にする、または削除する
- ・ <FORM>入力をチェックする
- ・ 適切な仮想 IIS ログファイル ACL を設定する
- ・ SSI の#exec によるコマンドシェル呼び出しを無効にする
- ・ 親パス (「..」によるパス指定) を無効にする
- ・ 使われていないスクリプトマッピングを削除する

3.5.2. IIS 5.0 の設定

IIS 5.0 (Windows2000 Server および Windows2000 Advanced Server) について、その導入時の設定上の注意を述べる。

(1) OS サービスパックの適用

- ・ Windows 2000 Service Pack 1 を必ず適用する

(2) セキュリティ修正プログラムの適用

- ・ 「MS00-086 『Web サーバによるファイル要求の解析』に対する対策」を適用する。

(3) MDAC の RDS 機能の抑止とサンプルアプリケーションの無効化・削除

- ・ RDS 機能は通常は必要無いので停止する。MSADC 仮想ディレクトリとその内容を

削除し、不要なレジストリキーも削除する。

- ・ IISAdmin フォルダ、Scripts フォルダ以外の仮想フォルダは不要。
- ・ 以下のフォルダを削除する。
 - IISHelp
 - IISSamples
 - MSADC
 - Printers
 - _vti_bin

(4) 不要なサービスの削除

- ・ 最低限必要なサービスは WWW サーバ、インターネットインフォメーションスナップイン、共通コンポーネントの 3 つのみ。
- ・ 「インターネットインフォメーションサービス」から操作を行う

(5) Administrator アカウント名の変更

- ・ Administrator アカウント名を変更することで攻撃者による管理者アカウントの特定を困難にすることができる。

(6) ユーザアカウントのパスワードの再設定

- ・ 9 文字以上のパスワードをつける。パスワードのハッシュ作成方法にデザイン上の特性があるため、8 文字以下のパスワードに比べて飛躍的に暗号的な強度を向上させることができる。
- ・ 先頭 7 文字には英字以外の記号を含める
- ・ 書き込みアクセスが可能な全てのユーザのパスワードについて再設定を行う。

(7) セキュリティテンプレート Hisecweb.inf の適用

- ・ テンプレートは Microsoft サイトからダウンロードする。

(8) IPSec ポリシーの構成

- ・ パケットフィルタリングを行う
- ・ 利用しないプロトコルやポートは全てブロックする
- ・ IPSec 管理ツールあるいはリソースキットの IPsecPol コマンドラインツールを使う

(9) アクセスコントロールリスト (ACL) の設定

- (10) ログ記録の有効化
- ・ W3C Extended ログ形式を使用する。
- (11) 使用しないスクリプトファイルタイプ拡張子へのマッピングの削除
- ・ 利用しないマッピングは必要無いので全て削除する。
 - ・ マッピングについては「ファイルの存在を確認する」よう変更する。パフォーマンスは低下するがより安全性は高まる。
- (12) ディレクトリトラバース対策
- ・ 「アプリケーションの構成」「アプリケーションのオプション」から「親のパスを有効にする」チェックボックスを外し、「..」(dot dot)を使ったパス指定が行えないよう設定する。
 - ・ デフォルトでは公開ディレクトリ Inetpub がシステムと同ドライブにインストールされる。システムとは別ドライブにドキュメントを置くよう設定を変更する。

3.5.3. 脆弱性対策

2000 年に報告された IIS の脆弱性とその対策方法について以下に説明する。

(1) IIS 4.0 の脆弱性

Hit-Highlighting 引数の形式不良に関する脆弱性 (CVE-2000-0097)

- ・ 手法：hit-highlighting (あるいは WebHits) 機能を実装する ISAPI アプリケーションは要求できるファイルを制限しない。故意に形式不良にした引数を用いてドキュメントを hit-highlight する要求することで、仮想ディレクトリを超えたアクセスが可能となる。
- ・ 影響：リモートからの攻撃者は Web サーバ上のファイルをアクセス権にかかわらず取得可能。
- ・ 予防対策：修正プログラムの適用 (MS00-006)

Internet Data Query ファイル要求に対するエラーメッセージの脆弱性 (CVE-2000-0098)

