パスワード今昔物語
2013年バージョン

すずきひろのぶ

スライドの概要

- パスワードを例にして、むかしと今とでは考え方が変化していることを理解してもらう
  - 時代によって考え方や方法論が変化していくことを知って欲しい
    - 昨日の常識は今日の常識ではないし、今日の常識は未来の常識ではないこと
- セキュリティには正解がないことを理解してもらう
  - 色々なアプローチがあるが何によって評価されるかでどれが最適なのかがかわってくる


BREAKING
NEWS

WordPress Sites Targeted by Mass Brute-force Botnet Attack

Original release date: April 15, 2013

Print Tweet Send Share

US-CERT is aware of an ongoing campaign targeting the content management software WordPress, a free and open source blogging tool and web publishing platform based on PHP and MySQL. All hosting providers offering WordPress for web content management are potentially targets. Hackers reportedly are utilizing over 90,000 servers to compromise websites' administrator panels by exploiting hosts with "admin" as account name, and weak passwords which are being resolved through brute force attack methods.

CloudFlare, a web performance and security startup, has to block 60 million requests against its WordPress customers within one hour elapse time. The online requests reprise the WordPress scenario targeting administrative accounts from a botnet supported by more than 90,000 separate IP addresses. A CloudFlare spokesman asserted that if hackers

http://www.us-cert.gov/ncas/current-activity/2013/04/15/WordPress-Sites-Targeted-Mass-Brute-force-Botnet-Attack

Wordpressサイトへの大規模な攻撃が発生

- Wordpressを運用しているサイトに対して管理者権限を略奪しようとする事案が発生し、既に90,000が侵入されているとの情報がUS-CERTからアナウンスされている
  - ユーザadminに対しパスワードを試す攻撃が行われている
    - 単純な辞書を使っているようだ
  - 90,000程度のボットを使い攻撃をしている
    - これはまれにみる大規模な攻撃

そもそもパスワードって何

- 知識を共有し、その知識を持っていることを確認することで、正当な相手であることを認める
  - 合言葉
    - 赤穂浪士の討ち入り（山・川）
    - 古代ギリシア軍では既に使われていたらしい

パスさせるワード

- ８文字程度の文字列をユーザに入力させ、コンピュータ内部の情報と付き合わせて同じかどうかを調べる
  - 同じであれば秘匿されている知識を共有しているとみなす

パスさせようとするワード

- 思い出しやすいワード
  - Hironobu1963
    - 絶対に忘れないようなパスワードだが類推可能である
  - 12345678
    - 誰が考えてもこれはひどいと思うだろうが、実際にこのパスワードを使う者は多い

```
Yahoo hack reveals most-used passwords
by Bridget Carey  July 12, 2012
c|net

http://news.cnet.com/8301-33692_3-57471417-305/yahoo-hack-
reveals-most-used-passwords/
```

Yahoo password leak case

- 2012年に米Yahooから450,000個のパスワードがリークした
  - おもしろそうな数字をあげてみる

パスワード数	凡例
2,295	123456, 1234567, 12345678,...
160	111111,000000,777777,...
780	password
233	password1, passoword2
437	welcome
106	Batman, superman, spiderman,...
27	ncc1701, ncc1701a,...

どれくらいのパスワードが分かったか

- 450,000のパスワード中137,559が判明した
- 106,873のパスワードがGmailやHotmailなど他のサイトで使いまわされていた

パスワードは一般ユーザに対して有効な方法なのか極めて大きな疑問を抱かざるを得ない

Cracklib

- 弱いパスワードをユーザに利用させないためのパスワードチェックライブラリ
  - 辞書によるパスワードの強度チェック
  - パターンによる強度チェック
  - 12文字未満のパスワードに対し警告
- Cracklibが警告を出すようなパスワードは弱いパスワード
  - 自分のパスワードファイルもクラックできる（安全性の確認のため）

ブルートフォース（しらみつぶし）

- 計算機の能力に依存する
  - スーパーコンピュータでも用いられている高性能グラフィック専用計算チップを使う手法（GPGPU）を一般でも利用可能な時代になっている
  - Ivan Golubev's Password Recovery Suiteのデータから現在の最高級グラフィックボードを使った時の性能推定

