Multi-Factor Authentication: Do I Need It? Should I Begin Using It? [And If I Do Need It, Why Aren't Folks Deploying It?]

Joe St Sauver, Ph.D. (joe@stsauver.com)
Farsight Security, Inc.
ACPENW 2015, Friday, May 8\textsuperscript{th}, 2015
10:00-11:15 AM, Altitude Room
Resort at the Mountain, Welches OR

https://www.stsauver.com/joe/acpenw-multifactor/
Introduction
This Is A Multifactor Talk, But We're Going To Begin By Talking About Passwords...

• Passwords are the default authentication technology today, used everywhere. So to motivate multifactor authentication, we need to begin by looking at passwords, and why, alone, they simply aren't good enough.

• I'd also note that in most cases multifactor authentication ADDS something to existing passwords, it DOESN'T eliminate the use of passwords. So passwords are still a highly relevant part of the overall multifactor authentication picture.

• I suspect that passwords, like the poor, will be with us forever. [with no disrespect meant to the poor or to Matthew 26:11]
Passwords Are Critical to IT Security

• If one or more of your passwords are compromised:
  -- **confidential materials may be disclosed** (resulting in you being sued/fired)
  -- **critical files may be surreptitiously modified or deleted,**
    (including grades in online grade books!)
  -- **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 school 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. But are they?
The Way These Things Often Start For Me: With ERRORS

• When I went to register for this very meeting, I found that I'd apparently accidentally made an "error" (I'm prone to making mistakes of all sorts pretty routinely)

• The mistake in this case was somewhat ironic, since I was scheduled to talk about multifactor authentication: I found out that I'd apparently tried to enter a deficient or inadequate password for the event registration web site.

• When I made that mistake, I saw the following error report (except for the red stuff, which I added)...
This Event's Own Registration Site

The following errors were detected:

- For a valid password, you must enter a minimum of 8 characters. Your password must contain at least one character from three of the following four sets: lower-case, upper-case, numeric and special characters.
(Just Like A Kid) I Often Find Myself Asking "Why?"

- **Why** do I need an **eight** character password? Why not seven, or nine, or twelve?
- **Why** does my password apparently need to contain characters from three out of four categories of characters (lowercase letters, upper case letters, numbers, special characters)?
- **If I REALLY need to protect this account, shouldn't I be using multifactor authentication for even MORE protection?**
- Are there other attack vectors that undercut the strength of the complex passwords we're relying on, instead? Such as perhaps the "I-forgot-my-password-so-please-reset-it-for-me" feature?
- **Why** have my email address be my account name? Why not allow me to pick an account name of my choice, and THEN collect my email address from me LATER?
- We'll talk about some of these things during today's session.
It's True: People **DO** Tend To Pick WEAK Passwords

- I know (apparently from personal experience :-) ) that people will often pick "weak" passwords if they're allowed to do so, including:
  
  ... passwords that are trivially short (6 character or less)
  ... passwords using only lower case letters
  ... passwords that are simply words found in the dictionary
  ... passwords that are a "clever" pattern on the keyboard
  
  (12345678, qwerty, etc.)

- Why are weak passwords a problem? Well, weak passwords can be guessed or otherwise obtained. When that happens, the user's privacy is violated, and those cracked accounts get abused.

- Let's talk a bit about some common account attack techniques.
1. Brute Force Attacks And Password Complexity
Password Strength

- **In an ideal world**, all possible passwords would be equally likely to be chosen (e.g., the set of human-selected passwords would be "uniformly distributed" across the field of possibilities)

- If you had an six character (or less) lowercase-only alphabetic password, you might have selected any of:
  - a, b, c, ..., z
  - aa, ab, ac, ... zz
  - aaa, aab, aac, ... zzz
  - [...] 
  - aaaaaa, aaaaab, aaaaac, ... zzzzzy, zzzzzz

- How many such combinations are there?
Number of Possible Combinations for a 1-6 Character Lowercase-Only Password

• 1 character, exactly: 26 (e.g., a, b, c, ... z)
• 2 characters, exactly: $26^2=676$ (e.g., aa, ab, ac ... zz)
• 3 characters, exactly: $26^3=17,576$
• 4 characters, exactly: $26^4=456,976$
• 5 characters, exactly: $26^5=11,881,376$
• 6 characters, exactly: $26^6=308,915,776$

• Total possibilities for a 1-6 character all lowercase password?

$$308,915,776 + 11,881,376 + 456,976 + 17,576 + 676 + 26 = 321,272,406$$
How Long Would It Take To Brute Force A Six Character Lowercase-Only Password?

![GRC's Interactive Brute Force Password “Search Space” Calculator](https://www.grc.com/haystack.htm)

**Brute Force Search Space Analysis:**

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Space Depth (Alphabet)</td>
<td>26</td>
</tr>
<tr>
<td>Search Space Length (Characters)</td>
<td>6 characters</td>
</tr>
<tr>
<td>Exact Search Space Size (Count)</td>
<td>321,272,406</td>
</tr>
<tr>
<td>Search Space Size (as a power of 10)</td>
<td>$3.21 \times 10^8$</td>
</tr>
</tbody>
</table>

**Time Required to Exhaustively Search this Password's Space:**

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online Attack Scenario:</td>
<td>3.72 days</td>
</tr>
<tr>
<td>(Assuming one thousand guesses per second)</td>
<td></td>
</tr>
<tr>
<td>Offline Fast Attack Scenario:</td>
<td>0.00321 seconds</td>
</tr>
<tr>
<td>(Assuming one hundred billion guesses per second)</td>
<td></td>
</tr>
<tr>
<td>Massive Cracking Array Scenario:</td>
<td>0.00000321 seconds</td>
</tr>
<tr>
<td>(Assuming one hundred trillion guesses per second)</td>
<td></td>
</tr>
</tbody>
</table>

Note that typical attacks will be online password guessing limited to, at most, a few hundred guesses per second.
What If We Made The Password Longer (But Still Used Only Lowercase Letters)?

- Making a password LONGER improves things IMMENSELY

- 7\textsuperscript{th} character: $321,272,406 \text{ (from the previous slide)} + (26^7) = 8,031,810,176$
- 8\textsuperscript{th} character: $8,031,810,176 + (26^8) = 208,827,064,576$
- 9\textsuperscript{th} character: $208,827,064,576 + (26^9) = 5,638,330,743,552$
- 10\textsuperscript{th} character: $5,638,330,743,552 + (26^{10}) = 146,805,426,396,928$
- 11\textsuperscript{th} character: $146,805,426,396,928 + (26^{11}) = 3,817,149,913,384,704$
- 12\textsuperscript{th} character: $3,817,149,913,384,704 + (26^{12}) = 99,246,106,575,066,880 \Leftarrow$ arguably, this is "quite a large number"
What If We "Went Totally Crazy" and Had a 36 CHARACTER-Long Lowercase Only Passphrase?

