第15日 ファイアウォール

概要

ファイアウォールはルータやサーバを含むさまざまな場所で必須のセキュリティ対策です。今ではクライアントのOSにも、ウィルス対策ソフトにも入っています。ただし、TCP/IPプロトコル、特に、アプリケーションポート番号とその送受信の仕組みをよく理解しないと、接続できない、あるいは、逆にトラブルに巻き込まれることになります。一方、よく理解すれば、サーバ以外のルータやクライアントなどのファイアウォールにも適用できます。

目標

本単元では、ファイアウォールの仕組みと操作などを学習します。
ポイントは以下のようなものです。

◎TCP/IPプロトコルについて、アプリケーションのポート番号を覚え、送受信の仕組みを理解する。また、TCPとUDPの接続の違いを理解する。
◎iptablesの用語と仕組みを理解する。特に、接続拒否のDROPとREJECTとの違いを知る。
◎iptablesの基本的、必須の設定例を理解し、覚える。
◎iptablesの動作確認やトラブルシュートで有効なtcpdumpの基本的な使い方を理解する。
◎iptablesのログの取り方を学習する。
◎その他、iptables制御に関係するポイントを知っておく。

1 ファイアウォール

本単元のファイアウォールとしては、Linuxカーネルの IPパケットフィルタルールのテーブルを設定及び運用管理するiptablesを利用します。

1.1 iptablesの構造と仕組み

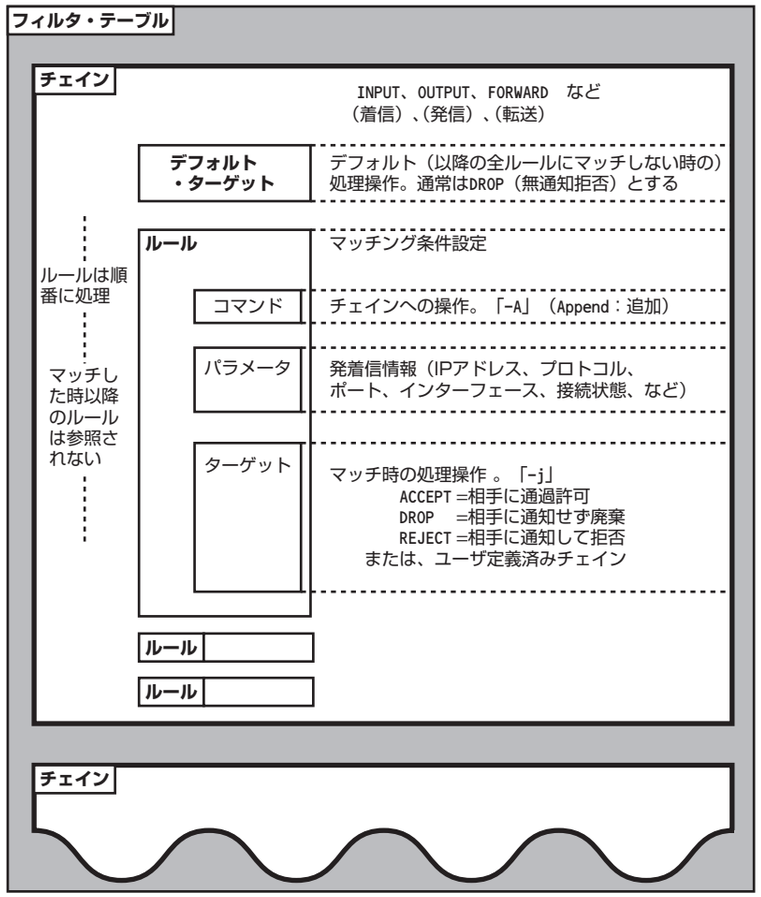
iptablesの一般的な構造は「チェイン」と呼ばれる方向性をもった「ルール」を連結した「テーブル」です。つまり、iptablesは受け取ったパケット1つ1つの方向性に対応した「テーブル」をチェインに従って処理（「ターゲット」）していきます。各用語の意味は表15-1のようなものです。

▼表15-1 iptablesの用語と意味・内容

名称	意味・内容
テーブル	ルール全体。代表的なものがfilter（フィルタ）処理
チェイン	ターゲット（マッチした場合の処理）となるパケットのルール（条件設定）の連結集合。代表的なものに以下のようなものがある ・組み込み済みチェイン INPUT …… 着信。自システムへの着信 OUTPUT …… 発信。自システムからの発信 FORWARD …… 転送。自システム経由の転送 ・ユーザ定義チェイン：新たに定義設定するチェイン
ルール	コマンド、パラメータ及びターゲットで構成される1エントリ行
ターゲット	ルールにマッチしたパケットに対して行う処理で、代表的なものに以下のようなものがある ACCEPT …… 通過を許可する DROP …… 相手に通知せずに廃棄（拒否）する REJECT …… 相手に拒否通知する（拒否する）