年	機材	MD5（秒）	SHA-1（秒）
2009	NVIDIA GeForce GTX-260	5億回	1億7,500万回
2013	NVIDIA GeForce GTX-690	約35億回（推定）	約11億回（推定）
同上	4枚ボード挿	約140億回（推定）	約44億回（推定）

文字の複雑さ

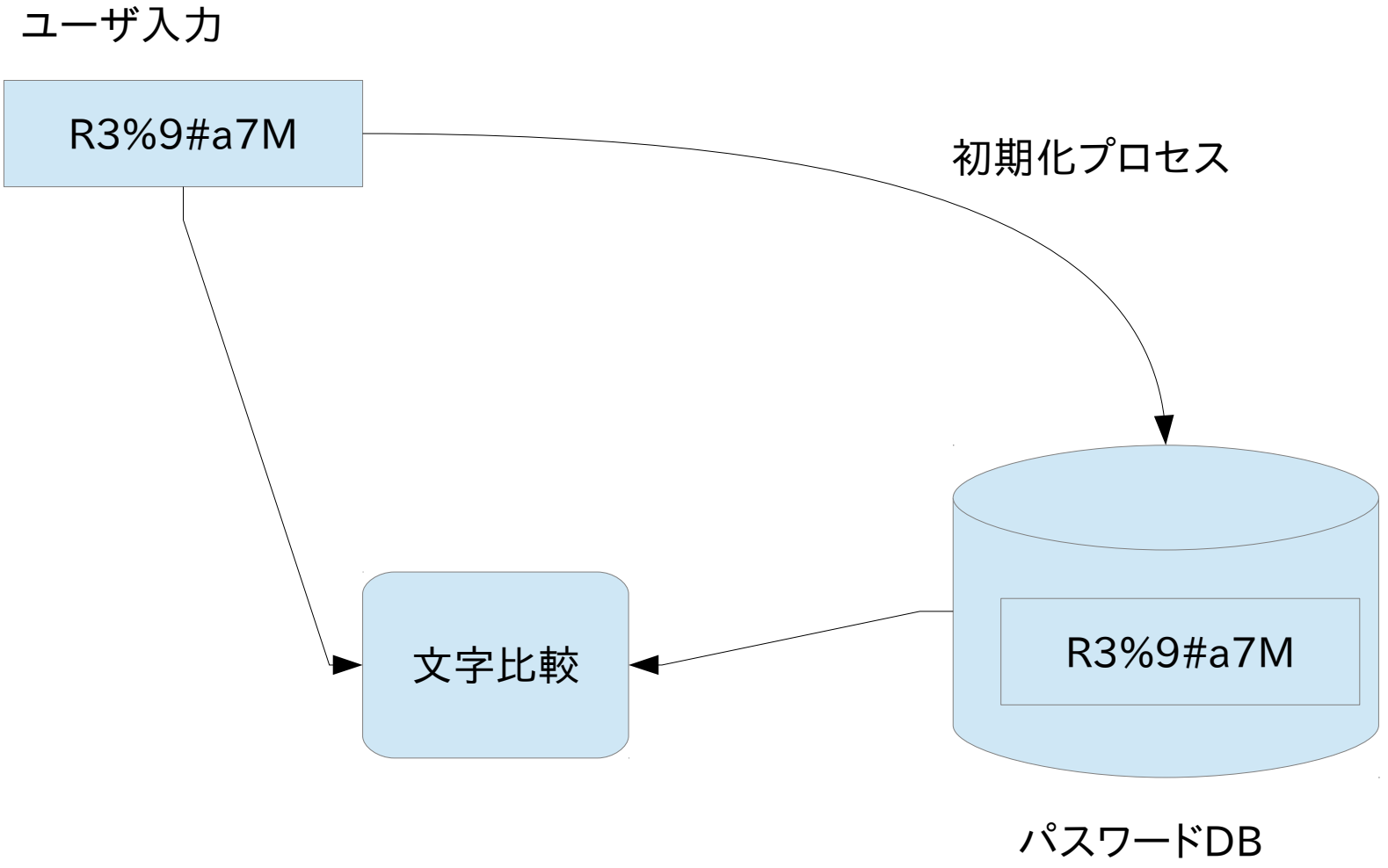
- 0-9A-Za-zのみで62文字
  - 8文字で約218兆3401億通り
  - 10文字で約84京通り