- ・ 手法：存在しない Internet Data Query ファイルをユーザが要求した場合に返されるエラーメッセージに、要求に含まれていた Web ディレクトリに対する物理的パスが示される。
- ・ 影響：リモートからの攻撃者は Web サーバのファイル構造に関する情報を取得可能。
- ・ 予防対策：修正プログラムの適用 (MS00-006)

チャンクエンコーディングされたポストに関する脆弱性 (CVE-2000-0226)

- ・ 手法：予約可能なバッファサイズに制限が無いため、POST または PUT 操作に対する極端に容量の大きいバッファを要求し、実際にはデータを送信せずにサーバ上のメモリを確保することが可能。
- ・ 影響：リモートからの攻撃者は Web サーバに対してサービス妨害攻撃をかけることが可能。
- ・ 予防対策：修正プログラムの適用 (MS00-018)

Link View サーバ側コンポーネントに関する脆弱性 (CVE-2000-0260)

- ・ 手法：Dvwssr.dll に未チェックのバッファが含まれる。不正な入力によりバッファオーバーフローを起こし、サーバをクラッシュさせることが可能。フォルダ上のアクセス権が不適切に設定されている場合や、.dll がアクセス権の程度が低いフォルダにコピーされている場合は、悪意あるユーザがコンポーネントを実行するおそれ

がある。

- ・ 影響：リモートからの攻撃者はサービス妨害や、サーバ上でシステムコンテキストで任意のコードを実行することが可能。
- ・ 予防対策：Dvwssr.dll を削除するか無効にする。修正プログラムの適用（MS00-025）

無効な URL リクエストによるサービス妨害に対する脆弱性（CVE-2000-0858）

- ・ 手法：IIS に一連の改竄されたリクエストを送ることで INETINFO.EXE により無効なメモリ要求が行われる。この結果全てのシステムリソースを使い果たし NT によって IIS が停止させられる。
- ・ 影響：リモートからの攻撃者はサービス妨害を試みることが可能。
- ・ 予防対策：ベンダの提供する修正プログラムの適用（MS00-063）。IIS を使用していなくても NT4.0 には適用が必要。SP6a 以降に修正内容は含まれる。

拡張 UNICODE を用いたディレクトリ横断に関する脆弱性（CVE-2000-0884）

- ・ 手法：拡張 UNICODE 文字で「/」および「¥」を入力すれば、「../」（dot dot）ディレクトリ横断（directory traversal）手法を使って IUSR_machinename 権限で参照可能な既知のファイルへのアクセス権を得ることができる。
- ・ 影響：リモートからの攻撃者は IUSR_machinename 権限でサーバ上の文書の取得、変更、削除が可能。また同権限で任意の命令の実行が可能。
- ・ ベンダの提供する修正プログラムの適用（MS00-078）。NT 4.0 は SP5 を先に適用する必要がある。あるいは IUSR_machinename アカウントにデフォルトで与えられている Everyone および User 権限を制限する。

（ 2 ） IIS 4.0 および IIS 5.0 の脆弱性

仮想化された UNC シェアに関する脆弱性（CVE-2000-0246）

- ・ 手法：IIS サーバ上の仮想ディレクトリが UNC シェアにマップされているときに、URL 要求の最後に特定の文字が含まれる場合、ISAPI 拡張子の処理が適切に行われず、UNC にマップされた仮想ディレクトリがサーバ側のスクリプトコード(ソースコード)を返す。
- ・ 影響：リモートからの攻撃者は Web サーバ上の ASP ソースコードおよび他のファイルを読むことが可能。
- ・ 予防対策：修正プログラムの適用（MS00-019）

スクリプトを持つ場合は修正プログラムを適用する (MS00-044)

正規化エラーによるファイルへの誤ったアクセス権の適用に関する脆弱性 (CVE-2000-0770)

- ・ 手法：CGI スクリプトと ISAPI 拡張を使って実装される形式のファイルにのみ影響する正規化エラーにより、Web サーバ上のファイルに対して特定の方法で改竄された URL を用いて要求を行うと不正なアクセス権が適用される。アクセス権の範囲は親フォルダかそれよりも上位のフォルダで設定されている範囲に限定される。Web フォルダ構造とサーバ上の物理フォルダ構造がミラーリングされていることが前提。
- ・ 影響：リモートからの攻撃者は、本来権限のないファイルの読み取りや実行が可能。
- ・ 予防対策：修正プログラムの適用 (MS00-057)