![GRC's Interactive Brute Force Password "Search Space" Calculator](https://www.grc.com/haystack.htm)

**oat meal is nutritious and delicious**

Brute Force Search Space Analysis:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Space Depth (Alphabet)</td>
<td>26 + 33 = 59</td>
</tr>
<tr>
<td>Search Space Length (Characters)</td>
<td>36 characters</td>
</tr>
<tr>
<td>Exact Search Space Size (Count)</td>
<td>5,729,232,372, 459,098,666, 549,656,927, 180,282,234,474,814,812, 596,145,669,014,757,320</td>
</tr>
<tr>
<td>Search Space Size (as a power of 10)</td>
<td>$5.73 \times 10^{63}$</td>
</tr>
</tbody>
</table>

Time Required to Exhaustively Search this Password's Space:

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Time Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online Attack Scenario: (Assuming one thousand guesses per second)</td>
<td>1.82 thousand trillion trillion trillion centuries</td>
</tr>
<tr>
<td>Offline Fast Attack Scenario: (Assuming one hundred billion guesses per second)</td>
<td>18.22 million trillion trillion trillion centuries</td>
</tr>
<tr>
<td>Massive Cracking Array Scenario: (Assuming one hundred trillion guesses per second)</td>
<td>18.22 thousand trillion trillion trillion centuries</td>
</tr>
</tbody>
</table>

Note that typical attacks will be online password guessing limited to, at most, a few hundred guesses per second.
Are Passphrases "The Answer?"

• While people have a hard time remembering random unpronounceable strings with numbers and weird characters and odd capitalization, most people CAN remember a relatively long "passphrase..."

• Pass phrases can be a more secure alternative, at least if:

  -- the site you're logging into allows you to use long passwords (some will, some may not)

  -- the site allows you to use words from the dictionary in your password (again, some may, some may not)

  -- your pass phrase isn't a well known cliché or otherwise overly-common ("a penny saved is a penny earned")
Note: That 36 Character Passphrase Was Made From JUST Lowercase Letters (Plus Spaces)

• It didn't have any upper case letters, or numbers or special symbols. That means that it, or a password like it) wouldn't work for the meeting website even though it WAS long, VERY LONG

• The lesson? We can (or SHOULD BE) allowed to make a trade-off between character set complexity and password/passphrase length
So What About MORE COMPLEX Character Sets?

• Instead of making the password long (but just lowercase), we could instead draw each potential character from a more complex character set, perhaps:

  26 lower case letters
  26 upper case letters
  10 numbers
  32 printable special symbols: space, !@#$%^&*()-_=+={[]}::"<>,.?/

  = 94 potential characters at each location
For Example: A Sample "Complex" Password

GRC's Interactive Brute Force Password “Search Space” Calculator
(NOTHING you do here ever leaves your browser. What happens here, stays here.)

Enter and edit your test passwords in the field above while viewing the analysis below.

Brute Force Search Space Analysis:

<table>
<thead>
<tr>
<th>Search Space Depth (Alphabet):</th>
<th>26+26+10+33 = 95</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Space Length (Characters):</td>
<td>13 characters</td>
</tr>
<tr>
<td>Exact Search Space Size (Count):</td>
<td>51,880,316, 927,184,027,554,126,495</td>
</tr>
<tr>
<td>(count of all possible passwords with this alphabet size and up to this password's length)</td>
<td></td>
</tr>
<tr>
<td>Search Space Size (as a power of 10):</td>
<td>$5.19 \times 10^{25}$</td>
</tr>
</tbody>
</table>

Time Required to Exhaustively Search this Password's Space:

<table>
<thead>
<tr>
<th>Online Attack Scenario: (Assuming one thousand guesses per second)</th>
<th>16.50 trillion centuries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offline Fast Attack Scenario: (Assuming one hundred billion guesses per second)</td>
<td>1.65 hundred thousand centuries</td>
</tr>
<tr>
<td>Massive Cracking Array Scenario: (Assuming one hundred trillion guesses per second)</td>
<td>1.65 hundred centuries</td>
</tr>
</tbody>
</table>

Note that typical attacks will be online password guessing limited to, at most, a few hundred guesses per second.
"Complex" Character Set: Possible Combinations By Password Size

- 1 complex character: $94^1 = 94$
- 1-2 complex characters: $94 + (94^2) = 8,930$
- 1-3 complex characters: $8,930 + (94^3) = 839,514$
- 1-4 complex characters: $839,514 + (94^4) = 78,914,410$
- 1-5 complex characters: $78,914,410 + (94^5) = 7,417,954,634$
- 1-6 complex characters: $7,417,954,634 + (94^6) = 697,287,735,690$
- 1-7 complex characters: $697,287,735,690 + (94^7) = 65,545,047,154,954$
- 1-8 complex characters: $65,545,047,154,954 + (94^8) = 6,161,234,432,565,770$ (as required by the meeting website)

- Versus 1-12 characters lower case only...
  $99,246,106,575,066,880$
What's The Biggest Problem With Complex Passwords?

- Simple: *people can't remember them.* This is particularly true for young children due to their relatively low level of cognitive development, but heck, poor memory can be a problem for adults, too.

- Do you remember the complex password I just showed you on the preceding slide?

- What about the longer pass phrase I showed you further back?

- We can make a memorable, long, lowercase-only passphrase that's just as strong as a shorter but hard-to-remember password using a complex character set!
Password Combinations By Password Length

A 12 char lowercase-only password is as strong as a 8 char complex one
2. Dictionary Attacks
On Passwords
Of Course, Not All Attacks On Passwords Are Exhaustive Attacks

- We know that some passwords are actually far less likely than others: the password "xhqwryyk", for example, is less likely to be chosen by a typical user than a dictionary word such as "hockey" (unless the user is relying on a password generator, such as pwgen or http://www.multicians.org/thvv/gpw-js.html):

  $ pwgen -1 12
  hai2ahch2oHi

- How many words do most users know? Estimates are ~40,000 words or less [see the graph on the following slide] Note, too, that young children (such as the ones you may work with) have vocabularies that are only a fraction of an adult's vocabulary.
Average English native speaker vocabulary by age (all participants)

Source:
Particularly Popular (BAD) Passwords

most popular passwords

Source: https://www.dragonresearchgroup.org/insight/sshpwauth-cloud.html
**Do YOU Use Any Of Those Passwords?**
(Or Any OTHER Really Lame Password?)

