

Passwords

Joe St Sauver, Ph.D.
Security Programs Manager, Internet2
(joe@uoregon.edu or joe@internet2.edu)

NWACC Security Conference, Portland, Oregon
Wednesday, November 18th, 2009, 1:00-2:15PM

<http://www.uoregon.edu/~joe/passwords/>

Disclaimer: All opinions expressed in this presentation are solely those of the author and do not necessarily represent the official opinion of the University of Oregon, Internet2, or any other entity.

The Format of This Talk; Acknowledgments and Disclaimer

- Yes, this is another one of those oddly formatted "Joe talks." For those who haven't seen one of my talks before, I make them verbose so they'll be readable after the fact for those who couldn't be here today, as well as for search engines, readers for whom english is a second language, the hearing impaired, etc. Please don't let my odd format shake you up. :-) I promise I won't read my slides to you, nor do you need to read them as I talk.
- With that out of the way, I'd like to thank Adrian Irish of the University of Montana and NWACC for the chance to talk today, although I should remind folks that all opinions expressed represent solely my own perspective, and do NOT necessarily represent the opinion of Adrian, UMT, NWACC, Internet2 nor the University of Oregon. 2

The Original Invitation

- Speaking of Adrian, when he'd originally asked me to come and present, we talked a little about potential topics for this meeting. Since I'd recently been at the annual Anti-Phishing Working Group (APWG) meeting and eCrime Researchers Summit in Tacoma, he suggested that I consider talking about phishing, or maybe malware.
- I mulled that over, then counter-proposed "passwords." Adrian assented, although he cautioned me that "this is a fairly technical, details oriented, hands on group."
- Those of you who know me, and the way I tend to approach things, know that he probably didn't need to worry too much about that particular point.
- Oh, and by the time we're done, you may believe I decided to cover Adrian's original suggestion, anyhow. :-)

Some Specifics of Our Coverage

- Oh yes, while you may know (or quickly notice), while I'm generally a Unix-oriented person, I'll try to remember to include at least a little Windows-related stuff in this talk, too. :-)
- I'll also tell you right up front that while I may mention specific products, those product references are meant by way of example, and should not be taken as denigrating equally capable alternative products (I just don't have room to mention every product in a talk of this length).
- Finally, I'm assuming you're only working domestically, so we won't go into export control issues that may come up when we talk about crypto-related stuff, although I will note that a growing number of American colleges and universities have overseas branch campuses, etc.

So How Does This Talk Fit Into Internet2's Overall Security Agenda?

- Because I serve as Internet2's Security Program Manager under contract through the University of Oregon, I try to make sure that the topics I cover when speaking to the community fit into Internet2's overall strategic plan, and into the security topics that we've identified for attention (see: www.uoregon.edu/~joe/security-tasks.pdf)
If you check out that list of security-related tasks, you'll see that item six is "Replacing traditional passwords" and item three is "Phishing."
- I'm always happy when I color within the lines. :-)
- Before we dive into today's topic in depth, however, let me just give you today's one slide sound-bite-length takeaway message...

Today's Talk's Sound-Byte-Length Message

- Traditional passwords are insufficiently secure. The time has come for you to replace traditional passwords with hardware crypto tokens (or some other more-secure-than-traditional-passwords authentication mechanism).
- So we've got an hour or so to talk a little about why I feel that way, and why I think you should, too. I also want to make sure I leave some time for questions and discussion at the end...

Some Authentication Options

- When it comes to authentication options, conventional options include:
 - something you know (such as a password)
 - something you have (such as a hardware crypto token)
 - something you are (biometrics, such as retinal scans)
- We might also consider “authentication architectures” where “what you can access” is controlled by “where you are,” whether that’s via an nnrp.access file in INN (limiting access by IP address range), or a closed point-to-point circuit-based network architecture.
- In reality, however, at least today, authentication is normally all about passwords.

Passwords Are Ubiquitous

- It is hard to think of an online service that doesn't use traditional passwords of one sort or another...
 - workstation, server, network device and mobile device logins including wireless auth, VPNs, etc.
 - networked applications such as your email (and instant messaging, and calendaring, and...)
 - many web sites (such as Amazon, eBay, Facebook, etc.)
 - campus course management systems (e.g., Blackboard)
 - campus administrative systems (with FERPA data)
 - online financial accounts (with GLB data)
 - medical and insurance-related sites (with HIPAA data)
 - etc., etc., etc.
- We truly use passwords everywhere.

So That Means All of Us Must Trust Passwords, Right?

- Quick informal poll: How many of you do trust passwords?

Put another way, knowing the sensitive assets that passwords protect, do you feel secure that passwords provide adequate protection for those resources and information assets?

-- Yes?

-- No?

-- Gee, Joe, I wish you hadn't asked that question!

- If some of us don't completely trust passwords, that may point to a problem, because...

Passwords Are Truly Critical to IT Security

- If one or more of your passwords are compromised:
 - confidential materials may be accessed or disclosed (resulting in you being sued/fired/arrested)
 - critical files may be surreptitiously modified or deleted, (including potentially irreplaceable data)
 - you may be denied access to your own resources (e.g., if the bad guys decide to “lock you out”)
 - your personal or institutional reputation may be damaged (for example if spam is sent from your account, your college may end up being blocklisted)
 - miscreants may take your money or even co-opt your identity
- I think passwords play a **critical** security role, so if we’re going to rely on them, then they’d **BETTER** be trustworthy

Thought Experiment: Would You Trust (Just) Traditional Passwords To Secure Access To...

- Industrial control systems ("SCADA"), including life safety-critical things? Apparently many people do, at least for things like transportation facilities, power plants and energy transmission facilities, chemical factories, etc.
- How about sensitive national security information, such as confidential diplomatic, defense, law enforcement, or intelligence-related information?
- In fact, since I mentioned defense systems, let's play "war games" for a minute: would you trust a password to adequately secure a thermonuclear weapon and its delivery system?
- I sure wouldn't, and apparently the Air Force didn't, either, although not for the reasons you might think. 11

Nuclear Weapons and “Passwords”

- Keeping Presidents in the Nuclear Dark

(Episode #1: The Case of the Missing “Permissive Action Links”)

Bruce G. Blair, Ph.D, CDI President, bblair@cdi.org

Feb. 11, 2004

Last month I asked Robert McNamara, the secretary of defense during the Kennedy and Johnson administrations, what he believed back in the 1960s was the status of technical locks on the Minuteman intercontinental missiles. [...] McNamara replied, in his trade-mark, assertively confident manner that he personally saw to it that these special locks (known to wonks as “Permissive Action Links”) were installed on the Minuteman force, and that he regarded them as essential to strict central control and preventing unauthorized launch.

When the history of the nuclear cold war is finally comprehensively written, this McNamara vignette will be one of a long litany of items pointing to the ignorance of presidents and defense secretaries and other nuclear security officials about the true state of nuclear affairs during their time in the saddle. What I then told McNamara about his vitally important locks elicited this response: “I am shocked, absolutely shocked and outraged. Who the hell authorized that?” What he had just learned from me was that the locks had been installed, but everyone knew the combination.

The Strategic Air Command (SAC) in Omaha quietly decided to set the “locks” to all zeros in order to circumvent this safeguard. During the early to mid-1970s, during my stint as a Minuteman launch officer, they still had not been changed. Our launch checklist in fact instructed us, the firing crew, to double-check the locking panel in our underground launch bunker to ensure that no digits other than zero had been inadvertently dialed into the panel. SAC remained far less concerned about unauthorized launches than about the potential of these safeguards to interfere with the implementation of wartime launch orders. And so the “secret unlock code” during the height of the nuclear crises of the Cold War remained constant at 00000000. [source: <http://www.cdi.org/blair/permissive-action-links.cfm>]

- [And if you want something “better” than PALs, how about this one: “British Nukes Were Protected With Bike Locks,” <http://news.bbc.co.uk/2/hi/programmes/newsnight/7097101.stm>]

Managing Risk

- This is the point where someone probably will speak up and say something like:

“But Joe!!! Security is really *all about managing risk!* Sure, passwords aren’t perfect, but in our case we’re not trying to secure nuclear weapons, we’re just trying to keep email accounts from sending spam or kids from logging in and changing their grades! Passwords are secure enough for that sort of thing!”

- Maybe, maybe not -- that’s what we’ll spend some time talking about today. Maybe by the time we’re done, I will have been able to change your mind.

Cost/Benefit Analysis

- This is also the point where other people may chime in:

“BUT JOE! I’d *love* to replace traditional passwords with something stronger/cooler, but we just can’t *AFFORD* it! When we do a cost/benefit analysis, the numbers just *DON’T* work!”

- We’ll also talk a little about *that* argument a little later in this talk.

The Insecurity of Passwords: Let Me Count The Ways I Love Thee

1. People Will Pick Weak Passwords

- We know from personal experience that if users are given the chance to do so, they'll routinely pick "weak" passwords, including:
 - password == username
 - trivially short (6 character or less) passwords
 - passwords that use only lower case letters
 - passwords that are words in the dictionary
 - passwords that are a pattern on the keyboard (12345678, qwerty, etc.)
- Why are weak passwords a problem? Well, weak passwords can be successfully attacked with either brute force attacks (password = a, b, c, d, ... z, aa, ab, ac, ad, ... az, ba, ... zzzzz), or via dictionary attacks (try all words found in the dictionary as potential passwords)

You Can (Try To) Force Users to Pick Strong Passwords

- For example, you can use modules such as pam_passwdqc to do password length and quality enforcement, see <http://www.openwall.com/passwdqc/>
- Users WILL complain if you impose sufficiently fascist password quality requirements, and visits to the helpdesk will go up if users find it "impossible" to pick a password that will "pass muster." [maybe try password generators?]
- Users may also employ "alternative channels" (such as manual password changes by privileged administrators) to overcome password policies which they don't "buy into."
- Recommendation: system admins and/or security admins (with prior mgmt approval!) should audit user passwords using commonly available password cracking tools...

Password Cracking Tools

- There are many password cracking tools in circulation, both “open source” and commercial.
- I never like to help the script kiddies by providing pointers, but note that these tools aren’t exactly “obscure.” See, for example, the results of Googling for password cracking tools

including

<http://sectools.org/crackers.html>

A Sample Tool From the Sectools.org Listing



The screenshot shows a web browser window with the title "John the Ripper password cracker". The address bar displays the URL <http://www.openwall.com/john/>. The page content includes the Openwall Project logo, a navigation menu with links like "/home", "Owl", "JtR", "Pro", "crypt", "pam", "passwdqc", "tcb", "phpass", "scanlogd", "popa3d", "msulogin", "Linux", "BIND", "advisories", "presentations", "services", "donations", "wordlists", "passwords", "news", "community", "lists", "wiki", "CVSweb", "mirrors", and "signatures", and a main heading "John the Ripper password cracker". Below the heading is a paragraph about the tool's purpose and supported platforms, followed by a link to "Order Openwall Wordlists CD (20+ languages) with delivery worldwide or download". Further down, there is information about the free Open Source version and links to the "John the Ripper Pro" homepage for Linux and Mac OS X. At the bottom, there are download links for the free version.

John the Ripper password cracker

<http://www.openwall.com/john/>

[Openwall Project](#)
bringing security into open environments

/home Owl JtR Pro crypt pam passwdqc tcb phpass scanlogd popa3d msulogin / Linux BIND / advisories presentations / services donations / wordlists passwords / news community lists wiki CVSweb mirrors signatures

John the Ripper password cracker

John the Ripper is a fast password cracker, currently available for many flavors of Unix, Windows, DOS, BeOS, and OpenVMS. Its primary purpose is to detect weak Unix passwords. Besides several crypt(3) password hash types most commonly found on various Unix systems, supported out of the box are Windows *LM hashes*, plus *many* more with contributed patches.

[Order Openwall Wordlists CD \(20+ languages\) with delivery worldwide or download](#)

John the Ripper is free and Open Source software, distributed primarily in source code form. If you would rather use a commercial product tailored for your specific operating system, please consider [John the Ripper Pro](#), which is distributed primarily in the form of "native" packages for the target operating systems and in general is meant to be easier to install and use while delivering optimal performance.