システム起動時のフィルタリング設定は他のサーバと同様に「/sbin/service iptables」というスクリプトで行われますが、このスクリプトは設定ファイル/etc/sysconfig/iptables内に定義済みのフィルタリング・ルールを使用します。インストール直後ではこの設定ファイルは存在しないので、iptablesコマンドでフィルタリング・ルールを1行ずつチェインに追加設定して作成したテーブルをiptablesのsaveコマンド（/sbin/service iptables save）で設定ファイルに格納する方法と、viなどで設定ファイルを新規作成する方法とがあります。本単元ではあらかじめこのルールファイルを作成しておいて、スクリプトを実行することにします。

iptablesルールファイルの一般的なパケットフィルタテーブルの構造は図15-1のようなものです。具体的には、図15-2のように、各チェイン毎にデフォルト設定をしておき、チェインのルールを記述していきます。



▲図 15-1　定義設定済みのルールファイル「/etc/sysconfig/iptables」の構造

*filter	フィルタテーブル宣言
:INPUT　～	着信チェインデフォルトターゲット（処理）
:OUTPUT　～	発信チェインデフォルトターゲット
:FORWARD　～	転送チェインデフォルトターゲット
:ユーザ定義　～	ユーザ定義チェインデフォルトターゲット
-A INPUT　～	着信チェインルール
・・・	・・・
-A OUTPUT　～	発信チェインルール
・・・	・・・
COMMIT	テーブル発効

▲図 15-2　「/etc/sysconfig/iptables」の具体的な記述

なお、iptablesスクリプト自体の処理は、iptablesルールファイルを参照して以下のような順序でテーブルを設定します。

- 現在のフィルタ設定をクリアする（デフォルト処理はそのまま残す）
- チェインの個々のルール設定を順番に行う

このような順序で作成したフィルタルールが、iptablesを通過する1つ1つのパケットに適用されます。着信、発信、あるいは、転送のためにiptablesに入ってきた各パケットは、指定したチェインのルールの条件にマッチしない場合、チェイン内の次のルールが評価されます。パケットがマッチした場合、ターゲットの値による処理が行われ、以降のルールは参照されません。また、最後のルールまでマッチしない場合には、最初に定義されている、それぞれのチェインに対応したデフォルトのターゲット（処理操作）が行われます。

1.2 TCP と UDP の処理の違い

iptablesルールの記述形式は以下のようなものです。表15-2に、詳細設定パラメータをまとめてあります。

【形式－－一般形】

コマンド　チェイン　IPアドレス　プロトコル　マッチング拡張
ポート　I/F　ターゲット

▼表 15-2　iptables ルールの詳細設定パラメータ

①コマンド	
	・**-A**：追加（Append）。ルール設定を追加する。 ・**-D**：削除（Delete）。チェインまたは指定済みのルール（番号）を削除する ・**-R**：置換（Replace）。指定済みのルール（番号）を新ルール設定で置き換える ・**-F**：クリア(Flush)。指定した(指定しなければすべての)チェインのフィルタ設定をクリアする。ただし、デフォルト設定は残る ・**-L**：フィルター覧の表示（List）
②チェイン	
	組み込み済みチェイン **INPUT**　：着信。自システムへの着信（外部から来てこのシステムで終結するもの） **OUTPUT**　：発信。自システムからの発信（このシステムから発信して外部に出て行くもの） **FORWARD**：転送。自システム経由の転送
	ユーザ定義チェイン：新たに定義設定するチェイン
③ IP アドレス	
	-s　発信 IP アドレス［/ ビットマスク］ **-d**　宛先 IP アドレス［/ ビットマスク］
④プロトコル	
	-p　プロトコル （**tcp** または **6**、**udp** または **17**、**icmp** または **1**、**all** または **0**、または無指定） icmp オプション **--icmp-type**　タイプ名 タイプ名）ICMP メッセージの種類（ICMP メッセージコード） 代表的な ICMP タイプの例 **echo-request**：エコー要求（ping）、**echo-reply**：エコー応答、 **destination-unreachable**：宛先到達不能 , **source-quench**：発信元送信抑制要求、**redirect**：リダイレクト

