第 11 部

Fire Wall 構築技術

第 1 章

はじめに

インターネットの接続は従来の大学や企業や政府の研究部門から，一般の企業，政府や地方公共団体，教育機関などに広がりを見せている．さらに，キャンパス LAN や企業内インターネットに代表されるインターネット接続組織内ネットワークの整備も進み，多くのインターネット接続組織は，組織内ネットワークとインターネットの間のネットワーク接続を行なっている．

このような状況で，特に企業や政府機関などでは，組織内ネットワーク上の情報や資源のインターネットからのアクセスを必要最低限に制約するための措置を施す必要性がでてきた．一方，従来は，多くの場合インターネットからオープンにアクセスできた大学のキャンパスネットワークでも，急増する外部からのアタックといったセキュリティ上の脅威を軽減し，事務部門や実験設備などのコンピュータの外部からの不正アクセスを防ぐ必要から，インターネットとキャンパスネットワークの接続点での何らかのアクセス制限の要求が高まってきている．

外部からのアクセスを許可されたもののみに制限するための機構として Firewall は効果的であり，広く使われている．WIDE プロジェクト Firewall Working Group では，Firewall の技術的な検討を行なっている．現在の Firewall Working Group の興味は以下の通りである:

- 現在の Firewall に関連する技術の調査と問題点の分析
- インターネットの利便性を確保しつつ，不正アクセスを確実に防止するための Firewall 技術の確立
- Firewall を含めた情報アクセスの制限 (部分的な開示) をおこなうアクセスコントロールのメカニズムに関する研究
- インターネットアーキテクチャにおける Firewall 技術の位置付けに関する検討

今年度の Firewall Working Group の報告では，93 年度に WG で調査した Firewall 技術のサーベイを中心に報告する．

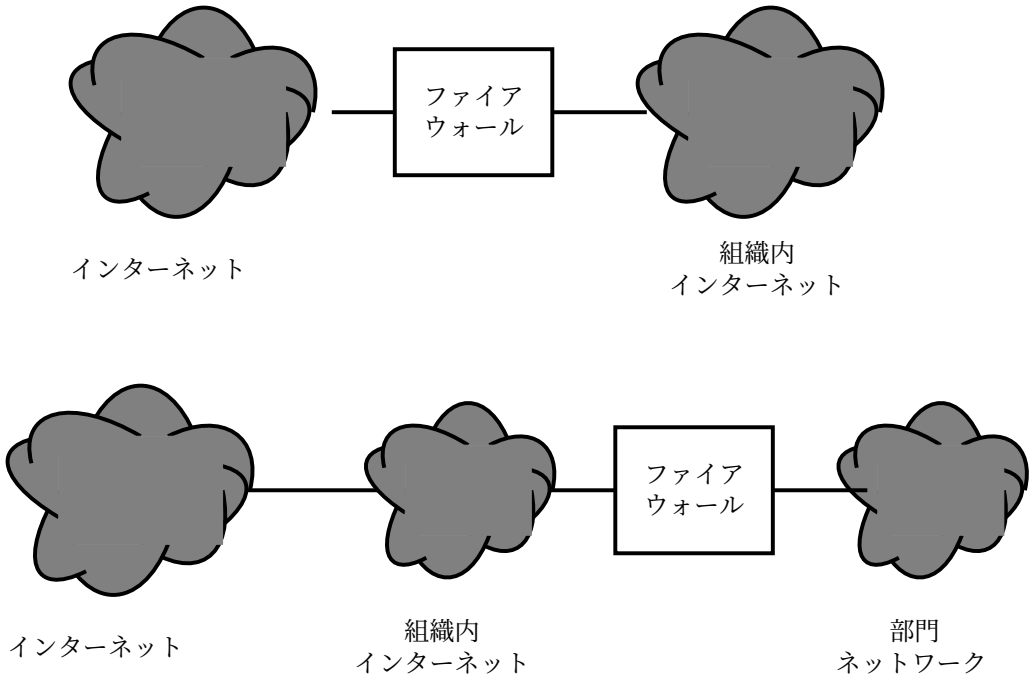

図 1.1: Firewall のイメージ

1.1 Firewall とは

従来，ネットワーク技術において Firewall(防火壁) とは，ブロードキャストストームのようなパケットの異常発生やネットワーク上の障害を，ネットワーク間接続装置によって，障害が発生したネットワーク内に局所化し，他のネットワークへの波及を防ぐための技術を指していた．しかし，最近では，内部のネットワーク資源の外部からのアクセスを部分的に制限したり，部分開示するといったアクセスコントロールをしつつ，外部からの不正なアクセスを遮断するために，内部ネットワークと外部ネットワークを接続する，ネットワーク間接続機能をいうようになってきている．

Firewall は多くの場合，インターネット接続組織の内部ネットワークとインターネットの接続点で運用することが多い (図 1.1) が，組織内部のセキュリティ要求の強いネットワーク[1]が，組織のバックボーンネットワークと接続するポイントに設置することもある．さらに，ポリシーの異なるネットワーク間に，お互いのポリシーを維持するために設置する場合もある．

[1] この場合 Firewall ホストは二つのインタフェースを持つ．しかし，パケットフィルタリングや経路制御により，外部からアクセス可能なホストを特定することにより，単一インタフェースのホストでファイアウオールホストを実現することも可能である．

Firewall の基本機能は一方のネットワークから他方へのアクセスを制限することであるが，アクセス制限を加えるレベルで，Firewall を大きく二つの方式に分類することができる:

- ネットワークレベルでの制限

  インターネットゲートウエイ (IP ルータ) で IP データグラムを一方のネットワークから他方に転送する際に，IP データグラムの送り先アドレス，発信元アドレス，送り先ポート，発信元ポート，プロトコル番号から転送をするかしないかを決定する．この機能はパケットフィルタリング機能として多くの IP ルータ上に実現されている．ポート番号やプロトコル番号によって特定のサービスだけにアクセスを限定することもできる．ネットワークレベルでの制限はパケットフィルタリングの他にアナウンスする経路情報を制限することにより行なうこともできる．

- アプリケーションレベルでの制限

  ネットワーク間接続をルータではなく，アプリケーション・ゲートウエイによって行なう方法である．一方から他方へのアクセスは，IP で直接行なうのではなくすべてアプリケーションゲートウエイを介しておこなう．アプリケーションゲートウエイはカーネル設定により IP 転送機能を禁止したホストにより実現するのが普通 (注) で，このホストを Firewall ホストと呼ぶ．

1.2　インターネット接続における Firewall の役割り

インターネット接続組織が Firewall を用いてインターネットに接続する主な意味は次の通りである:

- 外部からアクセスできるホストを限定することにより，安全でないホストを外部アクセスから守る．

- 外部からアクセスできるホストを限定することにより，そのホストに対して重点的にセキュリティ対策を施すことができる．

- 外部から内部へアクセスできるプロトコルを SMTP といった安全なサービスだけに限定することによって，外部からの不正アクセス，運用妨害，アタックから内部の資源を守ることができる．

- Firewall のログ機能によりアクセス記録をとることができ，これにより不正アクセスの痕跡を残すことができる．

1.3 Firewall の運用上の問題

本来インターネットはインターネットのノードがお互いに通信できることを前提に設計されたコンピュータネットワークである．Firewall は直接通信可能なホストと通信できないホストを区分けする技術であり，このことは，上記の前提を崩すことになる．Firewall の運用がつぎのような問題を引き起こす:

- DNS と電子メイルの問題

  外部からのアクセスが制限されているホストのアドレス情報を DNS を用いて外部にアナウンスすると，そのホスト向けの電子メイル配送に不具合をきたす．外に対してはアクセス可能なホスト情報のみをアナスンスする必要があり，この問題を回避するために，多くのインターネット参加組織では，組織内向けとインターネット向けの二つの DNS サーバを運用することがある[2]．

- 経路制御の問題

  ホスト単位に制限を加えると，そのホストの属するネットワークへの経路情報はあるので，そのホストへの通信は Firewall まで届くが，そこで IP データグラムが破棄される．Firewall の実現によっては破棄されるときに，ICMP でネットワーク到達不可能メッセージを発信元に送りかえすものもあり，この ICMP メッセージが発信元ホストに悪影響を与える場合がある．

- 到達可能なアドレスと到達不可能なアドレスの問題

  内側と外側の二つのインタフェースを持つ Firewall ホストの場合，外側のアドレスを送り先とした場合，外部からアクセス可能であるが，内部のアドレスを送り先とした場合，外部からアクセスは不可能である．

これらの問題は運用上の問題だけに留まらず，ネットワークアプリケーションのプログラミングにも影響を与える．Firewall 環境に考慮したアプリケーションプログラムのコーディング技法，注意事項については，取りまとめをおこなっているところである．

1.4 今後の研究課題

Firewall は完成された技術ではなく，多くの研究課題がある．

- 高信頼性の維持

  Firewall は少なくとも設定されたアクセス制御が正しく機能している必要がある．そのため，製品としてルータのパケットフイルタリングに頼ることは，実装が見えないことによる不安，設定や条件によるセキュリティの「穴」がないという保証がな

[2] JP ドメインの DNS サーバの系列化も同様な問題に起因する．

いなどの理由で，このルータのパケットフイルタリングだけに頼るのは危険であると考える運用者も多い．そこで多くの場合，経路となる複数のルータに同様のフィルタリングを設定したり，Firewall ホストにより，IP コネクションを完全遮断するといった対策をとる．

正しく Firewall が機能していることを検証する何らかの仕組みが望まれている．また，ファイアウオールの信頼性は，運用するオペレータのスキルに大きく依存する．また，スキルの高いオペレータであっても，日常運用における油断が，Firewall の低信頼化を招くことがある．このような人間系を含めて，Firewall の安全な運用をサポートする機構が必要である．さらに，完全な Firewall というのは現実にはむずかしく，そのためアクセスは記録として残し，すくなくとも異常なアクセスは事後であっても検知可能にしておく必要がある．さらにログから異常を自動発見する機能，発見をサポートする解析機能も必要であろう．

以上のように運用系を含めた Firewall システムの再検討が急務である．

- Firewall 実装技術

  現在の Firewall は，IP アドレス，ポート番号に頼ったアクセス制御と，サービスごとに個別のアプリケーションゲートウエイを利用する方法が基本である．しかし，実際にはユーザごと，情報ごとに，細かなアクセスコントロールを必要とする場合もあり，現在の Firewall はこの要求に答えていない．これらのアクセス制御を Firewall だけで行なうには無理があり，サーバプログラムやクライアントプログラムとのなにかしらの連携も必要になってくる．

  また，Firewall のアクセスコントロールはポリシー経路制御とも密接な関係がある．インターネットの情報アクセス制御，ポリシー制御のメカニズムを再検討し，そのなかでの Firewall の役割を考えなければならない．

- 性能の問題

  国内の場合 64Kbps 程度の専用線でインターネットと接続するケースが一般的である．この程度の速度であれば，一般の計算機上でのアプリケーションゲートウエイでも十分な処理速度を得ることができる．しかし，さらに高速な接続になると，現在のアプリケーションゲートウエイ方式では，十分な処理速度を得ることは難しい．また，要求が高度になるにつれ，必要とする処理量も増えることが予想される．インターネットの高速化に対応した Firewall 実装技術を確立する必要があるだろう．

- Firewall とインターネットアーキテクチャ

  すでに述べた通り，インターネットでは原則として任意のノード間が IP で相互接続できることを前提にして設計されている．一方で Firewall はこの任意ノード間の相互接続に制限を与える技術であり，そのために，運用上の/ネットワークアプリケーション作成上のつじつま合わせを必要とすることがある．インターネットにおける

Firewall の位置付けを明確にし，インターネットのアプリケーションの挙動やポリシ経路の問題を含めて，インターネットアークテクチャの見直しが必要となっている．

第 2 章

Firewall の典型例

この章では Firewall の典型例について紹介する。前章でも述べたように Firewall はインターネットと自組織のネットワークとの間のアクセス制限を行うことが目的である。はじめにアクセス制限の方針 (ポリシーと呼ばれることもある。) の例を紹介し, つぎにこれらのアクセス制限の方針の実装について述べる。

2.1 アクセス制限の方針の例

アクセス制限の方針は, インターネットから自組織のネットワークに向かってできることと, 自組織のネットワークからインターネットに向かってできること, に分けて考えるとわかりやすい。

インターネットから自組織のネットワークに向かって, 自組織のネットワークからインターネットに向かって “できること” というものにはつぎのようなサービス/機能がある。

以下ではこれらのサービス/機能を括弧内の呼び方で指す。

- ネットワークを介して計算機を使う。 (remote logon)
- ファイル転送を開始し, 転送を行う。 (File 転送)
- メールを送信/受信する (メール)
- Usenet ニュースを見る/書く (ニュース)
- (Gopher, WAIS, archie, WWW などの) 情報提供サーバへアクセスする (サーバアクセス)
- IP 到達可能である。 (IP 接続)

自組織のネットワークについては上のそれぞれの “できること” に関して組織内の人, 組織, 計算機などによって違いを設けるかどうかによって Firewall の実装がかわる。この区別の例としてはつぎにあげるものがある。

- 特定の計算機だけをゆるす。

表 2.1: アクセス制限の方針

	自組織のネットワークからインターネット	インターネットから自組織のネットワーク
remote login	特定の計算機だけゆるす 特定のサブネット上の計算機だけゆるす 特定の組織の計算機だけゆるす あらかじめ認めた人だけを許す	特定の計算機だけゆるす 特定のサブネット上の計算機だけゆるす 特定の組織の計算機だけゆるす あらかじめ認めた人だけを許す
ファイル転送	特定の計算機だけゆるす 特定のサブネット上の計算機だけゆるす 特定の組織の計算機だけゆるす あらかじめ認めた人だけを許す	特定の計算機だけゆるす 特定のサブネット上の計算機だけゆるす 特定の組織の計算機だけゆるす あらかじめ認めた人だけを許す
メール	特定の計算機だけゆるす 特定のサブネット上の計算機だけゆるす 特定の組織の計算機だけゆるす あらかじめ認めた人だけを許す	特定の計算機だけゆるす 特定のサブネット上の計算機だけゆるす 特定の組織の計算機だけゆるす あらかじめ認めた人だけを許す
ニュース	特定の計算機だけゆるす 特定のサブネット上の計算機だけゆるす 特定の組織の計算機だけゆるす あらかじめ認めた人だけを許す	特定の計算機だけゆるす 特定のサブネット上の計算機だけゆるす 特定の組織の計算機だけゆるす あらかじめ認めた人だけを許す
サーバアクセス	特定の計算機だけゆるす 特定のサブネット上の計算機だけゆるす 特定の組織の計算機だけゆるす あらかじめ認めた人だけを許す	特定の計算機だけゆるす 特定のサブネット上の計算機だけゆるす 特定の組織の計算機だけゆるす あらかじめ認めた人だけを許す
IP 接続	特定の計算機だけゆるす 特定のサブネット上の計算機だけゆるす 特定の組織の計算機だけゆるす	特定の計算機だけゆるす 特定のサブネット上の計算機だけゆるす 特定の組織の計算機だけゆるす

- 特定のサブネット上の計算機をだけゆるす。
- 特定の組織の所有する計算機をだけをゆるす。
- あらかじめ認めた人だけをゆるす。

以上をまとめたのが表 2.1である。

表 2.1の各欄はサービス/機能と、アクセスの方向 (インターネットから自組織のネットワークに向かってか, その逆向きか) によって分類されている。この各欄の中に人, 組織, 計算機で区別する場合の例が載っている。この表を使ってアクセス制限の方針を整理することができる。

以下にいくつかアクセス制限の方針の例を上げ, この表の形で示す。

例１

特定の人だけが自組織のネットワークからインターネットへ remote login, ファイル転送, メールの送受信ができる。ニュースは組織内のだれでも読み書きできる。　インターネットから自組織のネットワークへはメールの送受信, ニュースのやりとり以外はなにもできない。表 2.2が上の方針をまとめたものである。

例２

特定の人だけが自組織のネットワークからインターネットへ remote login, ファイル転送をできる。組織内のだれでもニュースは読み書きでき, メールの送受信ができる。インターネットから自組織のネットワークへは特定の人が remote login できる。表 2.3が上の方針をまとめたものである。

例３

表 2.2: アクセス制限の方針例 1

	自組織のネットワークからインターネット	インターネットから自組織のネットワーク
remote login	あらかじめ認めた人だけを許す	できない
ファイル転送	あらかじめ認めた人だけを許す	できない
メール	あらかじめ認めた人だけを許す	あらかじめ認めた人だけを許す
ニュース	だれでもできる	だれでもできる
サーバアクセス	できない	できない
IP 接続	できない	できない

表 2.3: アクセス制限の方針例 2

	自組織のネットワークからインターネット	インターネットから自組織のネットワーク
remote login	あらかじめ認めた人だけを許す	あらかじめ認めた人だけを許す
ファイル転送	あらかじめ認めた人だけを許す	できない
メール	だれでもできる	だれでもできる
ニュース	だれでもできる	だれでもできる
サーバアクセス	できない	できない
IP 接続	できない	できない