- I sure hope not!
- Those are the passwords that the Dragon Research Group (DRG) sees hacker/crackers try most often when the bad guys/gals are trying to guess the password of ssh (secure shell) accounts.
- Why do attackers try *those* passwords? The passwords on that list include some known default passwords (that may never have gotten changed), plus passwords that are empirically known to be popular with many average users (or popular with help desk staff members when setting initial user passwords).
- If you use one of those passwords (or any other simple password, including any word from a dictionary), it's just a matter of time until bad things happen. Many "password auditing tools" are in broad circulation.
Password Auditing Tools

• I never like to help the script kiddies by providing pointers, but note that these tools aren't exactly "obscure."

• See, for example, http://sectools.org/tag/pass-audit/
A Sample Tool From the Sectools.org Listing

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 lots of other hashes and ciphers in the community-enhanced version.

Openwall wordlists collection for password cracking (20+ languages)

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
- On Windows, consider Hash Suite (developed by a contributor to John the Ripper)

Download one of the latest official free versions (release notes):

- John the Ripper 1.8.0 (sources, tar.xz, 4.3 MB) and its signature
- John the Ripper 1.8.x extra charset files archive (tar.xz, 4.5 MB) and its signature
- John the Ripper 1.8.0 (sources, tar.gz, 5.2 MB) and its signature
- John the Ripper 1.7.9 (Windows binaries, ZIP, 2029 KB) and its signature
A Commercial Parallel GPU-Enabled Password Recovery Product From A Russian Company

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

- Hardware acceleration (patented) reduces password recovery time by a factor of 50
- Support for NVIDIA CUDA cards, AMD Radeon HD and Tableau TACC1441 hardware accelerators
- Linear scalability with no overhead allows using up to 10,000 workstations without performance drop-off
- Allows up to 32 CPUs or CPU cores and up to 8 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

Prices:

- Up to 5 clients - $599
- Up to 20 clients - $1999
- Up to 100 clients - $4999
- 100+ clients - contact us

- Purchase EDPR
- Download EDPR
- Download agent only
- System requirements for EDPR
- View the screenshot of EDPR
- Read EDPR Online Documentation

- Subscribe to the Password Recovery Software newsletter
- GPU Acceleration Frequently Asked Questions
- Size does matter. Advantages of distributed password recovery.
A Sample Open Source In-Situ Brute Force Network Login Auditing Tool

Medusa Parallel Network Login Auditor

JoMo-Kun / jmk "AT" foofus "DOT" net

- What?
- Why?
- How?
- Where?
- Who?
- Huh?

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 THC-Hydra? Here are some of the reasons for this application:
Can "Brute Forcers" Accidentally (or Intentionally) DDoS Your Users? Yes

 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 for 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 can help cut down on the network noise (but some attackers are thorough and will simply scan all open ports)
3. Password Sniffing Attacks
"You Can Observe A Lot By Watching" [Yoggi Berra]

- In addition to brute forcing passwords, or using dictionary attacks, you can also often sniff ("eavesdrop upon") passwords when they're transmitted over the network. This can be done with an Ethernet span port, a passive optical tap, a man-in-the-middle attack, or other means.

- If a password can be sniffed, it doesn't matter if it is long/strong or not, you're TOAST.

- "But Joe, everyone knows that passwords should only be sent over securely-encrypted channels! No one would send sensitive information in clear text over the wire, or to an insecure site!"

- Well, just to pick on "us" one (or two) more times...
## SSL Report: events.blufish.me (104.236.164.217)

**Assessed on:** Tue, 05 May 2015 14:39:29 UTC | HIDDEN | Clear cache

### Summary

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>100</td>
<td>0</td>
<td>80</td>
<td>90</td>
</tr>
</tbody>
</table>

- **Visit our documentation page** for more information, configuration guides, and books. Known issues are documented [here](#).
- **This server is vulnerable to the Poodle attack.** If possible, disable SSL 3 to mitigate. Grade capped to C. [MORE INFO »](#)
- **This server is vulnerable to the OpenSSL CCS vulnerability (CVE-2014-0224)** and exploitable. Grade set to F.
- **This server is vulnerable to the Heartbleed attack.** Grade set to F.
- **The server does not support Forward Secrecy with the reference browsers.** [MORE INFO »](#)
SSL Report: regonline.com (74.120.127.80)
Assessed on: Mon, 04 May 2015 13:12:37 UTC | Clear cache

Summary

Overall Rating

- Certificate: 100
- Protocol Support: 70
- Key Exchange: 90
- Cipher Strength: 80

Visit our documentation page for more information, configuration guides, and books. Known issues are documented here.

This server uses SSL 3, which is obsolete and insecure. Grade capped to B. MORE INFO

The server supports only older protocols, but not the current best TLS 1.2. Grade capped to B.

This server accepts the RC4 cipher, which is weak. Grade capped to B. MORE INFO

The server does not support Forward Secrecy with the reference browsers. MORE INFO
We MUST ALL Pay Attention To SSL/TLS ("Web") Crypto These Days

• SSL/TLS crypto has been a big mess for a long time. It's getting cleaned up, but it's a brutal process of vulnerability-after-vulnerability getting exposed and then corrected.

• If you don't stay patched up-to-date, and take steps to securely configure your "secure" web site(s), they won't BE secure (including lacking security to protect passwords sent to them):

• If you need recommendations for hardening your sites, I have some advice for hardening your SSL/TLS configuration in two talks I've previously given, see:

https://www.stsauver.com/joe/crypto-bcp/
"We're 100% Switched! We're Safe From Sniffing!"

• 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 Ethernet, and in such an architecture network traffic shouldn't be visible to eavesdroppers (the way it is on a shared network link).

• I would suggest that between ARP spoofing, MAC flooding, and other methods mean that 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 Use 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 use it, you just need to recognize that you may still have sensitive traffic that's at least briefly exposed.
• VPNs are not (complete) protection against sniffing.
"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 potentially stealing your passwords...

- Let me give you just a few examples...
Near-Side Endpoint: Hardware Keystroke Grabbers

The KeyGhost Hardware Keylogger is a tiny plug-in device that records every keystroke typed on any PC computer.

TimeDate Stamping KeyGhost SX
Click the link below to visit the KeyGhost SX website:
http://www.KeyGhost.com/SX

KeyGhost External Stand-alone Models
- KeyGhost Home Edition 128K Flash Memory - $89
- KeyGhost Std 512K Flash Memory - $99
- KeyGhost Pro 1 Megabyte Flash Memory - $149
- KeyGhost Pro SE 2 Megabyte Flash Memory - $199

KeyGhost USB Keylogger
Compatible with both PC and Mac USB keyboards. Click to read more...
Remote Endpoint: 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
"We'd LOVE to Encrypt, But We Just CAN'T!"

* Sometimes sites KNOW that they **should** encrypt, but intentionally **forbid it**. Why? Well...

