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

Countermeasure for SSH brute force attack with dynamic modification of the access control files

大隅　淑弘†，山井　成良†，井上　一郎二‡
Yoshihiro OOSUMI †, Nariyoshi YAMAI †, Ichiroji INOUE ‡

oosumi@cc.okayama-u.ac.jp, yamai@cc.okayama-u.ac.jp, iti-ino@cc.okayama-u.ac.jp

†岡山大学総合情報基盤センター

† Okayama University Information Technology Center

概要

近年，不正アクセスの侵入手口として，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多くなっている．計算機によっては，不特定の地域から SSH の接続を受け入れる必要のあるものがあり，また，登録ユーザの中には脆弱なパスワードを使用している者がある．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．本稿では，ログを監視することにより，SSH のパスワード認証に対する総当たり攻撃や辞書攻撃を検知し，不正なアクセスを動的に拒否するとともに，他の機器にも伝達して不正な攻撃を拒否する方式を提案する．

キーワード

SSH, 総当たり攻撃, 辞書攻撃

1. はじめに

現在では，サーバへのリモート接続サービスは，セキュアな SSH（Secure Shell）を使用するのが一般的である．従来標準的に使用されていた telnet では，通信内容が平文でネットワーク上を流れるため，悪意のある第三者に通信内容を盗聴されたり，また，なりすましによって，パスワードや個人情報を盗まれる危険性が非常に高いためである．SSH による接続サービスでは，鍵交換により暗号化通信路を確立してからサーバ，クライアント双方を認証し，データの送信を行うので，なりすましや盗聴などの危険を回避することができる．また，SSH の利用は，サーバへのログインだけでなく，ポートフォワーディングを利用して，POP や SMTP などを利用することが多い．最近では，ほとんどの ISP（Internet Services Provider）で OP25B（Outbound Port 25 Blocking）を行っていることもあり，重要なサービスになっている．このような理由から，telnet の接続サービスは禁止しても，SSH での接続を許可している場合が多い．ところが近年，SSH のパスワード認証に対する総当たり攻撃や辞書攻撃（以下，SSH の総当たり攻撃や辞書攻撃とする）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」では，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[1]．また，2006 年 8 月に発表された警視庁の分析レポートでは，2006 年 5 月からの 1 月間で SSH サービスに対する接続試行が，24 の国／地域，合計 105 の IP アドレスから行われ，試行された認証の回数は，1 サーバあたり 41,456 回，1 日あたり 1,382 回であったことが報告されている[2].

岡山大学総合情報基盤センターでは，教育・研究用計算機システムとして，メールサーバ，計算サーバ，アプリケーションサーバなど，多数の計算機が運用されており，学内のユーザにサービスを提供している．これらの計算機では，サービスの性質上，学内外の全ての地域から接続を受け付けるものがあり，SSH の接続サービスもその１つである．ところが，岡山大学においても SSH の総当たり攻撃や辞書攻撃が非常に多くなってきた.

従来から, SSH の総当たり攻撃や辞書攻撃に対しては, 多くの対策が行われてきたが，従来の方法では固定的な設定のため，変化する状況に対して動的に適用することができない．また，攻撃をしてくる相手はあらゆる手段で不正な侵入を試みるため，SSH だけではなく，POP や SMTP，IMAP，その他，ネットワーク上に提供しているサービスに対しても防衛対策をしておく必要がある．文献[3]では不正な接続を動的に抑制できるが，抑制する範囲は，その計算機の SSH だけに限られる.

本稿では，アクセスログを監視することにより，SSH の総当たり攻撃や辞書攻撃を検知し，不正な接続をピンポイントで自動拒否する方式を提案する．自動拒否は，SSH だけでなく他のサービスにも適用することができ，さらに，他のサーバやネットワーク機器に伝達して不正な攻撃から防衛するという特徴を持つ．なお，本稿で対象とした SSH は，一般的な OpenSSH であるが，アクセスログの記録とアクセス制御ファイルの参照もしくは iptables などによるアクセス制御ができる環境であれば, SSHD のプログラムやプロトコルには依存しない.

以下, まず 2 章では, 総当たり攻撃や辞書攻撃の特徴, 従来の対策方法と問題点について述べる. 次に 3 章では, 提案するアクセス制御ファイルの動的変更による対策方式について述べ，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，辞書攻撃と従来の対策

2.1. 総当たり攻撃，辞書攻撃の特徴

SSH の総当たり攻撃や辞書攻撃では，攻撃の特徴として次のようなことが挙げられる.