⑤マッチング拡張
コネクション（接続）状態 **-m state --state 状態** 状態）**NEW**：接続開始、**ESTABLISHED**：確立状態、**RELATED**：既存接続関係（TCP だけではなく UDP や ICMP などの送信パケットに関係する着信も含む）
⑥アプリケーション・ポート番号
--sport　発信ポート番号 **--dport**　宛先ポート番号 ポート番号：範囲指定（番号：番号）、サービス名（**telnet** とか **smtp** など）、否定（先頭に「**!**」付加）
⑦ I/F（インタフェースデバイス名）
LAN のインタフェース（**eth0** など）やループバック（**lo**）、指定しない場合は「すべて」 **-i** 着信用インタフェース・デバイス名 **-o** 発信用インタフェース・デバイス名
⑧ターゲット（処理操作）
・ **ACCEPT**：そのルールにマッチしたパケットの通過を許可する ・ **DROP**　：そのルールにマッチしたパケットの通過を許可しない（*） ・ **REJECT**：そのルールにマッチしたパケットに通過を拒否する（*） REJECT オプション **--reject-with REJECT** タイプ **icmp-net-unreachable**　：ネットワーク到達不能 **icmp-host-unreachable**　：ホスト到達不能 **icmp-port-unreachable**　：ポート到達不能（デフォルト） **icmp-proto-unreachable**：プロトコル到達不能 **icmp-net-prohibited**　：ネットワークアクセス禁止 **icmp-host-prohibited**　：ホストアクセス禁止 ・ **REDIRECT**：そのルールにマッチしたパケットの着信ポートの変更 ・ **LOG**：ログをとる LOG オプション **--log-level level**　：数値またはレベル名で指定されたレベルのログをとる **--log-prefix prefix**：指定したプレフィックスをログメッセージの先頭に付加する。長さは 29 文字まで **--log-tcp-sequence**：TCP シーケンス番号をログに記録する **--log-tcp-options**　：TCP パケットヘッダのオプションをログに記録する **--log-ip-options**　：IP パケットヘッダのオプションをログに記録する
⑨その他
-f（または、**--fragment**）：分割フラグメントの 2 パケット目以降を対象。否定は前に "**!**"

* REJECT と DROP の違い
REJECT：発信者に ICMP 宛先到達不能メッセージを送信。DROP では無通知。システムの負荷やセキュリティ（相手に情報を与えない）の点から DROP がよく使用される。

これ以降はこのようなパラメータを使用してiptablesの動作確認を行いますが、TCPとUDPのプロトコル上の違いから来る設定の違いが重要です。TCPではコネクションという状態（接続状態）があり、その間だけデータ送受信されます。そのため、接続できないならばデータ送受信もできません。逆にいえば、データ送受信させたくない場合には接続を受け付けず、データ送受信させたい場合には接続を受け付けるようにすればよい、つまり、最初の接続要求だけでアクセス制御が可能ということになります。一方、UDPではそのような接続という状態がないのですが、送信パケットに関する状況（状態）を監視していて関連する受信パケットを受け付けることができます（詳細は、「第26日運用管理技術　1 現実のファイアウォール」参照）。

iptablesでは、表15-2にあるようにTCPプロトコルでのTCP接続の状態や処理の設定パラメータを「state」で定義し、接続確立のための処理を「NEW」（接続開始）、TCP接続確立状態（確立された状態）を「ESTABLISHED」（確立状態）で定義しています。

例えば、以下の設定は、192.168.0.0/24からSSHサーバとの間のデータ送受信を許可するルールです。

```
-A INPUT -s 192.168.0.0/24 -p tcp -m tcp --dport 22 -m state --state NEW -j
    ACCEPT
```
（192.168.0.0/24 からの SSH サーバへの接続開始を許可）

```
-A INPUT -p tcp -m state --state ESTABLISHED -j ACCEPT
```
（接続確立済み状態での送受信は許可）

備考　TCP／UDP プロトコルを使用するアプリケーションプロトコル

TCPのみ使用：TELNET、FTP、メール（SMTP/POP3）、X（Xウィンドウ）など多数。
UDPのみ使用：SNMP（ネットワーク管理）、DHCP（自動IPアドレス割り振り）、など少数。
両方使用　　：DNS（通常UDP、ゾーン転送TCP）、NetBIOS（データ送受信TCP、他UDP）

1.3 iptablesの設定と動作確認

ここからiptablesのテストを行います。基本方針は以下の通りです。

- デフォルト設定をすべて無通知拒否（DROP）にしておく
- ループバックインタフェース（lo）経由の発着信は許可する
- DNS名前解決（UDP）は送受信ともに許可する
- サーバからの発信は基本的に許可する
- テスト用に特定のアプリケーションへのアクセスを禁止したり許可したりする

なお、サーバ側でtcpdump（パケットトラフィックの捕捉表示ツール）で送受信データを記録して、パケットの流れでフィルタリング処理を確認することにします（備考参照）。

重要注意

ここでiptablesによりアクセス制御を設定したので、今まで利用していたPOP3やNetBIOSなどがブロックされる。したがってこれ以降、再度、メール送受信やsambaなどを行う場合は（iptablesを止めて行うか）動作許可設定を行わなければならない。

注意すべきポートは、telnet（TCP/23）、メール受信（TCP/110）、ftp（TCP/21,20）、WWWブラウザからのVNC接続（TCP/5801）など。

備考 **tcpdump**

tcpdumpは、そのシステムが接続しているネットワーク上のトラフィック（パケットの流れ）を記録し、パケットのヘッダを時系列で表示する。このtcpdumpリストを分析することで、ネットワーク上のトラフィックを把握し、運用管理に役立てることが可能。tcpdumpでは、パケットのプロトコル（tcp、udp、icmpなど）やタイムスタンプ（時間）、発信識別子（発信システムの名前またはIPアドレスとアプリケーションの名前またはポート番号）、方向（>：左から右）、宛先識別子（宛先システムの名前またはIPアドレスとアプリケーションの名前またはポート番号）そして、パケットのヘッダ情報、をインタフェース（NICやループバックインタフェースlo、デフォルトは最初の物理インタフェース）毎に記録する。