  "*If users were to use strong encryption, then we **ourselves** couldn't monitor what people are doing on the network -- all that traffic would be opaque to our monitoring systems!*

* Is this true for YOUR site?

* Yes, I know, K12 is "special" – children need to be protected against online predators, etc.

* But is that reason enough to forgo strong end-to-end encryption? I don't think so.
"Good News! At Least We're Encrypting Our Wireless Links To Prevent Password Sniffing"

• This is one of those things that drive me crazy.

• 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 your wireless links still leaves traffic on all your other links vulnerable to being sniffed!

• And just for the record and as a matter of due diligence, if you're doing wireless encryption (and you should be), these days you really should be doing WPA2 with AES and unique per-user passwords (so-called "enterprise" mode, not personal mode)
"Evil Twin" Wireless Nodes

• Not to make you paranoid, but since we're here, let me also bring up another wireless issue relating to passwords: how do you know that the wireless network you connected 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 fake node, instead of your real one, he may be able to sniff all your traffic, including your passwords, even if your real nodes use strong encryption.

• 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)
Shoulder Surfing and/or Pinhole 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 "pinhole" wireless cameras focused 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 displayed on your screen. See http://www.3M.com/PrivacyFilters
The Malware Problem

• **Malware is the most common way that passwords get sniffed.**
• If your computer does get infected with some types of malicious software, that malware may snoop and report on whatever you type – including your secret passwords.
• It's therefore critical that you strive to keep your computer **malware free** (one good option is to use an operating system that doesn't get hit by malware much, since antivirus really isn't all that effective any more).
• Regardless of which operating system you use, make sure you keep everything patched up-to-date! (Try Secunia PSI on your personal Windows systems, see http://secunia.com/vulnerability_scanning/personal/ )
• On a Mac, an outline for updates can be found at: https://www.stsauver.com/joe/mac-software-updates.pdf
4. Users WILL Disclose Their Passwords
People ARE The Weakest Link!

• Users will disclose their passwords in many different ways.
• For example, phishing exists because people can be (and routinely are) "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 groups (such as elementary and secondary education, unfortunately) have cultures which reward trust and relatively unquestioning compliance when confronted with authoritatively presented demands:
  Phisher: "Tell me your password, immediately!"
  User: "Okay, okay! It's LetMeIn123, please 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:

  -- Principals may demand to know the password of each teacher's accounts so that they can "access critical files" while the teacher may be sick or out on vacation.

  -- Children will disclose their password to their parents

• Point is: many people don't really take to heart a "keep your password strictly to yourself" philosophy.

• Let's now talk about remembering passwords by writing them down (hopefully) without sharing them with all random visitors...
Important Note From Joe: this only works if you trust those who have access to the contents of your wallet, and you never lose your wallet...

Additional Important Note From Joe: this is NOT a recommendation that you should tape your username and password to the side of your computer where anyone walking by can see it!
A Writing-Passwords-Down Failure Example

London Railway Station System Passwords Exposed On TV Documentary
Writing Down Passwords In A School Setting

- In an elementary, middle school or high school setting, users might write their password down on the inside cover of a notebook or in their binder – but that may be prone to being noticed by classmates, or stolen, lost, or forgotten at home.

- Carrying ones passwords in one's purse or wallet is probably more secure, assuming you don't lose your purse or wallet (or have it lost or stolen).

- The problem? **Young schoolchildren often don't carry a purse/wallet!**
Another Problem With Writing Down Passwords and Saving Them in Your Wallet

bloguin.com/awfulannouncing/wp-content/uploads/sites/94/2013/03/Costanza-wallet.jpg
Online Password "Wallets"

• Some users simply store all their passwords in an encrypted password "safe" or "wallet."

• Some examples of such programs are shown on the next page

• For password safes to work best, users need either a dedicated system (such as a laptop that ONLY they use), or access to a smart phone, in which they can securely save their encrypted passwords.

• Does that sound like your student users? Do they have their own dedicated laptops or tablets, or do they have (and routinely have access to) a smart phone?
There Are Many Password Managers Available

The Best Password Managers

BY NEIL J. RUBENKING  AUGUST 22, 2014  COMMENTS

In these days of hacks, Heartbleed, and endless breaches, a strong, unique, and often-changed password for every site is even more imperative. A password manager can help you attain that goal.

<table>
<thead>
<tr>
<th>Name</th>
<th>Editor Rating</th>
<th>Lowest Price</th>
<th>Platforms</th>
<th>Password Strength Report</th>
<th>Secure Password Sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td>LastPass 3.0</td>
<td>🍓 🍓 🍓 🍓 🍓</td>
<td>$0.00</td>
<td>Windows, Mac, Linux</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>LastPass 3.0 Premium</td>
<td>🍓 🍓 🍓 🍓 🍓</td>
<td>$12.00</td>
<td>Android, iOS, Windows, Mac, Blackberry</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Dashlane 3</td>
<td>🍓 🍓 🍓 🍓 🍓</td>
<td>$9.95</td>
<td>Android, iOS</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>RoboForm Everywhere 7</td>
<td>🍓 🍓 🍓 🍓 🍓</td>
<td>$12.00</td>
<td>Android, iOS, Windows, Mac, Blackberry</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Intuitive Password 2.9</td>
<td>🍓 🍓 🍓 🍓 🍓</td>
<td>$12.00</td>
<td>Android, iOS, Windows, Mac, Blackberry</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Keeper Password Manager &amp; Digital Vault 8</td>
<td>🍓 🍓 🍓 🍓 🍓</td>
<td>$0.00</td>
<td>Windows, Mac, Blackberry, Windows Phone, Symbian, Linux</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Norton Identity Safe</td>
<td>🍓 🍓 🍓 🍓 🍓</td>
<td>$9.95</td>
<td>Windows, Mac</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>PasswordBox</td>
<td>🍓 🍓 🍓 🍓 🍓</td>
<td>$29.95</td>
<td>Windows, Mac</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>RoboForm Desktop 7</td>
<td>🍓 🍓 🍓 🍓 🍓</td>
<td>$29.95</td>
<td>Android, iOS, Windows, Mac, Kindle</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Sticky Password 7</td>
<td>🍓 🍓 🍓 🍓 🍓</td>
<td>$29.95</td>
<td>Android, iOS, Windows</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Online merchants and other supposedly secure websites can't be relied on to keep your personal information safe. Even sites using decent security practices may
Another Bad Form of Password Sharing: Using The Same Password on Multiple Sites

• Users may share passwords across multiple systems, using the same password for "everything" -- critical, highly-sensitive administrative systems, and for their online bridge club account etc., etc., etc.

• If a bad guy can crack the passwords on ANY of those systems, he'll be able to use that discovered password on ALL the other systems, including the highly sensitive ones.

