2007－DSM－47 (9)
2007／9／21

ホスト間連携を可能にするパスワード総当たり攻撃対策手法

大隅 淑弘†，山井 成良†

†岡山大学総合情報基盤センター

〒700-8530 岡山市津島中３丁目１番１号

E-mail：{oosumi,yamai}@cc.okayama-u.ac.jp

あらまし 近年，不正アクセスの侵入手口として，SSH などのサービスに対するパスワード総当たり攻撃が多くなっている．サービスによっては接続を受け入れる範囲を限定できないものがあり，また，登録ユーザには脆弱なパスワードを設定している者がある．このため，何度もの接続試行によってパスワードを破られ，計算機に不正に侵入される危険がある．本研究ではアクセスログを監視してこのような攻撃を検出し，ホスト間で連携して不正アクセスを防止する方式を提案する．

Technique of the countermeasure for brute force attack which can cooperate between the hosts

Yoshihiro OOSUMI †, Nariyoshi YAMAI †

† Information Technology Center, Okayama University

3-1-1, Tsushima-naka, Okayama, 700-8530, Japan
E-mail：{oosumi, yamai} @cc.okayama-u.ac.jp

Abstract Recently, as an invasion method of illegal access, brute force attacks for network services such as SSH, POP, and so on have been increasing. Since some users set vulnerable passwords for these network services, such brute force attacks have been real threats to unauthorized invasion. In this paper, we propose a countermeasure technique for brute force attack in collaboration with other hosts.

1. はじめに

計算機の利用はネットワークを経由することが多くなった．自宅や外出先からでも手元のパソコンからサーバにアクセスすれば，多くのサービスが利用できる．岡山大学でも学外から利用できる各種のサービスを提供しており，SSH（Secure Shell）や POP（Post Office Protocol）などはその代表的な例である．ネットワークからの利用では，パスワードや通信内容を盗聴されないために，暗号化した通信を行うのが一般的である．ところが近年，サーバに不正に侵入する手口として，SSH のパスワード認証に対する総当たり攻撃や辞書攻撃が非常に多くなっている．総当たり攻撃とは，何らかの規則によって文字列の組み合わせを作り，SSH での接続を何度も試行して計算機に不正に侵入する攻撃であり，辞書攻撃とは，パスワードとして使用されそうな文字列の集合を辞書として用意し，SSH での接続を何度も試行して，同様な不正侵入を試みる攻撃である．侵入された計算機では，情報漏洩やさらなる攻撃への踏み台にされることになる．2005 年 11 月に有限責任中間法人 JPCERT コーディネーションセンターから発表された「インターネットセキュリティーに対する JPCERT/CC 2005 年第３四半期活動報告」[1]でも，SSH サービスに対する総当たり攻撃の増加が報告され，注意が喚起されている．ここで，本稿ではこれ以降，パスワード認証に対する総当たり攻撃や辞書攻撃を総称して，総当たり攻撃と言うことにする．

これらの攻撃に対しては，従来からさまざまな対策が行われてきたが，従来の方法では固定的な対策が多

く，変化する状況に対して柔軟に対応できない．文献[2]では比較的うまく対策できるが，OS やソフトウエアの実装環境によっては利用できないなどの制約がある．また，攻撃者はポートスキャンなどにより，ネットワークに接続されたサーバを見つけ出してあらゆる方法で不正な接続を試みるため，多数の計算機を運用している組織では，それだけ多くの攻撃に晒されることになり，不正侵入の危険性も高くなる．

そこで本稿では，上記の問題の解決を図る対策手法を提案する．本方式では，syslog サーバを運用して組織内サーバのログを収集し，そこでアクセスログを監視することにより，組織内のどこで総当たり攻撃を受けても検知することができる．検知した攻撃者の IP アドレスは，すぐに各ホストに通知されて接続を拒否するため，まだ攻撃を受けていない場合でも事前に攻撃を防止することができる．本方式では，アクセスログから攻撃者の IP アドレスを取得し，ピンポイントで攻撃者からの接続だけを拒否するが，接続拒否の方法は iptables などの FireWall だけでなく，アクセス制御ファイルを利用するともでき，それぞれのプラットフォームに適した方法を選択することができる．また，これらの方法では，攻撃者からの接続をするのは SSH サービスだけでなく，他のサービスも同時に接続を拒否させることができる点も有利である．