1.3.1 設定例

上記前提で設定したフィルタテーブルの例がリスト15-1です。許可前提のアプリケーションは、TCP/SSH（ポート22）着信、TCP/SMTP着信（ポート25）（備考）、TCP/DNS（ポート53）着信、UDP/DNS（ポート53）着信、TCP/WWW（ポート80）着信、TCP/SSL-WWW（ポート443）着信、TCP/SMTPS（ポート465）着信、TCP/POP3S（ポート995）着信、そして、TCP/VNC（ポート5901）着信です。これを「/sbin/service iptables (re)start」で起動した後、リスト15-2のように実際の設定状況を確認します。

なお、iptablesの自動起動設定はデフォルトでONになっているので変更の必要はありません。

備考 **SMTPポートを許可して、POP3ポートを許可しないのは**

SMTPポートはメールクライアントからの送信の他に、外部SMTPサーバとの間のメール送受信に使用される。したがって、メールサーバでは外部からの着信許可は必須である。一方、POP3ポートはメールクライアント（一般に、LAN内部）との間の通信でしか利用されない。もし、外部との間で必要であれば、POP3Sを使用すればよい。

▼**リスト15-1** iptables（/etc/sysconfig/iptables）の設定

```
*filter                                          ←フィルタテーブル宣言
:INPUT DROP [0:0]                                ←着信デフォルトは無通知拒否
:FORWARD DROP [0:0]                              ←転送デフォルトは無通知拒否
:OUTPUT DROP [0:0]                               ←発信デフォルトは無通知拒否
:logdrop - [0:0]                                 ←無通知拒否のログ logdrop の定義

-A INPUT -i lo -j ACCEPT          ←ループバックインタフェース着信はすべて許可
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT ← TCP/SSH 着信を許可
-A INPUT -p tcp -m tcp --dport 25 -j ACCEPT ← TCP/SMTP 着信を許可
-A INPUT -p tcp -m tcp --dport 53 -j ACCEPT ← TCP/DNS 着信を許可
-A INPUT -p udp -m udp --dport 53 -j ACCEPT ← UDP/DNS 着信を許可
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT ← TCP/WWW 着信を許可
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT ← TCP/SSL-WWW 着信を許可
-A INPUT -p tcp -m tcp --dport 465 -j ACCEPT ← TCP/SMTPS 着信を許可
-A INPUT -p tcp -m tcp --dport 995 -j ACCEPT ← TCP/POP3S 着信を許可
-A INPUT -p tcp -m tcp --dport 5901 -j ACCEPT ← TCP/VNC 着信を許可
-A INPUT -j logdrop                               ←その他の着信は logdrop チェインへ

-A OUTPUT -j ACCEPT                              ←発信は許可

-A logdrop -j LOG                                ← logdrop チェインはログを取って
-A logdrop -j DROP                               ←無通知拒否

COMMIT                                           ←フィルタテーブルの発効
```

▼**リスト15-2** iptables 起動後の設定状況確認

コマンド「**iptables --list -v --line-numbers**」（リスト、詳細、行番号付き）で表示

```
Chain INPUT (policy DROP 0 packets, 0 bytes)        ←着信チェイン
num   pkts bytes target     prot opt in     out     source               destination
1      895 50748 ACCEPT     all  --  lo     any     anywhere             anywhere
2      429 23424 ACCEPT     tcp  --  any    any     anywhere             anywhere            tcp dpt:ssh
3        0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere            tcp dpt:smtp
4        0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere            tcp dpt:domain
5        0     0 ACCEPT     udp  --  any    any     anywhere             anywhere            udp dpt:domain
6        0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere            tcp dpt:http
7        0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere            tcp dpt:https
8        0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere            tcp dpt:smtps
9        0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere            tcp dpt:pop3s
10       8   921 logdrop    all  --  any    any     anywhere             anywhere

Chain FORWARD (policy DROP 0 packets, 0 bytes)      ←転送チェイン
num   pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy DROP 0 packets, 0 bytes)       ←発信チェイン
num   pkts bytes target     prot opt in     out     source               destination
1     1256 76008 ACCEPT     all  --  any    any     anywhere             anywhere

Chain logdrop (1 references)                        ← logdrop チェイン
num   pkts bytes target     prot opt in     out     source               destination
1        8   921 LOG        all  --  any    any     anywhere             anywhere            LOG level warning
2        8   921 DROP       all  --  any    any     anywhere             anywhere
```

1.3.2 動作確認

†1 iptables の自動起動設定（すべてオン）はインストール時になされている。「chkconfig --list iptables」で確認。

iptables起動後、クライアントからサーバアプリケーション宛に接続して、その動作を確認します[†1]。なお、実際の場ではこうした、拒否確認の他に、必ず許可確認も行っておきます。

