コンピュータセキュリティ 3－2
（1998. 11. 27）

ユーザ認証時のロックアウトの拡張

堤 俊之　　三戸 健史
日立ソフトウェアエンジニアリング(株)

本論文では、ファイル制御やプロセスアクセス制御などのセキュリティ機能を持たない OS 環境に、パスワードによるユーザ認証機能を持ったセキュリティシステムを導入した時に、必要とされるユーザ認証のロックアウト方式を提案する。本提案方式では、セキュリティシステムの端末にアクセス可能な不正ユーザが様々なパスワードの組合せを繰り返し試行する攻撃を防止することを目的としている。ダミーのユーザ認証を用いることで安易にロックアウトを外そうとする不正ユーザの攻撃を防止して、セキュリティレベルを高めることができた。

Extension of Lockout in User Authentication

Toshiyuki Tsutsumi　　Takeshi Mito
Hitachi Software Engineering Co., Ltd.

This paper presents a new lockout method for password authentication of the security application in the operating system which does not have access control function for file system nor process control system. The purpose of the new method is to prevent an illegal user from attempting to guess a password consecutively. The dummy authentication of the new method makes the security application safer against the attack that the illegal users try to make the lockout ineffective.

1.はじめに

本論文ではファイルアクセス制御やプロセスアクセス制御などのセキュリティ機能を持たない OS 環境に、パスワード認証機能を持ったセキュリティシステムを導入した時のロックアウトについて考察する。具体的には、セキュリティシステムへアクセスできる不正ユーザが、様々なパスワードの組合せを試す攻撃を防御するための仕組みについて述べる。

不正アクセスや通信内容の改ざん・盗聴からアプリケーションを守るために、今日多くのセキュリティシステムが開発されている。例えば、IPv4[1]にセキュリティ機能を付加した IPSEC[2]や電子メールシステムのセキュリティ機能を強化した S/MIME[3]、PGP[4]などである。このようなシステムは正規ユーザのみが使用できるようにするため、はじめにユーザの認証を行っている。

ユーザ認証を行うために必要なデータは、大きく次の３つに分類できる。

(1) ユーザ自身の身体的データ
(2) ユーザだけが保持しているデータ
(3) ユーザだけが知っているデータ

具体例としては(1)は指紋や虹彩などの生体情報、(2)は携帯メディアに納められた秘密鍵データ、(3)はパスワードなどである。このうち、もっとも一般的に利用されているのは(3)のデータである。これは、(1)では指紋や虹彩を読み取る特別なデバイスが、(2)では IC カードや IC カード読み取り機が必要であり、コストがかかるからである。しかし、最近ではセキュリティへの関心が高まり、IC カードや IC カード読み取り機の開発競争により価格も抑えられてきたので、(2)のデータを利用した認証も増えつつある。ところが、(2)を使った認証を単独で利用すると、IC カードを盗まれた場合、簡単になりすまされてしまう危険性がある。そこで、(2)を利用する場合でも、(3)と併用するのが

一般的であり、依然としてパスワードによる認証が必要とされている。

このパスワード認証に対して行われる攻撃を挙げると、「総当たり（Brute-force）攻撃」と「辞書（Dictionary）攻撃」がある。どちらの攻撃も、機械的に様々な文字の組合せをパスワードとして試すものである。これら攻撃は大量のパスワードを試すことにより正規のパスワードを発見するので、短期間のうちに多くのパスワードを試さなければならない。

このような攻撃を防ぐために、セキュリティシステムはロックアウト[5]を採用している。ロックアウトとはユーザ認証の連続試行回数を規定して、それ以上認証に失敗すると認証機能を一定期間利用不能にするものである。連続した認証機能の利用を制限すると、不正ユーザによる短期間の繰り返しユーザ認証を減らすことができる。こうして攻撃される機会を減らして、システムを安全にしている。