単純な計算

- 218兆個を140億個/秒（MD5）で割ると約15,571秒
  - 8文字ランダムからなるパスワードは平均7,785秒≒2時間強でクラックされる

システムの中でのパスワードの仕組み

レベル0
素朴な実装

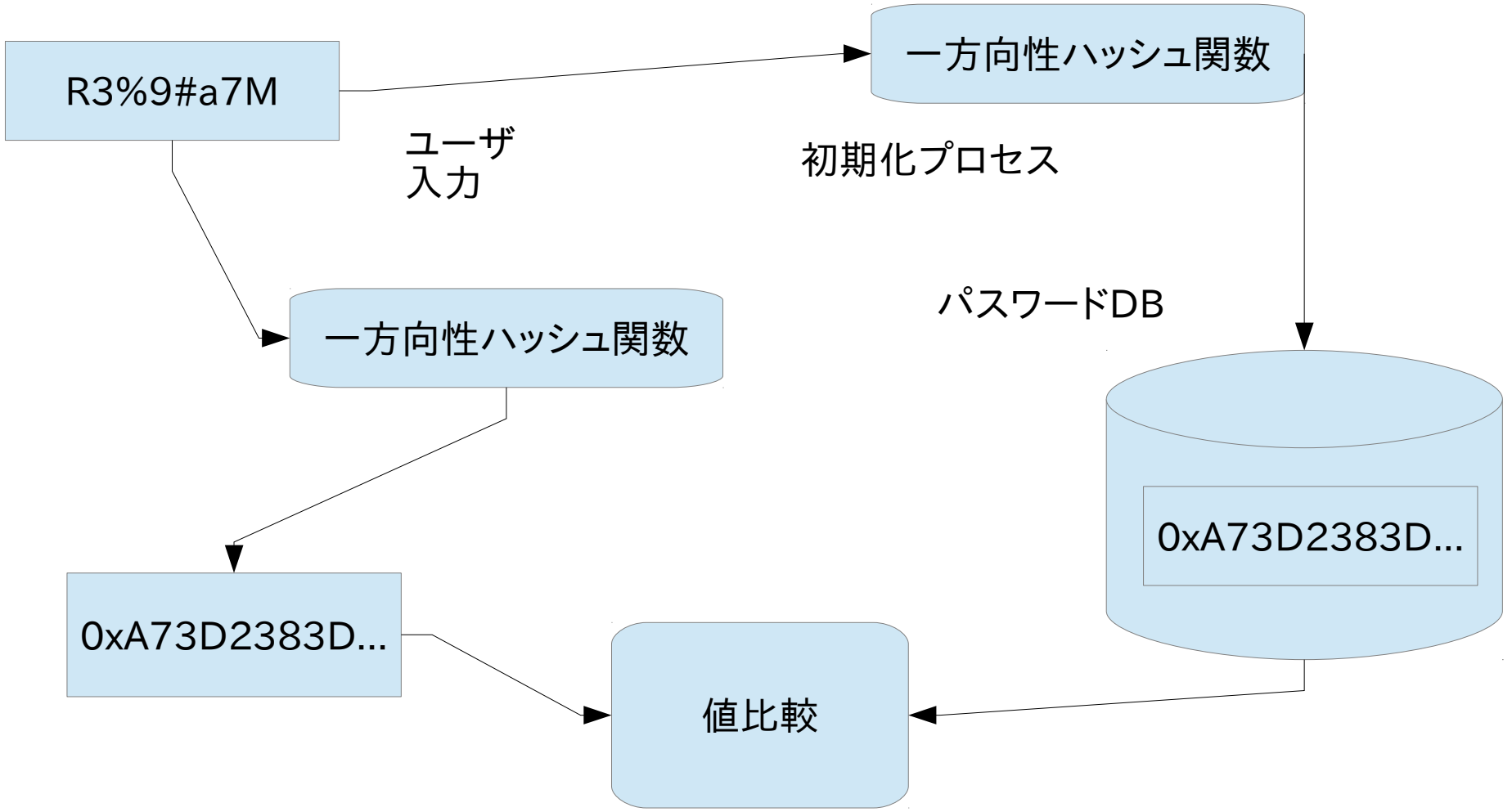


レベル0

- 利点
  - 単純に過去のデータを引き出して文字列比較をすれば良いので実装が簡単（安易ともいう）
    - 技術を知らない人はパスワードの仕組みはこんなイメージだと思われる
  - ユーザーがパスワードを忘れた場合、ユーザに利用しているパスワードを教えることができる
- 欠点
  - パスワードのDBの内容が盗まれたら全件のパスワードがそのままわかってしまう
    - サービスの存続に関わるレベルで大きな問題になる