接続および拒否の確認を行うアプリケーションは、ssh、telnet、メール受信（pop3）、そして、sambaです。Windowsとsambaサーバとの間では定期的にブラウズ情報のやりとりをしているので、あえてマイネットワークを開かなくても動作確認ができますが、すぐにログをとりたい場合には、sambaサーバを開く操作（ブロックされてエラーとなるが）を行います。

リスト15-3がiptablesフィルタブロック（拒否）のログで、NetBIOS着信、telnet着信、そして、pop3着信のブロックが記録されています。

同時にまた、リスト15-4のようにtcpdumpでログをとりパケットの動きを確認します。ここでも、NetBIOS、telnet、そして、pop3、のブロックが解析されています。また、sshの通過も記録されています。

▼**リスト15-3** iptablesフィルタブロックのログ（/var/log/messages）

```
【NetBIOS着信（137/138/139）のブロック】
Jul 18 17:05:48 h2g kernel: IN=eth0 OUT= MAC=00:e0:18:ef:b8:e8:00:02:55:17:75:12:08:00 SRC=192.168.0.8
    DST=192.168.0.18 LEN=64 TOS=0x00 PREC=0x00 TTL=128 ID=4791 DF PROTO=TCP SPT=1050 DPT=139 WINDOW=65535 RES=0x00
    SYN URGP=0
Jul 18 17:05:51 h2g kernel: IN=eth0 OUT= MAC=00:e0:18:ef:b8:e8:00:02:55:17:75:12:08:00 SRC=192.168.0.8
    DST=192.168.0.18 LEN=64 TOS=0x00 PREC=0x00 TTL=128 ID=4792 DF PROTO=TCP SPT=1050 DPT=139 WINDOW=65535 RES=0x00
    SYN URGP=0
Jul 18 17:05:57 h2g kernel: IN=eth0 OUT= MAC=00:e0:18:ef:b8:e8:00:02:55:17:75:12:08:00 SRC=192.168.0.8
    DST=192.168.0.18 LEN=64 TOS=0x00 PREC=0x00 TTL=128 ID=4793 DF PROTO=TCP SPT=1050 DPT=139 WINDOW=65535 RES=0x00
    SYN URGP=0
Jul 18 17:06:09 h2g kernel: IN=eth0 OUT= MAC=ff:ff:ff:ff:ff:ff:00:02:55:17:75:12:08:00 SRC=192.168.0.8
    DST=192.168.0.255 LEN=202 TOS=0x00 PREC=0x00 TTL=128 ID=4795 PROTO=UDP SPT=138 DPT=138 LEN=182
Jul 18 17:06:09 h2g kernel: IN=eth0 OUT= MAC=ff:ff:ff:ff:ff:ff:00:02:55:17:75:12:08:00 SRC=192.168.0.8
    DST=192.168.0.255 LEN=78 TOS=0x00 PREC=0x00 TTL=128 ID=4796 PROTO=UDP SPT=137 DPT=137 LEN=58
Jul 18 17:06:10 h2g kernel: IN=eth0 OUT= MAC=ff:ff:ff:ff:ff:ff:00:02:55:17:75:12:08:00 SRC=192.168.0.8
    DST=192.168.0.255 LEN=78 TOS=0x00 PREC=0x00 TTL=128 ID=4797 PROTO=UDP SPT=137 DPT=137 LEN=58
Jul 18 17:06:11 h2g kernel: IN=eth0 OUT= MAC=ff:ff:ff:ff:ff:ff:00:02:55:17:75:12:08:00 SRC=192.168.0.8
    DST=192.168.0.255 LEN=78 TOS=0x00 PREC=0x00 TTL=128 ID=4798 PROTO=UDP SPT=137 DPT=137 LEN=58

【telnet着信（23）のブロック】
Jul 18 17:04:56 h2g kernel: IN=eth0 OUT= MAC=00:e0:18:ef:b8:e8:00:02:55:17:75:12:08:00 SRC=192.168.0.8
    DST=192.168.0.18 LEN=64 TOS=0x00 PREC=0x00 TTL=128 ID=4788 DF PROTO=TCP SPT=1049 DPT=23 WINDOW=65535 RES=0x00
    SYN URGP=0
Jul 18 17:04:59 h2g kernel: IN=eth0 OUT= MAC=00:e0:18:ef:b8:e8:00:02:55:17:75:12:08:00 SRC=192.168.0.8
    DST=192.168.0.18 LEN=64 TOS=0x00 PREC=0x00 TTL=128 ID=4789 DF PROTO=TCP SPT=1049 DPT=23 WINDOW=65535 RES=0x00
    SYN URGP=0
Jul 18 17:05:05 h2g kernel: IN=eth0 OUT= MAC=00:e0:18:ef:b8:e8:00:02:55:17:75:12:08:00 SRC=192.168.0.8
    DST=192.168.0.18 LEN=64 TOS=0x00 PREC=0x00 TTL=128 ID=4790 DF PROTO=TCP SPT=1049 DPT=23 WINDOW=65535 RES=0x00
    SYN URGP=0

【pop3着信（110）のブロック】
Jul 18 17:06:08 h2g kernel: IN=eth0 OUT= MAC=00:e0:18:ef:b8:e8:00:02:55:17:75:12:08:00 SRC=192.168.0.8
    DST=192.168.0.18 LEN=64 TOS=0x00 PREC=0x00 TTL=128 ID=4794 DF PROTO=TCP SPT=1051 DPT=110 WINDOW=65535
    RES=0x00 SYN URGP=0
Jul 18 17:06:11 h2g kernel: IN=eth0 OUT= MAC=00:e0:18:ef:b8:e8:00:02:55:17:75:12:08:00 SRC=192.168.0.8
    DST=192.168.0.18 LEN=64 TOS=0x00 PREC=0x00 TTL=128 ID=4799 DF PROTO=TCP SPT=1051 DPT=110 WINDOW=65535 RES=0x00
    SYN URGP=0
Jul 18 17:06:17 h2g kernel: IN=eth0 OUT= MAC=00:e0:18:ef:b8:e8:00:02:55:17:75:12:08:00 SRC=192.168.0.8
    DST=192.168.0.18 LEN=64 TOS=0x00 PREC=0x00 TTL=128 ID=4804 DF PROTO=TCP SPT=1051 DPT=110 WINDOW=65535
    RES=0x00 SYN URGP=0
```