ファイル要求の解析に関する脆弱性 (CVE-2000-0886)

- ・ 手法：実行可能ファイル (.bat あるいは .cmd) のファイル名に OS コマンドを付加したリクエストを送信すると cmd.exe にそのまま渡され、ファイル処理後に付加された OS コマンドが IUSR_machinename 権限で実行される。
- ・ 影響：リモートからの攻撃者は IUSR_machinename 権限で任意の命令の実行、権限の向上、ファイルの取得/削除/追加/変更が可能である。攻撃者に通常の対話的ログオンユーザと同等の権利を与える。攻撃者は権限の向上を謀るための情報を収集することが可能。
- ・ 予防対策：ベンダの提供する修正プログラムの適用 (MS00-086)。2000年11月30日に IIS5.0 用の修正プログラムが更新されたため、最新の修正プログラムを再適用する必要がある。(NT 4.0 は SP5 を先に適用すること)(Windows 2000 は SP1 を先に適用すること。この修正は SP2 に含まれる)。日本語版修正プログラムが無い場合...bat および .cmd ファイルを削除あるいは Web からアクセスできないフォルダに移動し、IUSR_machinename アカウントによる cmd.exe へのアクセスを禁止する。

セッション ID クッキーの露呈に関する脆弱性 (CVE-2000-0970)

- ・ 手法：セキュアな Web セッションで発行されるセッション ID クッキーは SSL で保護されて送受されるが、その後ユーザがセキュアでない Web セッションに移動すると、全く同じセッション ID のクッキーがプレーンテキストで送られる。
- ・ 影響：リモートからの攻撃者は、対象ユーザと Web サーバ間の通信に関するコントロールが可能な場合に、セキュアな Web セッションを乗っ取ることが可能。
- ・ 予防対策：ベンダの提供する修正プログラムの適用 (MS00-080)

電話帳サービスにおけるバッファオーバーフローに関する脆弱性 (CVE-2000-1089)

- ・ 手法：電話帳サービスをインストールした IIS において、電話帳の更新要求を処理する URL 処理ルーチン部分のコードに適切なチェックが行われていないバッファが存在する。
- ・ 影響：ローカルなユーザはサービスの持つ権限 (IUSR_machinename あるいは IWAM_machinename) で任意の命令を実行可能。
- ・ 予防対策：ベンダの修正プログラムの適用 (MS00-094)

Front Page Server Extension におけるサービス不能攻撃に対する脆弱性 (CAN-2001-0096)

- ・ 手法：デフォルトでインストールされている FrontPage Server Extensions (FPSE) に脆弱性が存在する。FPSE が提供する browse-time 機能に対して不正な Web フォームを提示すると、IIS サービスが異常終了する。
- ・ 影響：リモートからの攻撃者はサービス妨害攻撃が可能。サーバの通常動作の再開にはサービスの再起動が必要 (IIS 5.0 は自動的に再起動がかかる)。
- ・ 予防対策：ベンダの提供する修正プログラムの適用 (MS00-100)。必要でなければ FPSE を無効にしておく。

(3) IIS 5.0 の脆弱性

「Translate:f」でのソースコードの露呈に関する脆弱性 (CVE-2000-0778)

- ・ 手法：HTTP GET リクエストに「Translate: f」ヘッダを付加して送ると、スクリプトエンジンが要求されたファイルに関して処理が必要であるかどうかを認識せずにソースコードをクライアントに返す。
- ・ 影響：リモートからの攻撃者は ASP ファイルや他のスクリプトのソースコードを取得可能。
- ・ 予防対策：ベンダの提供する修正プログラムの適用 (MS00-058)

3.6. Apache

Apache は UNIX 環境でフリー/商用を問わず広く使われている Web サーバソフトウェアである。Linux には標準の Web サーバとしてパッケージに含まれている。以下に Apache に関するセキュリティ上の対策について述べる。

3.6.1. Apache の設定

Apache の設定について重要な項目を以下に挙げる。

(1) インストール時

- ・ 最新のバージョンの Apache を入手して用いる。
- ・ Web 管理用のユーザアカウントを作成する：
Web サービスに無関係なシステムに対して管理作業時に誤った操作を施す危険を避けられる。
- ・ インデックス表示の禁止：
アクセス制御ディレクティブの<option>から "Indexes" を削除する