組織内の特定の計算機からインターネットへは remote login, ファイル転送, サーバアクセスできる。メールはだれでも送受信でき, ニュースもだれでも読み書きできる。インターネットから自組織のネットワーク上の特定の計算機に関して remote Login できる。表 2.4が上の方針をまとめたものである。

例 4

組織内の特定の計算機はインターネットへ IP 接続可能である．それ以外の計算機はメールの送受信はできる．表 2.5が上の方針をまとめたものである。

例 1 から例 4 にいくにしたがってアクセス制限が緩くなっていっている。一般的に企業は制限が厳しく, 大学は緩い。企業においてもインターネットの接続の価値と受容しなければならない危険とをくらべて, 自組織のネットワークからのアクセス制限は緩くしていく傾向にある。大学などでは無制限にインターネット接続していたところがインターネットからの危険を認識しインターネットからのアクセスに制限を設けだしたところもある。

2.2　Firewall の実装例

Firewall を実装の面から見るときは, 次の点に着目するとよい。

表 2.4: アクセス制限の方針例 3

	自組織のネットワークからインターネット	インターネットから自組織のネットワーク
remote login	特定の計算機だけゆるす	特定の計算機だけゆるす
ファイル転送	だれでもできる	できない
メール	だれでもできる	だれでもできる
ニュース	だれでもできる	だれでもできる
サーバアクセス	特定の計算機だけゆるす	できない
IP 接続	できない	できない

表 2.5: アクセス制限の方針例４

	自組織のネットワークからインターネット	インターネットから自組織のネットワーク
remote login	特定の計算機だけゆるす	特定の計算機だけゆるす
ファイル転送	特定の計算機だけゆるす	特定の計算機だけゆるす
メール	だれでもできる	だれでもできる
ニュース	だれでもできる	だれでもできる
サーバアクセス	特定の計算機だけゆるす	特定の計算機だけゆるす
IP 接続	特定の計算機だけゆるす	特定の計算機だけゆるす

- ネットワーク構成 (アドレスも含む)
- 経路制御
  - 経路情報のアナウンス
  - Firewall の経路制御表
- DNS
- IP Forwarding するかどうか、どういうものを転送するか

Firewall に計算機を使うときは次の点にも着目する。

- 接続できるポート
- 動いている daemon プログラム

図 2.1にネットワーク接続図をしめす。自組織のネットワークに Firewall がつながっている (このインタフェースを II と呼ぶ)。Firewall は別のインタフェース (インタフェース IE と呼ぶ。) を介して外部接続用のネットワーク (バリアセグメントと呼ばれることもある) につながっている。この外部接続用のネットワークにルータがつながっている。このルータが専用線などでネットワークプロバイダのルータにつながっている。

アドレス

自組織のネットワーク上の (全てまたは一部の) 計算機がインターネットと IP 到達可能にするためには自組織のネットワークの経路情報をインターネットに知らせなければならない。外部接続用のネットワークのアドレスは自組織のネットワークのサブネットを使ってもよい。自組織のネットワークの経路情報をインターネットに知らせている場合でも Firewall の経路制御表を操作して自組織のネットワーク上の一部の計算機だけを IP 到達可能にできる。

自組織のネットワーク上の計算機がインターネットと IP 到達可能でなくてよい場合は自組織のネットワークの経路情報をインターネットに知らせてはならない。　外部接続用のネットワークの経路情報はインターネットにしらせなければならない。 外部接続用のネットワークのアドレスは NIC (日本では JPNIC) より正式に取得する。(今後、自組織のネットワークをインターネットから IP 到達可能にする予定がなく, かつ, 他のプライベートネットワークとも接続する予定がないのなら) 自組織のネットワークのアドレスは [75] に従って指定されたアドレスを使うことが薦められている。

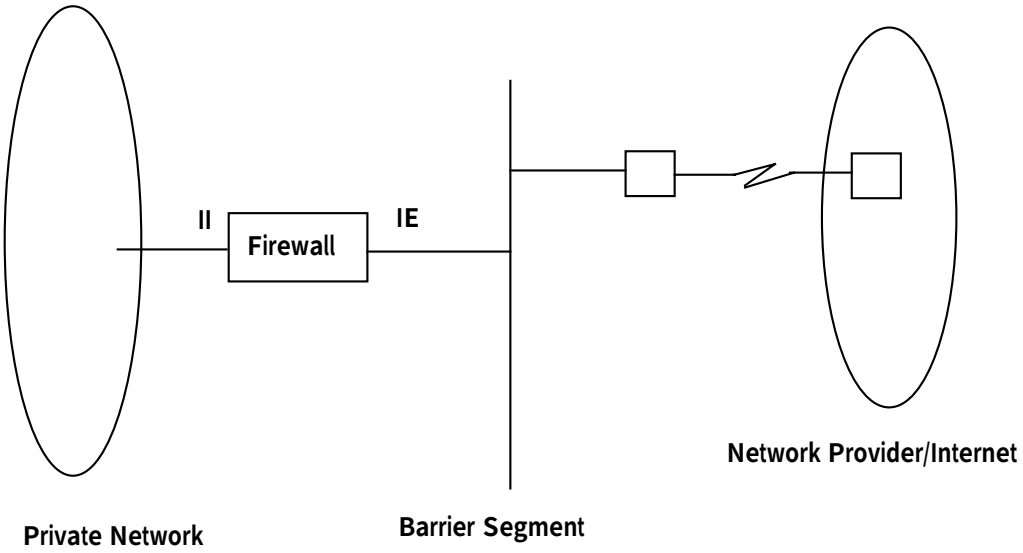

図 2.1: ネットワーク接続図

Firewall の経路表

Firewall の経路表は自組織のネットワークとはインタフェース II を通して通信するように, インターネットとはインタフェース IE を通して通信するように設定されなければならない。インターネットのネットワークの数が自組織のネットワークの数より多いので Default をインターネット側へむける。

ルータよりインターネット側の経路制御はネットワークプロバイダによって異なるのでここでは触れない。外部接続用のネットワークを介して複数のネットワークプロバイダに接続している場合は外部接続用のネットワーク上での経路制御によってネットワークプロバイダの使用方針を実現する。外部接続用のネットワークを介して一つのネットワークプロバイダとしかつながっていないときは外部接続用のネットワーク上での経路制御は必要ない。先に述べた情報を Firewall の経路表に静的に保持すればよい。

経路情報アナウンス

自組織のネットワーク上の (全てまたは一部の) 計算機がインターネットと IP 到達可能である場合は自組織のネットワークの経路情報を自組織のネットワークと他のネットワーク 境界点で生成し, インターネットに知らせる。この境界点は外部接続用のネットワークのアドレス, 専用線のアドレスにより表 2.6に示すようになる。

自組織のネットワーク上の (全てまたは一部の) 計算機がインターネットと IP 到達可能でない場合は外部接続用のネットワークの経路情報を外部接続用のネットワークと他のネットワーク 境界点で生成し, インターネットに知らせる。この境界点は外部接続用の

表 2.6: 自組織のネットワークの経路情報を生成する点

外部接続用のネットワークのアドレス	専用線のアドレス	生成する点
自組織のネットワークのサブネット	自組織のネットワークのサブネット	ネットワークプロバイダ
自組織のネットワークのサブネット	ネットワークプロバイダのサブネット	外部接続用のネットワーク上のルータ
ネットワークプロバイダのサブネット	ネットワークプロバイダのサブネット	Firewall

表 2.7: 外部接続用のネットワークの経路情報を生成する点

専用線のアドレス	生成する点
外部接続用のネットワークのサブネット	ネットワークプロバイダ
ネットワークプロバイダのサブネット	外部接続用のネットワーク上のルータ

ネットワークのアドレス, 専用線のアドレスにより表 2.7に示すようになる。

いづれの場合もこれらの設定/作業はネットワークプロバイダと相談し, 協調しておこなわなければならない。

DNS

Firewall があるときの DNS については [75] に解説がある。

IP Forwarding するかしないか, なにを転送するか

自組織のネットワーク上の (全てまたは一部の) 計算機がインターネットと IP 到達可能である場合は Firewall は自組織のネットワークから/あての (全てまたは一部の) パケットを転送しなければならない。全てにするか一部にするかはアクセス制限の方針から決まる。アクセス制限の方針として一部の計算機だけがインターネットと IP 到達可能にするのならその計算機あて/からのパケットだけを転送し, それ以外は廃棄する。アクセス制限の方針として特定の人だけインターネットに IP 到達可能にするとしていたとしても, パケットからは人を判別できないのでそれ以外の認証の仕組みが必要である。アクセス制限の方針で remote login のみ許す場合はパケットの TCP のポートまで判別して remote login 用の TCP ポート番号を持つパケットのみを通すようにしなければならない。

自組織のネットワーク上の (全てまたは一部の) 計算機がインターネットと IP 到達可能でない場合は Firewall でパケットを転送してはならない。

接続できるポート /動いている daemon

これらは Firewall が計算機であり, 自組織のネットワークがインターネットと IP 到達可能でない場合のみあてはまる。このときはインターネットから IP 到達可能なのは Firewall までであるので Firewall のどのポートがどこから接続可能であるかによってアクセス制限の方針を実装する。

remote login インターネットからの remote login をみとめないで, かつ, 自組織のネットワークからのインターネット上の計算機に remote login を許す場合、Firewall に login してそこからさらに目的の計算機に login するという方法をとる (これはかなり一般的である。)。このためには自組織のネットワークからの remote login だけを許可しなければならない。UNIX では inetd, telentd を改造したり inetd の代わりに xinetd を使ったりする。

ファイル転送 インターネットから開始されるファイル転送をみとめないで, かつ, 自組織のネットワークからのインターネット上の計算機に file 転送を許す場合 Firewall に転送してそこからさらに目的の計算機に転送するという方法をとるときは (これはかなり一般的である。) 自組織のネットワークからのファイル転送だけを許可しなければならない。このためには UNIX では inetd, ftpd を改造したり inetd の代わりに xinetd を使ったりする。

メールの送受信 特定の人/組織のみのメールを許可するような場合には通常のメール処理プログラムを改造する必要がある。メールの送受信を許している場合メールのポートを両側とも接続できるようにしておく。 UNIX では正しく設定された sendmail でよい。

ニュース 特定の人/組織のみのニュースを許可するような場合には通常のニュース処理プログラムを改造する必要がある。ニュースの送受信を制限していない場合ニュースのポートを両側とも接続できるようにしておく。

サーバへのアクセス Firewall をまたいでサーバへアクセスできるようにするには Firewall が router の場合はサーバが使うプロトコルのパケットは転送するようにしなければならない。この場合、自組織のネットワークからのみのアクセスを許すというアクセス制限を実装するのはむずかしい。Firewall が計算機の場合は Firewall 上で gateway daemon を動かして pakcket を中継する。Gateway daemon には一般 TCP 用として Socks, 一般 UDP 用として UDP Relay がある． Gopher 専用としては GopherGate がある。これらの gateway daemon は設定ファイルでアクセス制限を設定できるので IP アドレスなどで判別できるアクセス制限の方針は実現しやすい。

第 3 章

Tool と利用例の紹介

ここでは Firewall に関するいくつかの Tool と機器とを設定例などを含めて使用者の立場から紹介する。

3.1 Xinetd

4.2BSD をベースとした UNIX ワークステーションには、inetd と呼ばれる Network Service を提供するための Single process/Multi thread の daemon が用意されている。

inetd は Security 的には無力であり、起動されるプロセス側で Security に関して気を使う必要がある。

xinetd は Security 上の観点で inetd に対して以下にあげるの機能を追加している。

1. アクセス制御機能
2. ログ機能
3. その他の機能

3.1.1 configuration のしかた

上記の機能は、すべて/etc/xinetd.conf に記述することにより有効となる。xinetd.conf では以下の様な形式でサービス毎の挙動を指定する。

```
service ポート名
{
        キーワード  =    value
        キーワード  +=   value
        キーワード  -=   value
}
```

また、全体に関するデフォルト値を指定するために次の形式で指定することが可能である。

```
default
{
        キーワード  =    value
        キーワード  +=   value
        キーワード  -=   value
}
```

=はキーワードの値を value にし、+=はキーワードの値に value を追加し、-=はキーワードより value を削除する。

指定可能なキーワードを表 3.1.1にあげる。

3.1.2 アクセス制御機能

アクセス制御は Firewall に限らず Security を考える上で重要な機能である。
xinetd には以下にあげるアクセス制御の機能が組み込まれている。

- ホストによるアクセス制御 (ユーザによるアクセス制御)
- 時間帯によるアクセス制御
- 同時にアスセスできる数の制限

あるホストの FTP にたいしてアクセス制御を行なう場合の例をあげる。

```
service tftp
{
        flags           = INTERCEPT IDONLY
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = root
        group           = ftp
        server          = /usr/etc/in.ftpd
        server_args     = -a -L
        only_from       = 129.249.0.0
        only_from       = 131.221.0.0
        no_access       = 129.249.88.1
        access_time     = 10:00-11:50 12:50-17:36
        instances       = 10
        log_on_failure  += USERID
}
```

このホストは以下のようにアクセス制御されていることになる。

- identd に対して答えたもののみ許す (flags)
- アクセスできるのは 129.249/131.221 のホストであり 129.249.88.1 のみアクセスを許さない (only_from/no_access)
- 現地時間 10:00 より 11:50 と 12:50 より 17:36 までアクセスを許す (access_time)
- 一度に動ける ftpd は 10 個まで (instanmces)

各キーワードの設定方法を以下にあげる。

キーワード only_from は、ここに指定されたアドレスからのアクセスを許すという指定になる。一方 no_access は逆にここに指定されたアドレスからのアクセスを許さないという指定になる。

アドレスを指定するのは=の右側であるが、ここには以下のように記述する。

1. %d.%d.%d.%d

2. %d.%d.%d.{%d,%d,...} もちろん%d.%d.{%d,%d,...} でも良い

3. ネットワーク名 (/etc/networks からとか)

4. ホスト名

1 番めの形式の場合 1 番右側の数字が 0 の場合、ワイルドカードと解釈される。すなわち、129.249.88.0 は Subnet 129.249.88 のすべてのホストと解釈される。

アドレスとして 129.249.0.0 を指定した場合、129.249 のすべてのアドレスと解釈される (当然 0.0.0.0 を指定した場合は、すべての Internet アドレスと解釈される)。

1/2 番目の形式では、常に 8bit づつの区切りで記述しなければならないので 0 を指定する場合は注意が必要である。

3 番目の形式で指定する場合は/etc/netmasks 等を適切に指定してある場合は任意のネットマスクを認識する。

4 番目の形式でホスト名を指定する場合、指定されたホストのすべての IP アドレスが指定されたものとみなされる。

only_from/no_access のいづれも指定されなかった場合は、ホストアドレスによるアクセス制限はしないと指定されたことになる。

only_from/no_access の両方が指定された場合、両者のうちもっとも良くマッチしたものが採用される。

つまり上の例では以下の様に解釈される。

- 129.249/131.221 の全てのホストからのアクセスを許す

- が 129.249.88.1 からのアクセスは許さない。

アクセス可能な時間帯を指定する場合、access_time で指定する。例では現地時間の 10:00 より 11:50 まで 12:50 より 17:36 までのアクセスを許すことになる。

一度にアクセスできる数を制限する場合、instances で指定する。例では一度に 10 まで同時にそのサービスにアクセスすることができる。制限をしない場合は UNLIMITED を指定する。

3.1.3 ログ機能

ログの機能は xinetd ではかなり充実している。
xinetd ではログをとる時に

- syslog によるもの (facility として daemon,auth,user,local0-7 が指定可能)
- file に直接 xinetd が書くもの (ログファイルの大きさを指定できる)

の 2 種類の方法が指定できる。
指定は以下の様に指定する。SYSLOG と FILE を同時に指定することはできない。

```
log_type = SYSLOG local0 notice
log_type = FILE /var/log/xinetd.log 20K 21K
```

この例では SYSLOG の場合、「local0 の facility で notice レベルのログをとる」と指定している。
FILE を指定した場合、ログファイルの大きさの指定ができる。この例ではソフト的なリミット (ログファイルの大きさが越えたとログする) を 20K バイトにし、ハード的なリミット (ログをとることを中止する) を 21K バイトに指定している。この場合、ログがハードリミットを越えたことを alert のレベルで syslog に送る (facility は通常は daemon)。
また、xinetd ではログをとる状況を指定することができる。ログをとるイベントとしてはサービスが起動に成功した場合 (log_on_success) と失敗した時 (log_on_failure) の 2 つが設定できる。
各々に対して「何」をログするかを指定できる。
指定は以下のような形式で行なう。

```
log_on_success = USERID
log_on_failure = HOST USERID
```

表 3.1.3に log_on_success で指定できるアイテムを、表 3.1.3に log_on_failure に指定できるアイテムをあげる。
このうち、ATTEMPT/RECORD などを指定する場合 flags で INTERCEPT を指定する必要がある。

3.1.4 その他の機能

xinetd では、env/passenv を使うことにより環境変数を自由にサーバプロセスに渡すことができる。
例えば、SunOS などで Shared Library を用いている場合 LD_LIBRARY_PATH の指定により特定の Shared Library を使用する指定をすることもできる。
passenv を指定し、値を空に指定すると xinetd からは一切の環境変数が継承されない。