▼**リスト15-4** iptablesによる許可・拒否アクセスのtcpdumpログ

```
コマンド「tcpdump > tcpdump.log」を実行してログtcpdump.logを取り、後から"grep netbios"などにより抜き出して分析
【NetBIOSがブロックされたログ】（NetBIOSに応答しない）
17:05:48.659162 IP i6290nj.example.com.cma > h2g.example.com.netbios-ssn: S 3344655648:3344655648(0) win 65535
    <mss 1460,nop,wscale 1,nop,nop,timestamp 0 0,nop,nop,sackOK>
17:05:51.517401 IP i6290nj.example.com.cma > h2g.example.com.netbios-ssn: S 3344655648:3344655648(0) win 65535
    <mss 1460,nop,wscale 1,nop,nop,timestamp 0 0,nop,nop,sackOK>
17:05:57.533082 IP i6290nj.example.com.cma > h2g.example.com.netbios-ssn: S 3344655648:3344655648(0) win 65535
    <mss 1460,nop,wscale 1,nop,nop,timestamp 0 0,nop,nop,sackOK>
17:06:09.564883 IP i6290nj.example.com.netbios-dgm > 192.168.0.255.netbios-dgm: NBT UDP PACKET(138)
17:06:09.564988 IP i6290nj.example.com.netbios-ns > 192.168.0.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST;
    BROADCAST
17:06:10.314470 IP i6290nj.example.com.netbios-ns > 192.168.0.255.netbios-ns: NBT UDP PACKET(137): QUERY;
    REQUEST; BROADCAST
17:06:11.064491 IP i6290nj.example.com.netbios-ns > 192.168.0.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST;
    BROADCAST

【telnetがブロックされたログ】（telnetに応答しない）
17:04:56.792471 IP i6290nj.example.com.td-postman > h2g.example.com.telnet: S 1496055248:1496055248(0) win 65535
    <mss 1460,nop,wscale 1,nop,nop,timestamp 0 0,nop,nop,sackOK>
17:04:59.782634 IP i6290nj.example.com.td-postman > h2g.example.com.telnet: S 1496055248:1496055248(0) win 65535
    <mss 1460,nop,wscale 1,nop,nop,timestamp 0 0,nop,nop,sackOK>
17:05:05.798290 IP i6290nj.example.com.td-postman > h2g.example.com.telnet: S 1496055248:1496055248(0) win 65535
    <mss 1460,nop,wscale 1,nop,nop,timestamp 0 0,nop,nop,sackOK>

【pop3がブロックされたログ】（pop3に応答しない）
19:25:56.491946 i6290nj.example.com.1173 > h2g.example.com.pop3: S 346635864:346635864(0) win 65534 <mss
    1460,nop,wscale 1,nop,nop,timestamp 0 0,nop,nop,sackOK> (DF)
19:25:59.424151 i6290nj.example.com.1173 > h2g.example.com.pop3: S 346635864:346635864(0) win 65534 <mss
    1460,nop,wscale 1,nop,nop,timestamp 0 0,nop,nop,sackOK> (DF)
19:26:05.433588 i6290nj.example.com.1173 > h2g.example.com.pop3: S 346635864:346635864(0) win 65534 <mss
    1460,nop,wscale 1,nop,nop,timestamp 0 0,nop,nop,sackOK> (DF)

【sshが通過したログ】（ssh送受信OK）
17:04:15.584139 IP h2g.example.com.ssh > i6290nj.example.com.dcutility: P 3278221859:3278221959(100) ack
    4012357795 win 499
17:04:15.584640 IP h2g.example.com.ssh > i6290nj.example.com.dcutility: P 100:192(92) ack 1 win 499
17:04:15.584831 IP i6290nj.example.com.dcutility > h2g.example.com.ssh: . ack 192 win 64807
17:04:19.815102 IP h2g.example.com.ssh-mgmt > cumin.apnic.net.domain:  20132% [1au] A? ns5.apnic.net. (42)
17:04:20.032840 IP cumin.apnic.net.domain > h2g.example.com.ssh-mgmt:  20132*- 1/0/1 A licorice.apnic.net (58)
```

iptablesについては、第18日のIPsecでも再設定しますが、基本的な設定を覚えておきます。なお、第26日には実際の場でのファイアウォール設定を学習します。