(2) CGI

- ・ httpd.conf で CGI を実行するディレクトリを指定することができる。このディレクトリのアクセス権限の設定には注意すること。制限が適切でなければ意図しないプログラムのインストール、ソースコードやデータファイルの露呈といった危険が予想される。
- ・ ユーザディレクトリでの CGI の実行を許可すると、管理者が把握できない危険性がサーバ上に存在することになる。サーバ全体のセキュリティ上の弱点となる点には注意が必要である。
- ・ 外部から入手した CGI プログラムには、悪意あるコードが含まれている危険性や、脆弱性が存在する危険性がある。十分なコードのチェックを行うか、信頼のおける配布元からの供給を受けたコードを用いる必要がある。

(3) SSI

- ・ CGI に比べより大きな危険となる可能性がある。
- ・ SSI を利用しない場合は利用できないような設定を必ず行う。
- ・ SSI を実行するディレクトリは完全に把握する必要がある。
- ・ SSI を実行するファイルの拡張子は shtml に限定する。意図しない SSI の使用を防ぐ効果がある。
- ・ SSI を使う場合でも exec 関連の include は禁止する。これを許すと攻撃者がフォー

ムに書き込んだ SSI 行が SSI 解析前に適切に削除されなければ任意のコマンドを実行される危険がある。

3.6.2. Apache に関する脆弱性と影響を受けるバージョン

以下に Apache に関する脆弱性と、その脆弱性の影響を受けるバージョンを示す。

脆弱性の影響を受けない新しいバージョンの Apache プログラムを用いることが、これらの脆弱性に基づく攻撃への予防対策となる。

ディレクトリ内要一覧表示に関する脆弱性

- ・ 膨大な数のスラッシュを使った長いパス名のリクエストを送ることでディレクトリの内容一覧を取得可能。リモートからの攻撃者は情報収集が可能。
- ・ 対象バージョン：Apache 1.3.19 以前
- ・ 参考：BID:2503

PHP3 のファイル露呈に関する脆弱性

- ・ PHP が動作する Apache1.3 において、ディレクトリ横断の手法を用いたリクエストにより Web サーバのルート配下に置かれたファイルを取得できる。リモートからの攻撃者は既知のファイル名を持つファイルを取得可能。
- ・ 対象バージョン：Apache 1.3.6
- ・ 参考：CAN-2001-0042、BID:2060、XF:apache-php-disclose-files(5659)

Rewrite モジュールのファイル露呈に関する脆弱性

- ・ Apache 1.2 以降に含まれる mod_rewrite モジュールの脆弱性。正規表現のファイルネームを含むように RewriteRule ディレクティブが拡張されている場合に、リモートからの攻撃者はホスト上の任意のファイルを取得することが可能。
- ・ 対象バージョン：Apache 0.8.11、Apache 0.8.14、Apache 1.0、Apache 1.0.2、Apache 1.0.3、Apache 1.0.5、Apache 1.1、Apache 1.1.1、Apache 1.3.11win32、Apache 1.3.12
- ・ 参考：BID:1728、CVE-2000-0913、XF:apache-rewrite-view-files(5310)

SuSE Apache WebDAV のディレクトリ一覧表示に関する脆弱性

- ・ デフォルトの設定では WebDAV が有効になっている。PROPFIND HTTP リクエストを送ることで、リモートからの攻撃者は任意のディレクトリの一覧を取得可能。
- ・ 対象バージョン：Apache 1.3.12
- ・ 参考：BID:1656、CVE-2000-0869

Windows 版 Apache におけるルートディレクトリアクセスに関する脆弱性

- ・ リモートからの攻撃者はディレクトリの一覧を取得可能。ディレクトリの一覧表示が config で許可されていて、index ファイルが存在するときでも一覧が表示される。
- ・ 対象バージョン : IBM HTTP Server 1.3.3 win32、IBM HTTP Server 1.3.6.2 win32、Apache 1.3.12 win32 以前
- ・ 参考 : BID:1284、CVE:CVE-2000-0505