本稿で対象としたのは一般的な RedHat Linux 系のプラットフォームであるが，アクセスログの記録と iptables などの FireWall，あるいはアクセス制御ファイルなどによるアクセス制御ができればよく，RedHat などの Linux に限らず幅広く利用できる．SSHD については，プログラムやプロトコルのバージョンには依存しない．

なお，本稿では，SSH の総当たり攻撃について述べているが，本手法では SSH に限らず，他のサービスについても対応が可能である．

以下，まず 2 章では，総当たり攻撃の特徴，従来の対策方法と問題点について述べる．次に 3 章では，提案するホスト間の連携を可能にする攻撃対策について述べ，4 章および 5 章で実装と運用事例について説明する．

2. 総当たり攻撃と従来の対策

総当たり攻撃とは，何らかの規則によって文字列の組み合わせを作り，パスワード認証による接続を何度も試行して計算機に不正に侵入する攻撃であり，辞書攻撃とは，パスワードとして使用されそうな文字列の集合を辞書として用意し，パスワード認証による接続を何度も試行して，同様な不正侵入を試みる攻撃である．現在では，辞書攻撃と総当たり攻撃を組み合わせた攻撃が多くなっており，辞書は英単語だけでなく日本語のものも確認されている[1][3].

2.1. SSH の総当たり攻撃の特徴

SSH の総当たり攻撃や辞書攻撃では，経験的に次のような特徴が挙げられる．

(1) 攻撃に用いられるユーザ名は，root や admin, test など通常のユーザが使用しないユーザ名を使用することが多い．
(2) 試行するのはパスワードだけでなく，ユーザ名も変更しながら接続を試行する．
(3) できるだけ多くの試行をするために，ツールプログラムなどを使用して連続して接続を試行する．
(4) ポートスキャンなどによって攻撃対象を見つけ，手当たり次第に接続を試行する．

(1)については，root が最も多く，以下，admin, test, mysql, info, oracle, adam, ftp, postgres, apache などと続くという調査結果もある[3].

2.2. 従来の対策と問題点

通常，SSH の接続サービスは計算機ごとにスタンドアロンで運用されているため，SSH の総当たり攻撃に対しても各計算機で対策を施す必要がある．総当たり攻撃では，2.1 節 (4) のようにポートスキャンなどによって攻撃対象を探し出し，手当たり次第に攻撃を行うものが多い．このため，多くの計算機を運用している場合にはそれらの計算機すべてに同様な攻撃を受けることになり，それだけ多くの危険に晒されることになる．そして，どこかにセキュリティーの弱い計算機が紛れていると，不正に侵入されてしまうことになる．

また，従来からの SSH の総当たり攻撃への対策では固定的な設定によるものが多く，様々な状況に対して不十分であったり，過剰であったりして必ずしも最適とは言えない．必要なことは，正規のユーザはいつでもどこからでも便利に利用できて，不正な攻撃だけを排除することである．下記は従来からよく行われている対策方法の例である．

1. SSH の接続サービスを停止する
2. 解読されにくいパスワードを設定する
3. 接続できる IP アドレスの範囲を制限する
4. SSHD の listen port 番号を変更する[4].
5. 公開鍵暗号認証方式を用いる[5]
6. 攻撃に対して試行回数を制限する[2]

3. ホスト間の連携を可能にする攻撃対策

様々な計算機のサービスを提供する機関では，多くのサーバを運用しており，また，不特定な地域から接続を受け付けなければならないサーバが多い．このような場合には，2.2 節で述べた理由により，各サーバで単独に攻撃対策をするよりもホスト間で連携して対策を施す方が有利である．