Proceed to John the Ripper *Pro* homepage for your OS:

- [John the Ripper Pro for Linux](#)
- [John the Ripper Pro for Mac OS X](#)

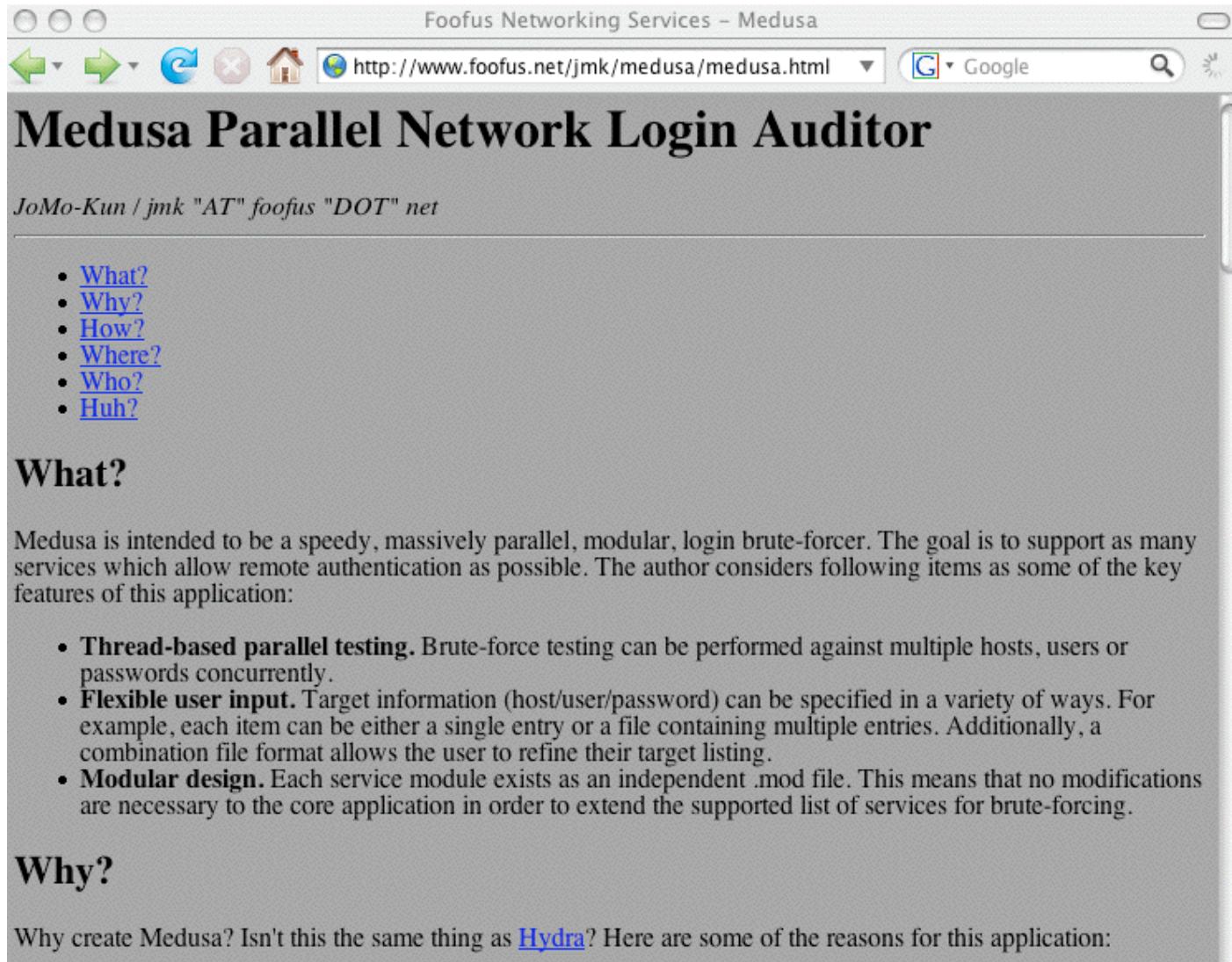
Download one of the latest free versions:

- [John the Ripper 1.7.3.4 \(Unix - sources, tar.gz, 798 KB\)](#) and its [signature](#)
- [John the Ripper 1.7.3.4 \(Irix - sources, tar.gz, 642 KB\)](#) and its [signature](#)

One Password Cracking Approach Particularly Worth Highlighting: Rainbow Table Methods

- Most computational algorithms, including password cracking algorithms, involve tradeoffs between time and space, e.g., I can use more CPU, or more RAM or disk storage, depending on how (algorithmically) I decide to attack a problem. Rainbow tables use precomputed stored hash chains (e.g., more storage) to dramatically reduce the CPU/clock time required to crack password hashes.
- For more information on Rainbow Tables, see:
<http://project-rainbowcrack.com/> and
<http://project-rainbowcrack.com/tutorial.htm> and
<http://www.ethicalhacker.net/content/view/94/24/>

A Sample Open Source In-Situ Brute Force Network “Auditing” Tool



The screenshot shows a web browser window titled "Foofus Networking Services – Medusa". The address bar displays the URL <http://www.foofus.net/jmk/medusa/medusa.html>. The main content area features a large title "Medusa Parallel Network Login Auditor" and a subtitle "JoMo-Kun / jmk "AT" foofus "DOT" net". Below the subtitle is a horizontal line and a bulleted list of questions: "• [What?](#)" [Why?](#) [How?](#) [Where?](#) [Who?](#) [Huh?](#)". The page content then branches into sections for "What?", "Why?", and "How?".

What?

Medusa is intended to be a speedy, massively parallel, modular, login brute-forcer. The goal is to support as many services which allow remote authentication as possible. The author considers following items as some of the key features of this application:

- **Thread-based parallel testing.** Brute-force testing can be performed against multiple hosts, users or passwords concurrently.
- **Flexible user input.** Target information (host/user/password) can be specified in a variety of ways. For example, each item can be either a single entry or a file containing multiple entries. Additionally, a combination file format allows the user to refine their target listing.
- **Modular design.** Each service module exists as an independent .mod file. This means that no modifications are necessary to the core application in order to extend the supported list of services for brute-forcing.

Why?

Why create Medusa? Isn't this the same thing as [Hydra](#)? Here are some of the reasons for this application:

Can "Brute Forcers" Accidentally (or Intentionally) DDoS Your Users?

The screenshot shows a Microsoft TechNet page titled "Domain Policy Settings". The URL in the address bar is <http://technet.microsoft.com/en-us/library/cc264456.aspx>. The page content discusses account lockout policies. A red box highlights a warning about the "Reset account lockout counter after" setting.

environment includes multiple versions of Windows, you will need to monitor for event IDs specific to each version, such as event ID 539.)

Reset account lockout counter after

This policy setting determines the length of time before the **Account lockout threshold** setting resets to zero. The default value for this policy setting is **Not Defined**. If the **Account lockout threshold** setting is defined, this reset time must be less than or equal to the value for the **Account lockout duration** setting.

If you leave this policy setting at its default value or configure the value to an interval that is too long, this may make your environment vulnerable to a DoS attack. An attacker could maliciously perform a number of failed logon attempts on all users in the organization, which will lock out their accounts as described earlier in this appendix. If no policy is determined to reset the account lockout, this is a manual task for administrators. Conversely, if a reasonable time value is configured for this policy setting, users are locked out for a set period until all of the accounts are unlocked automatically.

The recommended setting value of 15 minutes was determined as a reasonable amount of time that users are likely to accept, which should

More In-Situ Brute Forcing Points To Consider

- Are you limiting the number of password attempts you allow? For example, after three failed logins do you stop listening to additional login attempts from a source IP, but *only for a period of time?* [see preceding slide!]
- Do you log (and analyze!) all authentication failures?
- If you use the same authentication service for multiple network applications, do you track/limit/log login failures from ALL of them? Or are you only protecting one service (such as sshd) while allowing someone to flood "more obscure" services (such as POP3 or IMAP) with query traffic w/o those attacks being detected and handled?
- Moving services off their default ports, and/or using port knocking can help cut down on the network noise; if you'd like to read more about port knocking, see...

portknocking.org

The screenshot shows a web browser window displaying the homepage of portknocking.org. The page has a blue header bar with the title "PORTKNOCKING - A system for stealthy authentication across closed ports. : ABOUT : summary". Below the header is a navigation menu with links like "about", "firewall primer", "details", "knock lab", "download", "implementations", "documentation", and "FAQ". To the right of the menu is a main content area. On the left side of the content area is a sidebar with sections for "images", "resources", and "contact". The main content includes a "DEFINITION" section describing port knocking as a method of host-to-host communication using sequences of port numbers. It also features a "IMPLEMENTATIONS" section listing 36 implementations and a "PUBLICATIONS" section with references to academic papers and magazine articles. At the bottom, there's a proposal for implementing port knocking using OpenBSD's pf.

about

firewall primer
details
knock lab
download
implementations
documentation
FAQ

images
resources
contact

LINUX JOURNAL

Sys Admin
The journal for Linux and other systems administrators

PORTRUNNING - A system for stealthy authentication across closed ports. : ABOUT : summary

<http://www.portknocking.org/>

Perl prototype: v0.30

- pcplib support added; daemon no longer requires firewall log file

2004-Nov-14 18:59 | [...more](#)

NEW Net::Pcap support added to sniff packets directly [...more](#)

[summary](#) [features](#) [port forwarding](#) [port triggering](#) [obscurity](#) [critique](#) [requirements](#) [haha](#)

DEFINITION

Broadly, **port knocking** ([PK on wikipedia](#)) is a form of host-to-host communication in which information flows across closed ports. There are various variants of the port knocking method - information may be encoded into a port sequence or a packet-payload. In general, data are transmitted to closed ports and received by a monitoring daemon which intercepts the information without sending a receipt to the sender.

In one instance, **port knocking** refers to a method of communication between two computers (arbitrarily named here *client* and *server*) in which information is encoded, and possibly encrypted, into a sequence of port numbers. This sequence is termed the *knock*. Initially, the *server* presents no open ports to the public and is monitoring all connection attempts. The *client* initiates connection

IMPLEMENTATIONS

Catalogue has **36** implementations of port knocking (and related methods) as of 2009-Nov-03 11:22. [[download xml](#)]

PUBLICATIONS

When citing port knocking, please use

Christan Borss (2001) Listserv post to Braunschweiger Linux User Group (lug-bs@lk.etc.tu-bs.de) ([read](#)).

Barham P et al (2002) *Techniques for Lightweight Concealment and Authentication in IP Networks*. Intel Research Berkeley ([IRB-TR-02-009](#)).

Krzywinski, M (2003) *Port Knocking: Network Authentication Across Closed Ports*. SysAdmin Magazine **12**: 12-17.

[...more references](#)

Proposal to [implement port knocking using OpenBSD's pf](#). The writer states,

A Comercial Parallel GPU-Enabled Password Recovery Product From A Russian Company

ElcomSoft Distributed Password Recovery : High-performance distributed password recovery with NVIDIA GPU acceleration

http://www.elcomsoft.com/edpr.html

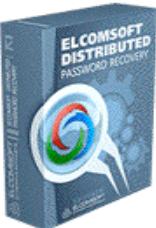
Google

ELCOMSOFT PROACTIVE SOFTWARE

HOME PRODUCTS DOWNLOADS PURCHASE SUPPORT PARTNERS PRESS ROOM ABOUT US PROMOTIONS US English

PASSWORD RECOVERY SOFTWARE

ElcomSoft Distributed Password Recovery



High-Performance Distributed Password Recovery

Break complex passwords, recover strong encryption keys and unlock documents in a production environment. Elcomsoft Distributed Password Recovery is a high-end solution for forensic and government agencies, data recovery and password recovery services and corporate users with multiple networked workstations connected over a LAN or the Internet. Featuring unique acceleration technologies and providing linear scalability with no overhead, Elcomsoft Distributed Password Recovery offers the fastest password recovery by a huge margin, and is the most technologically advanced password recovery product currently available.