• Really, really, not good. Just don't do it!
Storing Passwords on 3rd Party Sites  (For The 3rd Party Site To Use On Your Behalf)

• Sometimes users will even intentionally give login information, including passwords, to other sites to save and routinely re-use.

• A 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 (and it won't), it needs to know/save your un-hashed (raw and unsalted) password so it can ingest your email from those other accounts.
Example of POP Email Consolidation: Mail Fetcher

Check emails from other accounts using Gmail

If you have multiple email accounts, you can check them all in Gmail. GMail's Mail Fetcher feature first imports all your old messages to Gmail and then continues to bring in new messages sent to your other account. You can add up to 5 accounts, including Gmail and other email providers.

If you only want to get new emails brought into Gmail, try auto-forwarding your mail instead.

Tip: You can bring your contacts to Gmail too, learn how to import contacts.

Start importing emails
Cyber Bullying and Compelled Password Disclosure

• We must also recognize that if someone wants your password badly enough, they may be willing to use intimidation or even physical measures to get it.

• Do you think your weakest students are strong enough to resist coerced password disclosure if pressured by the class bully, or even a gang of classmates?
5. Besides Being Insecure, Passwords Are A Pain to Administer
If You Think About It...

- How much time, effort and money does your school's IT organization spend just trying to create, distribute, manage, reset and terminate accounts?
- 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 teacher's time worth? And how do you handle distance education students, if you have any?
- If you buy a commercial product to try to automate/simplify password resets, what did that cost? Does it work well for you?
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 insecure to you? It should...
- There are many ways that a bad guy could exploit that sort of password reset mechanism to steal your credentials, including:
  -- gaining control of the offsite email account, then requesting a password reset
  -- 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 )
  -- [etc]
This Event's Password Reset Solution

To email yourself a link to reset your password, click Continue. Please note that the password reset email will only be valid for 12 hours.

Email Address: [REDACTED]
"Trivial Pursuit" Passwords Resets

• Other sites allow password resets if the user can successfully regurgitate what I tend to 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 pet 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 or chocolate will often be good guesses )
For All Of The Preceding Reasons

• Plain old passwords in-and-of themselves just aren't good enough any more.

• You need something more – you need multifactor authentication.

• Let's talk about some multifactor options, just so we're all on the "same page"
6. Multifactor Options
Multifactor Authentication Leverages Two or More Of The Following

• Something you **know** (like a password or PIN)

• Something you **have** (such as a smart card, cryptographic key fob, or smart phone)

• Something you "**are**" (a biometric characteristic such as your fingerprint, your eye's iris, the characteristics of your voice, etc.)

• **Multifactor is NOT** using the same authentication technology twice (e.g., you can't just implement a second password and claim that you're now doing "multifactor authentication")
Classic Example of Multifactor Authentication

• Example: you use multifactor authentication every time you get money out of an automatic teller machine: you use both your ATM card (something you have), and something you known (your PIN).

• Another example: Multifactor authentication that leverages something you are (e.g., biometrics), is less common, but if you're a member of the Global Entry System Trusted Traveller Program you may remember that Customs collected a copy of your fingerprints (a biometric identifier) as part of enrolling in that program.

Types Of Multifactor We'll Consider Today

• Traditional cryptographic hard tokens (fading fast)

• Phone-based multifactor authentication (mainstream)

• PKI-based client certificates using USB-format PKI hard tokens, or smart cards (painful/nearly dead)

• Yubikey (emerging)

• Biometrics (special case)
7. Traditional Hard Cryptographic Tokens
Hardware Cryptographic Fobs

• Very traditional/"old school" "default" approach

• When you login, you enter your username and password the way you normally would, then you're prompted to enter a (periodically changing) security code from your hardware token. Successfully entering that code proves you're in possession of the token ("something you have")

• Security of this approach depends on controlling knowledge of the magic "seed" values associated with each token

• There have been problems when it comes to keeping those seeds secret...
RSA finally comes clean: SecurID is compromised

RSA Security will replace almost every one of the 40 million SecurID tokens currently in use as a result of the hacking attack the company disclosed back in March. The EMC subsidiary issued a letter to customers acknowledging that SecurID failed to protect defense contractor Lockheed Martin, which last month reported a hack attempt.

SecurID tokens are used in two-factor authentication systems. Each user account is linked to a token, and each token generates a pseudo-random number that changes periodically, typically every 30 or 60 seconds. To log in, the user enters a username, password, and the number shown on their token. The authentication server knows what number a particular token should be showing, and so uses this number to prove that the user is in possession of their token.

The exact sequence of numbers that a token generates is determined by a secret RSA-developed algorithm, and a seed value used to initialize the token. Each token has a different seed, and it’s this seed that is linked to each user account. If the algorithm and seed are disclosed, the token itself becomes worthless; the numbers can be calculated in just the same way that the authentication server calculates them.

arstechnica.com/security/2011/06/06/rsa-finally-comes-clean-securid-is-compromised/
"Value" Proposition?

• Hard cryptographic tokens are tangible object (like an office key); straightforward to account for, easy to issue at the same time you're issuing keys, etc.

• Easy to use: "When you need a code to login, check your fob"

• HOWEVER, software running on smart phone can perfectly emulate a hard token, with nothing extra to carry around.

• If you do go with soft tokens, there's no hardware to buy and distribute, or lose/break (and have to replace)

• Are the days of traditional hard tokens fading fast?
8. Phone-Based Multifactor
This Is The Year of Phone-Based Multifactor

• There's no doubt in my mind that this is the year of phone-based multifactor authentication

• If you're going to only think about one sort of multifactor technology, I'd suggest focusing on phone-based multifactor.

• Consider the following commercial phone-based multifactor rollouts...
2-step verification

Help keep the bad guys out of your account by using both your password and your phone.

Get Started

2-step verification adds an extra layer of security to your Google Account

In addition to your username and password, you'll enter a code that Google will send you via text, voice call, or our mobile app.

How it works

1. **Enter your password**
   Whenever you sign in to Google you'll enter your username and password as usual.

2. **Enter a code from your phone**
   Then, you'll be asked for a code that will be sent to you via text, voice call, or our mobile app.

Keep it simple
During sign in, you can tell us not to ask for a code again on that particular computer. You'll still be covered, because we'll ask for codes when you or anyone else tries to sign in to your account from other computers.
How do I set up two-step verification?

Set up two-step verification at My Apple ID (appleid.apple.com):

1. Select "Manage your Apple ID" and sign in.
2. Select "Password and Security."
3. Under Two-Step Verification, select Get Started and follow the onscreen instructions.

How does it work?

When you set up two-step verification, you register one or more trusted devices. A trusted device is a device you control that can receive 4-digit verification codes using either Find My iPhone notifications or SMS to verify your identity.