パスワード認証機能を持つシステムは、ロックアウトを実現するために必要なロックアウト情報を保持しなければならない。しかし、システムの稼動するOS環境がユーザ認証機能やファイルへのアクセス制御機能などのセキュリティ機構を持っていなければ、不正ユーザはロックアウト情報を削除･改ざんして、ロックアウトを解除できてしまう。そこで、本論文では、このようなOS環境でロックアウトを解除しようとする不正ユーザからシステムを守る一方式を提案する。

本論文では、2章で既存方式の説明し、その問題点をあげて目的を明確にする。3章では提案方式の概要を述べて、4章では処理の詳細を説明する。5章では提案方式に関するいくつかの考察を行う。

2.目的

2.1 従来方式

はじめに従来までのロックアウトについて説明する。

2.1.1 基本構成

図1に本論文で想定している システムの構成を示す。ユーザは識別情報の格納されている携帯メディアを持っている。識別情報とは、システムがユーザを認証するために必要となる情報である。携帯メディアは格納しているデータへのアクセス制御機能を含まないものとする。

システムは以下のデータで構成された「ロックアウト情報」をユーザ毎に保持している。

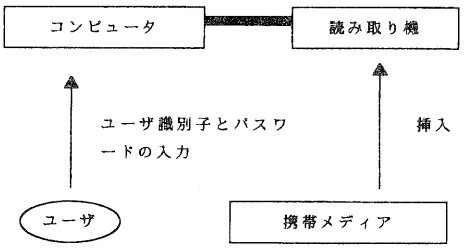

図1 従来ロックアウト方式の構成

(1) ユーザ認証失敗回数（FailNum）
ユーザ認証に失敗した回数である。MAXNUMをユーザ認証試行最大回数とすると、FailNumの値は0からMAXNUMの間である。初期値は0である。

(2) 最終ユーザ認証失敗時刻（FailTime）
最後にユーザ認証に失敗した時の時刻である。初期値は0である。

(3) ユーザ識別子（UserId）
ユーザ認証に失敗したユーザの識別子である。

2.1.2 ユーザ認証処理

ユーザはシステムを起動し、ユーザ識別子とパスワードを入力している。システムは予め登録されているパスワードを使って以下の処理を行う。

(1) FailNum≧MAXNUMならば、次の処理を行う。
(i) EndTimeをFailTime＋RSTTIMEにする。
EndTimeは、ロックアウト終了時刻である。
RSTTIMEはロックアウト時間である。
(ii) EndTime≧現在時刻 (*CurrentTime*)ならば、処理を終了する。これが、ロックアウト状態である。
(iii) FailNumを0にする。

(2) (入力パスワード)=(登録パスワード)ならば、次の処理を行う。
   (i) FailNum を 0 にする。
   (ii) ユーザにシステム利用許可を与える。
   (iii) ユーザが利用を終了したら、処理を終了する。
(3) FailNum の値を 1 増やす。
(4) FailTime を *CurrentTime* にする。
(5) パスワード入力を再度要求して(1)へ戻る。

2.2 ロックアウトの問題

ファイルシステムに対するアクセス制御機構やプロセスに対するアクセス制御機構を持たない PC マシン上では、ロックアウト情報を単一のファイルやプロセスに保持した場合、不正ユーザに削除されないようにすることは難しく、問題である。

また、不正ユーザがロックアウト情報だと思われるデータを削除した場合、システムを起動しロックアウトが解除されるか確認すれば、ロックアウト情報の正当性を検証できてしまうことも問題である。

2.3 目標

提案するロックアウト方式はロックアウト情報を改ざん・削除する攻撃（特に、EndTime を *CurrentTime* よりも古い時刻に設定する攻撃や削除する攻撃）をできるだけ防ぐことを目標としている。こうすることで、ユーザ認証機能を使って繰り返しパスワードを試す不正ユーザが、ロックアウトを不正に解除する攻撃を減らすことができる。

