パスワードシステムのセキュリティの検討

鬼丸智美　(指導教員: 金子 晃)

1　始めに

システムやネットワークセキュリティにおいて, 最も深刻な脅威の 1 つは不正侵入であり, それを防ぐのは第一にパスワードシステムである. 実際全てのマルチユーザシステムでは, ユーザの名前や ID だけでなくパスワードを要求するようになっている. そのパスワードシステムについて, その意義や問題点, また現在使われている技術について考察する.

2　侵入とは

侵入とは, 一般に hacker や cracker と呼ばれ, Anderson によって 3 つの種類に分類されている.

＊ Masquerader(なりすまし)
正規のユーザになりすまし, そのアカウントの持つ特権を利用する.

＊ misfeasor(失当行為者)
正規のユーザが, 自分のアカウントでは許可されていない特権のアクセスを利用したり, 特権的なアクセスを許可されたユーザがそれを悪用する.

＊ Clandestine user
システムの管理権を入手した者がシステムの監視から逃れてアクセスコントロールを行ったり, 監査記録を改竄したりする.

3　Password Protection

パスワードはユーザーの ID を認証し, ID は次のようにしてセキュリティを提供する.

＊ ID はシステムにアクセス権をもっているユーザかどうかを決定する.

＊ ID はシステムの管理者や, スーパーユーザといった特権を与えられたユーザかどうかを決定する.

＊ ID はいわゆる自由裁量のアクセスコントロールとして使われる.
(例：所有するファイルを ID のリストを利用し, 特定のユーザのみが閲覧できるようにする)

4　パスワードの暗号化

伝統的に, UNIX におけるパスワードは, 一方向性ハッシュ関数を使って暗号化を行っている. 元々, DES(Data Encryption Standard) と呼ばれる秘密鍵アルゴリズムを用いていたが, DES を含む暗号化の技術は, 合州国以外への持ち出しは禁止となっていた.
このため, オープンソース系の OS では, UNIX 互換性を持つ暗号ライブラリとして実装するために, 独自のライブラリを組み込んでいる.
例えば FreeBSD では, DES 仕様のパスワードスクランブラ (libcrypt) と, 問題になる実際の暗号化を行うライブラリ (libcipher) を別にしている. また, より強固なハッシュを行うため, RSA(RSA Data Security 社) の MD5(Message Digest Algorithm 5) を採用している. また, libcrypt 等のライブラリから呼ばれる重要な関数, crypt() では「a z,A Z,0 9,'.','/'」の中から, salt と呼ばれる 2 文字を使ってハッシュを行う. 結果的に暗号化されるパターンは 4096 通りあり, 並列処理を行ったとしても逆方向への解読は不可能となっている. しかし, この方式には, 文字列を同様の salt の 4096 通りにハッシュすることで, 言い当てることが可能だという大きな問題点がある. 要するに膨大な辞書を持っていて, なおかつ CPU パワーと時間を消費することによって, 当てずっぽうに解読出来るものであると言える.
これがパスワードクラッカーの常套手段であり, 後述のパスワード推測プログラム「Crack」がパスワード解析を行う方法としても用いられている.

5　Secure Hash Algorithm

SHA（Secure Hash Algorithm）は, 一群の関連したハッシュ関数である.
アメリカ国立標準技術研究所 (NIST) によってアメリカ政府標準のハッシュ関数 Secure Hash Standard (SHS) として, 160 ビット長の特徴値を出力する SHA-1(アルゴリズムは MD4 を元にしており, MD5 よりもビット数が大きいため攻撃に強いとされている) が採用されていたが, ハッシュ関数攻撃手法の発展とともに, 2005 年, 最新の暗号解析技術によって脆弱性が発見され, 期待される安全性を確保できないことが明らかになったため, NIST は 256 ビット長の特徴値を出力する SHA-2 への移行を推奨し, また SHA-1 に替わるハッシュ関数の公募を行うことを決定した. SHA-3（Secure Hash Algorithm 3）は, NIST が公募中の新しいハッシュ関数アルゴリズムである. NIST は, 世界中から 64 方式のアルゴリズムの中から, 51 方式を次世代標準ハッシュ関数の候補として認定し, SHA-3 の第一次選考が行われ, その結果, 第二次選考に進む候補として 14 方式が選定された. 日本からは株式会社日立製作所が, ベルギーのルーヴァン・カトリック大学と共同で開発したハッシュ関数「Luffa(ルッファ)」のみがこの第二次選考の候補として残っている.
今後は, 本年の夏頃に最終候補となる 5 方式が選出され, さらに 2 年かけて次世代標準ハッシュ関数, SHA-3 が決定される予定である.

6　脆弱なパスワード排除の主なツール

＊ Crack
Crack とは, Alec Muffett 氏によって書かれたパスワード推測プログラムであるが, 1996 年に Crack5.0 がリリースされて以来, バージョンアップはされていない. パスワード候補のハッシュ値をパスワードファイルの内容と照し合せることにより, 脆弱なログインパスワードを設定したユーザを素早く探し出す為に設計された. また, Crack の特徴として, 独自の Crypt() 関数を統合でき, なおかつ独自フォーマットのパスワードファイルも解析できる柔軟なところが挙げられ, Linux をはじめ FreeBSD, NetBSD, Solaris, Ultrix, OSF など, 多くの OS に対応している.