Then, any time you sign in to manage your Apple ID at My Apple ID or make an iTunes, App Store, or iBookstore purchase from a new device, you will need to enter both your password and a 4-digit verification code as shown below.

![Illustration of sign in process]

You enter your Apple ID and password as usual. We send a verification code to one of your devices. You enter the code to verify your identity and complete sign in.

After you sign in, you can manage your account or make purchases as usual. Without both your password and the verification code, access to your account will be denied.

You will also get a 14-digit Recovery Key for you to print and keep in a safe place. You will use your Recovery Key to regain access to your account if you ever lose access to your devices or forget your password.
Introducing Login Approvals
by Andrew Song for Facebook Engineering (Notes) on Thursday, May 12, 2011 at 9:58am 📚

Facebook has always been committed to both protecting our users’ account and information, as well as giving them more control over their Facebook experience. From our User Operations team, who work to re-secure compromised accounts, to the Engineering team that designs and implements new security features like login notifications, one-time passwords, and remote session management, everyone at Facebook is working to ensure users have a safe, enjoyable experience.

Even interns like myself are tasked with big projects to help improve account security. Instead of working on mundane tasks and simple problems, interns are given high-impact assignments that reach out to hundreds of millions users every time they use Facebook.

Today, we’re announcing our newest opt-in security feature that I’ve worked to build over the past few months: Login Approvals.

![Login Approvals](image)

**What is Login Approvals?**

Login Approvals is a security feature that requires you to enter a code that we text to your phone when you log in from an unrecognized computer. You can enable this feature in a few simple steps:

1. If you ever lose access to your phone, you can always return to a previously-recognized computer to regain access to your account.
2. Note: You’ll need to have your mobile phone with you to complete this process.
The Official Microsoft Blog

News & Perspectives

TechNet Blogs > The Official Microsoft Blog > Microsoft Account Gets More Secure

Microsoft Account Gets More Secure

17 Apr 2013 9:00 AM

Over the next couple days we will roll out a major upgrade to Microsoft account, including optional two-step verification to help keep your account more secure.

Microsoft has increasingly focused on delivering connected devices and services that are currently used by more than 700 million people around the world. A Microsoft account is the key that unlocks your experience across these products—from your Windows PC to your Windows Phone, from Xbox to Outlook.com, from SkyDrive and Skype to Office and much more.

Given this critical role for Microsoft account, we remain vigilant in working hard to protect your account, which is why we’re adding an option so you can enable two-step verification to further protect yourself. You should see this option show up in your account in the next few days. You can enable this capability at https://account.live.com/proofs/Manage.

One account connects your digital world

A Microsoft account makes your experiences on devices and services more personal and relevant. When you sign in to any device or service with your Microsoft account, your personal settings, contacts and other information meet you there. It keeps you connected to the
Logging in with your PayPal user name and password is secure—but if you’re looking for an additional layer of protection, the Security Key might be for you.

The Security Key generates a random security code that you enter along with your PayPal username and password. It’s easy to use, and it even works with your eBay account.

We can send your security code to your mobile phone by SMS message or you can order a credit-card size Security Key to carry with you.

- There’s no fee to use your mobile phone as your Security Key but standard text messages rates apply when you receive a secure code by SMS. Check with your mobile provider for details.
- The portable, credit-card size Security Key has a one-time fee of $29.95.

Get one today or visit the FAQs page for more information.
Yahoo! Introduces Stronger User Authentication – Second Sign-in Verification

Posted December 13th, 2011 at 10:00 am by HuongT Categories: General

Online account compromise occurs regularly and will probably continue in the foreseeable future. Often hackers employ botnets to hijack accounts from millions of unsuspecting consumers. In turn, these hijacked accounts are sold and used to initiate email scams, with messages seemingly sent from the legitimate account owners to their contacts. Scams can range from the less harmful “check this cool stuff” advertising spam to the more damaging “send money to help your friend” email. Worse still, the scam emails may contain malicious links that lead to the installation of malware on computers, which then makes the computers part of the ever-growing botnet networks.

To thwart account compromise, Yahoo! is introducing a stronger user authentication feature that aims to prevent account hijackers with a stolen password from accessing a person’s account. If you have a Yahoo! account, you can now further protect it by activating this new second sign-in verification feature from Yahoo! Account Info. As part of the process, you will be required to add a mobile phone number to your account and verify it via SMS.

Once the feature is turned on, any suspicious account sign-in attempt will be challenged by a second sign-in verification beyond the initial password validation. To confirm the legitimacy of the sign-in attempt, you or the hijacker will have to answer your account security question or enter a verification code that will be sent to your mobile phone. Presumably, only you, as the legitimate user, can sign in.
Point of Commonality: Mobile Phones

• If you look at the multifactor services that Google, Apple, and all the other commercial services are deploying, there's one factor that really jumps out: they're ALL deploying mobile phone-based solutions.

• Using a mobile phone-based solution has a couple of advantages:

  -- It is cost effective

  -- It also serves as a second independent channel, making it harder for malware to subvert your authentication process (even if malware is sniffing the traffic on your laptop and your broadband connection, hopefully it isn't also sniffing the traffic on your phone and its cellular connection)
Smart Phone Methods Are Not "All Created Equal"

- **Software Crypto Token:** Some just emulate a hardware one-time-crypto token in software on the smart phone, and you need to copy the six or eight digits that it displays when you want to login. Biggest advantage of this approach? No connectivity required.

- **Auth Code Pushed Via Text Message:** In this approach, when you go to login, a new numeric code is pushed to your phone. You enter the code to prove that you have possession of the phone. Biggest advantage of this approach? Works on any phone able to do text messaging (smart phone is NOT required)

- **Duo Push:** Proprietary approach that uses a cryptographically secure application running on the user's smart phone – just push "Approve" or "Deny" when asked if you'd like to proceed to login

- There are other options, too; see: https://www.stsauver.com/joe/phone-2fa.html
Example of the Duo Security Approve/Deny Choice
9. PKI, Client Certs And USB-Format PKI Hard Tokens
Understanding Client Certificates

• I don't really have time to give you a thorough introduction to client certificates today during today's brief session, but I do have a three hour tutorial from Security Professionals 2012 that you can go through at your leisure, see: "Client Certificates: A Security Professionals 2012 Preconference Seminar," https://www.stsauver.com/joe/secprof2012/

• For now, just think of client certificates as binding an identity (such as your email address) to a pair of cryptographic keys, one key that's publicly shareable, and a corresponding one that's secret. If you've got those credentials handy, you can use them for things like serving as a 2\textsuperscript{nd} factor for login, or for digitally signing or encrypting email.