3.疑似ユーザ認証方式

3.1 設計思想

疑似ユーザ認証方式は以下の方針を元に設計した。

(1) 不正ユーザにロックアウト情報を発見でき、ロックアウトを簡単に解除できたと思い込ませる。不正ユーザに正規のロックアウト情報を発見されると、その内容の改ざんや削除を防ぐことは非常に困難である。そこで、ロックアウト情報を通常データ形式と暗号データ形式の 2 つ用意して、通常データ形式のロックアウト情報を不正ユーザに発見しやすい場所に保持して、不正ユーザにロックアウトを解除し易いと思わせるのである。システムは、暗号データ形式のロックアウト情報を基にシステムの処理を決定する。

(2) 繰り返しロックアウト情報の改ざん削除を行う不正ユーザに対しては、不正回数に応じてペナルティを与える。これは、不正回数に応じてロックアウト時間を延長することである。このようにして、不正ユーザに対する正常なユーザ認証処理の利用期間を少なくできる。

3.2 疑似ユーザ認証

疑似ユーザ認証では、不正ユーザがどのようなパスワードを入力してもシステムの利用を許可することはない。システムは正しいパスワードであってもシステムの利用を許可しない。一定回数以上ユーザ認証に失敗するとロックアウト状態になる。不正ユーザが疑似ユーザ認証を繰り返し行うと、システムはロックアウト状態を長期化する。

図 2 では、横軸が時間の流れを示している。ここでは、ユーザがユーザ認証に失敗してロックアウトになるまでの認証試行最大回数を 3 回とする。図 2 では、ユーザは 6 回ユーザ認証を試みている（1-6）。

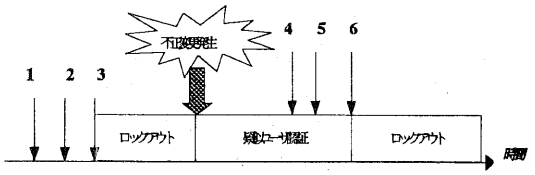

図 2 ロックアウトタイミング

不正ユーザが 3 回(1-3)パスワードを試して失敗し、ロックアウト状態になった時に、ロックアウ

ト情報を削除すると疑似ユーザ認証状態になる。ここで、不正ユーザが改めて3回(4-6)でパスワードを試して、正規なパスワードを入力しても、システムを利用することはできない。必ず3回目にロックアウト状態になる。

ロックアウト情報を削除する場合、改ざんによりロックアウトが解除されるならば、削除と同様に疑似ユーザ認証状態になる。改ざんによりロックアウトが解除されないならば、そのままロックアウトを続ける。

3.3 処理の概要

疑似ユーザ認証方式の処理の構造を図3に示す。

(1) ユーザの要求でシステムは起動する。
(2) パスワードが入力される。
(3) ロックアウト情報が改ざんされているか検証を行う。改ざんがあれば疑似ユーザ認証へ進む。改ざんがなければシステム状態の検証へ進む。
(4) システムの状態を検証する。ロックアウト状態ならば処理を終了する。正常状態ならばパスワードの検証へ進む。ロックアウト状態になってから一定期間が経過しているとロックアウト解除の処理が行われて元の状態に戻る。
(5) システムは入力されたユーザ識別子とパスワードを予め登録されているパスワードと比較検証する。同一ならば、システムはユーザに利用を許可する。異なるならばパスワード要求へ戻る。但し、定められた回数以上間違えた場合、ロックアウト情報の設定を行う。

3.4 構成の変更

疑似ユーザ認証方式の構成には、図 1 に、2.1.1 項で述べたロックアウト情報 FILEDATA と暗号化されたロックアウト情報 REGIDATA を加える。REGIDATA は FILEDATA と一部重複している以下の情報を含んだもので、システムの用意した鍵で暗号化されている。

(1) ユーザ認証失敗回数（FailNum）
(2) 最終ユーザ認証失敗時刻（FailTime）
(3) ユーザ識別子（UserId）
(4) 疑似ユーザ認証回数（DmmyNum）

システムが正常状態か疑似ユーザ認証状態か判断する。DmmyNum が 0 の時は正常状態である。DmmyNum が正整数の時は疑似ユーザ認証状態である。DmmyNum の値に比例して疑似ユーザ認証状態が長くなる。