1.4 その他

フィルタリング処理の着信では最低限のアクセスのみ受付するようにして、不要な処理を受付しないようにすることが重要です。例えば、telnetやpop3、ftpなどの平文送受信のアプリケーションはSSHやSSLなどの暗号化トンネル内で使用し、したがって、iptablesでは許可しないことです。また、SSHやSSLなどを使用するにしても、場合に応じて発信元を制限するといったことも重要です。

1 外部からのアクセスを必ず日常的に監視する

迷惑行為、ハッキング行為などへの拒否処理では、「REJECT」ではなく「DROP」にします。「REJECT」が相手に対して「パケットが到達しなかった（拒否した）ことを通知する（ICMP宛先到達不能メッセージを送信）」のに対して、「DROP」は「何も通知しない」からです。システムの負荷やセキュリティ（相手に情報を与えない）の点からも「DROP」がよく使用されます。唯一の例外が、クライアントから接続のあった際に、（一部の）サーバアプリケーションが発信元になって相手情報をとりに行くAUTH/IDENTプロトコルです。この処理が長引くとそのクライアントからサーバへの以降の接続処理が遅延するので、また、実際上不要な処理なので、このクライアントのAUTH/IDENTポート113への発信に対して「通知拒否＝REJECT」を返す設定にします。

2 iptables のログの区別

iptablesで拒否にも許可にもログ設定（-j　LOG）した場合、以下のログのように/var/log/messages にログされたエントリが ACCEPT（許可）したものに対するログか、DROP（無通知拒否）したものに対するログか、いずれであるかが区別しにくくなります。

```
Jul 18 18:40:22 h2g kernel: IN=eth0 OUT= MAC=00:e0:18:ef:b8:e8:00:02:55:
17:75:12:08:00 SRC=192.168.0.8 DST=192.168.0.18 LEN=64 TOS=0x00
PREC=0x00 TTL=128 ID=39227 DF PROTO=TCP SPT=1063 DPT=23 WINDOW=65535
RES=0x00 SYN URGP=0
```

そこで、許可と拒否を区別するログ識別子を設定して、すぐに判別できるように、/etc/sysconfig/iptablesの設定を変更します。

以下のように、組み込みテーブル（INPUT/OUTPUT/FORWARD）のデフォルト設定の下に2つの私用テーブルを作成（ACCEPT＋ログ、DROP＋ログ）し、それぞれのログにプレフィックス（先頭語）を付加して判別可能にします。

```
:logaccept - [0:0]          ←「ログを取り許可する」テーブル
:logdrop - [0:0]            ←「ログを取り拒絶する」テーブル
-A logaccept  -j  LOG  --log-prefix  [Accept-Log]: ←プレフィクス
-A logaccept  -j  ACCEPT    ←許可
-A logdrop  -j  LOG  --log-prefix  [Drop-Log]:     ←プレフィクス
-A logdrop  -j  DROP        ←無通知拒否
```


これによって、「logaccept」と「logdrop」を使用したルールは、例えば以下のように記録され判別が可能になります。

```
Jul 18 18:40:22 h2g kernel: [Drop-Log]:IN=eth0 OUT= MAC=00:e0:18:ef:b8:e
8:00:02:55:17:75:12:08:00 SRC=192.168.0.8 DST=192.168.0.18 LEN=64
TOS=0x00 PREC=0x00 TTL=128 ID=39227 DF PROTO=TCP SPT=1063 DPT=23
WINDOW=65535 RES=0x00 SYN URGP=0
```

3 iptablesの「再起動」

一般のサーバ・サービスでは、「再起動（restart）」は「停止（stop）」の後、「開始（start）」ですが、iptablesの「再起動」は「停止」して「開始」するわけではありません。

iptablesの「再起動」はデフォルト・ポリシーを除くすべてのルールをFLUSH（クリア）してから新しいルールを追加します。そのため、追加するルールが間違っていてエラー停止すると、デフォルト・ポリシーのみが残ります。そこで、デフォルト・ポリシーが重要です。デフォルト・ポリシーのINPUTがDROPならば、外部からのアクセスはすべて不可能で、ACCEPTならばすべてからアクセス可能になります。つまり、デフォルト・ポリシーのINPUTをDROPにしておけば、誰もアクセスできないのでセキュリティ上安全です（正当なユーザにとっては不満足ですが）。一方、逆のACCEPTであると、誰からも、つまり、攻撃者もアクセスできるのでセキュリティ上の問題が出てきます。