表 3.1: xinetd.conf で指定できるキーワード

id	サービスを識別するための識別子を指定する。
	普通はサービス名と同じ
type	RPC/INTERNAL/UNLISTED の任意の組合せ
flags	xinetd の動作に対する指定。
	REUSE/INTERCEPT/NORETRY/IDONLY がある
socket_type	stream/dgram/raw/seqpacket を指定する。
protocol	/etc/protocol の指定されているプロトコルのどれかを指定する
wait	yes/no を指定する。意味は inetd.conf の wait/nowait と同じ
user	サーバプロセスの uid を指定する
group	サーバプロセスの gid を指定する
instances	同時に動けるサーバプロセスの数を指定する
nice	サーバプロセスのプライオリティを指定する
server	サーバプロセスのパス名
server_args	サーバプロセスに渡す argument を指定する
only_from	アクセス制御機能
no_access	アクセス制御機能
access_time	アクセス制御機能
log_type	ログ機能
log_on_success	ログ機能
log_on_failure	ログ機能
rpc_version	RPC バージョン番号の指定
rpc_number	RPC プログラム番号
env	サーバプロセスに渡す環境変数を指定する
passenv	xinetd の環境変数のうちサーバプロセスに継承される
	環境変数を指定する
port	サーバのポートを指定する

表 3.2: log_on_success で指定できるもの

PID	サーバプロセスの pid
HOST	アクセスしたホストのアドレス
USERID	identd で調べた uid
EXIT	サーバプロセスの exit status
DURATION	サーバのセッションの持続時間

表 3.3: log_on_failure で指定できるもの

HOST	アクセスしたホストのアドレス
USERID	identd で調べた uid
ATTEMPT	何をしようとしたかのログ
RECORD	一部のサービスの時に相手側の情報のログする (login/shell/exec/finger のみサポート)

3.2 TCP Wrapper

TCP wrapper は、inetd が応答する、FTP や TELNET 等のネットワーク・サービスのリクエストに対する、アクセスの制限とアクセスの監視や通知を行なうツールである。これだけでは xinetd とあまり変わらないようであるが、TCP wrapper の場合、inetd は OS 標準のものそのまま利用し、tcpd という汎用のプログラムを使って、inetd から起動するプログラムに対して制限を行なう。また、TCP wrapper では、アクセス制限、ログ機能の他に、リクエストがあった場合に管理者にメイル送るといったコマンドも設定できる。

3.2.1 tcpd の設定

SunOS 4.x の fingerd(in.fingerd) を例に、tcpd の 2 種類の設定方法を説明しよう。なおアクセス制御等については、後程説明する。

1 番目の方法であるが、tcpd のコンパイル後、次の作業を root 権限で行なう。

```
# mkdir /usr/realdir
# mv /usr/etc/in.fingerd /usr/realdir
# cp tcpd /usr/etc/in.fingerd
```

これで、finger リクエストに対して tcpd が機能するようになる。但し、in.fingerd の移動先である /usr/realdir を、tcpd のコンパイル時に定義して組み込んでおく必要がある。TCP wrapper のパッケージでは、Makefile の REAL_DAEMON_DIR の変数で設定できるようになっているので、実際にコンパイルするときは次のようにすればよい。

```
% make REAL_DAEMON_DIR=/usr/realdir sunos4
```

この方法を使う場合、inetd や fingerd のプログラムはもとより inetd.conf も変更する必要はない。

2 番目の方法は、inetd.conf を変更する方法である。もともと SunOS 4.x の inetd.conf の fingerd の設定行は次のようになっている。

```
finger stream tcp nowait nobody /usr/etc/in.fingerd in.fingerd
```

tcpd を使う場合、次のように変更する。

```
finger stream tcp nowait nobody /usr/etc/tcpd in.fingerd
```

この場合、REAL_DAEMON_DIR には/usr/etc を設定して、tcpd をコンパイルしておく必要がある。

3.2.2 TCP wrapper のアクセス制御

TCP wrapper のアクセス制御は、次の 2 つのファイルで設定する。

- /etc/hosts.allow アクセスを許すコマンドやデーモンの組合せを記述する。
- /etc/hosts.deny アクセスを許さないコマンドやデーモンの組合せを記述する。

アクセス制御の優先順位は次のようになっている。

1. まず /etc/hosts.allow を検索する。/etc/hosts.allow の記述にアクセスが許可されていれば、それはアクセス許可となる。
2. /etc/hosts.allow で一致しなかった組合せについて、/etc/hosts.deny を検索する。もし検索して一致する組合せがあれば、そのアクセスは不許可となる。
3. /etc/hosts.allow と /etc/hosts.deny に記述の無い組合せはアクセス許可となる。

/etc/hosts.allow と /etc/hosts.deny の書式は同じである。基本的には、行単位に記述し、空行や # で始まる行は無視される。1 行が長くなる場合、\ を使って継続行の記述ができる。そして各行は“ : ”(コロン) で区切られた、3 つのフィールドからなる。

デーモンリスト : クライアントリスト : シェルコマンド

シェルコマンド部は省略可能なので、省略した場合、デーモンリストとクライアントリストのみとなる。デーモンリストは inetd から起動する argv[0] に相当するコマンド名 (デーモン名) であり、クライアントリストは、クライアントのホスト名、ドメイン名、ネットワーク番号や、ホストの IP アドレスを記述することができる。いずれも空白で区切って、複数記述できる。

次の例は、/etc/hosts.allow の記述例である。

```
in.fingerd : 133.4.0.0/255.255.0.0 192.234.32.0/255.255.255.0
in.telnetd : 133.4.0.0/255.255.0.0 EXCEPT 133.4.11.11
```

ここで、n.n.n.n/m.m.m.m の形式は、ネットワークとその有効マスクを表している。1 行目は、fingerd に対するコネクションは、 133.4.0.0 のネットワークと 192.234.32.0 のネットワーク内のホストからの接続を許可する設定である。 2 行目は telnetd に対する記述例であるが、133.4.0.0 のネットワークのホストからの接続は、133.4.11.11 のホストを除いて、許可するという意味である。もちろん、上記以外のホストからのアクセスについては、/etc/hosts.deny をどう記述するかによって挙動が変わるので、注意する。

クライアントリストには、ドメイン名での記述も可能である。foo.co.jp というドメインに所属するすべてのホストを指定するには、 .foo.co.jp のように先頭に“ . ”(ピリオド) を付けて示す。もし先頭に“ . ”を付けないと、ドメイン全部ではなく、foo.co.jp というホストとなる。

また次のキーワードも利用できる。

- ALL すべてにマッチする。
- LOCAL “ . ”を含まない文字列 (localhost など)
- UNKNOWN ホスト名やアドレスが不明のもの。
- KNOWN ホスト名やアドレスが明確なもの。UNKNOWN と KNOWN は、一時的なネームサーバーのトラブルによって動作が不安定になることがあるので、使う場合は、十分注意すべきであろう。
- EXCEPT “ リスト 1 EXCEPT リスト 2 ”と記述すると、リスト 1 に含まれるもののうちリスト 2 に含まれるものを除くとなる。また a EXCEPT b EXCEPT c は、(a EXCEPT (b EXCEPT c)) の意味になる。

ALL を使って、/etc/hosts.deny に次のように記述すると、tcpd を設定したデーモンは、すべて接続拒否となる。

```
ALL: ALL
```

tcpd のログ機能には、syslog に起動したアクセスのあったホストと、デーモンの名前を残す事ができるが、そのようなログよりは、シェルコマンドを組み合わせるのが強力である。シェルコマンドには、表 3.2.2に示すマクロが利用できる。

```
/etc/hosts.allow
    ALL: ALL : (echo %d connect from %h | /usr/ucb/Mail root) &
```

この例では、すべてのアクセスを許可するが、アクセスがあった場合は、すべて root 宛てに、そのデーモンとホストをメイルするという意味になる。

表 3.4: シェルコマンドで記述できるマクロ

%a	アクセスしたホストのアドレス
%c	アクセスした user@host の形式
%h	アクセスしたホスト名
%d	デーモンのプログラム名
%p	デーモンのプスセス ID
%u	アクセスしたユーザー名
%%	% 自身の記述

3.2.3 設定の注意

TCP wrapper は、FTP や TELNET 等のネットワークサービスに制限を加えるものなので、セキュリティの向上という点では、優れている。しかし、設定を誤ると、二度とネットワーク経由での修正ができなくなる可能性があるので、設定には十分注意する必要がある。

tcpd はその性格上、IP アドレスとホスト名の相互変換のために、ネームサーバに対するアクセスを厳重に行なうようなコーディングになっている。ネームサーバーを正しく設定してないと、ドメイン名を利用したコンフィグはできないので、ネームサーバーの設定に誤りが無い事を十分確認する必要がある。

TCP wrapper には、try という/etc/hosts.allow と/etc/hosts.deny の設定のテストツールが付属しているので、実際に運用を行なう場合は、十分にテストしてから運用するのがよいであろう。

3.3 UDPrelay

UDP 利用したもので、ユーザーの使用頻度が高いインターネット上のアプリケーションに、archie がある。Firewall を構築するためにゲートウェイで IP のフォワーディングを止めると、当然のことであるが、Firewall の内側からの archie の実行は、通常の手段では不可能になる。UDPrelay は Firewall 上で動作し、UDP の中継を行なうアプリケーションゲートウェイである。UDPreplay の設定を正しく行なうことで、Firewall の内側からの archie が実行可能となる。ここでは archie を使う場合の UDPrelay の設定を紹介する。

UDPrelay は、実際に UDP データを中継するサーバー部と、そのサーバーに UDP パケットを送るクライアント部に分けることができる。サーバー部 (UDPrelay) は、Firewall 上でデーモンとして動作し、クランアントから受けた UDP パケットを、必要に応じてフォワードする。

サーバーを動作させるには、/etc/udprelay.conf という設定ファイルを用意し、firewall のマシンで他の一般のコマンドのように起動する。inetd からの起動はできないので注意すること。

```
# udprelay &
```

実際に運用する場合は、rc.local 等から起動するように設定すべきであろう。

```
if [ -f /usr/etc/udprelay -a -f /etc/udprelay.conf ]; then
    udprelay &
fi
```

UDPrelay は、2 種類のコンフィグレーションを設定することができる。1 つは、UDPrelay 専用のクライアントライブラリを使うように、クライアント側プログラム (例えば archie) を変更して利用するのである。この場合、クライアントの UDP データは、クライアントライブラリによって encapsulate され UDPrelay の専用のポートに届く。 UDPrelay は、encapsulate されたデータを受け取り、展開して本来のサーバーにそのデータを中継する。この設定をする場合は、次の行を /etc/udprelay.conf に記述する。

```
encapsulate *.foo.co.jp 1092
```

この例は foo.co.jp ドメイン内の全マシンから UDPrelay へのアクセスを許すものである。

しかし、UDPrelay の専用クライアントを使うためには、アプリケーションを改造する必要がある。これには、udprelay に付属の、Rsendto と Rrecvfrom のライブラリを利用し、アプリケーションの初期化時に、UDPrelayinit() を呼び、sendto() を Rsendto() に、recvfrom() を Rrecvfrom() に置き換えればよい。

もう 1 つのコンフィグレーションは、UDPrelay の単純なリレー機能を利用することである。この方法を使って archie の UDP データをリレーする場合、/etc/udprelay.conf には次のように記述する。

```
relay  *.foo.co.jp  *  1525  archie.kyoto-u.ac.jp  1525  any
relay  *.foo.co.jp  *  1526  archie.wide.ad.jp     1525  any
```

1 行目の記述は、foo.co.jp ドメインの任意のマシンの任意のポートから、UDPrelay の動いているマシンのポート 1525(archie サーバーのポート番号) の UDP のデータを、archie.kyoto-u.ac.jp のポート 1525、つまり jp ドメインの archie サーバーに中継するものである。2 行めは同様に、1526 のポートの UDP データを、archie.wide.ad.jp の 1525 のポート、つまり jp ドメインのワールドワイドな archie サーバーに接続する方法である。

この方法を使った場合 archie コマンドでは、サーバーとして、Firewall のマシンを指定する事になり、あたかも Firewall 上のマシンが、 archie サーバーとなったように見える。また archie コマンドの接続ポートを変える事で、別の archie サーバーを検索することができるので、比較的簡単な設定であるといえる。

現在 UDPrelay 専用のクライアントプログラムは用意せず、後者の設定で運用している。archie コマンドは、デフォルトのサーバーを UDPrelay の動いているのマシンとし、−server オプションの変わりに、Firewall の UDPrelay へのポート番号を変えるための、−world オプションを用意して、ワールドワイドな archie サーバーを検索できるようにしている。

もともと UDPrelay を使った理由が、Firewall の内側から archie を実行したいという要望に基づくもので、UDP をリレーの機能で十分目的が果たせたためである。このとき手元で使っていた archie は、perl で記述されたものだったので、簡単には UDPrelay 専用のクライアントライブラリ (C で記述されている) を利用できなかったというのも理由の一つである。

3.4 Gophergate

3.4.1 目的と概要

Gophergate は、gopher プロトコル [142] を中継するためのアプリケーションゲートウェイで、主に次の 2 つの目的を達成するために作成された。

IP パケットを通過させないゲートウェイ上で gopher プロトコルの中継を行うことで、組織内のホストから外部の gopher サーバをアクセス可能にする。

データをキャッシュすることで、クライアントの応答性を向上させ、インターネット上に流れる無駄なトラフィックを削減する。

Gophergate は、ゲートウェイの計算機上でデーモンとして動作する。しかし、自身ではデータを保持せずに、すべて外部の gopher サーバへクライアントからの要求を中継することでサービスを実現する。

Gopher プロトコルでは、空文字列の SELECTOR (gopher プロトコルで、サーバ上のデータを指定するための文字列) をサーバに送ることで、最上位のディレクトリ情報を要求することができる。Gophergate は、空の SELECTOR を受け取ると、デフォルトのサーバの最上位ディレクトリの情報を返す。

実際のサーバから返ってきたディレクトリ情報は、そのままクライアントに渡すのではなく、自分自身がサービスできるような形に変換する。実際のサーバに関する情報を SELECTOR の中に埋め込むことで、後にクライアントから要求されたときに、実際のデータに関する情報を得ることができる。

一般に gopher で提供されるデータは、更新される頻度が比較的低いものが多い。しかし、クライアントはそれぞれ直接サーバをアクセスしてデータを読むため、1 つのデータがネットワーク上の同じ経路を何度も流れることも少なくない。そのため gophergate は、ゲートウェイ上でデータをキャッシュする機能を持つ。一度アクセスしたデータはゲートウェイのディスク上にキャッシュされ、二度目以降のアクセスに対してはキャッシュデータを送り返すことで、応答性を向上し、ネットワークのトラフィックを軽減する。

3.4.2 設定方法

Gophergate は、ゲートウェイ計算機上でデーモンとして動作する。起動は /etc/rc.local などから行い、現在のところ inetd から利用する方法は用意されていない。

設定ファイルはなく、すべてコマンドオプションとして指定する。利用できるオプションを表 3.5に示し、主な設定方法について説明する。

ポート番号の指定

Gophergate が使用するポート番号を指定することができる。指定がなければ 70 番が使用される。同じホストで gopher サーバが動作している場合には、当然それ以外の番号を使用しなければならない。

オプション	意味
-p port_num	ポート番号の指定 (指定がなければ 70 番を使用)
-h hostname	ホスト名の指定
-c directory	キャッシュディレクトリ
-s server	デフォルトの gopher サーバ (指定がなければ gopher.nic.ad.jp)
-l logfile	ログファイル
-a addr/mask	受け付け許可アドレス
-e expire	キャッシュの有効期間
-u user/uid	実行権限の指定
-i	ident プロトコルを使用
-I	ident サポートを強制
-C time	time 以上経過したキャッシュファイルを削除

表 3.5: gophergate のオプション

ホスト名の指定

Gophergate がクライアントに返すディレクトリエントリのホスト情報に使われる名前である。指定がなければ hostname コマンドの結果が利用される。hostname によって得られる名前が、組織内のホストからアクセスできない場合には、別の名前を指定する必要がある。

キャッシュディレクトリの指定

オプションでキャッシュディレクトリが指定されると、データのキャッシングが行われる。この指定がないとキャッシュは作成されない。

受け付け許可アドレス

Gophergate は、セキュリティ上の配慮からアクセスを受け付けるクライアントのアドレスを制限している。指定がない場合には、ゲートウェイの主 IP アドレスと同じネットワーク (natural mask による) に属するホストからのアクセスしか許さない。

これ以外の指定を行う必要がある場合には、-a オプションを使用して、アドレスとマスクを指定する。たとえば、255.255.255.0 のサブネットマスクを使っているネットワークで、133.4.10.0 と 133.4.20.0 の 2 つのサブネットからのアクセスのみ許すためには、次のように指定することができる。

```
-a 133.4.10.0/255.255.255.0,133.4.20.0/255.255.255.0
```