忘れたパスワードをメールで送り返してくれるようなシステムは意外と多いことに気がつくはず

レベル1
一方向性ハッシュ関数を導入する



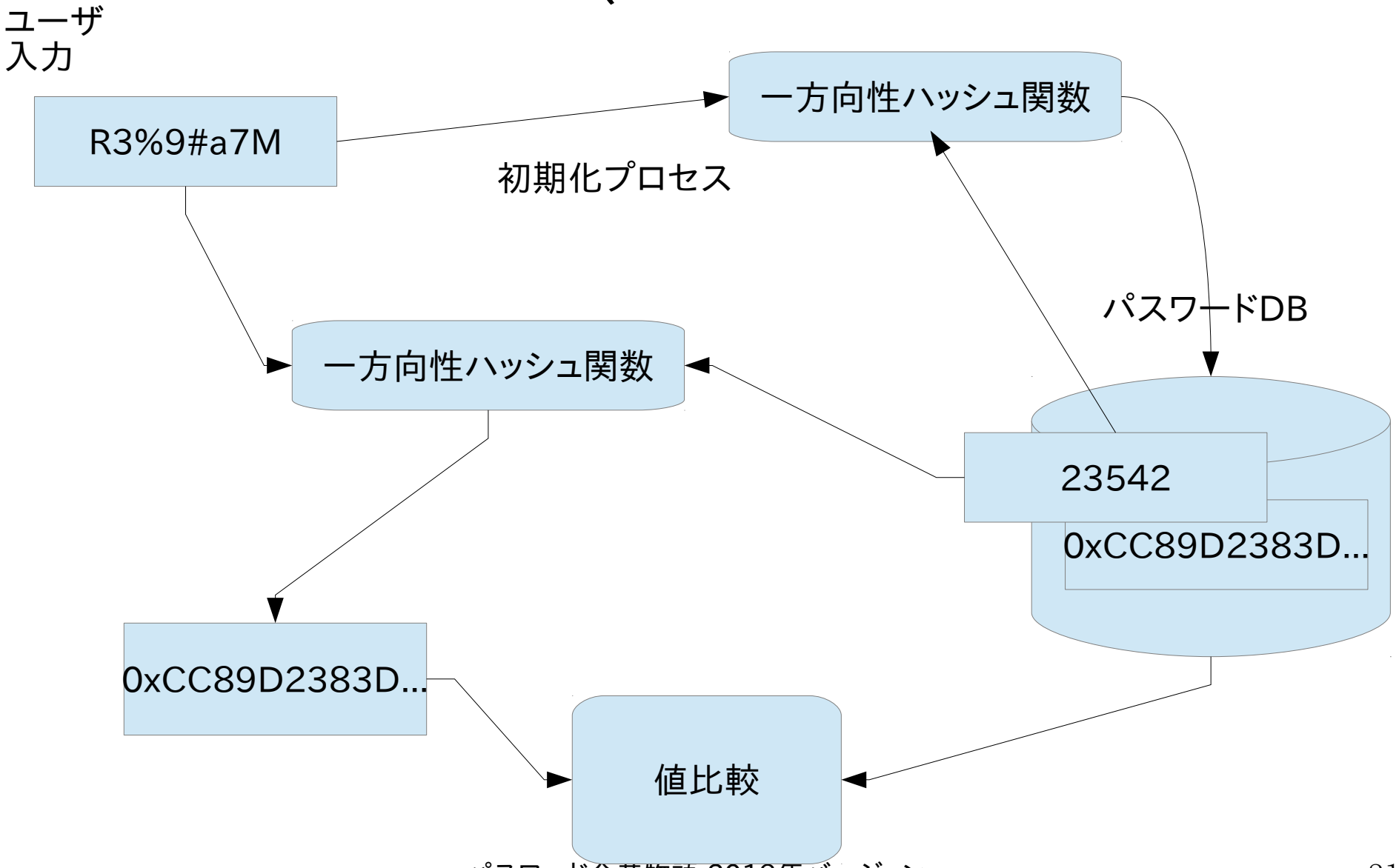
一方向性ハッシュ関数

- 入力した値を計算して値を出力する関数
  - 出力した値から入力した値を逆算することが困難である関数
    - *MD5, SHA-1,* SHA256
    - DES-MAC, AES-MAC
    - HMAC

レベル1

- 利点
  - パスワードDBを盗まれてもパスワードを見つけるにはパスワード解析をする必要がある
    - 安易なパスワードを利用するユーザは多く、あくまでも難しいパスワードを使っている場合という暗黙の条件がつく
- 欠点
  - 同じパスワードがあった場合、1つのパスワードが発見されれば、同じパスワードを使っているのも同時に発見されたことになる
    - 事前に処理したパスワード候補をハッシュ処理したものも用意することができるということ
  - 忘れたパスワードに対してはパスワードは再発行（再登録）という手順になる

レベル2
さらにSalt(ソルト）を加える



実装レベル2

- 利点
  - 実装レベル1の欠点である1つパスワードがわかっても、他のパスワードまでみつけることができない
- 欠点
  - 改めていうほどの欠点はない（標準的な実装）

パスワードの盗まれ方

パスワードの盗み方(1983)

- パスワードをメモしていたのを盗み見られる
  - ポストイットにパスワードを書いてディスプレイに張り付けておく
- ユーザに関係する単語を試す
  - 家族の名前や好きなものの名前

WarGames(1983)

パスワードの盗み方（2013）

- マルウェアをPCに植え付ける
  - まずマルウェアのあるサイトへ誘導するメールを送る
  - マルウェアのあるサイトにアクセスするとブラウザやPDFあるいはFlashの脆弱性をついて任意のコードが実行される
  - 最初は小さな実行コードを植え付けられるがそれはダウンローダーとして使われ、さらに機能の豊富なマルウェアをダウンロードする
  - PC内のファイルを精査しパスワードを記録しているファイルを外部に送信したり、あるいはキーロガーとしてPC内部に留まる