そこで本章では，syslog サーバを運用し，組織内のサーバのアクセスログを集めて監視することにより，組織内で発生した総当たり攻撃を検知し，ホスト間で連携して対策をする方式を提案する．本方式では，syslog サーバで検知した攻撃者の IP アドレスをすぐに各ホストに通知し，それぞれで接続を拒否するなどの対策が取れるため，まだ攻撃を受けていないホストも事前に攻撃を回避することができる．また，攻撃者の IP アドレスを取得しているため，SSH のサービスだけでなく，その他のサービスについても対策をすることができる．さらに，接続を拒否するのは攻撃者の IP アドレスだけなので，正規の利用者は利便性を損なうこともない．各サーバでの接続拒否の手段については，iptables などの FireWall の他にもアクセス制御ファイルを利用することができるため，より多くのプラットフォームで運用することができる．

3.1 節では，アクセスログ監視による動的アクセス制御について説明し，3.2 節では，ホスト間の連携について説明する．

3.1. アクセスログ監視による動的アクセス制御

アクセスログを監視することにより不正な攻撃を検知し，攻撃者の IP アドレスを取得して自動的に接続を拒否するなどの処置を行う．総当たり攻撃をしてくる悪意のある相手は，他にもあらゆる手段で攻撃をしてくると思われるため，接続を拒否するのは SSH に限らず，他のサービスも接続拒否することが望ましい．接続拒否をする方法としては，iptables などの FireWall の他にアクセス制御ファイルを利用することができる．

iptables は，Linux に実装された FireWall 機能であり，Linux カーネル 2.4 以降に組み込まれている．iptables では，コンピュータがやり取りするパケットを，あらかじめ定義しておいた「チェイン」単位に分類し，チェインごとにどのような処理を行うかを「ルール」として設定する[6].

アクセス制御ファイルは，TCP Wrappers（tcpd）が使用するアクアセス制御の設定ファイルであるが，現在では，tcpd だけでなく多くのネットワークサービスプログラムから利用することができる．hosts.allow には接続を許可するリストを，hosts.deny には禁止するリストを記載する．評価の順序は，最初に hosts.allow が参照され，ここに記載がないものは hosts.deny が参照される．どちらにも該当しないものはアクセスが許可される．

アクセスログ監視による動的アクセス制御は，syslog サーバと SSH などのネットワーク接続サービスをしているサーバで動作させておく．

以下，3.1.1-3.1.3 節で具体的な動作を説明する．

3.1.1. アクセスログ監視とパスワード認証エラー検出

本方式では，SSHD のアクセスログを常時監視し，パスワード認証エラーを検出して総当たり攻撃を検知する． RedHat 系の OS では/var/log/secure が SSHD のアクセスログファイルとなっているため，このログを常時監視してパスワード認証に失敗したものを抽出する．パスワード認証に失敗すると下記のログが書き出される．

- サーバに登録のあるユーザ名のパスワード違い
  Failed password for USER from IP port PORT
- サーバ登録のないユーザ名での接続
  Failed password for invalid user USER from IP port PORT

ここで取得した IP アドレスをキーにしてハッシュを作り，ログが書き出された時点のタイムスタンプと共に記憶しておく．また，接続に成功した IP アドレスは，比較的信頼できる相手として一定期間データベースに記録しておく．

3.1.2. 攻撃の判定

同じ IP アドレスから一定の時間内に何回のパスワード認証エラーがあったかによって攻撃を判定する．また，過去の一定期間内で接続に成功している IP アドレスは判定基準をやや緩くするため，3.1.1 節のデータベースを参照する．また，学内の IP アドレスも比較的信頼できる相手とする．本稿においては，パスワード認証エラーとなったものを次の条件で攻撃者と判定する．

《条件 1：繰り返しの接続試行 1》
2 分以内に 11 回
《条件 2：繰り返しの接続試行 2》
10 分以内に 20 回
《条件 3：比較的信頼できる相手》
10 分以内に 30 回

なお，各ホストに固有の判定条件が必要な場合は，各ホストで例外ルールを設定しておけばよい．

3.1.3. 接続拒否

攻撃と判定された IP アドレスは，iptables やアクセス制御ファイルに書き出して接続を拒否する．

iptables を利用する場合には，-s で攻撃者の IP アドレスを指定し，-j DROP で接続を拒否する．iptables の評価を最初に行うため，チェインの入力は insert コマンド（-I）で行う．insert コマンドではルール番号を省略すると，そのルールはチェインの先頭に挿入される．