キャッシュ有効期限の指定

キャッシュファイルの有効期限を指定することができる。作成されてからこの期間を過ぎたファイルは、たとえキャッシュディレクトリ中に存在しても参照されない。

ident プロトコルの利用

–i オプションを指定することで、ident プロトコルを使用することができる。Gophergate は、ident プロトコルを使ってクライアント側のユーザ名を求め、それをログ情報に書き出す。–I オプションが指定されると、クライアントが ident プロトコルをサポートしていない場合には接続を受け付けない。

3.4.3　Gophergate の仕組み

Gophergate は、空の SELECTOR を受け取ると、デフォルトで定義されている Gopher サーバからデータを手に入れる。この中に、次のようなエントリがあったと仮定する。

```
TYPE = 1
NAME = JPNIC Publications
SELECTOR = 1/ftp
HOST = gopher.nic.ad.jp
PORT = 70
```

Gophergate は、サーバから受け取ったこの情報を次のように書き換えてクライアントに渡す (gophergate は`gophergate.wide.ad.jp` というホスト上の 7777 番のポートで動いているとする)。

```
TYPE = 1
NAME = JPNIC Publications
SELECTOR = tab=#;#1#1/ftp#gopher.nic.ad.jp#70
HOST = gophergate.wide.ad.jp
PORT = 7777
```

HOST と PORT は自分自身のものに変更し、SELECTOR の中に実際のデータの情報を埋め込んでしまう。SELECTOR は `tab=#;` という文字列ではじまる。実際には、`#` の部分はどんな文字でもよく、もとのデータで使われていない文字が使用される。RFC1436 では、名前と SELECTOR に使用できる文字は、次のように定義されている。

```
UNASCII   ::= ASCII - [Tab CR-LF NUL].
User_Name ::= {UNASCII}.
Selector  ::= {UNASCII}.
```

したがってタブ、改行、復帰、ヌル以外のすべての文字が使われる可能性がある。また、クライアントと実際のサーバとの間に、2 つ以上の gophergate が介在することも原理的には可能なので、特定の文字を使用することはできない。

tab=#; の後にはここで定義されたタブ文字 (この例では #) で区切られた、以下に示す 5 つ以上のフィールドが並ぶ。

1 未使用
2 タイプを表す文字と名前
3 セレクタ
4 ホスト
5 ポート番号

ただし、2 番目のフィールドに名前情報を入れる必要はないので、通常はタイプを表す文字だけが入る。この情報を受け取ったクライアントは、次にホストgophergate.wide.ad.jpのポート 7777 番に次の SELECTOR を送る。

```
tab=#;#1#1/ftp#gopher.nic.ad.jp#70
```

これを gophergate が受けて、埋め込まれた情報にしたがって実際のサーバのデータをアクセスして送り返す。この例では、要求された情報はディレクトリであるから、上のような書き換えを再び行ってクライアントに送り返すことになる。

このようにして、ユーザには実際のホストにアクセスしているのと、まったく同様な環境が提供される。

3.4.4 キャッシュの仕組み

キャッシュは、オプションで指定されたキャッシュディレクトリの下に作成される。キャッシュファイルは、セレクタの中の/ をディレクトリの区切りとして、キャッシュディレクトリの下にディレクトリ構造を作る。最上位のディレクトリはホスト名とポート番号をコロン (:) で連結したものである。例として、次のデータを考える。

```
TYPE     = 0
SELECTOR = 0/ftp/INDEX
HOST     = gopher.nic.ad.jp
PORT     = 70
```

このデータのキャッシュファイルは、キャッシュディレクトリの下に、以下のようなパスで作成される。

```
gopher.nic.ad.jp:70/_0/_ftp/0INDEX
```

中間ディレクトリの名前の先頭にアンダースコア (_) が追加されているのは、SELECTOR としては、// と / を区別する必要があるからである。また、最後のファイル名の先頭には TYPE を表す文字を付加する。

キャッシュしなくない、あるいはしてはいけないファイルは、一度作成されたファイルの read の権限を落としておくことで、キャッシュを禁止することができる。また、特定のディレクトリ以下のキャッシュをすべて禁止するのは、ディレクトリへの書き込みや実行の権限を落とすことで実現できる。

3.4.5　問題点と課題

Gophergate は、作成されてまだ日が浅いため、まだ十分に使い込まれておらず、機能の拡張も継続して行われている。現時点で考えられる問題点として、以下のような点が挙げられる。

キャッシュの整合性 キャッシュと実際のデータの整合性を取る方法が存在しない。現在の gopher プロトコルでは対応できないが、gopher+ プロトコルを使ってキャッシュが最新であるかどうかを調べる仕組みを追加できる可能性はある。

サーバの意志の反映 gopher サーバの運用方針によっては、キャッシュをして欲しくないようなデータも存在するし、キャッシュの有効期間を指定できることもある。現在の枠組みでは、これらのサーバが持つ意志をゲートウェイに反映させる方法は存在しない。今後新しいプロトコルの開発も含め、考慮していかなければならない。

アクセス制限 現在は、アドレスとマスクの組でのみアクセス制限を行うことができる。ユーザや接続サーバなど、より極めの細かい制限を行う必要があるかもしれない。

他のプロトコルへの対応 Gophergate は、gopher プロトコルのみの中継とキャッシュを行うプログラムである。同様な機構を他のプロトコルに対しても提供することは十分考えられる。

3.5 Screend

専用ルータの代りに UNIX ワークステーションをゲートウェイとして使用する場合がある。専用ルータにはフィルタリング機能が装備されている場合が多く、プロトコルやホスト単位でアクセス制限を設定することができるが、UNIX ワークステーションの多くはフィルタリング機能が装備されていない。このため、アクセス制限を設けたいゲートウェイとして UNIX ワークステーションを用いることは一般的に不向きとされている。ここでは、UNIX ワークステーションでのフィルタリング機能の実装例として Screend[143] を紹介する。

3.5.1 Screend の構成

Screend は DEC 社の Western Research Laboratory に所属する Jeffrey C. Mogul 氏によって開発された。パッケージは anonymous ftp によって公開されていて、gatekeeper.dec.com から入手できるようになっている。Screend の一般公開が開始されたのは 1990 年の春だが、その後いくつかの細かい修正を経て、1993 年の初頭に公開されたものが広く普及している。

パッケージは大別すると 2 つの構成に分けられる。1 つはカーネルにフィルタリング機能を組み込むために必要な追加および修正に関する部分で、もう 1 つはカーネルに組み込まれたフィルタリング機能を制御するためのコマンド類である。

カーネルの修正

カーネルに対する修正は、既存のカーネルソースに対する修正と、新規に追加するモジュールの 2 つの部分に分けられる。Screend を使用するためには Screend のパッケージに含まれている 2 つのソースファイルをカーネルに組み込まなければならない。追加するファイル名と概略を表 3.6に示す。この追加されたソースコードの機能をカーネルに認識させるために、いくつかの既存のソースに対して修正を加える必要がある。この修正はパッチファイル形式でパッケージに含まれている。修正が必要なソースファイル名を表 3.7に示す。

ファイル名	概要
gw_screen.c	プロトコルに依存しない汎用フィルタ
ip_screen.c	IP プロトコルに依存した処理ルーチン

表 3.6: カーネルに追加するソース

コマンド

ファイル名	修正内容
net/if.c	Screend が使用する ioctl 命令を追加
netinet/ip_input.c	フィルタリング機能を経由して IP パケットを forward するように ipintr() ルーチンを修正

表 3.7: 修正が必要なソース

Screend のパッケージにはいくつかのコマンドが含まれている。中でも重要なものを表 3.8に示す。

コマンド名	機能
screenmode	フィルタリング機能を使用するかどうかを決定する
screend	daemon プロセスとして常駐し、カーネルのフィルタリングを制御したりログを取ったりする
screenstat	フィルタリングに関する統計情報を表示する

表 3.8: コマンド一覧

これらのコマンドはすべてソースコードで提供されているので、Screend の使用に先だってこれらのコマンドをコンパイルし、適当なディレクトリにインストールしておく必要がある。

3.5.2　Ultrix 上での Screend

Screend の配布パッケージには Ultrix 3.x で Screend を動かすためにカーネルに追加しなければならないモジュールのソースコード一式と、既存のカーネルのソースコードに対する変更部分がパッチファイル形式で含まれている。これらのファイルを付属のドキュメントの指示に従ってインストールすれば、フィルタリング機能が使えるようになる。しかし、残念ながら既存のカーネルモジュールに対する修正が必要なため、Ultrix 3.x で Screend を使うためには OS のソースライセンスを取得している必要があった。この問題を回避するために、Ultrix 4.0 からは Screend を使うために必要な修正が OS の中に取り込まれた状態で出荷されるようになった。Screend を使いたいユーザは OS をインストールするときにオプションパッケージの Screend を選択しカーネルを reconfig するだけで Screend が使えるので、OS のソースライセンスを取得する必要はなくなった。また、表 3.8に示したコマンド類も実行形式のファイルが提供されているので、自分でソースコードをコンパイルする必要もない。なお、これと同様の仕組みは AXP OSF/1 にも組み込まれている。

3.5.3 BSD/386 上での Screend

Screend はほとんどの部分がソースコードで配布されている。したがって、移植の対象となる OS のソースコードが入手できるのであれば、他機種への移植はそれほど難しくない。Ultrix 3.x のネットワーク関連のコードは 4.3BSD 相当の仕様になっているため、4.2BSD あるいは 4.3BSD に準拠した OS であればほとんど修正することなく Screend を移植できるだろう。ただし、NET/2 という名称で広く知られている 4.3BSD Reno 版以降の BSD では mbuf の構造が変わったため、ip_screen.c と gw_screen.c にかなりの修正を加えなければならない。カーネルに関してはこの他にも if.c と ip_input.c に対する修正が必要だが、if.c に対する変更は Screend 用の ioctl 命令を数行追加だけでよく、また ip_input.c に対する変更も ipintr() 内で ip_forward() を読んでいる部分を書き換えるだけなので、これらのソースコードへの変更はどちらもそれほど大変な作業ではない。

BSD/386 版の Screend のパッケージ構成は Ultrix 版のパッケージとほぼ同じ構成になっている。BSD/386 1.0 を使用しているのであれば、Ultrix 3.x に対するインストール手順とほぼ同じ手順の作業を行うことによって Screend を使えるようになる。1994 年の 2 月にリリースされた BSD/386 1.1 では、Screend をサポートするための仕組みがカーネルに装備された状態で出荷されている。BSD/386 1.1 の場合は既存のカーネルソースに対する修正を行う必要はなく、Screend のパッケージに含まれているソースコードをコピーしてカーネルを reconfig するだけで Screend が使えるようになっている。また、この BSD/386 対応版のパッケージでは既存の機能に加えて、TCP/IP のフラグによるフィルタリング機能が付加された。この機能によって、従来の Screend では実現が困難だった TCP/IP のセッションの方向によるアクセス制御が容易に行えるようになった。

3.5.4 使用例

ここでは BSD/386 に実装された Screend を例に取って簡単な設定例を紹介する。Screend を使うためには、ifconfig コマンドによってネットワークインターフェースが初期化される前に、screenmode コマンドを使用して screend が使えるようにしておく必要がある。ネットワークインターフェースを初期化した後で screend を起動すると、カーネルに実装されたフィルタリング機能が働きだすようになっている。BSD/386 の場合、これらの設定は/etc/netstart というファイルで行うのが一般的である。設定例を図 3.1に示す。

screend は、/etc/screend.conf に書かれている設定に従って動作するようになっている。設定ファイルのパスを明示的に指定したい場合には-f オプションを用いる。ネットワークインターフェースが初期化されてから、screend が起動されるまでの間にこのホストを通過しようとしたパケットは screend が起動されるまで forward されない。したがって、何等かの理由によって screend を止めた場合や screend の使用を中止する場合には、screenmode コマンドを使用してカーネルのフィルタリング機能を止めておかないとパケットが forward されなくなる。

'#' 以降の文字列と/* */によって囲まれた文字列はコメントとみなされる。すべての命令は ';' によって終ってさえいれば複数行にまたがって記述してもかまわない。default,

```
# enable IP screening before any ifconfig's are done.
screenmode on

# initialize interfaces
ifconfig ef0 inet 133.156.1.254 netmask 255.255.255.0
ifconfig ef1 inet 133.156.2.254 netmask 255.255.255.0
ifconfig lo0 inet 127.0.0.1

#  run screend after all ifconfig's are done.
screend -L /var/log/screend.log
```

図 3.1: /etc/netstart の設定例

notify, reject などのようなキーワードは小文字で記述しなければならない。ホスト名は名前で指定しても IP アドレスで指定してもいいが、screend が起動された時点ではまだ named が使えない場合が多いので、IP アドレスで記述しておいた方が安全である。ホスト名が reject や default などのような予約されているキーワードと同じ名前の場合には大文字でホスト名を指定すればよい。IP アドレスやネットマスクなどの数値は 10 進数でも 16 進数でも指定できる。screend.conf の設定例を図 3.2に示す。

screend.conf では、まず最初に screend のデフォルトの動作を設定しなければならない。つまり、基本的にすべてのパケットを forward して、ある特定のパケットだけを forward しないように設定するのか、基本的にすべてのパケットを forward しないようにして、特定のパケットだけを forward するように設定するのかを宣言しなければならない。この宣言には default コマンドを用いる。図 3.2の設定では基本的にパケットを forward しないと宣言し、その後に続く設定によって特定のパケットだけを forward するように設定している。デフォルトの動作を reject にした場合に notify オプションを指定すると、forward されなかったパケットに対して ICMP Host Unreach パケットが返送されるようになる。log オプションを指定すると、forward されなかったパケットがログファイルに記録される。ログは screend を起動したときに-L オプションで指定したファイルに記録される。

もし設定ファイル内で subnet コマンドを使用する場合には、default コマンドに引き続いて subnet コマンドでサブネットマスクの設定を行わなければならない。subnet コマンドは必要に応じて複数回使用できる。from や between コマンドなどのようなアクセスを制限するコマンドの詳細については Screend パッケージに付属しているドキュメントを参照してもらいたい。

3.5.5 性能評価と問題点

BSD/386 1.1 に Screend をインストールし、このホストを経由して NFS や ftp などのパフォーマンステストを行ってみたが、フィルタリング機能を使用したことによるパフォーマンスの低下は見た目には感じられなかった。正確な測定データがあるわけではないが、フィルタリング機能を使用したときと使用しなかったときの Round Trip 値の差が約 2ms であることから、フィルタリング機能を使うことによってパケットを forward する処理が約 1ms くらい遅くなっていると推測できる。実用上問題ない速度で処理されていると考えてかまわないだろう。

IP ヘッダにはいくつかのオプション機能があるが、このオプション機能の中には LSRR[1]や SSRR[2]のように Screend のフィルタリング機能を通り抜けて不正にアクセスする手段を提供してしまうものがある。現在の Screend の実装では IP ヘッダオプションをチェックして forward していいかどうかを調べるのが困難なため、IP オプションが含まれているパケットは無条件に forward しないように設計されている。この仕様が問題になるケースは少ないと思われるが、たとえば VIP ワーキンググループによって研究されている移動ホストをサポートするための研究では IP ヘッダのオプション領域を利用して IP ヘッダを拡張しているため、VIP パケットは Screend を使用したゲートウェイを通過することができないという問題が生じる場合もある。

[1] Loose Source Route
[2] Strict Source Route

```
/*
 * screend.conf の設定例
 */

# 基本的にパケットを forward しない
default reject notify log;

# subnet 命令のためにネットマスクを指定する
for 133.156.0.0 netmask is 255.255.255.0;

# 組織内のトラフィックは forward する
between net 133.156.0.0 and net 133.156.0.0 accept;

# 特定のホストだけ組織外へアクセスできるようにする
between host 133.156.2.250 and any accept;

# 特定の subnet だけ組織外へアクセスできるようにする
between subnet 133.156.10.0 and any accept;

# どのホストからでも NiftyServe への telnet だけは許可する
# r2.niftyserve.or.jp [192.47.24.133]
between host 192.47.24.133 tcp port telnet
                and net 133.156.0.0 accept;

# ICMP send-redirect 以外の ICMP パケットは forward する
from any icmp type-not redirect to
                any icmp type-not redirect accept;
```

図 3.2: screend.conf の設定例

3.6 Wellfleet のルータのパケットフィルタの設定について

3.6.1 概要

Wellfleet の製品では、パケットフィルタのことをトラフィックフィルタと呼ぶ。以下ではトラフィックフィルタという言葉を用いて説明する。Wellfleet のルータ (Version 5.75) のトラフィックフィルタには以下の特徴がある。

- インタフェースから入ってくるパケットについて通過権をチェックする。
- トラフィックフィルタの評価は、入力順ではなく Precedence 順に行なわれる。
- トラフィックフィルタの指定では、ソースとデスティネーションの両方に関して IP Address と Port 番号を指定できる。
- ルータ自体へのアクセス権も通常のトラフィックフィルタの機能を使って指定 する。
- 通過を許可したパケット以外のパケットの通過の禁止は、陽に指定しないと いけない。
- IP Address や Port 番号のリストを別に定義し、定義したリスト名でトラフィックフィルタが書ける。
- ルーティング情報のフィルタ機能とは別に、rip port からのパケットの受信を トラフィックフィルタの機能で別途許可しておかないといけない。