ScriptAlias ソース露呈に関する脆弱性

- ・ ScriptAlias ディレクトリが DocumentRoot 以下に置かれていた場合に、リモートからの攻撃者は cgi-bin ディレクトリの下プログラムソースコードや機密情報を取得可能。
- ・ 対象バージョン : Apache 0.8.14 以前、NSCA httpd 1.5a-export 以前
- ・ 参考 : BID:2300、CVE:CVE-1999-0236

MIME ヘッダによるサービス妨害攻撃に対する脆弱性

- ・ リモートからの攻撃者は 8000 バイトにおよぶ長い MIME ヘッダを大量に送信することで Web サービスをクラッシュさせることができる。通常の機能の回復にはサービスの再起動が必要。
- ・ 対象バージョン : Apache 1.2.5、Apache 1.3.1、MessageMedia UnityMail 2.0
- ・ 参考 : BID:1760

GET リクエストを用いたサービス妨害攻撃に対する脆弱性

- ・ リモートからの攻撃者は大量の「/」文字を含めた大量の GET リクエストを送信することでバッファオーバーフローを起こすことができる。サーバをフリーズさせてサービスの妨害を試みる事が可能。通常の機能の回復にはサービスの再起動が必要。
- ・ 対象バージョン : Apache 1.2.5 以前
- ・ 参考 : BID:2216、CVE:CAN-1999-0107

mod_cookies のバッファオーバーフロー - に関する脆弱性

- ・ Apache httpd におけるクッキー処理関連プログラムの脆弱性。mod_cookies.c 中の make_cookie 関数に適切なチェックを行っていないバッファが存在する。リモートからの攻撃者はバッファオーバーフローを起こすことで、サーバへのアクセス権を取得可能。
- ・ 対象バージョン : Apache 1.1.1 以前
- ・ 参考 : NAI:NAI-2、XF:http-apache-cookie、BID:1821、CVE-1999-0071

nph-test-cgi スクリプトに関する脆弱性

- ・ デフォルトでインストールされる nph-test-cgi プログラムに脆弱性が存在する。リモートからの攻撃者はサーバ上のファイルの一覧を取得可能。
- ・ 対象バージョン：NCSA NSCA httpd 1.5.2a 以前、Apache 1.1 以前、Netscape Commerce Server 1.12、Netscape Communications Server 1.1/1.12、Netscape Enterprise Server 2.0a
- ・ 参考：CERT:CA-97.07.nph-test-cgi_script、CVE:CVE-1999-0045、XF:http-cgi-nph、
 BID:686、

test-cgi スクリプトに関する脆弱性

- ・ デフォルトでインストールされる test-cgi プログラムに脆弱性が存在する。リモートからの攻撃者はサーバ上のファイルの一覧を取得可能。
- ・ 対象バージョン：NCSA NSCA httpd 1.5.2a 以前、Apache 1.0.5 以前
- ・ 参考：XF:http-cgi-test、BID:2003、CVE:CVE-1999-0070

phf スクリプトに関する脆弱性

- ・ CGI phf プログラムに致命的な脆弱性が存在する。リモートからの攻撃者はシェルメタキャラクタを利用して任意のコマンドの実行が可能。
- ・ 対象バージョン：Apache 1.0.3、NSCA httpd 1.5a-export
- ・ 参考：CERT:CA-96.06.cgi_example_code、XF:http-cgi-phf、CVE:CVE-1999-0067、
 BID:629

4. ファイアウォール

ファイアウォールに関する不正アクセス対策を述べる。

4.1. ファイアウォールの不正アクセス対策

ファイアウォールは内部ネットワークのセキュリティを実現するために不可欠ではあるが、たとえ管理を怠らず、適切な設定が施されていたとしても、ファイアウォールのみで組織のネットワーク内部のセキュリティを保つことは不可能である。

ファイアウォールは、外部から保護対象である内部のネットワークの情報へのアクセスと、その逆に内部から外部の情報へのアクセスに関して、基本的なセキュリティ方針の適用をある 1 地点で実現するものである。ファイアウォールはセキュアなネットワークを構築するための構成要素（部品）に過ぎず、効果的に使いこなすためには他の構成要素との連携が不可欠となる。

内部ネットワークと外部ネットワークの中間地点に置かれたファイアウォールにおいて、以下のようなネットワーク管理方針が取り決められる（あるいは方針が反映される）こととなる。