• The question is: how can you keep those credentials readily available for use, yet not have them be at constant risk of being stolen or misused by some hacker/cracker?
The Solution: PKI Hard Tokens and Smart Cards

• PKI hardware tokens normally come in two formats: smart cards (the size and shape of a credit card), and USB-format PKI hard tokens (which look just like a regular USB-format thumb drive).

• While these cards and tokens may not look like anything special, they actually are – these devices contain special tamper-resistant cryptographic storage and federally-certified cryptographic processing capabilities.

• Cryptographic credentials stored on PKI hard tokens can be configured to be useable but non-exportable, unlike cryptographic credentials stored as conventional files on a hard disk or thumb drive. This makes it difficult for hackers to steal your cryptographic credentials. Using a USB-format hard token or smart card also makes it easy to carry your credentials with you wherever you go.
Phone Based Multifactor vs. Client Certs on Hard Tokens

• While phone-based multifactor is focused just on two factor authentication, using client certificates enables two factor authentication while also enabling encryption and digital signatures.

• For example, if you have a client certificate you can digitally sign or encrypt email using S/MIME, and you can also digitally sign contracts, reports or other documents.

• Of course, you could also use an alternative encryption system, like PGP/GPG for encryption and signing, too
Smart Cards As A Basis for School ID Cards

• Another potential advantage of using client certificates on smart cards as your multifactor choice is the fact that smart cards can serve as the basis for a school ID card, containing not just the student's client certificate, but also all the things you'd normally find on a school ID card, such as the user's name and picture, an identification number, a bar code, year in school, etc.

• Of course, one complexity of deploying client certificates in smart card format is that you need to factor in the cost of deploying smart card readers, not just for school desktops and laptops, but also potentially for any home users who need to use smart card auth.

• Slick-sided mobile devices (such as smart phones or tablets) can also be a challenge if there's no easy way to add a smart card reader.
So Why Isn't Everyone Doing PKI?

• PKI tends to be fairly complex and isn't much fun to deploy.

• (Non)-integration with mobile devices has also really hurt PKI. (Where do you plug a USB key or smart card into an iPhone?)

• If someone suggests deploying PKI to you, you may want to run away from them.
10. Yubikey
What Is A Yubikey?

• Small USB device; two different basic form factors are available

• Normally either lives permanently in your USB port (nearly flush), or on your key chain

• ~$25 (quantity one)

• Emulates a keyboard, generating a code when you touch it

• Very promising and easy-to-use technology

image credit: http://www.thepowerbase.com/wp-content/uploads/2012/03/yubikey_nano.jpg
11. Biometrics
The "Forgotten Step Child:" Biometric Methods

• Use your fingerprints, an image of your iris, or other physical characteristics (such as a voice sample) to prove "you're you"

• Requires a means to collect biometric data:
  -- fingerprint or palm reader
  -- iris camera
  -- facial recognition camera
  -- audio capture device (microphone)
  -- etc.

• Integrated into...
  – Dedicated biometric systems
  – iPhone 6 TouchID
  – Windows Hello in Windows 10
  – Many others...

• Perfect for elementary and secondary school use? Or creepy and Big-Brother-ish?
In A California School District...

FACIAL RECOGNITION ‘BIOMETRICS’ COMING TO CALIFORNIA SCHOOL DISTRICT
But In Florida...

1002.222   Limitations on collection of information and disclosure of confidential and exempt student records.—

(1) An agency or institution as defined in s. 1002.22(1) may not:

(a) Collect, obtain, or retain information on the political affiliation, voting history, religious affiliation, or biometric information of a student or a parent or sibling of the student. For purposes of this subsection, the term “biometric information” means information collected from the electronic measurement or evaluation of any physical or behavioral characteristics that are attributable to a single person, including fingerprint characteristics, hand characteristics, eye characteristics, vocal characteristics, and any other physical characteristics used for the purpose of electronically identifying that person with a high degree of certainty. Examples of biometric information include, but are not limited to, a fingerprint or hand scan, a retina or iris scan, a voice print, or a facial geometry scan. Notwithstanding the provisions of this paragraph, a school district that used a palm scanner system for identifying students for breakfast and lunch programs on March 1, 2014, may continue to use the palm scanner system through the 2014-2015 school year.

What Do YOU Think About Biometrics in Oregon Elementary and Secondary Schools?

• Promising technology, well worth investigation and potential adoption? Or privacy invasive/creepy?

• Are there parallels in how Oregon handled RFID student tracking technology in the schools?

See http://www.oregonlaws.org/ors/339.890

[Anyone know if the State Board ever drafted the rules contemplated in that statute?]

• Let's move on and talk a little about non-adoption of multifactor authentication.
12. Non-Deployment
Deploying Multifactor At Scale

• There are different ways you can deploy multifactor authentication.

• You can deploy it just for special case situations (such as high risk users, or for particularly sensitive systems). You may already do that.

• We are particularly interested in seeing multifactor deployed at scale, e.g., ubiquitously -- for everyone, everywhere.

• But we're still not seeing the ubiquitous deployment we'd expected.

• WHY?
If Passwords Alone Are Awful, Why Aren't YOU Using Multifactor?

• No one knows better than you why you're not currently using MFA everywhere at your school. If you could simply tell us, that would be great.

• Unfortunately, we believe that at least in some cases, sites that aren't doing MFA may not have thought much about why they're not doing MFA (or at least they may not be able to articulate why).

• Therefore, we'd like to suggest a few potential reasons, and then see if any of these reasons resonate with you.

• Please speak up if any of these do strike a chord with you....
"MFA's On Our List, We're Just Really Busy"

• This may be the most common reason why at least some sites haven't done MFA yet: they're just really busy with lots of other projects. Is "I'm too busy" the main holdup for MFA at your site?

• Do you want to do MFA, but worry that deploying MFA would take too long or demand too much in the way of staff resources?

• If that's the case for your site, how long (order of magnitude) do you think deploying MFA would actually take? And what's a higher priority on your to-do list? How can we make deployment easier, or a higher priority?
"MFA's Too Expensive" (Or Is It?)

• Another commonly heard (historical) reason for not deploying MFA broadly was that it was "too expensive."

• That may have been true at one point, but these days the out-of-pocket cost for at least some MFA enterprise solutions is under a dollar per person per year. That's pretty cheap.

• Some of us may even have accounts from 3rd party cloud providers (such as Google) where we can enable use of MFA for free.

• And yet, somehow, many sites (and many users) still don't use it. So is money really the issue?
For Financial Comparison Purposes...

• Schools routinely spend $1/user/year (or more) on antivirus software. Why? Well, most sites worry a LOT about malware (bots, worms, trojan horses, etc.)

• But isn't phishing *just* as big a deal? Wouldn't it be worth $1/user/year to make (most) phishing go away, too?