アクセス制御ファイルでは，/etc/hosts.allow に EXCEPT の項目で記載する．

iptables を利用する場合とアクセス制御ファイルを利用する場合の特徴を次に示すが，各ホストの環境に適した方法を選択する．

- iptables では全てのパケットを監視し，非常に様々な動作を行うことができるが，アクセス制御ファイルでは，コネクションの受付だけが対象となる．このため，iptables では高トラフィック時には計算機の負荷もそれなりに発生するが，アクセス制御ファイルでは，コネクション確立後には計算機の負荷は発生しない．
- iptables では OS によっては利用できないものがあるが，その場合にもアクセス制御ファイルは利用できることが多い．

なお，iptables でもアクセス制御ファイルでも長期間の運用を続けると，拒否する IP アドレス数が無制限に増えるため，接続拒否をしておく IP アドレス数を決めておき，それを超えた場合は古いものから順に削除する．

3.2. ホスト間の連携

各ホストのログ収集のために syslog サーバを運用し，各ホストのアクセスログをネットワークからリアルタイムに収集しておく．この syslog サーバで 3.1.1-3.1.2 節によってアクセスログを監視しておけば，どのホストで総当たり攻撃があっても検知することができる．検知した攻撃者の IP アドレスは，すぐにネットワークを介して各ホストに通知し，各ホストで接続を拒否する．

また，攻撃者の IP アドレスや，ログから得られる他の情報を基に IPS(Intrusion Prevention System)やネットワークスイッチと連携した処置も可能となる．

3.2.1 節では，提案するシステムの構成と動作について説明する．

3.2.1. システムの構成と動作手順

（1）syslog サーバの運用と動作手順

各ホストのログを集めるために syslog サーバを運用する．syslog サーバでは，syslogd を-r オプションで起動する．syslogd は，多くのプログラムのロギング手段を提供する Daemon であるが，-r オプションにより，ネットワーク上でインターネットドメインソケットから syslog サービスを使用してメッセージを受信する機能が有効になる[7]．各ホストの syslogd が syslog サーバにアクセスログを送るように設定されていれば，syslog サーバには各ホストのアクセスログが収集されるため，このログを 3.1.1-3.1.2 節によって判定し，攻撃検知を行う．攻撃と判定された場合には，syslog サーバ自身の接続拒否動作を行うと同時に，各ホストに攻撃者の IP アドレスを通知する．

なお，syslog サーバでは収集したログの信頼性を確保する必要があるため，専用のサーバとして運用し，リモート接続の許可範囲や各種サービスプログラムなども必要最小限に制限しておくべきである．さらには，ログの記録も CD/DVD-RW メディアなど，記録された情報を簡単に書き換えられない方法を選択することが望ましい．

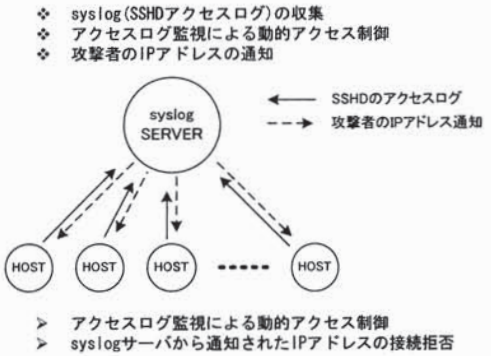

図-1 システムの構成と動作手順

（2）一般ホストの構成と動作手順

各ホストでは，syslog サーバにログを送るために，syslog.conf に設定を行い，syslogd を動作させる．具体的には，syslog.conf の selector フィールドに対象となるログの facility と priority を，action フィールドには先頭に@を付けて，syslog サーバを指定しておく．SSHD のアクセスログは，RedHat Linux 系の OS では /var/log/secure であり，ログの facility と priority はデフォルトで authpriv.info である．従って一般のホストは syslog サーバに authpriv.info などのログを送るように syslog.conf を設定しておく．syslog サーバから攻撃者の IP アドレスが通知された場合には，すぐに 3.1.3 節のアクセス拒否動作をする．