(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.

(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[2].

2.2. 従来の対策と問題点

SSH の総当たり攻撃や辞書攻撃には，従来から次のような対策方法が取られていたが, それぞれ問題点もあり, 必ずしも適切な対策とは言えない.

- SSH の接続サービスを停止する
  SSH の接続サービス自体を停止する方法である．これは最も確実な方法であるが，必要なサービスであれば停止することができない.
- 解読されにくいパスワードを設定する
  ランダムな文字列や記号を使用し，かつ長い文字列を設定することにより，解読されにくいパスワードを設定する．しかし，このようなパスワードは覚えにくく，ユーザ数が数千件，数万件ある状況では，全てのユーザが適切なパスワードを設定することは難しい.
- 接続の試行回数を制限する
  一度の接続でパスワード入力できる回数や同時接続可能数を減らしたり，一定数以上の接続を間引いたりする．あるいはログインに失敗した場合の再試行禁止時間を多く取る[4]．しかし，時間をかければそれなりの回数が試行できる.
- 接続できる IP アドレスの範囲を制限する
  アクセス制御ファイルや iptables で IP アドレスやドメイン名について接続できる範囲を最小限に制限する. しかし, 登録ユーザの出張や留学などが多いと, 接続を受け付ける範囲を常に最適に設定しておくことが難しい．また許可した範囲からでも攻撃を受けることがある.
- SSHD の listen port 番号を 22/tcp から他に変更する[5]

sshd_config を変更することにより, listen port を通常の 22/tcp から変更する．しかし，全ユーザにそれを通知する必要があり，また，一時的な対策にはなるが，変更後の port 番号を知られると再度 port 番号を変更する必要がある．

- 公開鍵暗号認証方式を用いる[6]
  パスワード認証を禁止して公開鍵暗号認証方式を用いる．この方法では，RSA 暗号化方式や DSA 暗号化方式で作成した公開鍵と秘密鍵のペアを使用するため，非常に強固であり利用者とホストの両方を認証することができる. しかしながら, 事前にサーバ，クライアントで設定を済ませておく必要があり，また，出張などで外出する場合には，設定を済ませたパソコンや秘密鍵を持ち歩く必要がある．そして，これを忘れると接続できなくなる．

3. アクセス制御ファイルの動的変更による対策

2 章で述べたように，SSH の総当たり攻撃や辞書攻撃における従来の対策では，事前に機能を制限しておくものが多く，変化する状況に常に最適に対応するには不十分である．また，対策が適用される範囲もその計算機の SSH 接続サービスに限られるため，他のサービスでも必要により個別に対策をしておく必要がある．さらに，多数の計算機を運用している状況下では，すべての計算機で最適な設定を維持するには，管理コストが高くなる．

そこで，本稿では SSHD の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，アクセス制御ファイルを動的に更新して自動拒否する方式を提案する．この方式では，接続を拒否するのは SSH に限らず，POP や IMAP など他のサービスにも同時に適用することができる．さらに，他のサーバや学内外との接続を監視する IPS (Intrusion Prevention System) ，ネットワークスイッチにも伝達して，不正な攻撃を事前に拒否することもできる．

なお, POP や IMAP を利用する環境においては, APOP あるいは POP3/IMAP over SSL/TSL などを用いることにより，パスワードや本文の盗聴を防止することは可能であるが，これだけでは総当たり攻撃や辞書攻撃を防止することができない点に注意する必要がある．

3.1. 本稿が対象とした計算機

本稿が対象とした計算機は，NEC TX7/i6010（以下 TX7）である．岡山大学では，総合情報基盤センター内に様々なサーバを運用しているが，全学向けのメール，スカラ計算，アプリケーションのサービスを提供するため, このTX7を運用している. TX7は8台のItanium2 CPUを搭載しており，OS は RedHat Linux AS2.1 をベースにした NEC IA-64 Linux R3.4 である．OS の理由により，iptables などの FireWall 機能は利用できない．登録ユーザ数は約 17500 件である．この計算機ではそのサービスの性質上，SSH，POP，SMTP（POP before SMTP）については，学内だけでなく，学外のあらゆる地域からの接続を受け入れる必要がある. SSH は OpenSSH-4.6p1 であり，2.2 節の公開鍵暗号認証方式の項で述べているような理由から，パスワード認証を許可しており，1 度の SSH 接続で 2 回までのパスワード入力が試行できる．