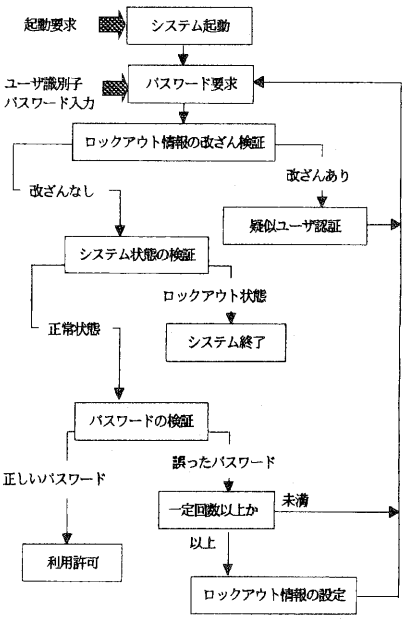

図 3 処理の概要

FILEDATA の FailTime や UserId を初期化する時には、システムは REGIDATA の対応する FailTime や UserId を初期化する必要がある。これは更新時も同じである。

システムは FILEDATA と REGIDATA をそれぞれユーザ毎に別の場所（異なるファイルやレジストリなど）で管理する。

3.5 表記方法

以下の説明では、FILEDATA にある変数には[F]を、REGIDATA にある変数には[R]を接尾語としてつける。特に[F][R]どちらも付いてない場合は、両方の変数に対するものである。

4.疑似ユーザ認証方式の処理

3章の処理概要で述べたのそれぞれの処理について説明する。

4.1 ロックアウト情報改ざん検証処理

(1) REGIDATAを復号する。
(2) (FailNum[F] ≠ FailNum[R])or(FailTime[F] ≠ FailTime[R])ならば、次の処理を行う。
  (i) DmmyNum[R]を1増やす。
  (ii) 0≦FailNum[F]<MAXNUMならば、疑似ユーザ認証処理へ進む。
  (iii) FailNumをMAXNUMにする。
  (iv) FailTimeを*CurrentTime*にする。
  (v) REGIDATAを暗号化する。
  (vi) 処理を終了する。
(3) (DmmyNum[R] > 0)and(FailNum[F] < MAXNUM)ならば、疑似ユーザ認証処理へ進む。
(4) システム状態検証処理へ進む。

4.2 疑似ユーザ認証処理

(1) FailNumを1増やす。
(2) FailTimeを*CurrentTime*にする。
(3) REGIDATAを暗号化する。
(4) パスワード検証処理を行わずに、パスワード要求処理へ戻る。正規のユーザ識別子とパスワードが入力されてもシステムを利用させない。

4.3 システム状態検証処理

(1) FailNum[F]<MAXNUMならば、パスワード検証処理へ進む。
(2) EndTime を FailTime[F] + (RSTTIME * (1+DmmyNum[R]))にする。
(3) EndTime<*CurrentTime*ならば、次の処理を行う。
  (i) FailNumを0に設定する。
  (ii) DmmyNum[R]を0に設定する。
  (iii) パスワード検証処理へ進む。
(4) REGIDATAを暗号化する。
(5) 処理を終了する。

4.4 パスワード検証処理

(1) (入力パスワード)=(登録パスワード)ならば、次の処理を行う。
  (i) DmmyNum[R]を0する。
  (ii) FailNumを0にする。
  (iii) REGIDATAを暗号化する。
  (iv) ユーザにシステム利用許可を与える。
  (v) ユーザが利用を終了したら、処理を終了する。
(2) ロックアウト情報設定処理へ進む。

4.5 ロックアウト情報設定処理

(1) FailTimeを*CurrentTime*にする。
(2) FailNumを1増やす。
(3) REGIDATAを暗号化する。
(4) パスワード要求処理へ戻る。

5.考察

5.1 ロックアウト情報の保存

本方式では、システムはロックアウト情報をファイルやレジストリなどのユーザの容易にアクセスできる領域に格納している。しかし、常駐プロセスを利用してメモリ中に格納すれば、より発見できにくくできるであろう。