なお，一般のホストでも 3.1 節のアクセスログ監視

による動的アクセス制御を行う．その理由は，syslog をネットワーク経由で syslog サーバに送る場合には udp が使用されるため，syslog サーバや途中のネットワークの状態によってはパケットが syslog サーバに届かないことがあるためである．従って，実際に攻撃を受けているホストでは，そのホストで攻撃を検知して通信を遮断する必要がある．

4. 試作システムの実装

前章で述べた提案方式に基づき，試作システムの実装を行った．試作システムの構成を図-2 に示す．
以下では試作システムの実装について述べる．

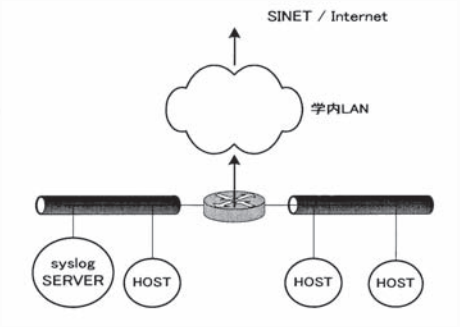

図-2 試作システムの構成

4.1. syslog サーバ，一般ホストの構成

syslog サーバ，一般ホストとも RedHat Linux 系の PC-UNIX で構成した．syslog サーバは CentOS 4.4，一般ホストは CentOS 5, CentOS 4.3, FeroraCore 4 である．試作システムの環境を表-1 に示す．

syslogサーバ			
CentOS 4.4	Openssh-3.9p1	iptables-1.2.11	perl-5.8.5
一般ホスト			
CentOS 5	Openssh-4.3p2	iptables-1.3.5	perl-5.8.8
CentOS 4.3	Openssh-3.9p1	iptables-1.2.11	perl-5.8.5
FedoraCore4	Openssh-4.2p1	iptables-1.3.0	perl-5.8.6

表-1 試作システムの環境

syslog サーバが収集するログの種類はプライオリティが info 以上のものである．なお，最近の syslogd では，ログファイルの肥大化を防ぐために，連続する同じログについては件数だけを書き出すものがあるため，これもカウントする．

4.2. アクセスログ監視と動的アクセス制御

アクセスログの監視と動的アクセス制御のプログラムは perl で作成し，perl モジュールは File::Tail [8], Proc::Daemon [9]を使用した．まず，File::Tail により SSHD のアクセスログである/var/log/secure を常時監視し，パスワード認証エラーを検出する．3.1.1 節−3.1.2 節の手順により，攻撃と判定されたものは iptables もしくはアクセス制御ファイルにより接続を拒否する．

iptables を利用する場合には iptables の I コマンドにより，攻撃者の IP アドレスを DROP としてチェインに追加する．

```
iptables -I INPUT -s IP_address -j DROP
```

アクセス制御ファイルで接続拒否する場合は，/etc/hosts.allow に EXCEPT の項目で記載する．図-3 に /etc/hosts.allow ファイルへの記載例を示す．なお，/etc/hosts.deny には ALL:ALL が記載してある．

```
#
# hosts.allow   This file describes the names of the hosts which are
#               allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
sshd.popper : ALL EXCEPT \
              /255.255.255.0,\
              0/255.255.255.0,\
              10,\
              76,\
              16,\
              1,\
              6,\
              0,\
              26
```

図-3 /etc/hosts.allow への記載例

また，接続を拒否する IP アドレスが規定の数を超えると，FIFO により古いものから削除する．iptables では delete コマンド（-D）を使用する．

```
iptables -D INPUT -s IP_address -j DROP
```

アクセス制御ファイルでは，hosts.allow に EXCEPT として記載されている IP アドレスを FIFO で削除する．

なお，過去 2 ヶ月以内で接続に成功している IP アドレスと学内の IP アドレスは，比較的信頼できる相手として判定基準をやや緩くしているが，IP アドレスの動的割当や踏み台による攻撃が予想されるため，あまり油断はできない．

アクセスログの監視と動的アクセス制御のプログラムは，syslog サーバと一般ホストで動作させる．

4.3. サーバ間の連携

syslog サーバから一般ホストへの伝達は，perl の IO::Socket [10]モジュールを使用した．syslog サーバでのアクセスログ監視により総当たり攻撃を検出すると，syslog サーバから一般ホストに対して IO::Socket を利用して攻撃者の IP アドレスを通知する．通知を受けたホストでは，iptables やアクセス制御ファイルなど，それぞれのプラットフォームに最適な方法によって接

続を自動的に拒否する.