なお，本方式では，Linux/UNIX/BSD 系 OS の基本機能を利用しているため，この TX7 以外の他の機種についても SSHD のアクセスログ記録とアクセス制御ファイルの参照あるいは iptables などが利用できる環境であれば，OS や SSHD のプログラム，プロトコルによらず適用できる．

3.2. アクセス制御ファイル

本方式では，接続の制御にはアクセス制御ファイルである，/etc/hosts.allow，/etc/hosts.deny を利用する．アクセス制御ファイルでは，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．最近は tcpwrappers（tcpd）だけでなく，いろいろなサーバプログラムが，コンパイル時にオプション指定しておけばアクセス制御ファイルを参照するようになる．

3.3. 動作手順

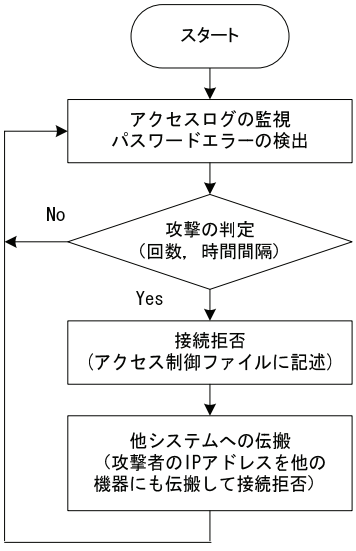

図-1 動作手順

動作手順を図-1 に示す．まず，デーモンプログラムを

起動して SSHD のアクセスログを常時監視する．ここでパスワード認証のエラーが検出されると，同じ IP アドレスから現在までのエラーの発生状況から攻撃の判定をする．攻撃と判定されたものは，アクセス制御ファイルに書き出して接続拒否をするとともに，他の機器にも攻撃者として伝達する．また，この接続拒否については，3 章で述べた理由により，SSH だけでなく同時に POP や IMAP についても接続を拒否する．なお，sendmail（SMTP）についてもコンパイル時に TCPWRAPPERS オプションを付けておけば，アクセス制御ファイルの参照が有効になり，spam メールなども拒否することができる．

3.4. アクセスログの監視とパスワード認証エラーの検出

SSH 接続のパスワード認証エラーを検知するために，SSHD のアクセスログファイルに逐次書き出される内容を監視する．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．このログファイルから SSH 接続で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．

・ 登録のあるユーザ名でのパスワード間違い

Failed password for USER from IP port PORT

・ 登録のないユーザ名での接続

Failed password for invalid user USER from IP port PORT

この 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今までの失敗回数，接続してきたユーザ名を記憶しておく．

3.5. 攻撃の判定

同じ IP アドレスから以下のようなパスワード認証エラーがあったときに攻撃と判定する．

（1）繰り返し接続を試行している場合

これはパスワード認証エラーの発生頻度によって判定する．パスワード認証エラーが検出されると，この IP アドレスをキーにして 3.4 節のハッシュ値を参照し，値のあるものについて前回との時間差，エラー発生回数，ユーザ名をチェックし，判定基準と比較する．

ここで，パスワード認証エラーの原因には，次のようなことが考えられる．

(a) 攻撃者が登録のあるユーザ名を使って総当たり攻撃あるいは辞書攻撃をした
(b) 攻撃者が登録のないユーザ名を使って総当たり攻撃あるいは辞書攻撃をした
(c) 正規のユーザがキータイプを間違えた
(d) 正規のユーザがパスワードを忘れて心当たりのものを何度か入力した

まず，(a)，(b)については，攻撃そのものなので短時間にできるだけ多くのパスワード入力の試行を繰り返す．(c)の入力間違いはせいぜい 6 回か 7 回程度であろう．(d)については，完全に忘れているのであれば，かなりの回数でパスワード入力を試行すると思われ，この場合は(a)と区別が付かない．ただし，思い出しながら入力するとなれば，短時間に連続して試行するとは考えにくい．過去のアクセスログについて，同じ IP アドレスからパスワード認証エラーの時間間隔を調査してみると，10 秒以内のものがほぼ 80%であった．以上より，本稿では，同じ IP アドレスから 10 秒以内の時間差で，連続して 11 回のパスワード認証エラーをしたものを攻撃者と判定する．