PCからサーバのパスワードを盗む

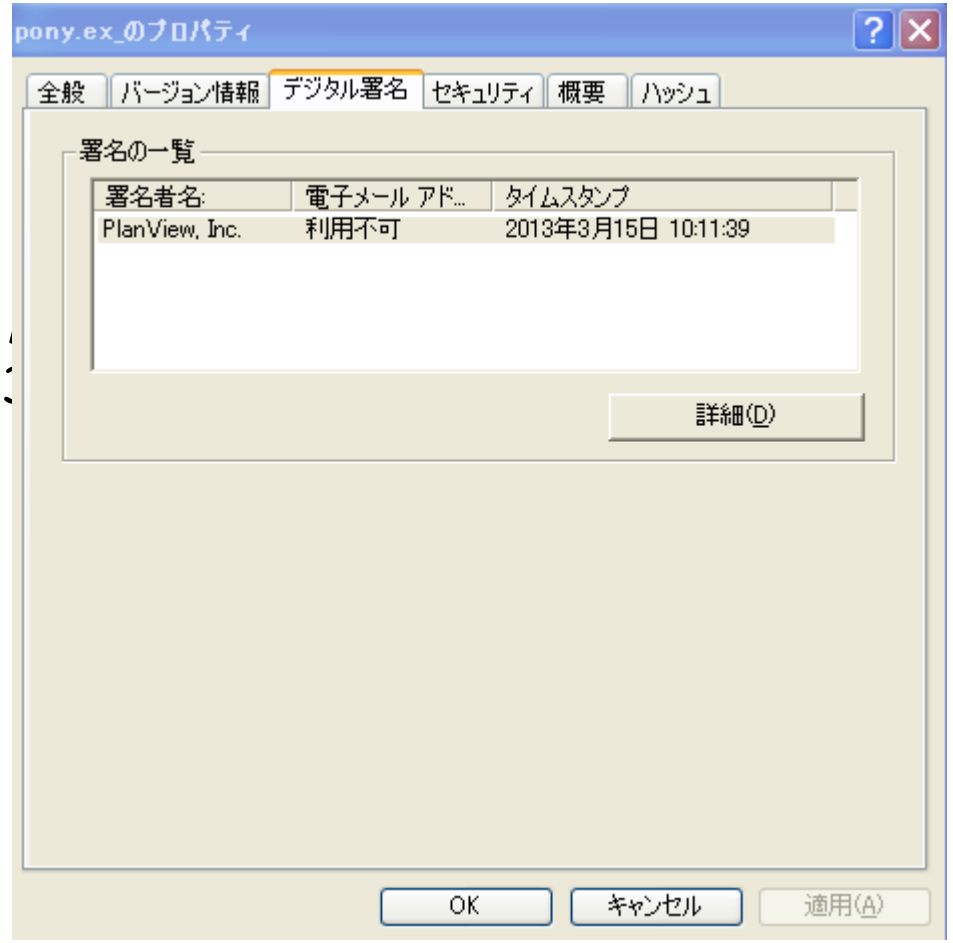
- Webサーバ等の遠隔サーバを管理するリモート端末やファイル転送のためのプログラムから盗む
  - ユーザの簡便さのために自動的に接続してくれる機能を持っているものがある
  - パスワードをファイルに取っていて、それを接続の際に使う

マルウェアがサーバー名、アカウント名、パスワードの情報が入っているファイルを外部に流出させる

PONY

参考元:IIJ SECT ブログ
BHEK2を悪用した国内改ざん事件の続報
https://sect.iij.ad.jp/d/2013/03/225209.html

FAR Manager Total Commander,
WS_FTP, CuteFTP, FlashFXP,
FileZilla, FTP Commander,
BulletProof FTP, SmartFTP,
TurboFTP, FFFTP, CoffeeCup FTP,
CoreFTP, FTP Explorer, Frigate3
FTP, SecureFX, UltraFXP,
FTPRush, WebSitePublisher,
BitKinex, ExpanDrive,
ClassicFTP, Fling, SoftX,
Directory Opus, FreeFTP,
DirectFTP (FreeFTP), LeapFTP,
WinSCP, 32bit FTP, NetDrive,
WebDrive, FTP Control, Opera



サーバ側はお手上げ

- どんなにサーバ側のパスワード管理を完全にしていてもクライアントがパスワードを漏らすので、サーバは自分を守ることができない
- ポストイットにパスワードを書いてディスプレイに貼っておくなんてかわいいもの
  - 少なくとも地球の裏側の匿名な誰かにパスワードは盗まれない
- 盗んだパスワードでWebサーバにログインし、好きなことができる
  - ブラウザでアクセスすると感染するマルウェアを仕込んでさらに犠牲者を増やす

1983 v.s. 2013

- パスワードを盗むのは自動化されている
  - 一度に多数を処理できるので、極めて効率的である
- 高速なネットワーク経由で行われている
  - 人的な接触やユーザの端末に近づく必要がないので、地球の裏側からも行うことが可能である
- ターゲットは必ずしもピンポイントではない
  - まずはできる所からを狙うのが一般的
    - もちろんターゲット型も存在しているが

横道の話なのですが

- 標的型・APT攻撃
  - 特定の人、あるいはグループを狙って執拗に攻撃をしかけてくる最近話題になっている
  - APT (Advanced Persistent Threat: 高度な継続的脅威）といったり、新しいタイプの攻撃と命名している所もある
- 本当に高度？本当に新しいタイプ？
  - 既存の攻撃法を踏襲しているだけなので高度ではない
    - マイクロソフトは誤解を助長させないためAPT攻撃とは呼ばない
    - 1983年当時は特定の人やグループに対して行っているので、むしろ先祖返りともいえる

現状のパスワード対策

安全なパスワード(?)