＊ CrackLib
CrackLib とは Crack と同様, Alec Muffett 氏によって書かれたプログラムで, 推測しやすいパスワードをユーザが選択することを防ぐ目的で作られた. CrackLib はパスワードについて, ユーザ名と gecos(ユーザー情報, 慣例的に ”フルネーム, オフィスの部屋番号, オフィスの内線番号, 自宅の電話番号, その他” の情報がカンマ ‘,’ 区切りで登録されていたが, 今では, このようなプライベートな情報をこのフィールドに登録するべきではない. 通常はユーザーのフルネームだけが登録される) エントリから単語を生成し, それらの単語のパスワードに対する照合チェック, パスワード内に単純すぎるパターンがないかどうかのチェック, またパスワードが辞書に載っていないかどうかのチェックといった複数のテストを実行し, パスワードが特定のセキュリティ指向の特性を満たしているかどうかを判定する.

＊ SATAN
SATAN とは「Security Administrator Tool for Analyzing Networks」の略で, 後述の COPS の作者である Dan Farmer 氏と Wiwtse Venema 氏によって書かれた, システム管理者のためのセキュリティ分析ツールである. SATAN はセキュリティホールのデータベースを持ち, それを元にセキュリティホールを発見する. 言い換えれば, 自分のマシンを攻撃し, セキュリティホールを検出する. それにより, 外部からの攻撃者が狙うような潜在的侵入経路を探す. 管理者は, 発見された弱点を修正することでセキュリティ強化を実施することが可能となる. SATAN が他のセキュリティツールと根本的に異なるのは, 実際に外側からのアタックを行って検出するというその性質, 過程であると言える. SATAN は自分のサイトだけでなく, 世界中のインターネット接続されているコンピュータ, ネットワークに対して, スキャンを行うことが可能である.
1995 年の春に登場した SATAN は, ネットワークに関わる人々だけに留まらず, 多くのマスコミでも取り上げられ論議をもたらした.
SATAN リリースに先立って, 二人の連名で書かれた論文「Improving the Security of Your Site by Breaking Into it(1993)」が発表され, 同様に物議をかもした. ドキュメントに記述されている詳細な手口やセキュリティホールの情報を, 一般に侵入の間口を開くものとして, 多くの人間が心配したためである. 結果的に, 世間一般のセキュリティに対する感心を高め, 各種ベンダーによる防御用ソフトウェアが開発されたという点で, とても重要だといえる. SATAN を実行する際の注意点として, NFS マウントされているシステム上や, Xhost 等のリモートセッションで実行されるべきではないという点がある. というのは, SATAN には独自の HTTP サーバが含まれており, ブラウザと通信を行う. MD5 ハッシュ関数による 32bit マジック COOKIE を認証のチケットとして発行することで, ある程度の安全を保ってはいるが, COOKIE のセッションが盗聴される可能性もあるためである.

＊ COPS
COPS とは「Computer Oracle and Password System」の略であり, UNIX システムに存在するセキュリティ上の脆弱点を検出する目的で米パーデュ大学が開発した監査ツールである.
実行した UNIX システムを検査し, ファイルのアクセス権限（モード）や問題のあるパスワードや SUID ファイル, アクセス制御の設定などを検出し, 報告する. COPS は大きなひとつのプログラムではなく, 十数個以上の小さなプログラム群によって構成されており, 必要に応じてそれらを実行する形をとる. COPS は古くから存在する UNIX のセキュリティチェックの定番であると言える. Linux や FreeBSD など, ほとんどの UNIX で動作する.

7 Crack の実装と実験結果

上記のパスワード推測プログラム Crack を FreeBSD 上に実装し, 架空のユーザを追加し, 各々のユーザのパスワードがどの程度破られるか実験を行った.
今回は辞書の拡張や制限の追加設定は行わず, Crack が始めから持つ辞書と gecos データでどれ程実用性があるかを検証した. 実際破られた結果は以下のようなものであった.
＊結果
一般的な人名（例：tomomi,akira,miny）
ユーザ名（例：onichan, 自分のユーザ名以外の登録されている他のユーザ名でも）
単純文字列（例：abc）
但し, 数字の組合せ 123 や, abc 以外のアルファベットの組み合わせ aaa など, またユーザ名に 1 文字追加した onichan2 などは推測不可能で, 辞書の充実や, パスワードポリシー自体を厳しくしなければ実際のクラッカーのレベルには及ばないようである.

まとめと今後の課題 パスワードシステムについて調査した. この知見をシステム強化に実現していきたい.

参考文献

[1] William Stallings: “Cryptography and Network Security”. Prentice Hall, 2006; Chapter18 INTRUDERS.

[2] Xiaoyun Wang,Yiqun Lisa Yin,and Hongbo Yu : “Finding Collection in the Full SHA-1”, CRYPT2005, LNCS 3621, pp.17–36.

[3] まえだひさこ: “PC-UNIX サーバのためのクラッカー撃退計画”. 翔泳社, 1999; sectin2 セキュリティツールの導入.

[4] Wikipedia ” SHA-3 ” “http://ja.wikipedia.org/wiki/SHA-3”

[5] HITACHI ニュースリリース “http://www.hitachi.co.jp/New/cnews/month/2009/01/0115a.html”

[6] CERT Advisory CA-1995-06 Security Administrator Tool for Analyzing Networks (SATAN) “http://www.cert.org/advisories/CA-1995-06.html”

[7] cops.1.04.README “http://ftp.cerias.purdue.edu/pub/tools/unix/scanners/cops/cops.1.04.README”

[8] crack.5.0.README “http://hpux.connect.org.uk/hppd/hpux/Sysadmin/crack-5.0/readme.html”