- ・ 外部ユーザがアクセスできる内部のサービスの種類
- ・ 内部ユーザがアクセスできる外部のサービスの種類
- ・ 一定の内部サービスへのアクセスが許可されている外部ユーザ

ファイアウォールを有効に機能させるためには、外部ネットワークからの全てのトラフィックをファイアウォールに集めて検査し、許可を与えて通過するように構成しなければならない。当然だがファイアウォール自体にも不正アクセス対策を施す必要がある。

4.2. フィルタリング方式

ファイアウォールの方式は、大きく分けてパケットフィルタリングとアプリケーション・ゲートウェイの 2 方式が一般的であったが、近年になって拡張や複合化が進められている。ユーザのセキュリティポリシーに適した方式を選択してファイアウォールを構築すべきである。以下に 2 つの方式を簡潔に説明する。

- ・ パケットフィルタリング：
IP パケットの発信元 IP アドレス、宛先 IP アドレス、アプリケーションの種類を示すポート番号、TCP や UDP などのプロトコル種別を基に、通過の可否の判断を行う方式。IP 層でのフィルタリングを行う。
- ・ アプリケーション・ゲートウェイ：
HTTP や FTP といったアプリケーションプロトコル毎にプロキシ（代理プログラム）を用意して内外のネットワークを切り離す。プロキシが用意されたアプリケーションに対してのみ、ファイアウォールを越えた通信が許される。

両者を比較すると、アプリケーション・ゲートウェイは送受信するパケットの内容を監視するためパケットフィルタリングより細かな制御が可能である。反面、アプリケーション・ゲートウェイはオーバーヘッドが大きく、パケットフィルタリングよりも通信速度が遅くなる。また、アプリケーションごとにプロキシを必要とするため、設定に手間がかかる。

アプリケーション・ゲートウェイ方式の一種に、サーキットレベル・ゲートウェイと呼ばれる方式もある。これはアプリケーションプロトコルの内容には感知せず、TCP/UDP のレベルでパケットフィルタリングとほぼ同様のチェックを行って特定のアプリケーションのみを通過させるゲートウェイソフト (socks, udprelay, plug-gw など) を利用する。アプリケーション・ゲートウェイに比べると、制御は粗いが、より高い汎用性を持つ。

パケットフィルタリングを拡張した方式には、ダイナミック・パケットフィルタリングと呼ばれる方式がある。これはパケットの内容をチェックし、アプリケーションレベルでのデータのやり取りを見て通過の可否を判断し、ポートの開閉のルールを変更する。要求と応答の対応が取れない場合などはパケットを遮断する機能を持っている。

4.3. 構築の基本姿勢

ファイアウォールを構築する際の基本姿勢は、組織の基本的なセキュリティ方針とほぼ等しい。以下の対照的な 2 つの姿勢のうち前者に基づいてファイアウォールを構築することが望ましいものとされる。

特別に許可するもの以外は全てを拒否する (推奨される姿勢):

基本的にはファイアウォールにおいて双方のネットワークからの全トラフィックをブロックし、利用するサービスやアプリケーションのみに他方のネットワークへの通路を開ける。内側から外側へのトラフィックについてもフィルタリングを行う。

慎重に考慮したサービスのみを管理運用するので非常にセキュアな環境が構築可能である。内外のネットワークにおけるサービス内容や利用されるアプリケーションを厳密に定義する必要がある。

特別に拒否するもの以外は全てを許可する:

セキュリティの実施が困難であり、管理負担が増える姿勢。基本的にはファイアウォールは全トラフィックを中継し、「特に危険なもの」があればそれらを閉ざす。柔軟な環境が構築可能であり、利用可能なサービスの幅も広いように感じられるが、潜在的な脅威を締め出すことができない。また、管理上、脅威が発見

される度に該当するサービスを調査して閉ざすという受け身の対応を採ることになるので、対応の遅れが致命的な損害に繋がりがち、拒否するトラフィックのリストが長くなると、ルールが複雑化し設定上のミスに繋がりがち。

ファイアウォールは単体のシステムではなく、境界の防衛に関するあらゆる要素を取り決めるセキュリティ方針の一部である。ファイアウォールによる防御を成功させるには、防御対象が明確でなければならない。セキュリティ分析、リスク評価、ニーズ分析を慎重に行い、結果に基づいてセキュリティ方針を決定する必要がある。