- むずかしい並びのパスワード
- 長いパスワード
- 頻繁に変えるパスワード

安全なようで安全ではないパスワード

- アルファベットの “i” を “1” / “O”を “0” に変える
  - “he110w0rld”
- 好きなフレーズの頭の文字を抜き出してパスワードを作る
  - “wysiwyg”

人間が思いつく程度の規則性があるなら自動生成可能なレベルなので辞書攻撃を拡張する方法が利用できる

安全なパスワードというけれど

- むずかしい並びのパスワード
  - 頭で考えるとバイアスが入るので安全なレベルのランダム性を確保できない
    - ここで「むずかしい」とはエントロピーが大きい（乱雑である）ということであるが、普通は意味が曖昧なまま「むずかしい」という言葉を捉えている
- 長いパスワード
  - 人間の記憶力を過大評価している
    - 多数の長いランダム列を記憶できるという前提に無理がある
    - 結果的に思い出しやすい（規則性のある・推定が容易な）文字列となる可能性の方がずっと高い
- 頻繁に変えるパスワード
  - 一番目と二番目の問題を何度も繰り返すことになることになり、結果は、あまり良いことにはならないであろう
    - 簡単に覚えられるパスワードを選択する＝安全ではないパスワードを使う可能性が極めて高い

むずかしい並び＝ランダムに生成する

- ランダムなパスワードのみ安全
  - 人が選ぶ場合、必ずなんらかのバイアスが入る
  - なのでツールを使って生成する
- ユーザがバイアスが入ったパスワードを使うのを避ける
  - システム側が自動的にランダムなパスワードを提供する
    - WordpressやMediawikiなどがこの考え方

サービス毎にパスワードを求められる現実

- 覚えられないからといってパスワードを使いまわすと、どこかでパスワードが漏れれば全滅する
  - 最近ではシステムがパスワードを生成し提供するようなシステムも多数ある
- 自分のFirefoxやChromeのパスワードリストには山ほど入っていますが何か
  - 覚えられる程度のパスワードは使っていない

1983年当時、そもそもコンピュータを使うということがまれで、使える場面も少なくせいぜい数個のパスワードを覚えば十分であった。パスワードはそんな時代の方法論がいつまでも使える方がおかしい。

Webサービスは覚えないのが勝ち

- 普通の人がすぐに思い出せるような文字列は簡単過ぎる
  - エントロピーが不十分
- 複雑で長いパスワードにして使い終わったら忘れる
  - 次回アクセスはパスワードを再発行する

一番安全な方法は自分も含めて
誰もログインできなくすること

- インターネット上にあふれるサービスを使う昨今、一度しか利用しないようなサイトでもユーザ登録をさせられることが多々あり、必然的に多くのパスワードを管理しなければならない
- ユーザIDがメールアドレスという簡単に推定できるケースも多く、その場合、アカウントを守るのはパスワードしかない
- パスワードは長期間安全であり、万が一サイト側の問題で漏れたとしてもそのサイト以外に影響がないようにするということを求められる

なぜにして未だにパスワード?

- 実装やスタートコストは安い
- とりあえず問題を深く考えず、これまでと同じ方法を取っていれば文句はいわれない
- 安易なパスワードの問題はユーザに責任を押し付け
  - ランダムなパスワードの自動発行やパスワードの強度をチェックしてくれるサイトはまだまだ少ない

むかしよりはパスワード管理のコストも増えている

- Yahooですらパスワードを流出させる
  - 漏れないコストも大変
  - パスワードが漏れると大変
- ちょっとしたサービスで一々パスワードを管理する負担は耐えられなくなってきている
  - TwitterやfacebookやOpenIDと連動してOAuthを使うというパターンが増えて来ている
  - OpenIDの方が個人的には押しなのだが、あまり知られていない（気がするのですけど、どうでしょうか？）