• And how much do we spend recovering from plain old password failures? Wouldn't it make more sense to spend a *little money* on MFA to prevent breaches rather than a *lot of money recovering* from phishing attacks?

• What do YOU think? Is MFA still too expensive? How much would you and your site be willing to pay for MFA?
"MFA's Too Big A Pain To Use" (Or Is It?)

• If you login many times a day and you needed to copy six or eight secret digits from a hard token each time you did so, I could easily see that quickly becoming a huge pain.

• These days however, MFA has become easier to use (just push "OK" on a smart phone while logging in, for example), and in other cases, use of risk-based approaches means that MFA won't "bug you" at all unless you're doing something "unusual" (or something particularly "significant").

• Thus MFA isn't as painful to use as it once was – is it? What do those of you in the audience think? Is MFA still too painful to routinely use? If so, in what way? How could we make it easier for you to use?
"It's Not About **Routine Use**..."

- Another thing I've heard is that MFA isn't too bad when it comes to *routine* use, its the problems that MFA can cause when things are *unusual* that worry folks:

  -- I forgot my MFA device at home, what do I do now?

  -- I just got a new phone. How do I *update the devices that the MFA system uses for me*?

  -- If I use a cloud-based MFA solution, what happens if our site gets DDoS'd and we can't access the cloud?

- Are exceptions really the roadblock? Do we need to focus on making sure that failure paths resolve painlessly?
"We Don't Have Anything Top Secret"

• This is another commonly heard comment... namely, that from a risk management POV, MFA is "overkill" for "regular users."

• At sites where this is the case, you'll often see "targeted deployments:" "we'll just do MFA for high risk accounts only" (or comments to the effect that "nobody's really interested in just plain old student accounts," etc.)

• In fact, however, we know that even a "mere" student account can still be leveraged to send spam, or it can be used as a stepping stone for attacks against higher-value assets. Even "just" student accounts really can matter.

• Or consider faculty/staff access that's able to be used to access/change student records or direct deposit information – employees likely really need MFA protection for their own accounts.
"We'll Do MFA When Everyone Else Does"

• This is what is sometimes referred to as the "herd phenomenon" or "critical mass problem" in education.

• That is, at least some sites aren't willing to adopt a new technology until it becomes a well accepted practice for education as a whole (or at least well accepted by their local peer cohort institutions).

• Of course, this has the potential to cause deadlocks unless/until you can get a critical mass of schools or districts willing to take a leadership role and set the example for others...

• If MFA is the right thing to do, and important, is your site willing to be an MFA leader rather than an MFA follower?
"I Can't Tell What Sort of MFA I Should Do!"

• Are there just too many MFA possibilities?

• Are you confused about what product or technology you should choose?
  
  – Traditional hard tokens?
  – Smart phone-based solutions?
  – PKI smart cards
  – Yubikey
  – Biometrics?
"MFA Can't Totally Prevent All Authentication Risks, So Why Bother?"

• Sometimes people are profoundly disappointed that MFA isn't a magic bullet that will perfectly protect all users against all possible authentication-related attacks.

• For example, hypothetically, at least some "man-in-the-browser" attacks may continue to work, even if users are using MFA (e.g., the user may *think* they're confirming access to their secured site, but in reality a third party may be intercepting the user's MFA input and using it for their own nefarious purposes)

• Are we really going to let a "quest for the perfect" prevent us from making genuine meaningful progress?
"Using MFA *Doesn't* Eliminate Passwords!"

• Some sites hate passwords and may have hoped that deploying "multifactor authentication" would somehow let them completely eliminate passwords.

• Because passwords normally remains as half of the MFA process, doing MFA usually *doesn't* mean that you'll be "eliminating passwords."

• This means that doing MFA actually means you end up with passwords (which you hate), PLUS potentially something else, too. That's not really what folks would prefer, I suspect.

• Did I capture this one correctly? Is this the "big deal" that's delaying deployment of MFA at your school?
MFA Doesn't *Have* to Include Passwords

• If you wanted to, you could try a password-less multifactor combination.

• One option might be something you have (like a phone-based method) plus some sort of biometric factor?

• But does your school allow your students to carry a phone during the school day?

• And are biometrics acceptable to your students, parents and employees?
"There Are Too Many Services That Need Protection With MFA!"

• For example, hypothetically you might want to secure "enable" access to your routers, and "root" access to large shared systems, and teacher access to your VPN, and web access to your student portal system, and...

• If you have to implement MFA support for campus services on a service-by-service basis, that can feel daunting.

• But what if you could secure broad chunks of your infrastructure in one fell swoop? You could, if you did some types of federated authentication...
Might MFA Actually Make Some People Feel Paradoxically Less Secure?

• Multifactor authentication is meant to, and generally does, eliminate at least some risks. Doing MFA should make us feel MORE secure.

• However, human minds are funny things. Is it possible that MFA paradoxically makes us feel LESS secure?

• After all, doing MFA may make users think more about the possibility that their accounts may be at risk:
  -- "Why do I need MFA?" Answer: "There must be really serious attacks going on against MY accounts! OMG!"
  -- "I feel better knowing that one of my accounts is secured with MFA... but what about all my other sensitive accounts??? All my other accounts don't use MFA! OMG!"
Ultimately, There Can Be Multiple Reasons Why a Site Doesn't Deploy Multifactor Authentication

• Some of those reasons may relate to people/processes/funding; other issues may relate to technology, or something else entirely.

• The critical reason (or mix of reasons) may vary widely from site to site, and may evolve over time.

• We know (or at least we've know we've repeatedly heard) that sometimes folks think that the decision to deploy (or to not deploy) multifactor authentication is one that rests with just one person, perhaps the school's lead cyber security person, or the Superintendent or Principal or School Board. We believe that's an oversimplified decision-making model.
Some of the People Who **May** Gate the Decision To Deploy (Or To NOT Deploy) Multifactor Auth

- **Superintendent/Board/Principal/Other Leadership:** Is infosec a district or school-wide priority? Is there interest in (and funding for) a multifactor initiative?

- **IT Staff:** Is the multifactor technology easy to integrate and use with the systems that IT staff administer?

- **Teachers:** Who will help them use it? Who will produce local documentation and do local trainings for them on it?

- **Parents:** How do parents feel? Better? (more secure?) Worse? (big hassle to fix a non-existent problem?)

- **Students:** Can we make it work for them, too?

- Who did I accidentally overlook?
Suggestion...

- If you're interested in deploying multifactor authentication at your school, but things haven't been going very fast to-date, could it be that you don't have all the right folks talking (yet)?

- Would it help to be able to trade notes and share war stories with others from the community?
Thanks For the Chance to Talk Today!

• Are there any questions?