詳細に取り決めたセキュリティ方針がなければ、慎重にファイアウォールを構築しても裏をかかれて内部ネットワーク全体が侵害される可能性がある。

4.4. 弱点

ファイアウォールはセキュリティを達成する万能手段ではなく、以下のような弱点や対応不可能な状況が存在する。

4.4.1. オープンなポートを通じた攻撃

ファイアウォールは定められたルールに従ってパケットフィルタリングを行い、ルールで認められていないパケットを遮断するが、利用のための必要最低限のパケットは通過させざるを得ない。ファイアウォールにより遮断されないサービスとしてはネットワーク管理上のシンプルなプロトコルのほかに Web、メール、DNS が挙げられる。基本的にはファイアウォールには開いたポートを通して行われる内部のサービスへの攻撃を防ぐ能力は無い。この弱点を利用し ICMP、UDP でトンネリングを行うトロイの木馬ソフトウェアによりファイアウォールは無効化されてしまう。

また、Web、メール、DNS 等の解決のために許可されたポートを通して、ネットワーク内部の脆弱性を攻略するために不正な入力を送る攻撃手法が数多く存在する。単純な機能のみを備えるファイアウォールでは、このような攻撃のための通信を制限することはできない。

4.4.2. 動的なポート番号の割り当て

ある範囲のポート番号の中からアプリケーションに動的にポート番号を割り当てるような利用方法を、ファイアウォールを越えて行う場合は、ファイアウォールで塞ぐべきポートと開くべきポートを特定することができない。このような利用を行うためには非常に緩いファイアウォール設定を行う必要がある。例としては、DCOM (Distributed Component Object Model) を活用するシステムが挙げられる。

4.4.3. 監視上の対象の制約

ファイアウォールはネットワーク構造と配置された位置によってその効果に制約を受ける。

例えば、インターネットとの接点に配置されたファイアウォールは、ローカルにログイン可能な攻撃者が行う内部ネットワークのマシン間のアクセスに関する制限を加えることはできない。当然だが、ファイアウォールは物理的なアクセスを制限できないので、フロッピーディスク、CD-R、リムーバブルディスク等のメディアからの情報の漏洩も全く防げない。

ファイアウォールは、ファイアウォールを通過しない通信に対しては何ら防御手段とならない。例えば、ネットワーク内部のマシンからダイヤルアップが行われた場合は、PPP 等の手段で全く別の経路からネットワーク外部に接続できる。このような接続は、ファイアウォールを完全に迂回するため、どんなに慎重に構築したファイアウォールがあっても全く効果を持たないのと同様の結果となる。このような接続はサイト全体のセキュリティを著しく低下するものであり、厳しく監視する必要がある。

ウイルスに汚染されたファイルや不正プログラムのフィルタリングはオーバーヘッドがかかるため普通はファイアウォールでは行わない。これらはアプリケーションを扱うサーバ（メールサーバ等）や各クライアントにおいて検知される。ウイルス検知ソフトウェアと連携可能なファイアウォールであってもファイアウォール自身は検知を行わない。

4.5. 脆弱性対策

ファイアウォールの脆弱性に関する不正アクセス手法の例を示す。

Check Point Firewall-1 パケット不正フラグメンテーション攻撃（CVE-2000-0482）

- ・ 手法：不正にフラグメント化されたパケットを Check Point Firewall-1 に直接送信するか、ルーティングさせることで、パケットのログ生成にプロセッサ時間の 100% を消費させることができる。Firewall-1 のルールベースでこの攻撃を回避することができない。
- ・ 影響：Firewall-1 の CPU タイムを占有し、サービス不能にする。
- ・ 予防対策：(1) Firewall-1 4.1 用の service pack 2 を適用する。(2) FireWall-1 module でコンソールログ表示を停止するために、\$FWDIR/bin/fw ctl debug -buf を実行するか、起動時に実行されるよう \$FWDIR/bin/fw/fwstart に追加する。