Features and Benefits

- ★ NVIDIA GPU acceleration (patent pending) reduces password recovery time by a factor of 50
- ★ Linear scalability with no overhead allows using up to 10,000 workstations without performance drop-off
- ★ Allows up to 64 CPUs or CPU cores and up to 32 GPUs per processing node
- ★ Broad compatibility recovers document and system passwords to various file formats ([click for the complete list of formats](#))
- ★ Brute-force and dictionary attacks
- ★ Distributed password recovery over LAN, Internet or both
- ★ Console management for flexible control from any networked PC

Prices:

Up to 20 clients	- \$599
Up to 100 clients	- \$1,199
Up to 500 clients	- \$2,399
Up to 2500 clients	- \$4,999
2500+ clients	- contact us

[Purchase EDPR](#)

[Download EDPR 2.90.215 server/console and agent](#)

[Download EDPR agent only](#)

[View the screenshot of EDPR](#)

[Read EDPR Online Documentation](#)

[GPU Acceleration Frequently Asked Questions](#)

[\[Whitepaper\] Size does matter. Advantages of distributed password recovery.](#)

[Subscribe to the Password Recovery Software newsletter](#)

Shadow Password Files

- Of course, offline password cracking tools require password hashes as input, and these days most operating systems store password hashes as shadow files, accessible only by privileged users.
- Shadow files are still of great interest to crackers, being among the top 10 targets seen by the Honeynet Project (see www.honeynet.org/book/export/html/14)
- Hmm... what might a cracker do? Well, let's assume he/she manages to obtain access to your backup server, or to your unencrypted backup media, including a copy of /etc/shadow ... Suddenly, (s)he's got plenty of fodder to feed to his/her favorite password cracking tool(s).
- Now you know one reason why I tend to harp on the importance of encrypting and/or controlling access to your backups!

Passwords Encryption Algorithms

- You should insure that you're using a strong password encryption algorithm (assuming your operating system gives you that choice).
- Some systems may still be storing passwords encrypted with the traditional "crypt" based on 56-bit DES encryption. That's pretty bad. You shouldn't use it.
- There are alternative password encryption algorithms available on at least some systems, such as Redhat Linux, including MD5 (identified by the leading \$1\$ in the printable form of the password file), and Blowfish (tagged with \$2\$ or \$2a\$), however I'd recommend SHA256 (\$5\$) or SHA512 (\$6\$), which may be the default on (some) distros. See people.redhat.com/drepper/SHA-crypt.txt and kbase.redhat.com/faq/docs/DOC-15806 and httpd.apache.org/docs/2.2/misc/password_encryptions.html

LAN Manager Password Hashes on Windows

- On many versions of Windows, password hashes are stored in multiple formats: LM hash (for "compatibility"), plus other formats. LM hash format hashes are vulnerable to fast brute force attacks. For example, LMCrack says, "The design goal of LMCrack was to walk a large key space based on a dictionary style attack rather than on a comprehensive brute force attack and to complete the task in under 5 minutes. The result is a program that utilises a database of pre-computed hashes, which can search an effective key space of 3 trillion passwords in less than 60 seconds with an average success rate of 50+%." (ouch)
- See "How to prevent Windows from storing a LAN manager hash of your password in Active Directory and local SAM databases," support.microsoft.com/default.aspx?scid=KB;EN-US;q299656& (or tinyurl.com/no-lan-man-hashes)

Windows Syskey

- “How to use the SysKey utility to secure the Windows Security Accounts Manager database”

<http://support.microsoft.com/kb/310105/>

“The Microsoft Windows 2000, Microsoft Windows XP, and Microsoft Windows 2003 Security Accounts Management Database (SAM) stores hashed copies of user passwords. This database is encrypted with a locally stored system key. To keep the SAM database secure, Windows requires that the password hashes are encrypted. Windows prevents the use of stored, unencrypted password hashes.

“You can use the SysKey utility to additionally secure the SAM database by moving the SAM database encryption key off the Windows-based computer. The SysKey utility can also be used to configure a start-up password that must be entered to decrypt the system key so that Windows can access the SAM database. This article describes how to use the SysKey utility to secure the Windows SAM database.”

Sometimes People Don't Even Bother Changing Default Device Passwords

- Not changing default/well known passwords is an extreme example of “picking bad passwords”, but it is still all too common.
- You may or may not be aware that lists of default/well known passwords for particular devices are in widespread circulation -- if you didn’t know this, you do now.
- It is absolutely critical that any or all default accounts/passwords get changed (or disabled) when devices are put on the wire.

One Example of A Default Password List

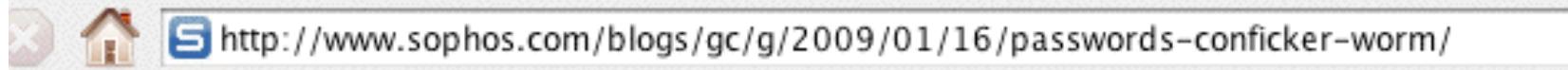
Default Password List

The screenshot shows a web browser window with the title "Default Password List". The address bar displays the URL "http://www.phenoelit-us.org/dpl/dpl.html". The main content is a table with the following columns: Brand, Model/Type, Port/Protocol, Username, and Notes. The table contains numerous entries for brands like Adtran, Alcatel, and Allied Telesyn, listing various routers and switches with their respective default credentials and notes.

Brand	Model/Type	Port/Protocol	Username	Notes
adtran	T3SU 1Q/2G 1G	Telnet	n/a	(none)
adtran	Express 5110/5200/5210	Telnet	n/a	adtran
adtran	Agent Card	Telnet	n/a	ADTRAN
adtran	TSU Router Module/L128/L768/1.5	Telnet	n/a	(none)
adtran	T3SU 300	Telnet	n/a	adtran
Alcatel	PBX	4400	Port 2533	kermit
Alcatel	PBX	4400	Port 2533	dhs3mt
Alcatel	PBX	4400	Port 2533	at4400
Alcatel	PBX	4400	Port 2533	mtch
Alcatel	PBX	4400	Port 2533	mtcl
Alcatel	PBX	4400	Port 2533	root
Alcatel	PBX	4400	Port 2533	dhs3pms
Alcatel	PBX	4400	Port 2533	adfexc
Alcatel	PBX	4400	Port 2533	client
Alcatel	PBX	4400	Port 2533	install
Alcatel	PBX	4400	Port 2533	halt
Alcatel	Office 4200		Multi	n/a
Alcatel	OmniStack 6024		Telnet	admin
Alcatel	Omnistack/Omniswitch		Telnet/Console	diag
Alcatel	Omnistack/omniswitch		Telnet	diag
Alcatel	Timestep VPN 1520	3.00.026	Permit config and console	root
Alcatel	OXO	1.3	Multi	(none)
Allied	Telesyn		Multi	manager
Allied Telesyn	AT-8024(GB)		Console	n/a
Allied Telesyn	AT-8024(GB)		HTTP	manager
Allied Telesyn	AT Router		HTTP	root

Some Malware Includes Password Attack Code Targeting Poor Password Choices

Passwords used by the Conficker worm | Graham Cluley's blo



Passwords used by the Conficker worm

It's not possible to emphasise enough the importance of using sensible passwords on your network.

Not just on the areas of your network that you don't want your users to traipse through, but also on the default network shares that are present on installations of commonly used operating systems like Windows NT/2000/XP/2003.

One of the ways in which the Conficker worm (also known as Confick or Downadup) uses to spread is to try and batter its way into ADMIN\$ shares using a long list of different passwords.

As you can see in the list below, it relies upon computers using poorly chosen passwords such as dictionary words, "password", "qwerty" or sequences of letters or repeated numbers:

2. People Will Disclose Their Passwords

- Users will disclose their passwords in many different ways.
- For example, phishing exists because people can be “socially engineered,” into revealing their passwords.
- If you have users whom you’ve trained to be cynical, skeptical and defiant, they may (properly) refuse to reveal their passwords when receiving phishing attacks. Unfortunately, some regions of the country (e.g., the Midwest), and some groups (including higher education, unfortunately) have cultures which reward trust and unquestioning compliance when confronted with authoritatively presented demands:
Phisher: “Tell me your password immediately!”
User: “Okay, okay! It’s LetMeIn123, don’t ‘disable’ me!”
- Small bribes can also work wonders...

Passwords revealed by sweet deal

More than 70% of people would reveal their computer password in exchange for a bar of chocolate, a survey has found.

It also showed that 34% of respondents volunteered their password when asked without even needing to be bribed.

A second survey found that 79% of people unwittingly gave away information that could be used to steal their identity when questioned.

Security firms predict that the lax security practices will fuel a British boom in online identity theft.

Security shock

The survey on passwords was carried out for the Infosecurity Europe trade show due to take place at Olympia in London from 27-29 April.

The survey data was gathered by questioning commuters passing through Liverpool Street station in London and found that many were happy to share login and password information with those carrying out the research.

As well as people simply telling the questioners their passwords or saying they would hand them over in exchange for some confectionery, a further 34% revealed the word or phrase they used when asked if it had anything to do with a pet or child's name.

Family names, pets and football teams were all used by those questioned to provide inspiration for a password.

The survey found that, on average, people have to remember four passwords, though one unlucky respondent had to remember 40.

Many adopt very unsafe tactics to remember these login names. Some of those questioned simply use the same password for every system they must log on to.

Those that used several passwords often wrote them down and hid them in a desk or in a document on their computer.

Almost all of those questioned, 80%, said they were fed up with passwords and would like a better way to login to work computer systems.

Source: <http://news.bbc.co.uk/2/hi/technology/3639679.stm>

Sharing Passwords

- Sometimes users will “voluntarily” share their password with others (even without chocolate!)... For example:
 - An executive delegates triage of her email to her administrative assistant; sometimes the administrative assistant is out sick or on vacation, so the executive’s password is also shared with the backup admin assistant, and of course the password’s never changed...
 - Supervisors may demand to know the password of subordinates accounts so that they can “access critical files” the subordinate may be working on when the subordinate is sick or out on vacation.
 - Spouses often will routinely trust each other with their passwords, even though they shouldn’t (sorry, honey, we’ve been married 25 yrs but you simply don’t need to know!)

Writing Passwords Down

- If password requirements are sufficiently complex, users have little choice but to record them if they're to have a hope of remembering all them.
- The yellow sticky note on the side of the monitor is the stereotypical password repository site (although others may favor the top drawer of their desk).
- This presents obvious risks if you have visitors (and even if you don't think you have visitors, you probably at least have custodial staff occasionally servicing your office)
- Carrying ones passwords in one's wallet is probably at least marginally more secure, although it isn't ideal. (You can read one person's take on the wallet as an option at <http://askthegeek.kennyhart.com/index.php/2009/02/04/why-your-wallet-is-the-best-password-manager/>)

Sharing Passwords With Any/All Users of Their Computer

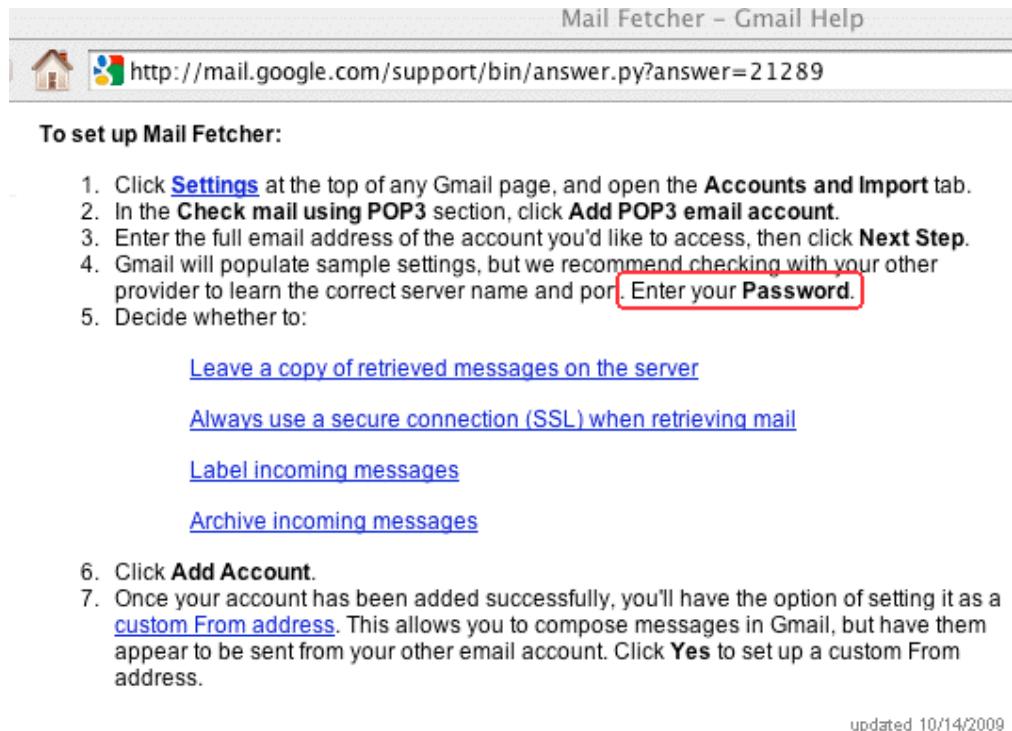
- We all know people who routinely save their passwords in applications such as email clients, so that anyone who physically is at their computer can automatically “login” as them just by sitting down at their computer and starting the application with the saved password.
- Requiring a username and password to login to the computer, and/or consistently employing a locking screen saver, and or/or using RF proximity cards (such as the “walk away security” cards from Xyloc) , may help control that exposure, but saving passwords in apps is still an evil practice, and one that causes a lot of forgotten password issues (“I can’t remember what I set my password to, I just saved whatever it was in my client”).