《条件 1》

さらに，この判定条件を意図的あるいは偶然に回避してしまった場合を考える．つまり，10 秒以上の時間差あるいは 10 回以内のエラーでは，攻撃者として検出しないことになる．このため，さらに 1 時間程度の間に 15 回《条件 2》あるいは 1 日の間に 30 回《条件 3》のパスワード認証エラーをしたものも攻撃者と判定する．

なお，以上の判定基準に合致しなかったものは，前回接続時間，接続回数，接続試行ユーザ名に今回のものを加えてからハッシュ値を更新しておく．

（2）システム管理ユーザ名や攻撃で使用されるユーザ名で接続している場合

前項 2.1 のとおり，総当たり攻撃や辞書攻撃では，接続するユーザ名として，システム用あるいはテスト用のユーザ名がよく使用される．また，この他にも経験的に明らかに攻撃と思われるユーザ名が使用されることがある．このため，これらのユーザ名で接続を試行した場合は，3 回の SSH 接続，つまり，5 回目のパスワード認証エラーで攻撃と判定する．正規のユーザが間違って root などの管理アカウントで接続しても，2 度の SSH 接続については，攻撃の判定をしない．《条件 4》

（3）ユーザ名を頻繁に変更している場合

この場合も攻撃と思われる．正規のユーザでも何度かはタイプミスをすることもあり得るが，そうそう何度も変更するとは考えられない．本稿では，5 回のユーザ名変更までは許容し，6 回目の変更を攻撃と判定する．《条件 5》

以上をまとめると，同じ IP アドレスから，次のパスワード認証エラーがあったときに攻撃と判定する．

《条件 1：繰り返しの接続試行》
10 秒以内の時間差で 11 回
《条件 2：繰り返しの接続試行》
5 分以内の時間差で 15 回（1 時間程度で 15 回）
《条件 3：繰り返しの接続試行》
1 時間以内の時間差で 30 回（1 日で 30 回）
《条件 4：システム管理ユーザ名等で接続》
1 分以内の時間差で 5 回

《条件 5：ユーザ名を頻繁に変更》
1 分以内の時間差でユーザ名を 6 回変更

3.6. 接続拒否

攻撃と判定された IP アドレスは, アクセス制御ファイルに書き出して接続を拒否する．このファイルへの書き出しは，/etc/hosts.allow に EXCEPT の項目で記載する．hosts.allow に記載する理由は, 前項 3.2 のように評価が最初に行われるためである．仮に hosts.deny に記載したとすると，他のサーバ管理者が hosts.allow に ALL:ALL などと書いてしまうと hosts.deny の拒否設定が無効になる．

アクセス制御ファイルに記載できるアドレス数は無限ではない．また，攻撃を仕掛けてくるホストはいつまでも同じとは考えにくいため，拒否するアドレス数を最大 80 件とした． 80 件になれば FIFO で自動的に 10 件を削除する．

なお，本方式では，アクセス制御ファイルを利用したが，iptables などの FireWall を利用することもできる．

3.7. 他のサーバやネットワーク機器への伝達

攻撃者の IP アドレスは, TX7 だけでなく他のサーバにも伝達し，同様に接続を拒否するようにした．現在，総合情報基盤センターでは，TX7 以外にも 6 カ所の遠隔キャンパスでメールサーバを運用している．これら 6 台のメールサーバも RedHat 系の Linux であり，sendmail と POP を運用している．SSH はシステム管理者だけが学内から接続でき，sendmail も基本的に学内からの接続のみ受け付ける[7]が，POP については，学内外のあらゆる地域からの接続を受け入れる必要がある．このため，POP における総当たり攻撃や辞書攻撃によってユーザ名，パスワードの盗難の危険がある．これら 6 台のメールサーバへも攻撃者の IP アドレスを伝達して接続を拒否する．

この他にも，IPS や学内外を接続するネットワークスイッチと連携して接続を拒否することもできるが，現有機器固有の理由により，現在は試作したシステムの機能に含めていない．

4. 実装

本方式は TX7 に試作システムとして実装し, 運用している．実装したサーバの接続構成を図-2 に示す．プログラムは perl で作成した．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．3.4 節－3.5 節の手順により攻撃と判定されたものは，アクセス制御ファイルの /etc/hosts.allow に EXCEPT の項目で記載して接続を拒否するとともに，このハッシュは削除する．判定の条件を満たさなければハッシュ値を更新しておく．/etc/hosts.allow ファイルへの記載例を図-3 に示す．なお, /etc/hosts.deny には ALL:ALL が記載してある．