Checkpoint Firewall-1 の内部アドレスが漏洩する脆弱性（CVE-2000-0181）

- ・ 手法：CheckPointのFirewall-1は内側アドレスを外側のネットワークに対してもらしてしまう脆弱性が存在する。通常の負荷状態（CPU使用率で40%、200以上のアクティブなコネクション程度）でFirewall-1は内部のアドレスを使用しての接続の中継を行ってしまう。このアドレスはルーティング不可のものであろうと、内部のものであろうと転送されてしまう。このパケットは正しく外側のアドレスを持っているが、同じソースポートを使用する。

- ・ 影響：ファイアウォールの背後に存在するマシンのアドレスおよび接続中継に利用するマシンの内部アドレスを特定し、クライアントへの攻撃に有益な情報を得ることができる。
- ・ 予防対策：サービスパックを適用する

IPFilter Firewall Race Condition エラー (CVE-2000-0553)

- ・ 手法：ファイアウォールの "return-rst", "keep state" ルール設定をオーバーラップさせるとアクセス制限を回避可能である。
- ・ 影響：攻撃者はアクセス制御を回避することができる。
- ・ 予防対策：なし。

5. その他

その他のいくつかのサービスについて述べる。

5.1. DNS

重要なサービスである DNS について述べる。

5.1.1. ゾーン転送設定のミス

システム管理者による DNS サーバの誤った設定によって、信頼できないユーザ（攻撃者）にゾーンファイルのコピーを提供してしまう可能性がある。外部 DNS 情報と内部のプライベート DNS 情報を分離するための DNS 機構を利用していない場合、内部ネットワークのホスト名、IP アドレス、HINFO レコードに記載された OS タイプ等の情報が漏洩する危険性がある。

以下のような対策が挙げられる：

- ・ ゾーントランスファーの実行を、権限を持つサーバに限定する。Windows NT 環境では DNS ゾーントランスファーの制限として通知オプションの強化を行う。
- ・ ファイアウォールやルータにおいて TCP ポート 53 への権限の無い接続を拒否し、このポートへの接続を侵害行為と想定してログに残す。
- ・ 攻撃者に有利な情報を与えないために HINFO レコードの情報を全て削除する。

5.1.2. BIND への攻撃

DNS の UNIX 環境への実装プログラムである BIND は攻撃者に最も狙われやすいサービスプログラムの一つとして知られている。BIND に関してはいくつかの危険度の高い脆弱性が報告されている。該当するバージョンのプログラムを使用する場合は最新のバージョンへのプログラムの更新が必要である。

弱点のあるバージョンの BIND で DNS の再帰が有効になっている場合に、攻撃者は再帰検索を実行するネームサーバのキャッシュを操作可能である（この手法は PTR レコードの偽造と呼ばれる）。IP アドレスをホスト名に割り当てるプロセスを攻略できる。

攻撃者により偽の情報がキャッシュされるため、ホスト名検索の信頼関係が崩れ、サービス妨害や他の深刻な被害が起き得る。

侵入に利用され易く危険度の高い BIND の脆弱性のいくつかについて、以下に攻撃手法の説明と対策を挙げる。

BIND の Transaction シグニチャにおけるバッファオーバーフロー脆弱性（CAN-2001-0010）

- ・ 手法：transaction signature (TSIG) を扱うコードの中にバッファオーバーフローに関する脆弱性が存在する。無効な TSIG を扱う際に、ある既知の値でメモリを上書き

可能である。named の持つ権限（典型的には root 権限）で任意のシェルコードを実行できる。

- ・ 影響：リモートからの攻撃者は SU 権限を取得可能。
- ・ 対象バージョン：BIND 4.9.8 未満、BIND 8.2.3 未満
- ・ 対策：脆弱性の影響を受けない最新のバージョン（8.2.3 以降、あるいは 9.1 以降）へのアップグレードが強く推奨される。（このほか各ベンダが修正プログラムを適用している）

nslookupComplain 関数におけるバッファオーバーフロー脆弱性（CAN-2001-0011）

- ・ 手法：syslog に対するエラーメッセージの受け渡しのための、ローカルに定義された文字配列に適切なチェックが行われないバッファが存在する。
- ・ 影響：攻撃者は改竄した DNS クエリを送ることで、DNS に対するサービス妨害や、DNS の動作する権限（通常はルート権限）でのコードの実行が可能。
- ・ 対象バージョン：BIND 4.9.7 以前
- ・ 対策：脆弱性の影響を受けないバージョンへのアップグレード。