Another Form of Password Sharing: Password Reuse on Multiple Sites

- Other users will share passwords across multiple systems, using the same password for critical, highly sensitive administrative systems, and for their online bridge club account (and for <fill in the blank>). If I can crack their password on any of those systems, I'll be able to get into ALL of their accounts, including the highly sensitive ones.
- Sometimes users will even give login information, including passwords, to other sites to save and routinely re-use.

Classic example of this is email POP consolidation, where a free web email account may offer to use your credentials to periodically fetch other email by logging in to a remote system as you (I kid you not). Obviously, unless it is going to prompt you for your password each time it does this, it needs to know/save your unhashed (raw) password.³⁸

Example of POP Email Consolidation: Gmail



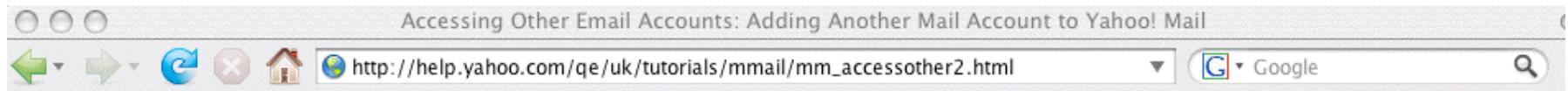
The screenshot shows a web browser window titled "Mail Fetcher – Gmail Help". The address bar displays the URL <http://mail.google.com/support/bin/answer.py?answer=21289>. The page content is a guide for setting up Mail Fetcher in Gmail:

To set up Mail Fetcher:

1. Click [Settings](#) at the top of any Gmail page, and open the [Accounts and Import](#) tab.
2. In the [Check mail using POP3](#) section, click [Add POP3 email account](#).
3. Enter the full email address of the account you'd like to access, then click [Next Step](#).
4. Gmail will populate sample settings, but we recommend checking with your other provider to learn the correct server name and port. [Enter your Password](#).
5. Decide whether to:
 - [Leave a copy of retrieved messages on the server](#)
 - [Always use a secure connection \(SSL\) when retrieving mail](#)
 - [Label incoming messages](#)
 - [Archive incoming messages](#)
6. Click [Add Account](#).
7. Once your account has been added successfully, you'll have the option of setting it as a [custom From address](#). This allows you to compose messages in Gmail, but have them appear to be sent from your other email account. Click [Yes](#) to set up a custom From address.

updated 10/14/2009

Example of POP Email Consolidation: Yahoo



9. In Add Account Step 3, if you want Yahoo! Mail to retrieve messages from the external account, provide the information that follows, then click the **Setup Mail Server** button. Otherwise, click the **Skip This Step** button.
- **Mail server**—the remote server where your external account is stored, including that account's email messages.
 - **Username**—the one you use to access your other account, typically the part of your email address before the @ symbol.
 - **Password**—the one you use to access your other account.
 - **Indicator**—the colour Yahoo! Mail uses to mark messages that you receive from this account.

Step 3 Setup Mail Server

This step is optional. You can enter this information later on the Add Account screen.

Please enter the mail server, username and password for this account. Entering this information will enable you to receive mail from this account.

Mail Server: Only POP mail servers are supported at this time.

Username: Note: The username is different than your email address before the at sign (@).

Password: Your password is not displayed for security.

Indicator: This color will be used to mark messages.

“Rubber Hose Cryptography”

- We must also recognize that if someone wants your password badly enough, they may be willing to use extreme physical measures, such as torture, to get it. This is sometimes sardonically referred to as “rubber hose cryptography.”
- A (somewhat) more civilized version of “rubber hose cryptography” would be judicially mandated password production (presumably under threat of contempt of court).

See, for example, the case of Sebastien Boucher...

Court: self-incrimination privilege won't protect password

The privilege against self-incrimination, a federal court has ruled, does not bar prosecutors from forcing a defendant in a child pornography case to decrypt his laptop hard drive—reversing a 2007 decision that found the demand to enter a password equivalent to compelled testimony.

By Julian Sanchez | Last updated March 2, 2009 9:30 AM CT

A federal district court in Vermont has ruled that the Fifth Amendment right against self-incrimination does not bar the government from requiring Sebastien Boucher, who faces charges of possessing child pornography, to decrypt his laptop hard drive. A lower court had previously quashed a subpoena compelling Boucher to enter his password, reasoning that this was tantamount to requiring a defendant to testify against himself.

Boucher, a Canadian citizen who legally resides in the U.S., was stopped while returning to the country in 2006. Immigration officials searched his laptop at the border, and found thousands of image files that a border agent judged, on the basis of their file names, to be probable adult and child pornography. After viewing several images, border guards seized the laptop and shut it down—at which point Boucher's Pretty Good Privacy encryption kicked in, locking down the Z drive on which the files were contained.

Federal prosecutors initially sought a grand jury subpoena ordering Boucher to provide the password that would allow them to decrypt his hard drive. They later amended their request—presumably in hopes of avoiding Fifth Amendment concerns—clarifying that they would ask Boucher to enter the password himself in front of the grand jury. Despite this, a magistrate judge ruled in 2007 that the act of entering the password, even if the password itself was not disclosed to the government, was "testimonial" and therefore could not be compelled without offending the Fifth Amendment.

The author of the 2007 opinion, Judge Jerome Niedermeier, distinguished the order to enter the password from the superficially similar requirement that a defendant produce the key to a locked safe on the grounds that asking for the password was a demand that Boucher reveal the contents of his mind. Even though the password itself might not be incriminating, either disclosing or entering it would entail "implicit statements of fact, such as admitting that evidence exists, is authentic, or is within a suspect's control."

See <http://arstechnica.com/tech-policy/news/2009/03/court-self-incrimination-privilege-stops-with-passwords.ars>

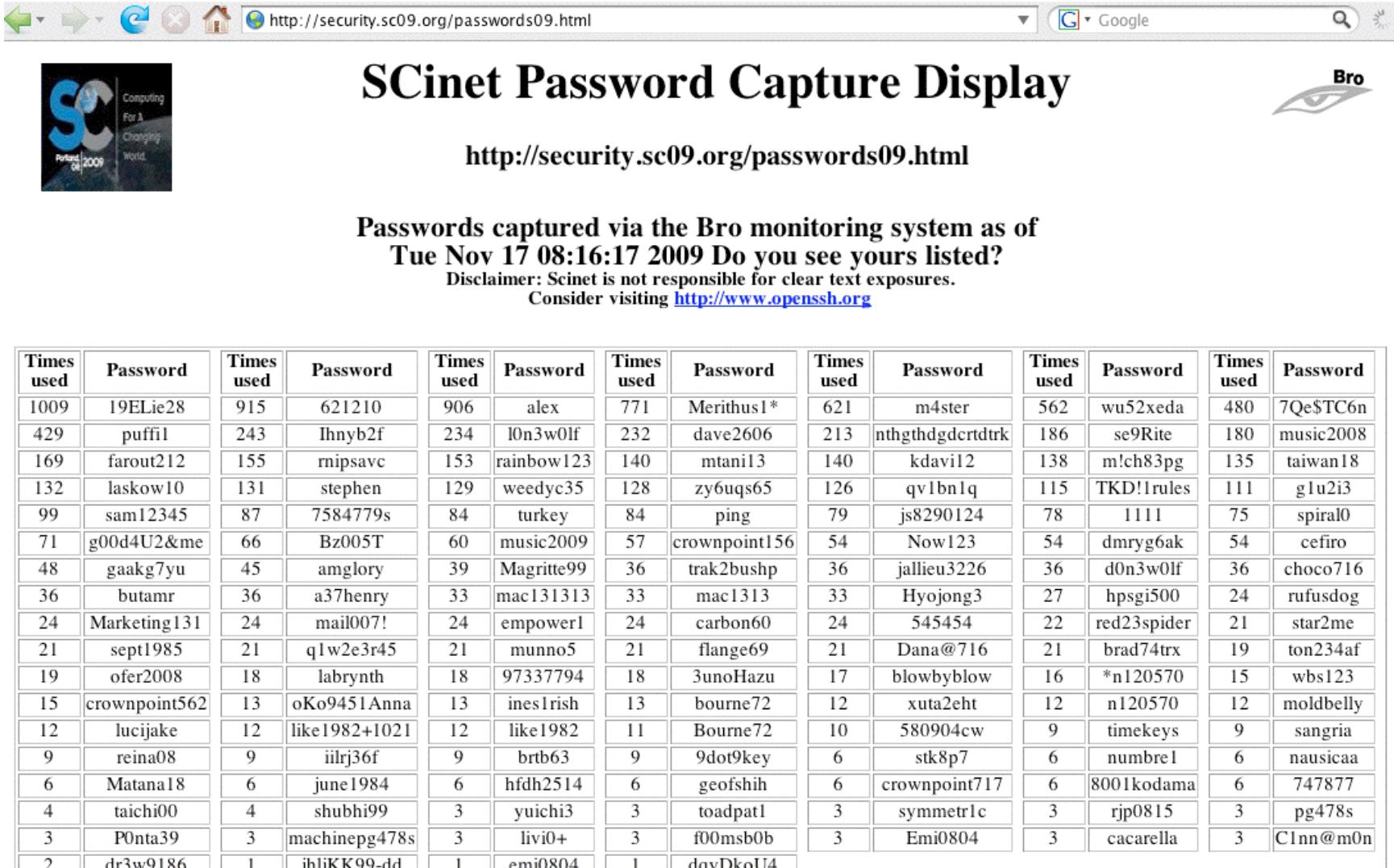
Password “Safes”

- Some users, overwhelmed by password proliferation, don't even attempt to memorize all the passwords they need to juggle, they simply store them in an online encrypted password “safe” or password “wallet.” One example of such a product is KeePass (see <http://keepass.info/>).
- While this is convenient, obviously this is a case of putting all one's eggs in one's basket. Should the password safe or password wallet be compromised, you're likely to experience severe problems.
- I would be particularly wary of password safes which tightly integrate with browsers or other applications; if you use a password safe, I'd suggest using one that merely displays your username and password, at which point you can manually enter those values into applications as you deem appropriate.

3. Passwords Will Be Sniffed

- In addition to people picking weak passwords, and sharing those passwords in various ways, passwords will also be captured, or “sniffed.”
- “But Joe, everyone knows that passwords should only be sent over encrypted channels! You’d have to be an idiot to send your password in clear text over the wire!”