3.6.2 設定例

ネットワークの構成は図 3.3のようになっていると仮定する。

ルータに接続するとトップメニュー (図 3.4) が表示される。トップメニューで 3 番を選ぶと、Configuration Editor メニュー (図 3.5) が表示される。ここで 7 番を選ぶと DoD Internet Router メニュー (図 3.6) が表示される。

1. リストの定義

   リストを定義しなくてもトラフィックフィルタの設定はできるが、定義しておいたほうが、わかりやすいし、設定が簡単になることが多い。

   DoD Internet Router メニュー (図 3.6) で 1 番を選ぶと Lists メニュー (図 3.7) が表示される。Lists メニュー (図 3.7) で 1 番を選び、IP Address Lists の一覧 (図 3.8) を表示する。ここで例えば 1 番の List の内容は図 3.9のようになっている。IP アドレスの指定は単体、範囲指定、またはその混在ができる。それぞれの List の意味は表 3.9の通りである。

   次に、Lists メニュー (図 3.7) で 2 番を選ぶと、IP Port Lists メニュー (図 3.10) が表示される。ここで 1 番の IP Port List の内容は図 3.11のようになっている。この

リスト名	意味
CENTER	社内からゲートウェイ LAN へ接続する特定ホスト群
GW_LAN	ゲートウェイ LAN 上の任意のホスト群
SERVER	社内メール／ニュースサーバ
SOCKSGATE	socks サーバ
OUTSIDE	ルータの社外側インタフェースのアドレス
INSIDE	ルータの社内側インタフェースのアドレス

表 3.9: IP アドレスの List の意味

リスト名	意味
FTP_TELNET	ftp と telnet のサーバの Port 番号
GT1023	1024 以上の Port 番号
RIP	rip の Port 番号
MAIL_NEWS	smtp と nntp の Port 番号
SOCKS	socks の Port 番号
TELNET	telnet のサーバの Port 番号

表 3.10: Port 番号の List の意味

Port List は telnet と ftp のサーバ側の Port 番号を指定したものである。IP Port 番号の指定も単体、範囲指定、またはその混在ができる。

2 番の IP Port List の内容は図 3.12のようになっている。これは telnet や ftp のクライアント側の Port 番号を指定したものである。一般的には 1024 以上の任意の Port 番号が使われるが、一部の 1024 番以上の Port で待っているサーバの番号を除外するようにしている。それぞれの List の意味は表 3.10の通りである。

2. トラフィックフィルタの設定

ここでは、以下のパケットのみ通過するように設定する。

(a) 社内の特定のホスト群 (CENTER) からゲートウェイ LAN 上の任意のホスト群 (GW_LAN) への telnet と ftp

(b) 社内の特定のホスト群 (CENTER) とゲートウェイ LAN 上の任意のホスト群 (GW_LAN) との間の icmp

(c) 社内メール／ニュースサーバ (SERVER) とゲートウェイ LAN 上の任意の ホスト群 (GW_LAN) との間の smtp と nntp

通過する packet	1 番の Filter	2 番の Filter
(a)	3 番	3 番
(b)	4 番	4 番
(c)	6 番,7 番	6 番,7 番
(d)	8 番	8 番
(e)	-	5 番
(f)	5 番	-
(g)	2 番	2 番
(h)	1 番	1 番

表 3.11: 通過する packet と各インタフェースの Traffic Filter の対応

(d) 社内任意ホストから socks サーバへの socks

(e) 社内の特定のホスト群 (CENTER) から Wellfleet ルータの社内側インタ フェース (INSIDE) への telnet

(f) Wellfleet ルータの社外側インタフェース (OUTSIDE) からゲートウェイ LAN 上の任意のホスト群 (GW_LAN) への telnet

(g) 任意のホストからの rip の packet の受信 (通過) を許可する

(h) その他の全ての packet の通過を禁止する

DoD Internet Router メニュー (図 3.6) で 2 番を選ぶと Network Interface Definitions メニュー (図 3.13) が表示される。ここで 1 番を選ぶと、1 番のインタフェースの設定内容 (図 3.14) が表示される。さらに 1 番を選ぶと、1 番のインタフェースの Traffic Filter のリスト (図 3.15) が表示される。また、Network Interface Definitions メニュー (図 3.13) で 2 番を選ぶと、2 番のインタフェースの設定内容 (図 3.16) が表示され、ここで 1 番を選ぶと、2 番のインタフェースの Traffic Filter のリスト (図 3.17) が表示される。

上記の通過する packet 内容と 1 番のインタフェース、2 番のインタフェースのそれぞれの Traffic Filter の対応は表 3.11の通りである。

図 3.18は 1 番のインタフェースの 1 番の Traffic Filter の詳細な設定内容である。これではソース、デスティネーション、プロトコルに関係なく全ての packet を落す (Drop) 設定になっている。他のすべての Traffic Filter に該当しなかった場合、一番最後に packet を落す必要があるため、Precedence を 1 とし、一番最後に評価する必要がある。この内容は、2 番のインタフェースの 1 番の Traffic Filter の設定内容と全く同じである。

図 3.19は 1 番のインタフェースの 2 番の Traffic Filter の詳細な設定内容である。これでは、任意のホストの RIP の Port からきて任意のホストの RIP の Port へ送られ

る全ての packet を通す (Accept) 設定にしている。この内容も、2 番のインタフェースの 2 番の Traffic Filter の設定内容と同じである。

図 3.20は 1 番のインタフェースの 3 番の Traffic Filter の詳細な設定内容である。2 番のインタフェースの 3 番の Traffic Filter の設定内容とは、ソースとデスティネーションが反対になっている。

図 3.21と図 3.22は 1 番のインタフェースの 4 番の Traffic Filter の詳細な設定内容である。Wellfleet では、ICMP packet を直接指定することができないので、IP packet 中の bit 位置と bit 長とその値によって ICMP packet を指定している。1 番のインタフェースの 4 番の Traffic Filter の設定内容とは、ソースとデスティネーションが反対になっている。

1 番のインタフェースの 5 番の Traffic Filter と 2 番のインタフェースの 5 番の Traffic Filter は、ルータから外への telnet と、内からルータへの telnet なので、設定内容が対象にはなっていない。

1 番のインタフェースと 2 番のインタフェースの 6 番と 7 番の Traffic Filter は、smtp と nntp のそれぞれのサーバ間の接続を認めるためのもので、例では、社外側のサーバはホストを特定せず、ゲートウェイ LAN 上のすべてのホストと通信できる設定にしている。また、これらメール／ニュースサーバは両方向から接続する必要があるため、両方向に接続できるように設定している。その他のサービスは片方向の接続だけなので一方向だけから接続できる設定になっている。

3.6.3　CISCO のルータのパケットフィルタの特徴

CISCO のルータ (Version 8.2(5)) のパケットフィルタには次の特徴がある。

- インタフェースから出ていくパケットについて通過権をチェックする。
- パケットフィルタは入力順に評価される。
- IP アドレスはソースとデスティネーションの両方に関して指定できるが、 Port 番号はデスティネーションに関してのみ指定できる。
- ルータ自体へのアクセス権はパケットフィルタの機能とは別の機能で指定する。
- パケットフィルタで許可指定されていないパケットは全て禁止指定がしてあると みなされる。

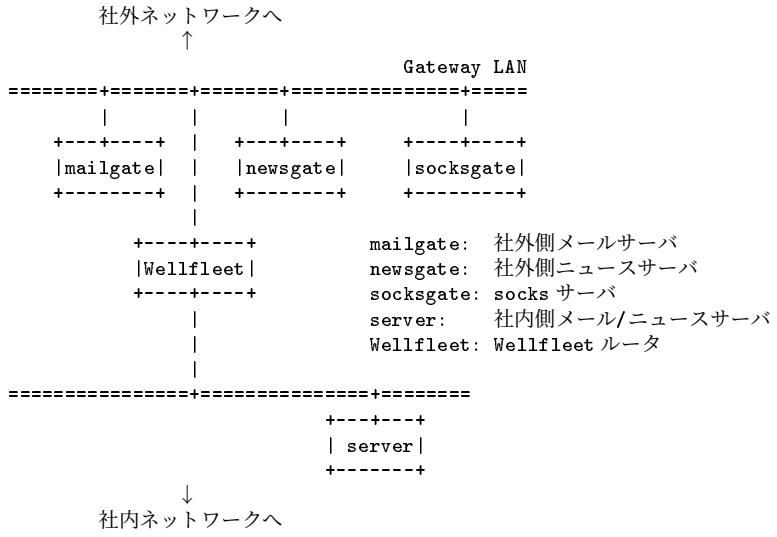

図 3.3: ネットワークの構成

```
-------
Wellfleet Communications, Inc.    WIDE_FNET                5-Apr-1994  15:18:12

============================-  SESSION 2 - MGR MODE -============================
                                 Main Menu

               1.  Statistics Screen Menu
               2.  Network Control Language Interpreter
          -->  3.  Configuration Editor
               4.  Event Log
               5.  LOGOUT









PRESS: ? for help, Down, Up, <- to exit, <RETURN> to select


-------
```

図 3.4: トップメニュー

```
-------
Wellfleet Communications, Inc.    WIDE_FNET                5-Apr-1994  15:19:34

============================-  SESSION 2 - MGR MODE -============================
Configuration Editor 1.19              Current File : CONFIG


1.  System (1)
2.  Software (1)
3.  Lines (2)
4.  Circuits (2)
5.  Circuit Groups (2)
6.  Bridge (0)
7.  DoD Internet Router (1)
8.  DECNET IV Routing Service (0)
9.  SNMP Sessions (2)
10. Xerox Routing Service (0)
11. IPX Routing Service (0)
12. AppleTalk Router (0)
13. X.25 Network Service (0)


Enter Selection (0 for Previous Menu) : ____


-------
```

図 3.5: Configuration Editor メニュー

```
-------
Wellfleet Communications, Inc.    WIDE_FNET                5-Apr-1994  15:20:32

============================-  SESSION 2 - MGR MODE -============================
Configuration Editor 1.19              Current File : CONFIG
Auto Enable          : Yes             Global Broadcast     : Yes
RIP Network Diameter : 15              Mode                 : Router/Host
Management Priority  : Low             Non Local ARP Source : Drop and Log
Bootp Gateway Support   : No           Bootp Gateway Auto Enable  : Yes
Bootp Gateway Max Hops  : 3            Suppress Authentication Traps : Yes


1. Lists (1)
2. Network Interface Definitions (2)
3. Static Routes (1)
4. OSPF (0)
5. EGP Configuration (0)
6. TCP Configuration (0)
7. TFTP Configuration (1)
8. BOOTP Configuration (0)
9. Import Route Filters (0)
10. Export Route Filters (9)
Enter Selection (0 for Previous Menu) : ____


-------
```

図 3.6: DoD Internet Router メニュー

```
-------
Wellfleet Communications, Inc.    WIDE_FNET                5-Apr-1994  15:21:47

============================-  SESSION 2 - MGR MODE -============================
Configuration Editor 1.19              Current File : CONFIG


1. IP Address Lists (6)
2. IP Port Lists (6)









Enter Selection (0 for Previous Menu) : ____

-------
```

図 3.7: Lists メニュー

```
-------
Wellfleet Communications, Inc.    WIDE_FNET                5-Apr-1994  15:22:11

============================-  SESSION 2 - MGR MODE -============================
Configuration Editor 1.19              Current File : CONFIG


IP Address Lists
    List Name
    ---------
 1. CENTER
 2. GW_LAN
 3. SERVER
 4. SOCKSGATE
 5. OUTSIDE
 6. INSIDE




Action (-> for selections) : Previous Display

-------
```

図 3.8: IP Address Lists の一覧

```
-------
Wellfleet Communications, Inc.    WIDE_FNET                5-Apr-1994  15:23:42

---------------------------  SESSION 2 - MGR MODE -----------------------------
Configuration Editor 1.19              Current File : CONFIG

List Name : CENTER


                List Members
    IP Address (low)      IP Address (high)
    -----------------     -----------------
 1. 133.160.39.0          133.160.39.255
 2. 133.160.53.0          133.160.53.255
 3. 133.160.32.39
 4. 133.160.63.53




Action (-> for selections) : Previous Display


-------
```

図 3.9: 1 番の IP Address List リストの内容

```
-------
Wellfleet Communications, Inc.    WIDE_FNET                5-Apr-1994  15:25:22

---------------------------  SESSION 2 - MGR MODE -----------------------------
Configuration Editor 1.19              Current File : CONFIG


  IP Port Lists
    List Name
    ---------
 1. FTP_TELNET
 2. GT1023
 3. RIP
 4. MAIL_NEWS
 5. SOCKS
 6. TELNET




Action (-> for selections) : Previous Display


-------
```

図 3.10: IP Port Lists の一覧

```
-------
Wellfleet Communications, Inc.    WIDE_FNET                5-Apr-1994  15:26:22

============================-  SESSION 2 - MGR MODE -============================
Configuration Editor 1.19              Current File : CONFIG

List Name : FTP_TELNET


             List Members
    IP Port (low)      IP Port (high)
    --------------     --------------
 1. 20                 21
 2. 23




Action (-> for selections) : Previous Display


-------
```

図 3.11: 1 番の IP Port List の内容

```
-------
Wellfleet Communications, Inc.    WIDE_FNET                5-Apr-1994  15:26:38

============================-  SESSION 2 - MGR MODE -============================
Configuration Editor 1.19              Current File : CONFIG

List Name : GT1023


             List Members
    IP Port (low)      IP Port (high)
    --------------     --------------
 1. 1024               2048
 2. 4058               5999
 3. 6001               6666
 4. 6668               65535
 5. 2050               4056




Action (-> for selections) : Previous Display


-------
```

図 3.12: 2 番の IP Port List の内容

```
-------
Wellfleet Communications, Inc.    WIDE_FNET                5-Apr-1994  15:28:38

============================-  SESSION 2 - MGR MODE -============================
Configuration Editor 1.19              Current File : CONFIG
Auto Enable          : Yes             Global Broadcast     : Yes
RIP Network Diameter : 15              Mode                 : Router/Host
Management Priority  : Low             Non Local ARP Source : Drop and Log
Bootp Gateway Support   : No           Bootp Gateway Auto Enable  : Yes
Bootp Gateway Max Hops  : 3            Suppress Authentication Traps : Yes


                 Network Interface Definitions
    Internet Address     Subnet Mask          Circuit Group
    ----------------     ----------------     ----------------
 1. 133.160.1.1          255.255.255.0        G_E21
 2. 133.160.2.1          255.255.255.0        G_E22




Action (-> for selections) : Previous Display


-------
```

図 3.13: Network Interface Definitions メニュー

```
-------
Wellfleet Communications, Inc.    WIDE_FNET                5-Apr-1994  15:29:10

============================-  SESSION 2 - MGR MODE -============================
Configuration Editor 1.19              Current File : CONFIG
Internet Address : 133.160.1.1
Subnet Mask      : 255.255.255.0       Receive Broadcast  : Network and Subnet
Circuit Group    : G_E21               Transmit Broadcast : All Ones

Address Resolution : ARP               RIP Supply : Yes
Normal ARP         : Yes               RIP Listen : Yes
Proxy ARP          : No                Default Route Supply : No
Host Cache         : No                Default Route Listen : No
UDP Checksum Off   : No                Poisoned Reverse/Split Horizon : Poison
                                       RIP Interface Cost   : 1

Address Mask Reply   : No              ASB Flood  : No
MTU Discovery Option : No              Source Route (Token Ring) : No
Load Balancing : No
Bootp Packet Reception : Yes           Bootp Packet Transmission : Yes
1. Traffic Filters (8)

Enter Selection (0 for Previous Menu) : ____


-------
```

図 3.14: 1 番のインタフェースの設定内容

```
-------
Wellfleet Communications, Inc.    WIDE_FNET                5-Apr-1994  15:29:36

============================-  SESSION 2 - MGR MODE -============================
Configuration Editor 1.19              Current File : CONFIG

                            Traffic Filters
    Precedence     IP Dest (low)     IP source (low)     Action
    ----------     -------------     ---------------     ------
 1. 1                                                    Drop
 2. 10                                                   Accept
 3. 10             CENTER            GW_LAN              Accept
 4. 10             CENTER            GW_LAN              Accept
 5. 10             OUTSIDE           GW_LAN              Accept
 6. 10             FDM               GW_LAN              Accept
 7. 10             FDM               GW_LAN              Accept
 8. 10                               FLOODGATE           Accept



Action (-> for selections) : Previous Display


-------
```