5. 運用事例

試作したシステムは，アクセスログ監視と動的アクセス制御はすでに運用をしているが[11]，その他の機能については運用を開始したところである．今後，動作の検証ができれば，岡山大学総合情報基盤センター（以下，センター）の計算機システムで実稼働する予定である．センターではすでにsyslogサーバを運用しており，センターが運用するサーバについて，ログの収集を行っているが，実稼働においては，このsyslogサーバを利用する予定である．

なお，本方式では，総当たり攻撃に対して最初から接続を拒否するものではなく，繰り返される接続試行の特徴によって判定をしているため，何度かはパスワード入力を試行されてしまう．つまり，サーバの登録ユーザに脆弱なパスワードを設定している者があると，判定中にもパスワードを破られて侵入される危険がある．このため，岡山大学総合情報基盤センターでは，全ての登録ユーザについて，パスワードの脆弱性を定期的にチェックし，脆弱なパスワードは強固なものに変更してもらうか，あるいは強制的に変更している．脆弱性のチェックにはJohn the Ripper [12]を使用している．

6. まとめ

本稿では，syslogサーバを運用し，ホストで間連携してパスワード総当たり攻撃の対策を行う方式を提案した．アクセス制御の方法も，iptablesなどのFireWallだけでなくアクセス制御ファイルが利用できるため，より多くのプラットフォームで運用することができる．

ネットワークを経由した計算機の利用は，今後もますますその重要性を増してゆく．本稿で述べたSSH接続サービスを始め，POPやIMAPの他にも様々なサービスが開発され普及するものと思われる．このようなネットワーク社会においては，便利で安全にネットワーク上の計算機を利用することは非常に重要な課題であり，本稿による対策手法は，有効な手段の1つと言える．

今後の課題として，総当たり攻撃についてはSSHだけでなく，POPやIMAPなどでも検知する手法を検討する．また，他のプラットフォームへの移植についても検討する．

文献

[1] 有限責任中間法人 JPCERT コーディネーションセンター:インターネットセキュリティーに対するJPCERT/CC 2005年第3四半期活動報告，pp.1-3，2005年11月7日

[2] 佐藤裕介：iptablesのipt_recentでsshのbrute force attack対策，http://www2s.biglobe.ne.jp/~nuts/labo/inti/ipt_recent.html

[3] 警視庁：分析レポート SSHサービスに対する攻撃について，pp.1-10，平成18年8月17日

[4] Security Note: SSH のポートを開けてブルートフォース攻撃を防ぐ，http://security-note.net/2007/01/ssh.html

[5] 新山祐介，春山征吾：OpenSSH セキュリティー管理ガイド，株式会社秀和システム，pp.52-73，2001

[6] ITpro：iptablesとはITpro，http://itpro.nikkeibp.co.jp/article/Keyword/20070207/261226/

[7] www.linux.or.jp 管理グループ (Webmasters)：Man page of SYSLOGD，http://www.linux.or.jp/JM/html/sysklogd/man8/syslogd.8.html

[8] CPAN：File::Tail，http://search.cpan.org/~mgrabnar/File-Tail-0.99.3/Tail.pm

[9] CPAN：Proc::Daemon，http://search.cpan.org/~ehood/Proc-Daemon-0.03/Daemon.pm

[10] CPAN：IO::Socket，http://search.cpan.org/~gbarr/IO-1.2301/IO/Socket.pm

[11] 大隅淑弘，山井成良，井上一郎二：アクセス制御ファイルの動的変更によるSSH総当たり攻撃への対策，第11回学術情報処理研究集会論文

[12] Openwall Project：John the Ripper password cracker，http://www.openwall.com/john/