Passwords From SC2009, In PDX This Week



http://security.sc09.org/passwords09.html

SCinet Password Capture Display

http://security.sc09.org/passwords09.html

**Passwords captured via the Bro monitoring system as of
Tue Nov 17 08:16:17 2009 Do you see yours listed?**

Disclaimer: Scinet is not responsible for clear text exposures.
Consider visiting <http://www.openssh.org>

Times used	Password	Times used	Password	Times used	Password	Times used	Password	Times used	Password	Times used	Password	Times used	Password	Times used	Password
1009	19ELie28	915	621210	906	alex	771	Merithus1*	621	m4ster	562	wu52xeda	480	7Qe\$TC6n		
429	puffi1	243	Ihnyb2f	234	l0n3w0lf	232	dave2606	213	nthgthdgdcrttdrk	186	se9Rite	180	music2008		
169	farout212	155	rniipsavc	153	rainbow123	140	mtani13	140	kdavi12	138	m!ch83pg	135	taiwan18		
132	laskow10	131	stephen	129	weedy35	128	zy6uqs65	126	qv1bn1q	115	TKD!1rules	111	g1u2i3		
99	sam12345	87	7584779s	84	turkey	84	ping	79	js8290124	78	1111	75	spiral0		
71	g00d4U2&me	66	Bz005T	60	music2009	57	crownpoint156	54	Now123	54	dmryg6ak	54	cefiro		
48	gaakg7yu	45	amglory	39	Magritte99	36	trak2bushp	36	jallieu3226	36	d0n3w0lf	36	choco716		
36	butamr	36	a37henry	33	mac131313	33	mac1313	33	Hyojong3	27	hpsgi500	24	rufusdog		
24	Marketing131	24	mail007!	24	empower1	24	carbon60	24	545454	22	red23spider	21	star2me		
21	sept1985	21	q1w2e3r45	21	munno5	21	flange69	21	Dana@716	21	brad74trx	19	ton234af		
19	ofer2008	18	labrynth	18	97337794	18	3unoHazu	17	blowbyblow	16	*n120570	15	wbs123		
15	crownpoint562	13	oKo9451Anna	13	ines1rish	13	bourne72	12	xuta2eht	12	n120570	12	moldbelly		
12	lucijake	12	like1982+1021	12	like1982	11	Bourne72	10	580904cw	9	timekeys	9	sangria		
9	reina08	9	iilrj36f	9	brtb63	9	9dot9key	6	stk8p7	6	numbre1	6	nausicaa		
6	Matana18	6	june1984	6	hfdh2514	6	geofshih	6	crownpoint717	6	8001kodama	6	747877		
4	taichi00	4	shubhi99	3	yuichi3	3	toadpat1	3	symmetrlc	3	rjp0815	3	pg478s		
3	P0nta39	3	machinepg478s	3	livi0+	3	f00msb0b	3	Emi0804	3	cacarella	3	C1nn@m0n		
2	dr3w9186	1	jhkjKK99-dd	1	emi0804	1	dqyDkoU4								

And If The Smart Guys From SC Screw Up...

- SO WILL YOUR (EVEN SMARTER) USERS.

I promise, your users WILL send their passwords over unencrypted links.

- I'm also willing to bet that at least some of you enable or facilitate those mistakes by permitting plain text password-using daemons (such as telnet, ftp, etc.) to continue to run on your local systems, or by allowing unencrypted web login pages.
- You need to go on a personal crusade to stomp out any and all remaining plain text-using protocols running on your campus network.
- FTP (as used by many web page development and publishing tools) is a particularly probable source of exposure.

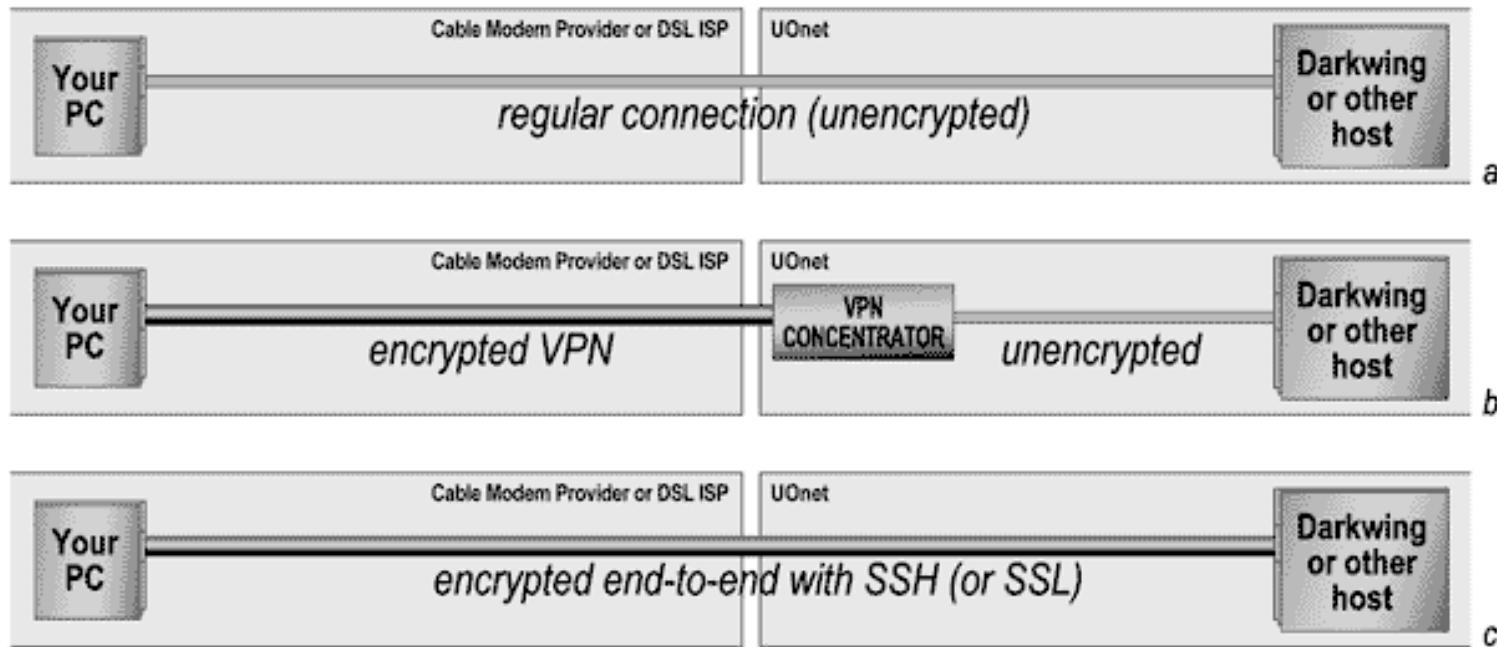
“BUT JOE! We’re 100% switched! We’re Safe!”

- From time-to-time I run into people who believe that they don't have any sniffing exposure because their network architecture is 100% switched, and in such an architecture network traffic shouldn't be visible to eavesdroppers (the way it would be on a shared network link).
- I would suggest that between arp spoofing, mac flooding, and mac duplication (among other methods), there's a good chance that a bad guy or bad gal can still arrange to see network traffic even in a fully switched environment.
- For those who insist on details, see for example
<http://monkey.org/~dugsong/dsniff/> or
<http://www.oxid.it/cain.html>
- Switched networks do NOT provide sufficient protection against network traffic sniffing

“BUT JOE! We have a VPN!”

- Another technology that's often trotted out as a solution to the problem of sniffing is the use of VPNs.
- Virtual private networks provide an encrypted tunnel between the end user's workstation and the VPN concentrator. As far as they go, they're fine, *they just don't go far enough*: VPN's are NOT “end-to-end secure,” they're only “one-end-to-VPN-concentrator secure”, and a bad guy can still attempt to sniff the VPN'd traffic after it exits the VPN in clear text.
- Don't get me wrong, VPNs can help reduce the traffic exposure problem, and if you've got one, I'd certainly encourage you to still use it, you just need to recognize that you still have traffic that's exposed.
- VPNs are not (complete) protection against sniffing.

VPN vs. End-to-End Encryption



“BUT JOE! We *DO* Encrypt End-To-End!”

- Excellent! I’m delighted to hear that you’re using ssh and SSL/TLS to minimize your exposure to sniffing on the wire!
It is an important step!
- Unfortunately, you’re still not safe from eavesdroppers snarfing your passwords...
- Let me give you just a few examples...

Hardware Keystroke Grabbers

http://www.keyghost.com/USB-Keylogger.htm

We welcome

KeyGhost USB Keylogger
World's first keylogger for Mac and PC USB keyboards.
Simply plug it in and record keystrokes.
Works with 100% of all USB keyboards!

KeyGhost Headlines

KeyGhost USB Keylogger
"Customer satisfaction guaranteed" 12-Month Manufacturers Warranty.
[Order now](#)

High-capacity and compact.
NEW! Plug-style USB KeyGhost devices released.

KeyGhost devices are always designed in consultation with leading law enforcement and government officials.

You can be certain that a KeyGhost product will always work as described.

NATO Classification Information
KeyGhost LTD (NATO supplier) NCAGE Code E1969

The plug-style KeyGhost USB devices look similar to USB Thumb Drives and record all keystrokes typed on any USB keyboard (Mac or PC).



NEW! QIDO - Qwerty to Dvorak USB Adapter.

QIDO
Qwerty-In > Dvorak-Out

Our latest development in human interface technology, the Patent Pending QIDO is a small plug-in device which gives you instant control of your keyboard layout in hardware. Decrease the risk of typing related RSI by quickly learning the

- KeyGhost USB Keyloggers work by recording USB traffic in hardware. There is no software to install to record or retrieve keystrokes on PC or Mac.

- Easy to use, just plug KeyGhost to your keyboard and record all USB keystrokes typed on that keyboard. Can be connected regardless whether PC is turned on or off.

Trojan'd ssh/sshd

- If I can get root on a box you login to, a cracker can install a trojan'd sshd, and having done that, he can then collect username/password pairs at his leisure, even if you consistently use end-to-end encryption.
- Think this doesn't happen?

“The Stakkato Intrusions,”
www.nsc.liu.se/~nixon/stakkato.pdf

Shoulder Surfing and/or Video Cameras Watching Keyboards

- This can be a very low tech attack (as simple as your seat mate watching your fingers while you login to your laptop on an airplane or at a conference), or a very sophisticated high tech attack (perhaps using clandestinely-installed ceiling-mounted miniature wireless cameras focussed on office workstation keyboards).

Either way, the bad guys can still “get” your password.

- While you’re protecting your keystrokes from nosy neighbors, you may also want to consider installing a 3M privacy filter to reduce snooping of what’s shown on your screen. See <http://www.3M.com/PrivacyFilters>

Don't Forget About Malware Targeting Passwords

- Passwords (including online banking credentials!) are a prime target for major malware infestations these days, including:
 - Clampi/Ligats/Ilomo/Rscan
 - Zeus/Zbot/WSNPOEM/NTOS/PRG
 - Koobface
 - Taterf
- For a discussion of my perspective on these and other malware threats and how you might mitigate them, see "Coping With Malware and Other Sorts of Automated Abuse," www.uoregon.edu/~joe/malware/malware.pdf
- Speaking of mitigation, why isn't everyone doing the right thing, and encrypting all their traffic?

"We'd LOVE to Encrypt, But We Just CAN'T!"