図 3.15: 1 番のインタフェースの Traffic Filters のリスト

```
-------
Wellfleet Communications, Inc.    WIDE_FNET                5-Apr-1994  15:29:10

============================-  SESSION 2 - MGR MODE -============================
Configuration Editor 1.19              Current File : CONFIG
Internet Address : 133.160.2.1
Subnet Mask      : 255.255.255.0       Receive Broadcast  : Network and Subnet
Circuit Group    : G_E21               Transmit Broadcast : All Ones

Address Resolution : ARP               RIP Supply : Yes
Normal ARP         : Yes               RIP Listen : Yes
Proxy ARP          : No                Default Route Supply : No
Host Cache         : No                Default Route Listen : No
UDP Checksum Off   : No                Poisoned Reverse/Split Horizon : Poison
                                       RIP Interface Cost   : 1

Address Mask Reply   : No              ASB Flood  : No
MTU Discovery Option : No              Source Route (Token Ring) : No
Load Balancing : No
Bootp Packet Reception : Yes           Bootp Packet Transmission : Yes
1. Traffic Filters (8)

Enter Selection (0 for Previous Menu) : ____


-------
```

図 3.16: 2 番のインタフェースの設定内容

```
-------
Wellfleet Communications, Inc.    WIDE_FNET                5-Apr-1994  15:29:36

============================-  SESSION 2 - MGR MODE -============================
Configuration Editor 1.19              Current File : CONFIG

                            Traffic Filters
    Precedence     IP Dest (low)     IP source (low)     Action
    ----------     -------------     ---------------     ------
 1. 1                                                    Drop
 2. 10                                                   Accept
 3. 10             GW_LAN            CENTER              Accept
 4. 10             GW_LAN            CENTER              Accept
 5. 10             INSIDE            CENTER              Accept
 6. 10             GW_LAN            FDM                 Accept
 7. 10             GW_LAN            FDM                 Accept
 8. 10             FLOODGATE                             Accept



Action (-> for selections) : Previous Display


-------
```

図 3.17: 2 番のインタフェースの Traffic Filters のリスト

```
-------
Wellfleet Communications, Inc.    WIDE_FNET                5-Apr-1994  15:31:47

============================-  SESSION 2 - MGR MODE -============================
Configuration Editor 1.19              Current File : CONFIG

Precedence : 1
IP Dest (low) :
IP dest (high) :                                          Effect : Ignore
IP source (low) :
IP source (high) :                                        Effect : Ignore
Protocol : Ignore
Action : Drop

1. User Defined Fields (0)
2. Next Hop Assignment (0)




Enter Selection (0 for Previous Menu) : ____


-------
```

図 3.18: 1 番のインタフェースの 1 番の Traffic Filter の内容

```
-------
Wellfleet Communications, Inc.    WIDE_FNET                5-Apr-1994  15:36:21

============================-  SESSION 2 - MGR MODE -============================
Configuration Editor 1.19              Current File : CONFIG

Precedence : 10
IP Dest (low) :
IP dest (high) :                                          Effect : Ignore
IP source (low) :
IP source (high) :                                        Effect : Ignore
Protocol : UDP or TCP
Action : Accept
UDP/TCP Dest Port (low) : RIP            (high) :         Effect : Match
UDP/TCP Source Port (low) : RIP          (high) :         Effect : Match
1. User Defined Fields (0)
2. Next Hop Assignment (0)




Enter Selection (0 for Previous Menu) : ____

-------
```

図 3.19: 1 番のインタフェースの 2 番の Traffic Filter の内容

```
-------
Wellfleet Communications, Inc.    WIDE_FNET                5-Apr-1994  15:32:22

============================-  SESSION 2 - MGR MODE -============================
Configuration Editor 1.19              Current File : CONFIG

Precedence : 10
IP Dest (low) : CENTER
IP dest (high) :                                          Effect : Match
IP source (low) : GW_LAN
IP source (high) :                                        Effect : Match
Protocol : TCP
Action : Accept
UDP/TCP Dest Port (low) : GT1023         (high) :         Effect : Match
UDP/TCP Source Port (low) : FTP_TELNET   (high) :         Effect : Match
1. User Defined Fields (0)
2. Next Hop Assignment (0)




Enter Selection (0 for Previous Menu) : ____

-------
```

図 3.20: 1 番のインタフェースの 3 番の Traffic Filter の内容

```
-------
Wellfleet Communications, Inc.    WIDE_FNET                5-Apr-1994  15:33:08

---------------------------  SESSION 2 - MGR MODE -----------------------------
Configuration Editor 1.19              Current File : CONFIG

Precedence : 10
IP Dest (low) : CENTER
IP dest (high) :                                          Effect : Match
IP source (low) : GW_LAN
IP source (high) :                                        Effect : Match
Protocol : Ignore
Action : Accept

                    User Defined Fields
    Header            Offset     Length     Effect
    ------            ------     ------     ------
 1. Network           72         8          Match



Action (-> for selections) : Previous Display


-------
```

図 3.21: 1 番のインタフェースの 4 番の Traffic Filter の内容

```
-------
Wellfleet Communications, Inc.    WIDE_FNET                5-Apr-1994  15:34:01

---------------------------  SESSION 2 - MGR MODE -----------------------------
Configuration Editor 1.19              Current File : CONFIG

Header : Network
Offset : 72
Length : 8
Effect : Match


                  Values
    Low Value (hex)      High Value (hex)
    ---------------      ----------------
 1. 01




Action (-> for selections) : Previous Display


-------
```

図 3.22: 1 番のインタフェースの 4 番の Traffic Filter の内容 (続き)

第 4 章

ドキュメント、ツールの一覧表

本章では、Firewall を構築するにあたって必要となるツールや、参考となるドキュメントを集めて紹介する。

4.1 ツールの一覧

各ツールは次のタグづけをして紹介してある。

T:	title	(必須)	名称
A:	author	(必須)	作者
D:	date	(オプション)	最新のものが作られた日
V:	version	(オプション)	最新バージョン
K:	keyword	(必須)	キーワード
F:	function	(必須)	機能
E:	environment	(オプション)	動作環境 (ハード、ソフト)
U:	url	(必須)	URL
N:	note	(オプション)	備考 (caveats(警告), bugs, limitation(制限) なども含む)

4.1.1 パスワード関連

T: npasswd
A: Clyde Hoover
K: password
F: 単純なパスワードを認めない passwd 互換パッケージ。
U: ftp://ftp.dit.co.jp/pub/security/tools/npasswd.tar.gz

T: npasswd_enhanced
A: Paul Leyland
K: password
F: 単純なパスワードを認めない passwd 互換パッケージ。npasswd の Ultrix 版
E: Ultrix
U: ftp://black.ox.ac.uk/src/security/npasswd_enhanced.shar.Z

T: npasswd_jpl

A: Dan Kegel

K: password, crack

F: 単純なパスワードを認めない passwd 互換パッケージ。 npasswd に対して crack V3 を適用した。

U: ftp://black.ox.ac.uk/src/security/npasswd_jpl.tar.Z

T: npasswd_boulder

A: Todd Miller

K: password, crack

F: 単純なパスワードを認めない passwd 互換パッケージ。 JPL 版に対して crack V5 を適用するなどの変更を加えている。

U: ftp://ftp.dit.co.jp/pub/security/tools/npasswd_boulder.tar.gz

T: BP+ (passwd+ version beta)

A: Matt Bishop <Matt.Bishop@dartmouth.edu>

D: 92/06/30

V: beta

K: passwd, chsh, chfn

F: パスワードや login shell などを変更する為のプログラム群 パスワードの aging 機能もある.

U: ftp://dartmouth.edu/pub/security/BP+.tar.Z

N: passwd+ のβバージョン

T: passwd+

A: Matt Bishop <Matt.Bishop@dartmouth.edu>

D: 92/06/02

V: ALPHA TEST REVISION 4

K: passwd, chsh, chfn

F: パスワードや login shell などを変更する為のプログラム群 変更履歴などのログ取り機能もある.

U: ftp://dartmouth.edu/pub/security/passwd+.tar.Z

N: これのβバージョンが BP+

T: anlpasswd

A: Mark Henderson <Support@mcs.anl.gov>

D: 93/10/28

K: passwd, perl, NIS

F: 破られやすいパスワードには変更できないようにするプログラム. NIS にも対応している.

U: ftp://info.mcs.anl.gov/pub/systems/anlpasswd-2.2.tar.Z

N: (O'Reilly の本に載っている) Larry Wall による passwd の改良版.

T: shadow-3.3.1

A: John F. Haugh, II <jfh@rpp386.cactus.org>

D: 93/08/08

V: release 3.3.1

K: shadow password, login

F: login 等を置き換え shaddow password の機能を実現する. パスワードの aging や 16 文字のパスワードを許す機能などもある.

U: ftp://ftp.uu.net/usenet/comp.sources.misc/volume38/shadow
ftp://ftp.uu.net/usenet/comp.sources.misc/volume39/shadow

N: authentication をユーザ定義できるので crypt を入れ換えることも可. CrackLib もサポートされているので破られやすいパスワードをはじくことも可.

T: yppapasswd-v1.0

A: Matthew Scott <knight@cs.wwu.edu>

V: 1.0

K: passwd, chsh, chfn, perl, NIS

F: yppasswdd を置き換え, 破られやすいパスワードへの変更をはじく.

U: ftp://ftp.wg.omron.co.jp/pub/security/tools/yppapasswd-v1.0.tar.gz

N: (O'Reilly の本に載っている) Larry Wall による passwd を元にしている.

4.1.2 システムのチェック

T: binaudit

A: Matt Bishop <Matt.Bishop@dartmouth.edu>

D: 92/05/19

V: 3.1.3

K: change, setuid, setgid, owner, mode

F: /bin 中やその他指定されたファイルが，変更されたかどうか調べたり，setuid, setgid のファイルを探して変更されているかを見つけるツール。メールでのレポートも可能。

U: ftp://ftp.omron.co.jp/pub/security/tools/binaudit.tar.gz

T: cert-ftp-check

A: df@death.cert.sei.cmu.edu

D: 90/12/20

K: anonymous ftp, owner, mode

F: anonymous ftp が正しく設定されているかを，ファイルの owner，mode などを調べることによって確認し，問題点を指摘する。

U: ftp://ftp.omron.co.jp/pub/security/tools/cert-ftp-check.shar.gz

T: chkacct

A: Philip R. Moyer <prm@ecn.purdue.edu>

D: 93/02/28

V: 1.3

K: dot files, user files, mode

F: 各ユーザの /.[A-z] を始めとする各ファイルのパーミッションなどを調べて問題点を指摘・修正する。各個人が自分のアカウント，ファイルを守るために使用するもの。

U: ftp://ftp.omron.co.jp/pub/security/tools/chkacct.tar.gz

T: COPS_104

A: Dan Farmer <df@death.cert.sei.cmu.edu>

D: 92/03/10

K: owner, mode, passwd, rc, crontab, ftp, NFS

F: 広範囲にシステムのチェックするツール．permissions/modes，passwd，rc や crontab，setuid，実行ファイルの CRC，user startup file，a-ftp，tftp，mail の alias，hosts.equiv，NFS 他。

U: ftp://cert.sei.cmu.edu/pub/tools/cops/1.04/cops_104.tar.Z

T: COPS_104+

A: Dan Farmer <dan.farmer@sun.com>

D: 93/10/14

K: permissions, modes, passwd, rc, crontab, ftp, NFS

F: 広範囲にシステムのチェックするツール．permissions/modes，passwd，rcや crontab，setuid 他．cops_104 に比べてレポートの改善や，問題箇所の自動修正等が付加された。

U: ftp://black.ox.ac.uk/src/security/cops_104+.tar.gz

N: CHANGES 等の記載が無く，cops_104 からの拡張部分や author など判りにくい。

T: Internet Security Scanner

A: Christopher Klaus <cklaus@hotsun.nersc.gov>

V: 1.21

K: password, sendmail, aliases, anonymous ftp, rpc, NIS, rexd, bootparam

F: passwd 的なコマンドにパスワードが crack 可能かどうか調べる機能を入れる時に利用可能なライブラリ。

E: システムのセキュリティホールを見つけ出す。

U: ftp://ftp.wg.omron.co.jp/pub/security/tools/iss-1.21.shar.gz

N: 時々、ポートソケットを open できないことがある。SunOs4.1.1-3 以外のマシンではあまりテストされていない。時々、原因不明の Core を吐くことがある。

T: cpm

A: CERT Coordination Center, Software Engineering Institute, Carnegie Mellon University

D: 94/02/03

V: 1.0

K: network interface, promiscuous mode

E: ioctl(2) が SIOCGIFCONF をサポートしていなければならない。4.1.3 以前の SunOS および多くの BSD から派生したプラットフォームでは大丈夫なはず。

F: ネットワークインターフェースが promiscuous モードかどうか調べる

U: info.cert.org:/pub/tools/cpm.tar.Z

N: ioctl(2) が SIOCGIFCONF をサポートしていなければならない。

T: tripwire

A: Gene Kim <gkim@cs.purdue.edu>, Gene Spafford <spaf@cs.purdue.edu>

D: 93/12/15

V: 1.1

K: file sytem, checksum, MD5, MD4, MD2, Snefru, SHA, CRC-32, CRC-16

E: 386BSD 0.1, Apple A/UX 3.x, AT&T SVR3.2.3, AT&T SVR4.0.3, CMU Mach 2.x ConvexOS 9.1, Cray Unicos 6.1.6, DEC Ultrix 4.x, Encore Umax V 2.4.1.P3 FPS FPX 4.3.3, HP/UX 8.x, HP/UX 9.x, IBM AIX 3.x, BSDI beta, MIPS EP/IX 1.4.3, NeXT Mach 2.x, NeXT Mach 3.x, OSF/1 1.0.4, Pyramid DC/OSx 1.1, Pyramid OSx 5.1, Sequent Dynix 3.x, Sequent dynnix/PTX 2.0.x SunOS 4.0.3, SunOS 5.x, SCO Xenix 03.02.00, SCO XENIX 2.2.6, SGI IRIX 4.x, SGI IRIX 4.0.5

F: ファイルシステムの不正な変更を調べる。

U: ftp://ftp.wg.omron.co.jp/pub/security/tools/tripwire-1.1.tar.gz

T: tiger

A: Douglas Lee Schales <Doug.Schales@net.tamu.edu> 他

D: 93/12/20

V: 2.2.2

K: cron, mail aliases, passwd, NFS exports, inetd, PATH, setuid, .rhosts, .netrc, load-module, digital signature

E: SunOS 4.x, SunOS 5.x, NeXT 3.x, IRIX 4.x, AIX 3.x, UNICOS 6.x, Linux 0.99.x, HP/UX

F: COPS と同様にシステムのセキュリティホールを見つけ出す。

U: net.tamu.edu:/pub/security/TAMU/tiger-2.2.2.tar.gz

N: IRIX 4.x, AIX 3.x, UNICOS 6.x, Linux 0.99.x, HP/UX については部分的なコンフィギュレーションファイルしかない

4.1.3 モニタリング

T: NeTraMet

A: Nevil Brownlee <n.brownlee@auckland.ac.nz>

D: 94/01/14

V: 2.1

K: network traffic flow, Internet Accounting Architechture, IP, DECnet, EtherTalk, IPX, RFC1272, NeMaC

F: トラフィック解析のツール．対象とするパケットを Ethernet アドレス，プロトコルアドレス (IP, DECnet, EtherTalk, IPX)，ポート番号などを組合せたルールとして記述できる．NeMaC (NeTraMet Manager/Collector) 付き．

E: SunOS と IRIX の Makefile, PC(DOS) の実行ファイルが含まれている．

U: ftp://ccu1.auckland.ac.nz/iawg/NeTraMet (0700-2000 GMT)
ftp://ftp.delmarva.com/pub/nms
ftp://ftp.funet.fi/pub/networking/management/NeTraMet

N: Internet Accounting Architechture の実装。RFC1272 "Internet Accounting Background" 参照。

T: Watcher
A: Kenneth Ingham <ingham@ariel.unm.edu>
D: 91/01/
K: system monitoring
F: システムの状態がルールで記述した条件 (ディスク容量，負荷...) を満たしたときにそれを報告するプログラム。
U: ftp://ariel.unm.edu/pub/unix/Watcher.tar.Z
N: 1987 Usenix "Keeping Watch over the Flocks at Night (and day)".

T: IsOn
A: Phil Dietz <pdietz@cse.unl.edu>, Mike Gleason <mgleason@cse.unl.edu>
D: 93/12/14
V: 5.0
K: login, logout, finger, rusers
F: 注目したあるユーザのログイン，ログアウトをモニタするツール.
U: news:<1993Dec14.042025.3559@sparky.sterling.com>
ftp://ftp.elelab.nsc.co.jp/pub/netnews/comp.sources.misc/volume41/ison
ftp://ftp.omron.co.jp/pub/security/tools/ison-5.0.shar.gz
N: comp.sources.misc v41i031

T: nocol
A: Vikas Aggarwal <vikas@jvnc.net>
D: 93/11/01
V: 3.0
K: network monitoring, ICMP, DNS, SNMP, modem, appletalk, novell, BGP
F: ネットワーク監視パッケージ．ICMP，DNS，SNMP trap，modem line， appletalk & novell routes and services，BGP peers などをモニタできる。display agent は任意個設定可能。
U: ftp://ftp.jvnc.net/pub/jvncnet-packages/nocol/nocol.tar.Z
N: 表示は curses 使用．perl インターフェイスも用意されている。

T: record_arp
A: Stephane Bortzmeyer <bortzmeyer@cnam.cnam.fr>
D: 93/07/16
K: SNMP, LAN, arp, MAC address, Ethernet address, network address, IP address
F: SNMP エージェントの ARP テーブルをダンプすることによって，MAC アドレス (例えば Ethernet アドレス) とネットワークアドレス (例えば IP アドレス) を得る perl スクリプト.
E: Tricklet が必要。
U: ftp://cnam.cnam.fr/pub/CNAM/MISC/record_arp.tar.Z
N: Tricklet は ftp://dutepp0.et.tudelft.nl/pub/Tricklet から入手可能。

T: Swatch

A: Stephen E. Hansen <hansen@Sierra.Stanford.EDU>, E. Todd Atkins <atkins@EE-CF.Stanford.