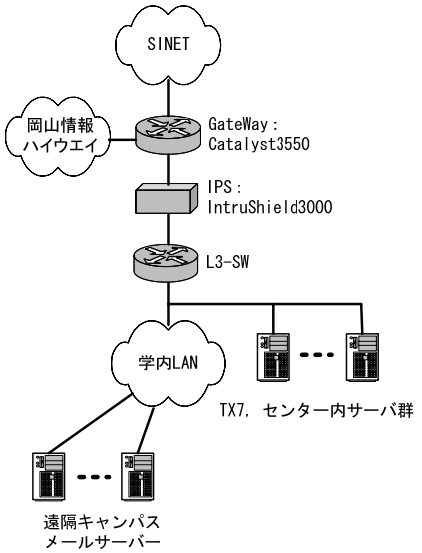

図-2　実装したサーバの接続構成

```
#
# 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 \
              [redacted]/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

次のユーザ名は，TX7 のシステム用ユーザ名のため, これで接続要求をした IP アドレスは攻撃者として，3.5 節（2）による判定をする．

root, bin, daemon, adm, lp, sync, shutdown, halt, mail, news, uucp, operator, games, gopher, ftp, nobody, mailnull, rpm, xfs, wnn, ntp, rpc, gdm, rpcuser, nfsnobody, nscd, ident, radvd, postgres, apache, squid, named, pcap, pvm, piranha, netdump, amanda, junkbust, mailman, fax, mysql, ldap, sshd

次のユーザ名は，現在までの経験により，攻撃者と考えられるため，同様に 3.5 節（2）による判定をする．

test, guest, admin, user, webmaster, staff, alias, recruit, sales, delta

また，今後も攻撃者と思われるユーザ名があれば，随時攻撃者の条件に組み入れる．

なお, 接続元が学内と学外でも判定基準を変えており，学内からの接続には基準を若干緩くしている．学内といえども，踏み台にされた計算機からの攻撃が予想される

ため，完全には許容できない．

他のサーバへの伝達は，perl の io::socket [10]モジュールを使用した．すなわち，TX7 で検出した攻撃者の IP アドレスを 6 台のメールサーバに通知し，それぞれ同様に hosts.allow ファイルに記載して攻撃を防止する．こちらも接続拒否アドレスが 80 件になれば FIFO で自動的に 10 件を削除する．

5. 運用事例

試作システムは，2007 年 1 月から運用をしている．最近の状況では，2007 年 6 月 20 日から 7 月 20 日の間で接続拒否された IP アドレスは 108 件であった．平均的に 1 日に 3 件～4 件程度のため，アクセス制御ファイルに 80 件を記載すると，その IP アドレスでは 20 日間程度は接続を拒否されることになる．

また，攻撃ではないにもかかわらず，攻撃と判定されて接続拒否されたものは運用開始以来で 5 件あった．これらは，パスワードを忘れて何回も入力を試行したものや，ユーザ名に anonymous をセットして sftp クライアントソフトで接続をしたものであった．

本方式では，SSH の総当たり攻撃や辞書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 [11]を使用している．

6. まとめ

本稿では，SSH のパスワード認証に対する総当たり攻撃や辞書攻撃を検知し，アクセス制御ファイルを動的に更新することによって，ピンポイントで接続を拒否するとともに，他のサーバやネットワーク機器にも情報を伝達して，不正な攻撃を防止する方式を提案した．

SSH による接続サービスは，通常のリモート接続だけでなく，1 章で述べたような SSH のポートフォワーディングによるメール利用や VPN 通信なども重要なサービスになっている．このように SSH の接続サービスは利用範囲が広く，今後も多くの利用が見込まれており，利便性を損なわず安全にサービスを提供するには，本稿による方式は有効であると言える．

また，本稿では接続拒否にアクセス制御ファイルを使用したが，同様な機能は iptables でも実現できることを示した．

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] 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

[3] 鈴木聡，湯浅富久子 : ブラックリストを用いた PAM 遅延モジュールによる SSH への攻撃抑制，情報処理学会研究報告，No.2006-DSM-040，pp.1-5，2006

[4] 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

[5] 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

[6] 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

[7] 山井成良，宮下卓也，大隅淑弘，林信彦 : 岡山大学における電子メールシステムのセキュリティ対策，情報処理学会研究報告，Vol.2002 No.82，pp.61-66，2002

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