4 Gnomeのログインでハングアップ

GUIのGnome（Xウィンドウ）でログインする際、CentOS 5.5ではloインタフェース経由の発着信が許可設定されていないと、ハングアップすることがあります。

評価チェック

ファイアウォールを完了したならば、評価ユーティリティ eval を実行して下さい。

```
/root/work/evalsh
```

ファイアウォールが完了したので、eval は［FIREWALL］のチェックを終えて、次の単元の［SSH V2］のエラーを表示します。もし、［FIREWALL］のところで、［FAIL］が表示された場合には、メッセージに従って、本単元の iptables の学習に戻って下さい。
［FIREWALL］を完了したならば、次の単元に進みます。

要点整理

本単元では、iptables を例にファイアウォール（パケットフィルタリング）を学習しました。覚えておくポイントは以下のようなところです。

- TCP/IP アプリケーションの基本的な仕組みとポート番号。
- iptables のフィルタテーブル宣言から、チェインデフォルト設定、各チェインのルール設定、そしてフィルタ発効までの詳細かつ具体的な設定。
- iptables ルールの記述形式と詳細設定パラメータ。
- TCP 接続とその後のデータ送受信の個別の設定と、UDP アプリケーションの設定。
- フィルタログの２つの切り分け（許可と拒否）設定。
- フィルタログの確認と tcpdump によるパケットの流れの確認方法。
- iptables 再起動時の「フラッシュ」の意味。

これらは、インターネット接続時の最前線に置かれるものなので、厳密かつ正確に把握しておくことが重要です。

学習者の眼

確認作業も怠りなく

本単元のメインテーマはファイアウォール設定です。ファイアウォールはセキュリティ関連の中でも重要な機能の 1 つであり、サーバ管理者としては当然知っておきたい機能でした。

セキュリティの重要性が再認識されている現在では、クライアント PC でもセキュリティソフトが導入されているので、多少は心得ているつもりでした。ただ、今回学習して改めて確認できたことはよかったです。また、詳細な仕組みや設定内容に関してはまったく知らなかったので､今回学習できたことはとてもためになりました。特にフィルタリング設定等は、詳細部分まで理解して作業することが重要であるとわかりました。

以下に、実際の設定、確認作業に関する注意点、問題点を挙げます。

●ファイアウォールの設定時のポイント

作業量はそれほど多くありませんが、iptables の設定が複雑でした。

(1) iptables のファイルの所在

iptables は元々存在しないので自分で作成（/etc/sysconfig/iptables）しました。最初は所定のディレクトリに iptables が存在しなくて迷いましたが、自分で作成して iptables が起動でき、問題ありませんでした。

(2) iptables の設定ミス防止

iptables のフィルタリング設定時に指定するプロトコルやアプリケーションポートの設定は、慎重に実施して下さい。意図しないフィルタリングや、全くフィルタリングされていなかった等、設定直後に気付けばすぐに修正できますが、あとになれば設定ミスだと気付くのに時間がかかるかと思います。

iptables 起動（service iptables restart 実施）後、"iptables --list -v --line-numbers" コマンドを実施し、動作中のフィルタリング設定の確認を実施すると設定ミスにすぐに気付くかと思います。

●ファイアウォールの動作確認時のポイント

次に、ファイアウォールの動作確認ですが、"/var/log/messages" と "tcpdump" から動作確認します。

(1) iptables フィルタブロックのログ確認

"/var/log/messages" を開きながらブロック設定されている動作を確認すると分かりやすいかと思います ("tail -f /var/log/messages" で出力させると便利です)。以下は動作確認の手段です。

① NetBIOS の着信ブロック

クライアント PC よりデスクトップ上の［マイネットワーク］-［ワークグループのコンピュータを表示する］より同一ネットワークに所属する PC 一覧が表示されます。この時点で "/var/log/messages" にブロックログが出力されます。

② https の着信ブロック

第 13 日目で SSL-Web 環境を構築したので、それを利用します。Web ブラウザより URL（https://www.example.com/secure.shtml）を入力します。この時点で "/var/log/messages" にブロックログが出力されます。なお、SSL-Web には接続不可となります

③ POP3 着信ブロック

使用しているメーラー（OutlookExpress6 を使用）でメール送受信実施します。この時点で "/var/log/messages" にブロックログが出力されます。なお、メール受信は不可となります。

※第 13 日で SSL メールのポート番号（smtps:465,pop3s:995）に変更している場合、元に戻してから実施します。

④ ssh 着信ブロック

第 14 日で作成した ssh バッチファイルを実施します。この時点で "/var/log/messages" にブロックログが出力されます。なお、ssh 接続は不可となります。

(2) tcpdump ログの確認

以下の順で実施します。

① tcpdump > tcpdump.log 実施

②上記確認動作（着信ブロック動作）

③強制終了（Ctrl ＋ C）実施

④ログファイル（tcpdump.log）の内容確認実施

本単元は確認事項が多いので、ログ出力内容や確認箇所を把握しておくことが重要です。

ファイアウォール関連のログは本番のサーバ管理業務でも参照することが多いので、ここで訓練しておくとよいかと思いました。