D: 93/11/14

V: 2.0

K: syslog, monitor

F: syslog の出力を監視するプログラム。設定ファイルにより、パターンに対する動作を指定する。例えば「『FAILURE』という文字列が現れたら、あるコマンドを実行する」ということをさせる。

E: Perl 言語の処理系

U: ftp://sierra.stanford.edu/pub/sources/swatch-2.0.tar.gz

N: 1993 年の USENIX LISA VII で発表された。

T: Tcplist

A: John DiMarco <jdd@cdf.toronto.edu>

D: 93/05/18

V: 1.1

K: TCP, connection, netstat, RFC931

F: そのホスト上にその時点で存在する全ての TCP の接続を表示する。netstat と違うのは、RFC931(Authentication Sever) の機能により、接続の両端のプロセスの持ち主を表示すること。

```
himazu@aria.isl.mei.co.jp:1023  root@chorus.isl.mei.co.jp:login
```

のように表示される。

E: SunOS4.1.X, IRIX4.X

U: ftp://ftp.omron.co.jp/pub/security/tools/tcplist-1.1.shar.gz

N: README には ftp://ftp.cdf.toronto.edu/pub/tcplist に最新版があると書いてあるが、94/04/04 現在そのディレクトリーは存在しない。

T: Tcplocate

A: Peter Eriksson <pen@lysator.liu.se>

D: 92/12/03

V: 1.0

F: 以下のいずれかの形式で使用する。

(a) tcplocate 相手の IP アドレス (10 進表記) ポート番号
(b) tcplocate local ポート番号

指定した TCP 接続を使用している自分のホスト側のプロセスのプロセス ID、ユーザー ID を表示する。例えば netstat で

tcp 0 0 132.182.27.26.6000 132.182.1.2.3669 ESTABLISHED

のように表示される場合

```
# tcplocate 132.182.1.2 3669
USER        PID CONNs
himazu      9341   1
# tcplocate local 6000
USER        PID CONNs
himazu      9341   6
```

と表示される。

E: SunOS4.X, Dynix2.X, RISC/OS4.X, HP-UX7.X HP-UX8.X, Ultrix4.X

U: ftp://ftp.omron.co.jp/pub/security/tools/tcplocate-1.0.tar.gz

N: Pident を改造したもの。

T: Spar

A: Doug Schales <Doug.Schales@net.tamu.edu>

D: 93/10/28

V: 1.2

F: lastcomm の機能拡張版。AWK のような文法で出力行の選択、加工が行なえる。より高速。

例: `spar -e 'cmd=ls && user=daemon && tty=ttyp0 {print}' lastcomm`

E: SunOS4.X, SunOS5.X, IRIX4.X, NeXT3.X, HPUX7.X, AIX3.X ANSI C 準拠の C コンパイラー (GCC など)

U: ftp://net.tamu.edu/pub/security/TAMU/spar-1.2.tar.gz
ftp://ftp.omron.co.jp/pub/seciryty/tools/TAMU/spar-1.2.tar.gz

N: TAMU のセキュリティーツールの 1 つ。

T: netlog

A: Douglas Lee Schales <Doug.Schales@net.tamu.edu>, David K. Hess, David R. Safford <dave.safford@net.tamu.edu>

D: 94/01/05

V: 1.02

K: logging, tcp, udp

F: ネットワークのログを行う複数のツールから構成される: TCP のコネクション要求, UDP の疑似的なセッション, ICMP のログの収集を行うコマンド, 収集したログの解析コマンド, NNstat の簡易版, 特定のポートで通常と異なるやりとりの監視ツール

E: SunOS 4.x(NIT 使用), SunOS 5.X(DLPI)

U: ftp://net.tamu.edu/pub/security/TAMU/netlog-1.02.tar.gz

N: ANSI C を必要 (GCC でも可)

T: tcptop

A: Steven Grimm <koreth@hyperion.com>

D: 93/12/11

V: 1.0

K: etherfind, tcp

F: etherfind の出力から TCP のコネクションを top の様に表示する。

E: SunOS 4.X(etherfind や ypwhich コマンドを絶対パスで呼び出している)

U: ftp://ftp.dit.co.jp/pub/security/tools/tcptop-1.0.tar.gz 他

N: top と異なり対話的なコマンドは一切受け付けず, キー入力すると画面が乱れたままである.

4.1.4 認証

T: SRA (Secure RPC Authentification)

A: David R. Safford <dave.safford@net.tamu.edu>

D: 93/11/20

V: 1.3

K: RPC, TELNET, FTP

F: 認証つきの RPC およびその上の TELNET/FTP のサーバ/クライアント

E: net-2 ベースなのでポータブル。Solaris 1.x/2.x および Linux 用バイナリあり。

U: ftp://ftp.wg.omron.co.jp/pub/security/tools/TAMU/srasrc-1.3.tar.gz
ftp://net.tamu.edu/pub/security/TAMU/srasrc-1.3.tar.gz

T: authd-3.01

A: Vic Abell <abe@mace.cc.purdue.edu>

D: 91/02/07

K: authd

F: RFC931 のインプリメンテーション。TCP コネクションの owner を与える。

U: ftp://ftp.wg.omron.co.jp/pub/security/tools/authd-3.01.tar.gz

N: obsoleted

T: idlookup-1.2.tar.gz

A: Peter Eriksson <pen@lysator.liu.se>

D: 92/07/03

K: identd

F: ident プロトコルにより、TCP connection の identifier を得て、表示す る単純なツール

U: ftp://ftp.wg.omron.co.jp/pub/security/tools/idlookup-1.2.tar.gz

T: pidentd-2.3alpha2

A: Peter Eriksson <pen@lysator.liu.se>

D: 94/02/23

K: identd

F: RFC1413 のインプリメンテーション。TCP コネクションの owner を与える。カーネルから情報を直接得ている。多数の OS をサポート 2.3alpha2 では、DES encription サポート

E: Dynix, SunOS3.X, 4.X, 5.X, HP-UX[7-9].X, RISC/OS 4.5x, Ultrix 4.X, *BSD, bsd/386 など

U: ftp://ftp.lysator.liu.se/pub/ident/servers/pidentd-2.3alpha2.tar.gz

4.1.5　アクセスコントロール

T: Secure tftpd

A: Kazuhiko Yamamoto <kazu@csce.kyushu-u.ac.jp>

D: 94/01/16

V: 1.00

K: tftp, tftpd, access contorl, change root, log

F: セキュリティ強化をはかった tftpd. アクセスできるディレクトリの制限機能, ホスト名/ドメイン名によるアクセス制御機能, 不正なアクセスの記録を取る機能を有する。

E: Tahoe based OS

U: ftp://pub/Security/tools/xtftpd/xtftpd-1.00.tar.gz

T: xinetd

A: Panos Tsirigotis <panos@cs.colorado.edu>

D: 93/07/08

V: 2.1.4

K: inetd, access-control

F: inetd の拡張版でアクセス制御/ログ機能が充実している

E: 4.2BSD の後継者

U: ftp://mystique.cs.colorado.edu

T: TCP/IP daemon wrapper package

A: Wietse Venema <wietse@wzv.win.tue.nl>

V: Version 6.3

D: 94/03/23

K: log_tcp, inetd, tcp_wrapper

F: inetd が扱うサービスのログを syslog(3) を使って取ったり、アクセス・コントロールをしたりする。inetd にまず起動され、ログ、アクセス・コントロール・チェックの後、本体を起動する。 Version 6.0 以前は log_tcp という名称だった。

U: ftp://ftp.win.tue.nl/pub/security/tcp_wrappers_6.3.shar.Z

4.1.6　Application Level Gateway

T: UDP packet relayer

A: Tom Fitzgerald <fitz@wang.com>

V: Version 0.2

D: 93/10/30

K: UDP, relay, archie

F: firewall 上で UDP を中継する。archie の中継に有効

U: ftp://ftp.wang.com/pub/fitz/udprelay-0.2.tar.Z

T: tcpconnect

A: 歌代 和正 (Kazumasa Utashiro) <utashiro@sra.co.jp>

D: 93/10/28

V: 0.8

K: application gateway

F: 特定の TCP ポートへの接続を別のホストのサービスに転送する。inetd と 併用することで、firewall に対する sendmail や nntp の接続を組織内のサーバに転送することが可能。また、標準入力からデータを入力することも可能で、tcpreceive と組み合わせてネットワークワイドなパイプを実現できる。

E: Perl 言語で記述されているため、perl が動作する環境なら利用可能。

U: ftp://ftp.sra.co.jp/pub/lang/perl/sra-scripts/tcpconnect-0.8

T: tcpreceive

A: 歌代 和正 (Kazumasa Utashiro) <utashiro@sra.co.jp>

D: 94/04/27

V: 1.2

K: application gateway

F: TCP ポートからデータを受け取り標準出力へ出力するコマンド。tcpconnect と併用して、異なるホスト上でパイプを実現することができる。

E: perl が動作する環境なら利用可能。

U: ftp://ftp.sra.co.jp/pub/lang/perl/sra-scripts/tcpreceive-1.2

N: 1.2 から ident プロトコルを使って接続するユーザを指定することが可能。

T: tcprelay

A: 歌代 和正 (Kazumasa Utashiro) <utashiro@sra.co.jp>

D: 92/04/13

V: 1.2

K: application gateway

F: TCP の転送を行うコマンド。tcpconnect とほぼ同様な機能を持つが、 inetd を使わずに自分でポートを作成する。また、ftp のプロトコルを理解し、PORT コマンドにしたがって動的にポートを作成するため、ゲートウェイを介して ftp を利用できる。

E: perl が動作する環境なら利用可能。

U: ftp://ftp.sra.co.jp/pub/lang/perl/sra-scripts/tcprelay-1.2

N: 単に中継するだけなら tcpreceive と tcpconnect を組み合わせるだけで実現できる。

T: gophergate

A: 歌代 和正 (Kazumasa Utashiro) <utashiro@sra.co.jp>

D: 94/04/27

V: 1.5

K: gopher, application gateway, cache

F: gopher プロトコルを中継するためのゲートウェイ。firewall 上で動作さ せることによって、組織内のホストから外部のサーバを利用することができる。また、データのキャッシュを行うため、応答性を向上させネットワークのトラフィックを抑える効果もある。

E: perl が動作する環境なら利用可能。

U: ftp://ftp.sra.co.jp/pub/lang/perl/sra-scripts/gophergate-1.5

N: 利用実績が十分でないため、今後機能の改善、セキュリティ、キャッシュ の使い方などに関する考察が必要。

4.1.7　フィルタリング

T: ip_fil

A: Darren Reed <avalon@cairo.anu.edu.au>

D: 94/04/22

V: 最新バージョン

K: Packet filter

F: Sun Workstation を packet filter 機能付きのルータにするためキット。カーネルにフィルタリング機能を与えるためのパッチ及び、設定ファイルに従ってカーネルにフィルタリングのパラメータを渡すためのコマンドからなる。

E: SunOS 4.1.2 でテスト済。SunOS 4.1. なら動くだろう。

U: ftp://coombs.anu.edu.au/pub/net/kernel/ip_fil.tar.Z

N: ip_forward() ではなく、ipintr(), ip_output() でフィルタリングの条件をチェックするため、ip_fil ルータ発・着のパケットについてもフィルタ可能である。

T: IPACL

A: Gerhard Fuernkranz <fuer@siemens.co.at>

D: 93/01/13

V: 最新バージョン

K: Packet filter

F: Lachamann Streams TCP を持つ SVR4/386 カーネルにおいて入ってくる、もしくは出ていく TCP/UP のパケットの始点・終点のアドレス及びポートを元にフィルタリングを行なう。STREAMS のモジュールとして実装されている。

E: SVR4/386 kernel with Lachamann Streams TCP

U: ftp://ftp.wg.omron.co.jp/pub/security/tools/ipacl.tar.gz

N: STREAMS の module として実装されている。TCP/UDP と IP の間に push されるため、フィルタリング機能付のルータとして利用することはできない。

T: netaccess

A: Kirk Smith <ks@purdue-ecn.ARPA>

D: 85/02/02

V: 最新バージョン

K: connect

F: 外部ネットワークにアクセスできる人を制限することが目的。各ホストのカーネルに入れて、特定のグループに属する人にだけが任意のネットワークに connect(2) をかけられるようにし、それ以外の人は natural netmask 単位で特定のネットワークに対するものしか許さないようにするしかけ。

E: 4.3BSD

U: ftp://ftp.wg.omron.co.jp/pub/security/tools/netaccess.shar.gz

T: Drawbridge

A: David K. Hess, Douglas Lee Schales, and David R. Safford (Texas A&M University)

D: 93/06/28

V: 1.1

K: packet filter, filter bridge

F: 2 枚の ehternet card を持つ PC 上で動作して、フィルタリング機能を持つブリッジを提供する。制御は専用のフィルタリング記述言語を用いて行ない、NIT 機能を持つ Workstation から命令をダウンロードする。

E: Filter Bridge “Filter” は PC 上で動作する。それを制御する “Filter Manager” 及び、フィルタリング記述言語のコンパイラ “Filter Compiler” は Sun Workstation 上で動作する。

U: ftp://net.tamu.edu/pub/security/TAMU/drawbridge-1.1.tar.gz

N: TCP/IP は incoming と outgoing の両方、UDP/IP は incoming のみをフィルタする。それ以外の protocol についてはブリッジする。

T: Network Filter

A: Kazuhiko Yamamoto <kazu@csce.kyushu-u.ac.jp>

D: 94/07/07

V: 1.03

K: filtering, screening, IP address, port

F: IP パケットを転送する際に, IP の始点アドレス, および, TCP/UDP の終点ポート番号により, フィルタリングを行なう. フィルタリングのエントリは, 専用コマンドにより, 動的にカーネルに入れることができる。

E: SunOS 4.1.x

U: ftp://ftp.csce.kyushu-u.acjp/pub/Security/tools/filter/network-filter-1.0.3.tar.gz

N: Tahoe のネットワークコードを SunOS へリンクするため, SunOS の Loose Source Routing オプションを並べ変える機能などが失われる。

4.1.8 暗号関連

T: DES 暗号化フィルタ (kmdes)

A: 森 公一郎 <kmori@lsi-j.co.jp>

D: 93/06/18

V: 1.00

K: DES, ECB モード，CBC モード，暗号鍵

F: SunOS 4.1 の des(1), des_crypt(3) と同じ仕様の DES package。 libdes の内容は ecb_crypt(), cbc_crypt(), des_setparity() で，DES の ECB モードと CBC モードをサポート．デフォルトは CBC モード。

E: MS-DOS, UNIX

U: ftp://ftp.wg.omron.co.jp/pub/security/tools/kmdes-1.00.tar.gz

N: 著作権は放棄．再配布，変更，変更したものの配布は自由.

T: libdes

A: Eric Young <eay@psych.psy.uq.oz.au>

D: 93/10/08

V: 3.01

K: DES，ECB モード，CBC モード，OFB モード，CFB モード，kerberos

F: des(1) 及び IT's project Athena の des_crypt(3) 互換の library。DES の ECB, CBC, OGB, CFB, TRIPLE ECB, TRIPLE CBC モードをサポート。libdes を kerberos version 4(eBones-p9.tar.Z) に組み込むことにより kerberos の利用が可能.

E: MS-DOS(Microsoft C v 5.1 and Turbo C v 2.0), BSD and SYSV UNIX(most32bit and 64bit version), VMS

U: ftp://ftp.wg.omron.co.jp/pub/security/tools/libdes.tar.gz

N: フリーソフトウェアで，Free Software Foundation の "GNU General PublicLicense" と "Artistic License" のもとで，再配布，修正が可能.

T: CRYPT BREAKERS WORKBENCH

A: Robert W. Baldwin <baldwin@xx.lcs.mit.edu>

D: 93/10/08

V: 3.01

K: DES，ECB モード，CBC モード，OFB モード，CFB モード，kerberos

F: unix の crypt で暗号化されたファイルを cipher-text only attack するためのシステム。目的は、crypt 程度の暗号強度のツールで大事な情報を暗号化させないようにすること。

U: ftp://ftp.wg.omron.co.jp/pub/security/tools/cbw.tar.gz

N: フリーソフトウェアで，Free Software Foundation の "GNU General PublicLicense" と "Artistic License"のもとで，再配布，修正が可能.