また、ロックアウト情報を安全なリモートシステムに保持できれば、情報の改ざん・削除はできなくなる。しかし、この場合は以下の点に気を付けた配信プロトコルの設計が必要である。

(1) メッセージ改ざん
(2) リプレイ攻撃
(3) 応答性能

5.2 疑似ユーザ認証付システムのインストール

不正ユーザが疑似ユーザ認証付システムを自分用にカスタマイズできる環境にインストールできると、ロックアウト情報の格納場所を発見される可能性が高くなる。

したがって、安全なインストール方法を検討する必要がある。例えば、特別なパスワードを入力

しないとインストールできないような仕組みが組み込まれていると、インストール媒体だけを盗まれてもシステムを不正に利用されることが防げる。

5.3 トロイの木馬

トロイの木馬攻撃は非常に危険である。トロイの木馬を仕掛けられた状態では、ユーザ識別子やパスワードを守ることは非常に困難である。したがって、トロイの木馬を仕掛けられないような対策を講じる必要がある。例えば、コンソールあるいはネットワークからのあらゆる不正アクセスを監視するなどの方法が考えられる。

5.4 ロックアウト解除方式

不正ユーザがREGIDATAやFILEDATAを改ざん削除して繰り返すと、ロックアウト時間も不正の改ざん削除の回数に応じて長くなる。この特徴を利用すると、不正ユーザが正規ユーザをユーザ認証から締め出すことができる。

この攻撃を防ぐために、システムは以下の処理を用意している。

(1) 疑似ユーザ認証状態で入力パスワードを検証する。
(2) パスワードが正しければ、DmmyNumを0にする。

上記方法で、RSTTIME 時間待てば正常状態に戻り、正しいユーザはシステムを利用することができる。

しかし、この方法では、ユーザが正しいパスワードを入力しているにもかかわらず、ユーザ認証の失敗通知を行うので、ユーザは自分のパスワードを間違えたと思ってしまう可能性が高い。

パスワードを入力直後にロックアウトに入れば正規ユーザはロックアウト時間待つしかないので問題ないが、ロックアウトの入らずに、もう一度正しいパスワードを入力した場合、必ず失敗するので、さらに自分のパスワードを疑う結果になる。

ここで、正しいパスワードが入力された時は、強制的にロックアウトする方法も考えられるが、攻撃者がパスワード発見の目安にできるのでそのまま利用することは難しい。さらに、ユーザがユーザ識別子とパスワードを入力できる状態でないと適用できない問題もある。

6.まとめ

パスワード認証を利用したセキュリティシステムの使用する情報を不正ユーザが簡単に改ざんしたり削除したりできないようにするための方式を提案した。本方式は次の利点がある。

(1) ユーザがパスワードを入力する前にシステムに関連した情報を操作した場合、それを検知することができる。
(2) 不正を繰り返し行えば、不正の回数に応じてシステムへのアクセス禁止期間を長くすることができる。

ロックアウト情報の格納場所、パスワード認証付システムのインストール、トロイの木馬攻撃に対する対策、疑似ユーザ認証状態の解除方式について議論した。

また、以下の検討事項がある。

(1) 本方式は絶対的にロックアウト情報を守れない。ロックアウト情報が複数有ることに気づかれたら、破られてしまう。
(2) 疑似ユーザ認証状態で正規ユーザが正しいパスワードを繰り返し入力しても、疑似ユーザ認証状態を解除することができない。

7.参考文献

[1] Postel, J.: Internet Protocol, RFC791 (1981).

[2] Atkinson, R.: Security Architecture for the Internet Protocol, RFC1825 (1995).

[3] Dusse, S., Hoffman, P., Ramsdell, B., Lundblade, L. and Repka, L.: S/MIME version 2 Message Specification, RFC2311 (1998).

[4] Zimmermann, P.: PGP User's Guide (1992).

[5] Common Criteria Implementation Board: Common Criteria for Information Technology Security Evaluation Version 2.0, Part2: Security functional requirements (1998).