- One such argument normally goes something like this...
 - A. We'd love to use end-to-end encryption, but after thinking about it, we just "can't"
 - B. Why not?
 - A. If everyone were to use strong encryption we ourselves couldn't monitor what people do on the network -- all that traffic would be opaque to our monitoring systems!
 - B. But what's the bigger risk: having a bad guy sniffing your passwords, or having an employee who's screwing around checking out his favorite team on ESPN, eh? 55

Simple Network Management Protocol

- SNMP is a mechanism for remotely administering and monitoring network devices such as switches, routers, etc.
- Read (and/or write) access to SNMP managed devices is controlled via SNMP “community strings” (e.g., passwords).
- Because SNMP was developed long ago, and is often used to interact with relatively “simple” network devices, encryption was not a mandatory part of the protocol. Thus SNMP (pre-SNMPv3) community strings are totally vulnerable to sniffing attacks, and unfortunately, deployment of SNMPv3 remains relatively limited.
- *Recommendation:* If you use SNMP, try to make sure it is SNMPv3, use community strings which aren’t “public” and “private,” use a dedicated out-of-band network for SNMP management and monitoring, and block off-site SNMP traffic (port 161 and 162, tcp and udp) at your border.⁵⁶

“Good News! We’ve Encrypted Our Wireless Links To Prevent Password Sniffing”

- Just for the record and as a matter of due diligence, we all know that WEP really doesn’t offer any security against eavesdropping, and that WPA (TKIP) isn’t much better, right? If you’re doing wireless encryption, these days you really need to be doing WPA (AES) or WPA2. (See, e.g., <http://www.smallnetbuilder.com/content/view/24251/100/> and <http://www.networkworld.com/news/2009/082709-new-attack-cracks-common-wi-fi.html>)
- Better yet, why encrypt traffic ONLY on your wireless network? Traffic on ALL network segments is vulnerable to being sniffed, so ALL traffic should be encrypted end-to-end ALL the time! Encrypting JUST wireless links leaves all your other traffic vulnerable to being sniffed!

“Evil Twin” Wireless Nodes

- Not to make you paranoid, but let me also bring up another wireless issue: how do you know that the wireless access point you’re connecting to is the **real** wireless access point you intended to connect to?
- You should be aware that a bad guy could potentially put up an “evil twin” wireless access point -- using the same SSID your production access points normally use -- and in most cases your users wouldn’t be able to tell the difference. If the bad guy can con you into using his node, he may be able to sniff all your traffic, including your passwords, even if your real nodes have been secured.
- 802.1x can help to address some of these issues, but deploying 802.1x is not painless (and is not a “Uncle Bob” or “Aunt Sue”-ish project for home wireless networks)₅₈

4. Passwords Won't Get Changed

- Most sites encourage (read: “require”) users to change their password at least once every $\langle N \rangle$ months, where $\langle N \rangle$ might be as little as one month, or as much as twelve months or even more.
- There is often confusion about the origin and purpose of this periodic password change requirement, sometimes even among security staff, and some frankly view it as a pointless or even counter-productive policy (although folks still go through the motions to at least keep the auditors happy).
- I believe that periodic password changes are useful, and SHOULD be required.

Let me tell you some reasons why.

Periodic Password Changes Limit The Window for Brute Force Attacks

- If you never change your password, an attacker conducting a brute force attack on your password may have a protracted window during which he or she can concentrate on cracking your password, confident that you haven't changed your password during that time.
- If you do change your password, the attacker will need to restart their cracking effort because cracking your old password typically won't help the attacker deduce your new password (unless your current password is derived from your previous password in some ascertainable way).
- The permissible time between changes is thus determined at least in part by the length and complexity of the passwords being used.

Password Changes Can Terminate Unknown Access by Parasitic “Silent Riders”

- Assume you have a user who buys a new computer, and sells their old one, failing to nuke-and-pave its disk before doing so.
- Also assume that the user has configured various applications to take advantage of saved passwords, perhaps for dialup, wireless, or VPN access.
- The purchaser of the old system thus gets not just a system, but also the former owner's network access, and will be able to continue using those saved credentials unless/until the former owner changes their passwords, thereby rendering the saved credentials invalid.
- If password changes don't take place, the parasitic silent rider can continue their unauthorized access forever... 61

Password Changes Make It Harder For Users To Reuse Passwords on Multiple Sites

- Assume that at least some users will be prone toward using the same password on all the sites they frequent.
- Requiring periodic password changes may result in at least modest password diversity (following the change, the user may have the same password on all their sites except for the one that insisted on a password change).
- Of course, there's also the possibility that users will simply update ALL their passwords to the new value that one site may insist they pick, but sometimes you just can't keep people from shooting themselves in the foot. :-;

Password Changes Make It Harder For Users To Hardcode-and-Forget Their Passwords

- If I don't think that I'll ever have to change my password, I may very well just save it in my browser and all my other applications, and then happily forget about it.
- On the other hand, if I know that every three months (or whatever), I'll have to change my password, and I know that I'll need to know my old password to pick a new one, I'm less likely to hardcode-and-forget my passwords.

Users Need to Know HOW They Should Legitimately Be Changing Their Password

- With phishing emerging as a growing problem at many sites, users may be continually presented with bogus requests to “confirm” their current password.
- One step toward hardening users to resist that sort of phishing is insuring that users know how they should be legitimately changing their password.
- Periodic password changes cause users to practice that process so they have an established mental baseline against which to compare illegitimate phishing efforts.
- Routinizing password changes also effectively prevents “ask the admin to change it for me” password changing channels simply because there’d be too many requests (the admins would quickly get tired of doing them!) ⁶⁴

Requiring Password Changes Provide A “Teachable Security Moment”

- Users need ongoing encouragement and education when it comes to system and network security.
- For example, users need reinforcement when it comes to basic security training, including things as basic as:

“Never disclose your password to anyone.”
- Most users, however, will not attend formal training sessions, and may never interact with security staff one-on-one. Things like quarterly password changes at least force users to think (at least a little!) about security at least a few times a year.

Some Final Thoughts on Password Changes

- When you initially assign passwords to new accounts, are those values random (for example, h9KU#rIZ%3a), derived (perhaps the user's DOB), or a constant (ChangeMeNow)? If passwords seldom get changed, I sure hope you're assigning truly random passwords!
- Do you require users to change their passwords when they first login? What if they only use POP, IMAP or some other service that doesn't support compulsory password changes, rather than traditional shell service?
- You may find it interesting to check to see how many users have NEVER changed their password from its initial value (you may also want to check for users who aren't using their accounts at all)

5. Password Administration Is A Huge PITA, Insecure, and Very Expensive, Too

- How much do you spend just trying to create, distribute, manage, reset and terminate accounts? That is, what are you spending on various identity management projects this year?
- If your help desk is like many, a major part of its workload is handling forgotten passwords. How much do you pay to staff that operation?
- If you require staff or students to appear in person with picture identification when password resets are required, what's a half hour of a tenured professor's time worth?
- If you buy a commercial product to try to automate/simplify password resets, what did that cost? Does it work well for you? Is it **really** secure?

Email As a Password Reset Channel

- We're all familiar with sites that allow you to send a one-time password reset link to another email account (which you provided at the time you signed up).
- Does that process feel secure to you? It shouldn't...
- There are many ways that a bad guy could exploit that sort of password reset mechanism to steal your credentials, including:
 - sniffing (unencrypted) email to get a reset link (miss the first one? just ask for it another time...)
 - hijacking email traffic for a site by injecting a bogus DNS MX record (see Dan Kaminsky's "forgot my password" scenario/discussion at slides 24-39 of <http://www.blackhat.com/presentations/bh-dc-09/Kaminsky/BlackHat-DC-09-Kaminsky-DNS-Critical-Infrastructure.pdf>)

“Trivial Pursuit” Passwords Resets

- Other sites allow password resets if the user can successfully regurgitate what I call “personal trivia.”
- Classic examples of “personal trivia” include:
 - date and/or place of birth
 - mother’s maiden name
 - last four digits of your social security number
 - your driver’s license number
 - name of your first employer
 - name of your favorite dog or cat
 - favorite flavor ice cream, etc., etc.
- Unfortunately, a lot of personal trivia answers are
 - (a) easily forgotten; or (b) are a matter of public record due to things like genealogical databases, and social networking sites; or (c) have low information entropy (e.g.: favorite ice cream flavor? Vanilla’s a safe bet) ⁶⁹

A Biometric Approach to Password Resets



products overview
address change
survey
locator
-> password reset

vPortal

Password Reset

"I forgot my password." Unfortunately, this statement costs many IT departments up to \$30 per user per year. As long as passwords are required to access systems like voicemail and email, password reset requests will continue to plague enterprise help desks. In fact, the problem is so pervasive that an estimated 30% of all incoming calls to the average enterprise help desk are password related.

Secure Self-Service is the Answer

Using biometric voice authentication technology, Versay's Password Reset application enables users to reset their own passwords quickly and securely. After a simple one-time enrollment, users can access the system 24 hours a day, 7 days a week over any telephone to reset their password. By automating the authentication process, your agents are freed to focus on value added sales and service interactions, your employees and customers have faster access to information and services, and your business benefits from the reduced labor and costs. And, since biometric voice authentication is more secure than PINs and passwords, the risk of fraud and identity theft is significantly reduced. Benefits of self-service reset include:

- **Decreased Costs:** Research shows that agent-handled password resets cost upwards of \$30 per user. At just a fraction of the price, Versay's Password reset application delivers a quick return on investment.
- **Increased Security:** Biometric voice authentication is significantly more secure than PINs and passwords and eliminates the potential for human error in manual password reset procedures.

6. The Ten-Ton Gorilla in the Room

- Even if passwords had no technical security issues or practical security issues, there are policy driven reasons why passwords are no longer good enough.
- For example, consider the Payment Card Industry Data Security Standard (PCI-DSS) 8.3 (Version 1.2.1, July 2009). It requires:

"Incorporate two-factor authentication for remote access (network-level access originating from outside the network) to the network by employees, administrators, and third parties."

- Trust me, two factor authentication is in your future.

Two Factor Authentication

The Most Common Alternatives To Just Traditional Passwords

- Two factor authentication combines two or more types of authentication, such as:
 - Hardware cryptographic tokens (crypto key fobs)
 - Biometrics (such as thumbprint scanners)
 - Smart cards or USB thumb drives with certs
- Today, let's just consider hardware crypto tokens.
- Given all the problems that traditional passwords represent, why haven't sites rushed out to deploy hardware crypto tokens?

Cost Is One of The Primary Reasons Why There's Resistance to Replacing Passwords

- As expensive (and insecure) as passwords may potentially be, the cost of alternatives is one of the primary reasons why there's resistance to replacing passwords with something else.
- For example, some hardware cryptographic tokens may run \$20 to \$30+ per year per person (e.g., \$50 to \$60+ for a crypto token with a 2-3 year life from one vendor).
- If we assume that we'd like to issue crypto fobs to everyone on campus, if we're at a campus with 25,000 students, faculty and staff, doing the math that implies a cost of \$500,000 to \$750,000 per year just for tokens!
- That's a non-trivial expense, particularly if passwords are "free" (even though we know they're NOT really "free₇₄")

Sample Cost/Token For One Edu Site

OBTAI N CORRECT PREREQUISITES (Completed by applicant)

BEFORE completing this form, please verify that you have obtained the following:

- UCD Login ID
- Kerberos Password
- DES Security Token (note: not required if requesting only access to class roster via the web or online grading)