T: pgp - Pretty Good Privacy encryption system

A: Philip R. Zimmermann

D: 93/06/02

V: 2.3a

K: pgp, DES, RSA 公開鍵暗号，電子署名，メッセージダイジェスト

F: 公開鍵暗号パッケージ．平文の暗号化，暗号文の復号化，電子署名およびその検証，RSA 暗号鍵の作成が可能．公開鍵配布局 (PGP Public keyserver) があり，電子メールで公開鍵の登録，取得が可能.

U: ftp://ftp.wg.omron.co.jp/pub/security/tools/pgp23A.tar.Z.gz

N: フリーソフトウェアで，Free Software Foundation の "GNU General PublicLicense のもとで，再配布，修正が可能。

T: PGP Function Library

A: Pr0duct Cypher

V: 1.0

K: PGP, DES, RSA 公開鍵暗号，電子署名，メッセージダイジェスト，ZIP

F: PGP 互換の暗号化アプリケーションを簡単に実現するライブラリを提供．対象とするアプリケーションは，point-and-click PGP，電子署名を用いたセキュアなメールプログラムやニュースリーダ，暗号鍵配布局，ディジタルキャッシュサーバ，ログイン認証など。

U: ftp://ftp.wg.omron.co.jp/pub/security/tools/PGPTools.tar.gz

T: mailcrypt.el

A: Jin Choi <jsc@mit.edu>

D: 93/08/15
V: 1.2
K: ripem, pgp, RMAIL, VM
F: RIPEM と PGP でメイルを暗号化するための lisp パッケージで, RMAIL と VM をサポートしている。
E: Emacs, comint.el(cmushell などに附属)
U: ftp://black.ox.ac.uk/src/security/mailcrypt.el.Z

T: pgptalk
A: <miron@extropia.wimsey.com> (原作<yenne@ccwf.cc.utexas.edu>)
D: 93/06/08 (?)
V: 2.0
K: talk, ytalk, pgp
F: talk に複数ユーザなどの拡張を施した ytalk に PGP による暗号化を追加。
E: curses
U: ftp://black.ox.ac.uk/src/security/pgptalk.2.0.tar.gz
N: ANSI C または非 ANSI C 化するツールが必要.

4.1.9 その他のツール

T: chrootuid1.2.shar.Z
A: Wietse Venema <wietse@wzv.win.tue.nl>
D: 93/08/16
V: 1.2
K: chroot, setuid, su
F: chroot(2), setuid(2) した上でプログラムを起動し，制限された環境下でコマンド実行を可能とする。
E: UNIX, syslog
U: ftp://ftp.win.tue.nl/pub/security/chrooduid1.2.shar.Z
N: set-uid でインストールしてはいけない。

T: cookiegen.tar
A: Marc VanHeyningen <mvanheyn@indiana.edu>
D: 92/05/15
V: Version 1.2
K: xauth
F: 次に示すような簡単な方法で，xauth の設定を行なうためのプログラム。

makeseed — cookiegen <display-name> — xauth source -

makeseed の例： perl -e 'printf int(unpack("

E: UNIX, X-Windows

T: fsh.1.0.tar.gz
A: Mathew Lim
D: 90/10/02
V: Version 1.0
K: shell
F: false shell. ログインさせたくないユーザの shell にする。/.fshrc または /usr/local/etc/fshrc の内容を表示して終了。syslog(3) も行なう。
E: UNIX, syslog

T: secure_mount.shar
A: Brendan Kehoe <brendan@cs.widener.edu>
D: 91/12/06
V: Version 1.0
K: mount, setuid
F: フロッピーや CD-ROM などのマウントを root 以外が行なえるようにするもの。
E: UNIX, Perl
U: (alt.sources に流れた？)

T: wipe-1.0b.shar.gz
A: Calvin Clark <ckclark@MIT.EDU>
D: 93/10/28
V: 1.0beta
K: file, rm, unlink
F: UNIX でファイルの消去 (unlink) をするとき、unlink(2) だとディスク中には中身が残る。wipe はファイルの中身を全部 zero にするので内容は完全に消える。
E: UNIX
U: ftp://rtfm.mit.edu/pub/wipe
N: Bugs: Uses getopt().

T: sudo
A: Jeff Nieusma <nieusma@rootgroup.com> and David Hieb <davehieb@rootgroup.com>
D: 91/12/3
V: 1.2
K: setuid, root
F: コマンドを root 権限で実行することを許す。 /etc/sudoers でユーザとコマンドを制限する。
U: ftp://ftp.wg.omron.co.jp/pub/security/tools/sudo.v1.2.tar.gz

T: super
A: Will Deich <will@surya.caltech.edu>
D: 93/7/7
V: 2.0
K: setuid, root

F: コマンドを root 権限で実行することを許す。super.tab でユーザとコマンドを制限する。

U: ftp://ftp.wg.omron.co.jp/pub/security/tools/super-2.0.gz

holefixes は雑多なファイルがいっしょになっているので、中に入っているファイルを個別に解説した。

T: fingerfix (in holefixes)

A: Keith Bostic <bostic@okeeffe.Berkeley.EDU>

D: 88/11/4

K: finger, Internet Worm

F: Internet Worm に使われた finger のセキュリティホールをふさぐパッチ。Internet Worm worm へのパッチつき。

U: ftp://ftp.wg.omron.co.jp/pub/security/tools/holefixes.tar.gz

T: ftpfix (in holefixes)

A: Keith Bostic <bostic@okeeffe.Berkeley.EDU> and Pete Cottrell <pete@brillig.umd.edu>, edited by Steven D. Miller <steve@umiacs.UMD.EDU>

D: 88/11/10-18

K: anonymous FTP

F: anonymous FTP のセキュリティホールをふさぐパッチ。

E: 4.2BSD, 4.3BSD, 4.3BSD-Tahoe

U: ftp://ftp.wg.omron.co.jp/pub/security/tools/holefixes.tar.gz

T: SunFTPPatch (in holefixes)

A: John Kimball <jkimball@src.honeywell.com>

D: 88/11/17

K: ftpd, SunOS

F: 4.3BSD-Tahoe の ftpd を SunOS 3.X で使うためのパッチ。

E: SunOS 3.X

U: ftp://ftp.wg.omron.co.jp/pub/security/tools/holefixes.tar.gz

T: sendmailfix (in holefixes)

A: Keith Bostic <bostic@okeeffe.Berkeley.EDU>,
edited by Steve D. Miller <steve@mimsy.umd.edu>

D: 88/11/4

K: sendmail, Internet Worm

F: Internet Worm に使われた sendmail のセキュリティホールをふさぐパッチ。

U: ftp://ftp.wg.omron.co.jp/pub/security/tools/holefixes.tar.gz

T: wormfixes.sun (in holefixes)

A: Chuq Von Rospach <plaid!chuq@sun.com>
edited by Steve D. Miller <steve@umiacs.UMD.EDU>
and Bill Lewandowski <wrl@Ford-cos2.ARPA>

D: 88/11/10
K: SunOS, Internet Worm
F: SunOS 用の公式 Internet Worm 対策パッチ。
E: Sun-[234], OS 3.x and 4.0 および 386i OS 4.0
U: ftp://ftp.wg.omron.co.jp/pub/security/tools/holefixes.tar.gz

T: fingernote (in holefixes)
A: Tony Nardo at APL <trn@warper.jhuapl.edu>,
edited by Steve D. Miller <steve@umiacs.UMD.EDU>
D: 88/11/12
K: fingerd
F: fingerd の小さいが重要なセキュリティホールについてのメモ。
E: SunOS 3.X
U: ftp://ftp.wg.omron.co.jp/pub/security/tools/holefixes.tar.gz

T: smrsh
A: Eric Allman
D: 93/11/05
K: sendmail, restricted shell
F: sendmail の program mailer として使うための小さな restricted shell。
U: ftp://ftp.wg.omron.co.jp/pub/security/tools/holefixes.tar.gz
N: CERT advisory CA-93:16.security.vulnerability 参照

T: tfpd
A: Jim Guyton <guyton@rand-unix>
D: 92/01/15
K: tftpd
F: 4.3BSD-Tahoe の tftpd を改造して chroot するようにしたもの。
U: ftp://ftp.wg.omron.co.jp/pub/security/tools/tftpd.shar.gz

T: uucp-security
A: dan <zen@death.Corp.Sun.COM> and Bill Davidson <davidsen@crdos1.crd.GE.COM>
D: 91/10/30
K: uucp
F: uucp のセキュリティチェック (permission など) 用ドキュメントおよびツール。
U: ftp://ftp.wg.omron.co.jp/pub/security/tools/uucp-security.tar.gz

T: resolv+
A: Bill Wisner <wisner@uunet.uu.net> 他
D: 92/10/21
V: 2.1.1
K: resolver, DNS, NIS

F: BIND 4.8.3 の resolver に, DNS, NIS, hosts ファイルからアクセス する順番や, どれを使用するかを指定可能な機能を追加した。指定は/etc/host.conf または環境変数で指定できる。IP アドレスからホスト名を得る時に, 得られたホスト名から IP アドレスの再チェックや, このチェックに失敗したときに syslog へ出力するといった機能を持つ。

E: UNIX, SunOS 4.X では共有ライブラリを置き換え可能

U: ftp://ftp.dit.co.jp/pub/security/tools/resolv+2.1.1.tar.Z 他

4.2 ドキュメントの一覧

各ツールは次のタグづけをして紹介してある。

T:	title	(必須)	タイトル
A:	author	(必須)	著者
D:	date	(オプション)	発表された日
P:	page	(必須)	ページ数
K:	keyword	(必須)	キーワード
S:	summary	(オプション)	Abstract をさらに要約したもの
U:	url	(必須)	URL
N:	note	(オプション)	その他、備考 (発表された場所／学会など)

4.2.1 Firewall 関連

T: "The Design of a Secure Internet Gateway"

A: Bill Cheswick, AT&T Bell Labs <ches@research.att.com>

D: 91/09/10

P: 9

K: security,internet,gateway,mail,TCP

S: AT&T での firewall について書かれている。外部へアクセスする場合に用いている ptelnet, pftp などの proxy package や, HHA を用いてのログインなどについても紹介されている。

U: ftp://research.att.com/dist/internet_security/gateway_slides.ps

N: Proceedings of the 3rd USENIX UNIX Security Symposium, Baltimore, September 14-16, 1992.

T: "Network (In)Security Through IP Packet Filterring"

A: D. Brent Chapman, Great Circle Association <brent@greatcircle.com>

P: 14

K: security,filter,gateway,IP,packet,router

S: セキュリティの道具として、IP パケットフィルタリングの有効性について述べられている。また、現在のパケットフィルタリングの実装の多くが含んでいる問題点を明らかにし、その問題点の解決方法について議論され ている。

U: ftp://ftp.greatcircle.com/pub/firewalls/papers/chapman/pkt_filtering.ps.Z

N: Proceedings of the 3rd USENIX UNIX Security Symposium, Baltimore, September 14-16, 1992.

T: "X Through the Firewall, and Ohter Application Relays"

A: G.Winfield Treese, Alec Wolman

D: 93/03/03

P: 20

K: security,X,application,internet,firewall

S: アプリケーションに特化した「中継」を構築するための技術をいくつか 紹介している。また、アプリケーションに必要とされる機能を提供しながら セキュリティ要求を満足する方法について議論されている。

U: ftp://crl.dec.com/pub/DEC/CRL/tech-reports/93.10.ps.Z

N: Cambridge Research Lab Technical Report 93/10, DEC, May 3, 1993

4.2.2 Cracker,Worm 関連

T: "There Be Dragons"

A: Steven M. Bellovin, AT&T Bell Labs <smb@ulysses.att.com>

D: 92/08/15

P: 16

K: attack,trap,detect,logging,counterintelligence

S: cracker がどのように攻撃してくるかを調査するために、ATT.COM の インターネットゲートウェイに様々な trap プログラムや ログ機能を入れた経験談が書かれている。

U: ftp://research.att.com/dist/internet_security/dragon.dvi
ftp://research.att.com/dist/internet_security/dragon.ps

N: Proceedings of the 3rd USENIX UNIX Security Symposium, Baltimore September 14-16 1992 でも発表されている。

T: "An Evening with Berferd In Which a Cracker is Lured, Endured, and Studied"

A: Bill Cheswick, AT&T Bell Labs <ches@research.att.com>

D: 92/07

P: 11

K: cracker,security hole,sendmail

U: ftp://research.att.com/dist/internet_security/berferd.dvi
ftp://research.att.com/dist/internet_security/berferd.ps

S: Berferd 事件について、cracker の侵入口、侵入手段を調査するために、 AT&T の bastion host の ftp や telnet 等の標準的なサービスにおとりをしかけ、追跡をした記録について具体的に書かれている。

T: "The Greatest Cracker-Case in Denmark": The Detecting, Tracing and Arresting of Two International Crackers

A: Jorgen Bo Madsen <Jorgen.Bo.Madsen@uni-c.dk>

D: 92/09/15

P: 24

K: cracker,trace,detect,arrest

S: 1991 年 1 月、デンマークで起きた cracker 侵入事件で、cracker の証拠を集め るための行動追跡の方法等について書かれている。この追跡によって、cracker は警察に逮捕されるに至った。

N: Proceedings of the 3rd USENIX UNIX Security Symposium, Baltimore, September 14-16, 1992.

T: "Are Computer Hacker Break-Ins Ethical?"

A: Eugene H. Spafford, Purdue <spaf@cs.purdue.edu>

D: 90/07/ (Revised 91/04)

P: 15

K: incident,intrusion,break-in,ethics

U: ftp://ftp.cs.purdue.edu/pub/spaf/security/ethics.PS.Z

S: コンピュータへの侵入についての倫理について書かれている。以前に発生 した事故が引き起こした論議について取り上げ、侵入を正当化しようとす る理由を分析し、反証している。

N: Primary appearance: The Journal of Systems and Software, 17(1), pp. 41-48, Jan 1992. Also, Purdue TR994.

T: "The Internet Worm Program: An Analysis"

A: Eugene H. Spafford, Purdue <spaf@cs.purdue.edu>

S: The original Internet Worm analysis.

U: ftp://ftp.cs.purdue.edu/pub/spaf/security/IWorm.PS.Z

N: Primary appearance: ACM Computer Communication Review, January 1989, 19(1), pp. 17-57, Also, Purdue TR 823.

D: 88/12/08

P: 表紙+40

K: Worm

S: 1988 年 11 月 2 日の Worm プログラムのデータと機能の詳細の説明。 Worm プログラムによって悪用された欠陥の調査結果

T: "The Internet Worm Incident"

A: Eugene H. Spafford, Purdue <spaf@cs.purdue.edu>

S: The timeline and effects paper on the Worm.

U: ftp://ftp.cs.purdue.edu/pub/spaf/security/IWorm2.PS.Z

N: Primary appearance: Proceedings of the European Software Engineering Conference 1989 (Lecture Notes in CS #387, Springer-Verlag), pp. 446-468, Sept. 1989, Also, Purdue TR 933.

P: 表紙+18

S: 1988 年 11 月 2 日の Internet Worm 事件についての解説

K: Worm,Virus,Security

T: "Computer Viruses: A Form of Artificial Life?"

A: Eugene H. Spafford, Purdue <spaf@cs.purdue.edu>

S: The paper on computer viruses as a form of artificial life.

U: ftp://ftp.cs.purdue.edu/pub/spaf/security/ALife.PS.Z

N: Primary appearance: Artificial Life II, Studies in the Sciences of Complexity, vol. Xii (eds. D. Farmer, C.Langton, S. Rasmussen, and C. Taylor), Addison-Wesley, pp. 727-747, 1991, Also, Purdue TR 985. The current version has been updated and will appear in the first or second issue of the "Journal of Artificial Life" in 1994.

P: 23

T: "A Tour of the Worm"

A: Donn Seeley, Department of Computer Science

N: November 2, 1988, 15pages USENIX Winter Conference Proceedings, Jan. 1989

4.2.3　暗号関連

T: "Limitations of the Kerberos Authentiction System"

A: Steven M. Bellovin, AT&T Bell Labs, Michael Merritt, AT&T Bell Labs

D: 91/01/

P: 15

K: kerberos,authentication,limitation,weakness

U: ftp://research.att.com/dist/internet_security/kerblimit.usenix.ps

S: MIT の Project Athena によって導入された Kerberos について、主に Version 4 に焦点をあて、問題点を指摘している。Kerberos を一般的な環境で利用するための改良点、およびプロトコル自体の問題点について指摘している。Version 5 では、これらの指摘のいくつかは改善されている。

N: USENIX Conference Proceedings Dallas(Winter) Jan, '91 (A version of this paper was published in the October, 1990 issue of Computer Communications Review.)

第 5 章

おわりに

1993 年の秋の合宿から Firewall Working Group は活動をはじめた。研究初年なので、State of the Arts の確認のためのサーベイと、WIDE プロジェクト参加組織どうしでのノウハウの確認とが活動の中心となった。第 29 回 IETF(1994 年 3 月) の総会でインターネットのセキュリティの一側面として firewall がとりあげられた。 インターネットアーキテクチュアのなかでの firewall の関心が高まっている。 WIDE プロジェクト Firewall Working Group では、興味がある点を追求し、活動の成果をインターネット全体に還元することをめざしたい。