OAuth/OpenID

- OpenID
  - OpenIDを使い自分が正しいユーザであることを認証プロバイダによって証明してもらう
    - Google がOpenIDのIdentity providerなので実はAndroidユーザはデフォルトでOpenIDを持っているようなもの
- OAuth
  - 既に利用しているサービスを利用させることによって自分がそのサービスの利用者であることを証明する
    - 最近twitterやfacebookのOAuthを使うサービスが非常に増えている

また横道ですいません
twitterやFacebookのOAuth

- OAuthのインタフェースは最悪
  - twitterやfacebookにログインしていないでOAuthを使うサービスを呼び出した時、twitterやfacebookにログインするようにとIDとPasswordの入力を催促する
  - これはフィッシングの恰好の餌食なので絶対に入れないこと
  - 一回閉じて、twitterやfacebookに正規の方法でログインしてから、再度やるようなくせをつけてください
    - でも、たぶんそれでもフィッシングされるとは思うけど
- OAuthは利用権限を委譲する
  - 有用なサービスのように見せかけて個人の情報を抜き取るなどされていてもわからない

とはいえ、まだまだパスワードの時代は続く

パスワードマネージャを使う

- システムにパスワードマネージャーをインストールし、そこにパスワードをしまいこむ
  - さらに安全度を高めるには、パスワードを入力するシステムとパスワードを管理するシステムは分離されていると良い
    - セキュリティベンダーはPC用のものを提供している
    - 紙のパスワード帳に書き込み貯金通帳と一緒に管理するのもこれはこれで正しいパスワード管理である
    - スマホのパスワード管理帳なども今はある
    - 文具メーカーがパスワード管理専用のルーツを発売

なぜ認証が高度化しなかったのか

- ブラウザとWebサーバという単純な通信なので、そこに新しいセキュリティモデルを加えるのは大変な作業となる
  - 端末ベースだとSSHのような公開鍵暗号法を使った認証が普通に使えるので、そこまでひどくない
- ワンタイムパスワードなどは銀行で使われるが、デバイスコストや認証サーバー側のコストなどが高く、それに見合うだけのサービスでなくてはならない
- パーソナル・コンピュータという割には個人的に固定的に所有するハードウェアではなくハードウェア側に認証を固定的に置くことがむずかしかった
- セキュリティに関して色々な方法論はあるが、複雑に特許がかかっており、これをクリアするのは至難の技である

パスワード認証はもう限界に来ている

- パスワードベースから公開鍵暗号法による認証ベースに置き換わる必要がある
  - SSHやVPNでは長い期間の実績があるので、それをどう生かしていくのか
- BYODの時代に対応する必要がある
  - Googleの二段階認証プロセスは利用するデバイスが固定的であることを前提としている
  - スマートフォンなど個人所有の機材を認証デバイスとして使える時代になるので状況はかわるかも
- OpenIDやOAuthはどこまで耐えられるのか
  - OAuthは簡便だが十分な安全を満たす仕様であるのか？

パスワードの明日はどっちだ

- パスワードに変わる認証の研究はたくさんあれど、方向性が定まらない
  - 必要なのは優れたものではなく、デ・ファクトの位置づけになるなれるもの
- マルチパスでの認証もあるはずだけど
  - スマートフォン側に認証専用アプリを入れてPCと連動させるなど
    - でも誰かが特許を取っていればちょっと作ってみんなに使ってもらうか、というようなことは出来ない

まとめ

- パスワードは古くから使われている認証方法であるが故に、深く根を下ろし、そこから簡単には抜け出せなくなっている
- パスワードが盗まれる方が悪いというユーザに責任転嫁できるモデルだったが、サーバ自体からパスワードが流出する時代になって、そうもいっていられなくなった
- Webサービスが花盛りの今日、ユーザ登録などが非常に多くなり、たくさんのパスワードを管理しなくてはいけなくなったが大量のパスワードを管理する限界に既に達している
- ユーザがパスワード管理に気をつけるという運用論のように見せかけた精神論では限界が来ている
- OpenID/OAuthは問題もあるが破綻しているパスワードを使うモデルよりは少しはマシである
- しかし新しい実装を行おうにも複雑な特許のからみがあり、おいそれとは思い付きを実装しオープンソースで配布できない
- パスワードを使うという限界と転換期にあると思われるが、しかしながら次の方向性が見えないという大変な時代