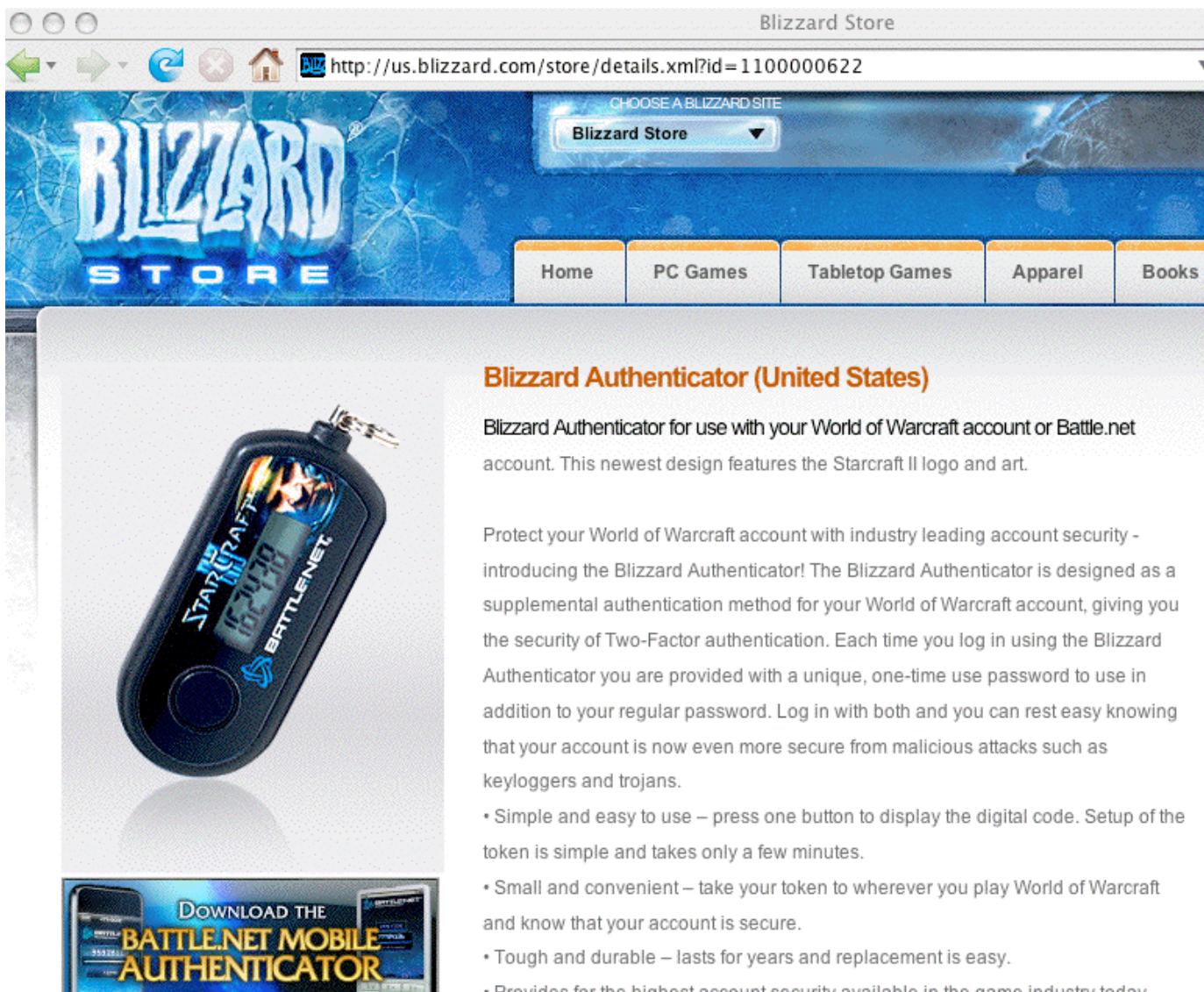
If you *do not* have one or more of these items, see below to obtain each item before proceeding. Please remember that the DES Security Token is *required* when requesting view or maintenance/update access to BANNER, issue Permission to Add/Drop numbers, or online scheduling for department schedulers. Review the information below to determine how to obtain each of these items.

To obtain your UCD LoginID and Kerberos Password, visit IT-Express in Shields Library, or go to <https://computingaccounts.ucdavis.edu/>

To obtain your DES Security Token:

- 1) You must have your UCD LoginID to purchase a security token
- 2) Complete a purchase order form, requesting the required number of tokens and the description 'DES Card(s).' The cost of the token is \$64.00.
- 3) Indicate the UCD LoginID of the person who will be using the token on the purchase order form.
- 4) Specify Yes or No for department pickup; the UCD Bookstore / Computer Shop can send the token(s) via campus mail or contact you for pickup.

Some Sites Have Deployed Hardware Tokens At Significantly Lower Costs (WOW=\$6.50)



The screenshot shows a web browser displaying the Blizzard Store at <http://us.blizzard.com/store/details.xml?id=1100000622>. The page features a large blue header with the "BLIZZARD STORE" logo. A dropdown menu "CHOOSE A BLIZZARD SITE" is set to "Blizzard Store". Below the header is a navigation bar with links for Home, PC Games, Tabletop Games, Apparel, and Books. The main content area is titled "Blizzard Authenticator (United States)". It includes a photograph of a black hardware token with "STAR CRAFT II" and "BATTLE.NET" printed on it, along with a digital display showing a six-digit code. A descriptive text block explains the product's purpose as a Two-Factor authentication method for World of Warcraft accounts. Below this is a bulleted list of benefits: "Simple and easy to use – press one button to display the digital code. Setup of the token is simple and takes only a few minutes.", "Small and convenient – take your token to wherever you play World of Warcraft and know that your account is secure.", "Tough and durable – lasts for years and replacement is easy.", and "Provides for the highest account security available in the game industry today."

Blizzard Authenticator (United States)

Blizzard Authenticator for use with your World of Warcraft account or Battle.net account. This newest design features the Starcraft II logo and art.

Protect your World of Warcraft account with industry leading account security - introducing the Blizzard Authenticator! The Blizzard Authenticator is designed as a supplemental authentication method for your World of Warcraft account, giving you the security of Two-Factor authentication. Each time you log in using the Blizzard Authenticator you are provided with a unique, one-time use password to use in addition to your regular password. Log in with both and you can rest easy knowing that your account is now even more secure from malicious attacks such as keyloggers and trojans.

- Simple and easy to use – press one button to display the digital code. Setup of the token is simple and takes only a few minutes.
- Small and convenient – take your token to wherever you play World of Warcraft and know that your account is secure.
- Tough and durable – lasts for years and replacement is easy.
- Provides for the highest account security available in the game industry today.

PayPal SecurityKey: \$5

PayPal Security Key



Get an extra layer of protection with the PayPal Security Key - it's easy to use and portable, so you can access your account with confidence from just about anywhere.

► What is it?

The PayPal Security Key creates random temporary security codes that help safeguard your PayPal account when you log in. It comes in 2 types, each with different advantages:

1. **Security key:** You carry this small credit-card sized device with you. It creates a unique security code on the go.
2. **Mobile phone security key:** You can sign up to get security codes sent by text message to your mobile phone.

► Already have a security key?

[Activate Now](#)

It will also work with your eBay account.

[Click Here »](#)



► How much does it cost?

The mobile security key has no costs, except your mobile provider's standard text messaging charges. Check with your mobile provider for details.

Each security key is \$5, and there's no monthly service fee or additional cost. Replacement keys are the same price.

E*Trade: Free For Bigger Customers

E*TRADE FINANCIAL – Home

<https://us.etrade.com/e/t/estation/pricing?id=1220801000#Find>

Learn about the Digital Security ID

E*TRADE FINANCIAL already maintains the highest levels of online security available. Now we're exceeding today's standards with our E*TRADE Complete™ Security System, including:

- **FREE optional Digital Security ID** that makes unauthorized log-on virtually impossible.

PLUS:

- **Electronic documents** to help prevent identify theft via mail and paper trails.
- **SmartAlerts** you configure to inform you of any account activity.
- **Security specialists** to investigate attempted fraud and resolve issues.

A photograph of a physical digital security device. It has a circular face with a purple and yellow stylized 'X' logo in the center. To the right of the logo is a small digital screen displaying the number '032848'. Below the screen is a small blue label that reads 'Secured by RSA'.

Digital Security ID
Secured by **RSA**

Six-digit personal access code
changes every 60 seconds.

[Back to top](#)

Understand how the Digital Security ID works

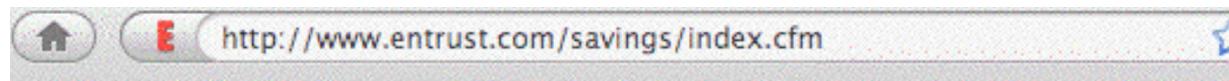
When you sign up for the Digital Security ID, we send you an electronic device that is like a **combination lock for your account(s)**, except that the **combination changes every 60 seconds**. Our two-factor authentication solution, which you can deploy on your own computer, uses the device to provide you with a **unique personal access code** - a random, six-digit number that changes every 60 seconds. Used in combination with your user ID and password when you log on at *etrade.com*, your personal access code prevents unintended or unauthorized users from logging on to your E*TRADE account(s).

[Back to top](#)

Find out what it will cost me

The E*TRADE Complete™ Digital Security ID will be provided at no cost to customers with \$50,000 or more in combined E*TRADE Securities and E*TRADE Bank accounts, or who place 10 or more trades per month. Customers with accounts that have multiple account holders will be provided up to two devices at no cost. All other E*TRADE customers will be charged \$25 per device. A \$25 charge may be imposed for each additional or replacement Digital Security ID. Your Digital Security ID has an average life of three years.

Sample Token Vendor Competing On Price...



Entrust IdentityGuard Mini Token

Versatile and convenient, Entrust now offers the security of one-time-password tokens to provide strong authentication for enterprises, governments and consumers.

Small form, big introduction

Who would have thought that the introduction of something as small as the new Entrust IdentityGuard Mini Token could have such a huge impact on the authentication market. The Entrust IdentityGuard Mini Token and its industry-first \$5 price tag eliminates past barriers to leveraging hardware tokens as part of a versatile authentication platform.

With its smart, simple, one-button design, the Entrust IdentityGuard Mini Token offers easy-to-use capabilities that can be deployed alone or in combination with other authentication methods as part of the Entrust IdentityGuard platform — a highly respected versatile authentication solution.

Affordable, smart pricing from Entrust

Priced at \$5 per token, the Entrust IdentityGuard Mini Token demonstrates that secure, reliable hardware authentication can be had at an attractive price. The real value for organizations is the ability to leverage affordable tokens in parallel with the full spectrum

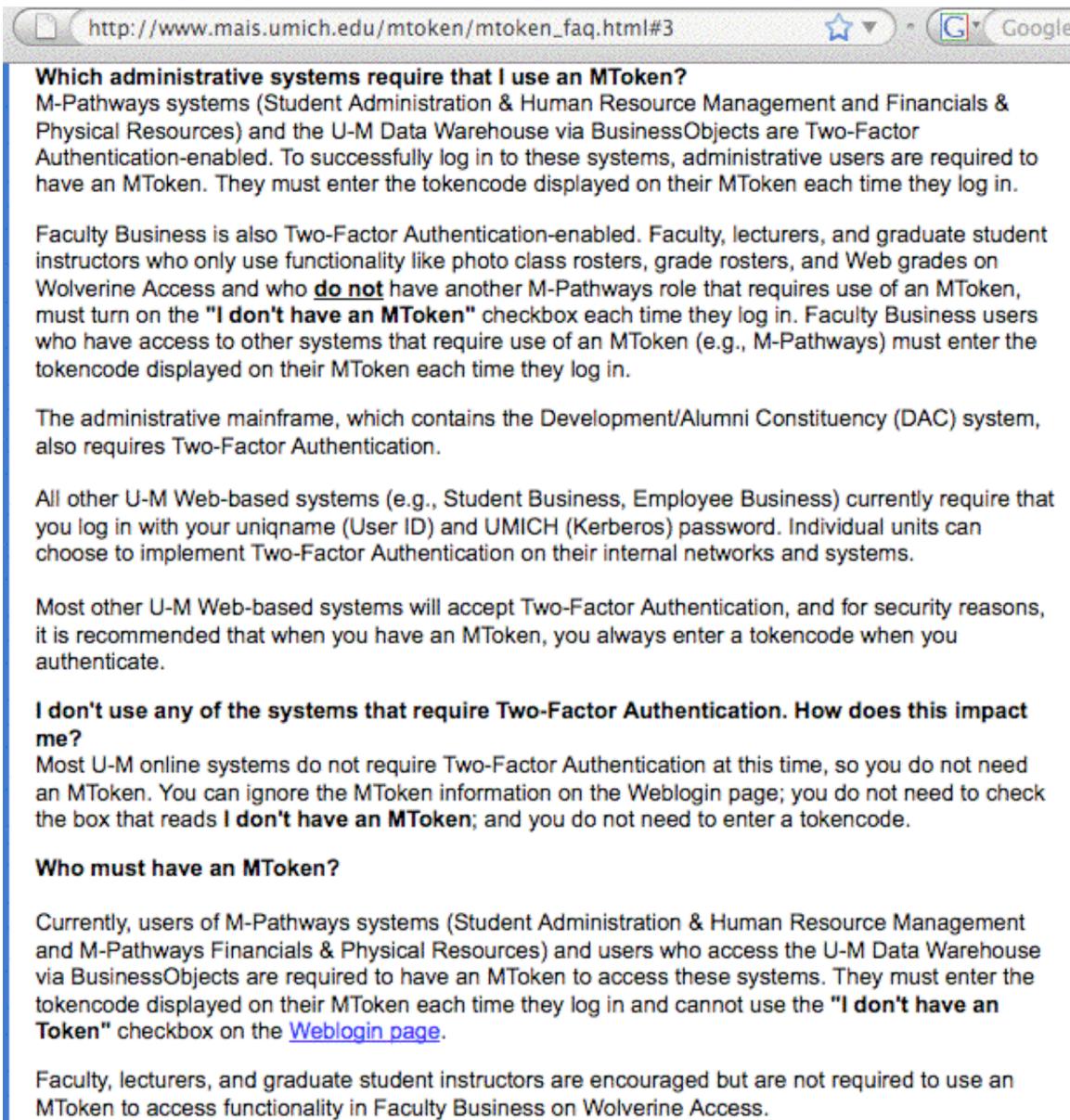


Entrust IdentityGuard Tokens
QUANTITY ONE: \$5

Incremental Deployment Is Also A Possibility

- If you can't afford to deploy hardware crypto tokens everywhere due to cost, potentially consider an incremental deployment strategy.
- For example, perhaps you may want to consider initially deploying hardware crypto tokens just for administrative system users, or privileged users, or even just for all faculty/staff.
- By limiting the number of tokens you initially deploy, you can:
 - focus on the most security-sensitive areas first
 - keep the total cost for the project low
 - gain experience deploying and administering tokens
 - help campus users become familiar with tokens and how they work (many users may never have heard of them)

Sample EDU Incremental Deployment



http://www.mais.umich.edu/mtoken/mtoken_faq.html#3

Which administrative systems require that I use an MToken?
M-Pathways systems (Student Administration & Human Resource Management and Financials & Physical Resources) and the U-M Data Warehouse via BusinessObjects are Two-Factor Authentication-enabled. To successfully log in to these systems, administrative users are required to have an MToken. They must enter the tokencode displayed on their MToken each time they log in.

Faculty Business is also Two-Factor Authentication-enabled. Faculty, lecturers, and graduate student instructors who only use functionality like photo class rosters, grade rosters, and Web grades on Wolverine Access and who do not have another M-Pathways role that requires use of an MToken, must turn on the "I don't have an MToken" checkbox each time they log in. Faculty Business users who have access to other systems that require use of an MToken (e.g., M-Pathways) must enter the tokencode displayed on their MToken each time they log in.

The administrative mainframe, which contains the Development/Alumni Constituency (DAC) system, also requires Two-Factor Authentication.

All other U-M Web-based systems (e.g., Student Business, Employee Business) currently require that you log in with your uniqname (User ID) and UMICH (Kerberos) password. Individual units can choose to implement Two-Factor Authentication on their internal networks and systems.

Most other U-M Web-based systems will accept Two-Factor Authentication, and for security reasons, it is recommended that when you have an MToken, you always enter a tokencode when you authenticate.

I don't use any of the systems that require Two-Factor Authentication. How does this impact me?
Most U-M online systems do not require Two-Factor Authentication at this time, so you do not need an MToken. You can ignore the MToken information on the Weblogin page; you do not need to check the box that reads **I don't have an MToken**; and you do not need to enter a tokencode.

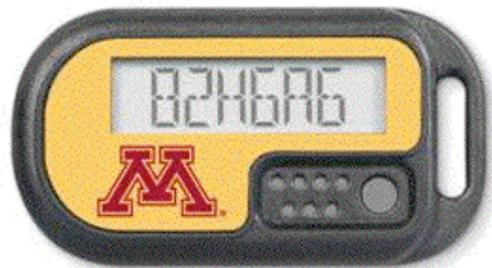
Who must have an MToken?
Currently, users of M-Pathways systems (Student Administration & Human Resource Management and M-Pathways Financials & Physical Resources) and users who access the U-M Data Warehouse via BusinessObjects are required to have an MToken to access these systems. They must enter the tokencode displayed on their MToken each time they log in and cannot use the "I don't have an Token" checkbox on the [Weblogin page](#).

Faculty, lecturers, and graduate student instructors are encouraged but are not required to use an MToken to access functionality in Faculty Business on Wolverine Access.

Example #2 (They've Deployed ~8K Tokens)

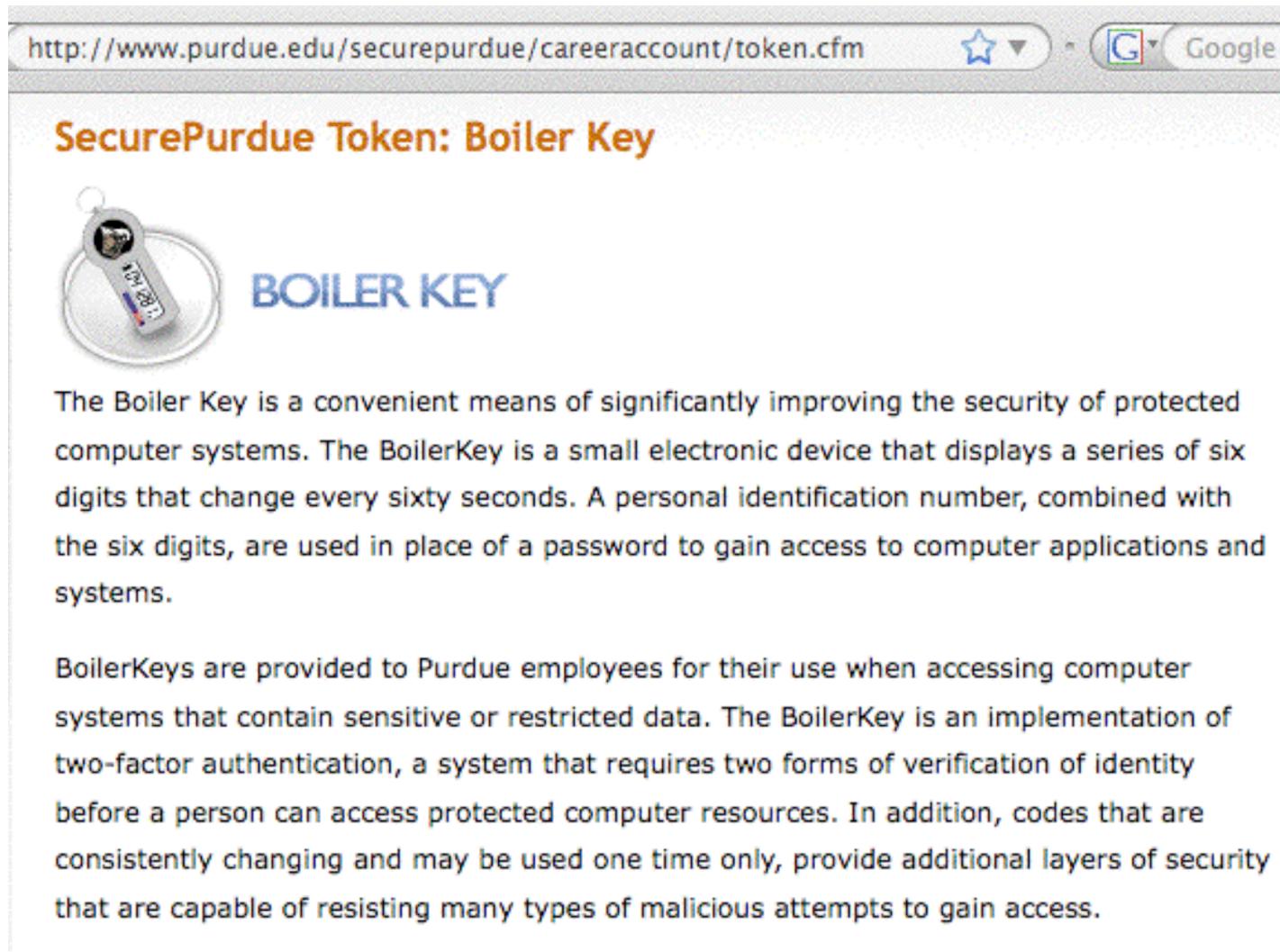


M Key Frequently Asked Questions (FAQ)



[What is Two Factor Authentication?](#)
[Why is the University implementing the M Key system?](#)
[How does it work?](#)
[Who will use it?](#)
[Can I still use my Enterprise password?](#)
[How do I request an M Key?](#)
[What is the "REMOVE" sticker on the M Key?](#)

Example #3



The screenshot shows a web browser window with the URL <http://www.purdue.edu/securepurdue/careeraccount/token.cfm> in the address bar. The page title is "SecurePurdue Token: Boiler Key". On the left, there is an image of a physical Boiler Key device, which looks like a small electronic key fob with a screen displaying numbers. To the right of the image, the text "BOILER KEY" is written in blue capital letters. Below the image and text, there is a detailed description of the Boiler Key's function and purpose.

The Boiler Key is a convenient means of significantly improving the security of protected computer systems. The BoilerKey is a small electronic device that displays a series of six digits that change every sixty seconds. A personal identification number, combined with the six digits, are used in place of a password to gain access to computer applications and systems.

BoilerKeys are provided to Purdue employees for their use when accessing computer systems that contain sensitive or restricted data. The BoilerKey is an implementation of two-factor authentication, a system that requires two forms of verification of identity before a person can access protected computer resources. In addition, codes that are consistently changing and may be used one time only, provide additional layers of security that are capable of resisting many types of malicious attempts to gain access.

Another Option: Allow *Individuals* To Opt-In

- Interest among users varies when it comes to security
- You may want to take advantage of those differences by allowing (but not requiring) users to opt-in to better-than-traditional-password authentication, with participation contingent upon the user self-funding purchase of a hardware crypto token. (Note that this is the WoW, eBay, etc. model)
- The trade-off here is that administration of this sort of onesie-twosie adoption will be more manual/labor intensive, and the cost per token may also be higher since volumes will likely be lower.
- Remaining non-crypto-fobbed accounts will also continue to be the target of hacker/crackers.

Please Don't Make My Pants Fall Down

- Assume each of the following services decided to replace their traditional passwords with hardware crypto fobs:
 - your general purpose university account
 - other specialized university systems (perhaps core networking devices, or a departmental HPC cluster, say)
 - your bank or credit union
 - your brokerage
 - your health care provider
 - your favorite Internet online auction site
 - your favorite online gaming site
 - etc., etc., etc.
- Also assume that no two sites use the same token (or even the same brand of tokens!)
- Soon we may all be carrying around huge rings of hardware crypto fobs! We *need* to get federated! ⁸⁵

Tokenless/Phone-Based Second Channel

- If we assume that everyone has a cell phone, is that the ultimate tokenless (and scalable!) "2nd factor?" (actually, 2nd channel)
- User goes to login as they normally would with username and password
- As part of that login process, a magic code gets sent to user's pre-registered cell phone as a text message
- User is prompted to enter the code they've been sent
- User enters the provided magic code
- Authentication then proceeds as normal
- Example implementation: see www.phonefactor.com

Conclusion/Summary

- Traditional passwords have many security issues, but they are still all-too-widely used
- The time has come for you and your site to replace them with something stronger, such as hardware crypto tokens
- You may NEED to do so for compliance reasons (PCI-DSS)
- Leading online sites (such as eBay, gaming sites, and brokerages) are already deploying hardware tokens, and so are leading educational institutions (at least for selected groups)
- Cost remains a potential issue, but can be potentially finessed
- Scalability is the other issue we need to keep in mind; second channel authentication (e.g., magic codes sent to user cell phones via text messages) may be one answer¹⁷

Thanks For the Chance to Talk Today!

- Are there any questions?