Welcome to

3. User Accounts

KEA Kompetence Computer Systems Security 2023

Henrik Kramselund he/him han/ham xhek@kea.dk @kramse

Slides are available as PDF, kramse@Github

3-user-accounts.tex in the repo security-courses

Goals for today

Todays goals:

- Talk about user accounts in general

Photo by Thomas Galler on Unsplash

Plan for today

Subjects

- What are user accounts – user ID
- Securing Administrative User Accounts
- Securing Normal User Accounts
- Databases: RDBMS, PostgreSQL, Deadlocks

Exercises

- Databases - discussion about Relational Database Management System RDBMS Model and NoSQL

Reading Summary

MLSH SectionI: Setting up a Secure Linux System

Chapter 1: Running Linux in a Virtual Environment

Chapter 2: Securing Administrative User Accounts

Chapter 3: Securing Normal User Accounts

Separation of duty ns function

> **Separation of duties** (SoD; also known as Segregation of Duties) is the concept of having more than one person required to complete a task. In business the separation by sharing of more than one individual in one single task is an internal control intended to prevent fraud and error.

Quote from https://en.wikipedia.org/wiki/Separation_of_duties

> **Separation of function**. Developers do not develop new programs on production systems because of the potential threat to production data.

Computer Security, Matt Bishop, 2019

Danish: Funktionsadskillelse

Accuracy vs disclosure

Lipner five commercial requirements:

1. Users will not write their own programs, but use existing production software.
2. Programmers develop and test applications on a nonproduction system, possibly using contrived data.
3. Moving applications from development to production requires a special process.
4. This process must be controlled and audited.
5. Managers and auditors must have access to system state and system logs

Available from

https://csrc.nist.gov/CSRC/media/Publications/conference-paper/1982/05/24/proceedings-5th-seminar-dod-computer-security-initiative/documents/1982-5th-semina

pdf

The Biba Model

Ken Biba (1977) proposed three different integrity access control policies.

1. The Low Water Mark Integrity Policy
2. The Ring Policy
3. Strict Integrity

- All assume that we associate integrity labels with subjects and objects, analogous to clearance levels in BLP.
- Only Strict Integrity had much continuing influence. It is the one typically referred to as the "Biba Model" or "Biba Integrity."

Example page 178 mentions that this was implemented in FreeBSD

Lipners Integrity Matrix Model

Lipner provides two security levels, in the following order (higher to lower):

- Audit Manager (AM): system audit and management functions are at this level.
- System Low (SL): any process can read information at this level.

He similarly defined five categories:

- Development (D): production programs under development and testing, but not yet in production use
- Production Code (PC): production processes and programs
- Production Data (PD): data covered by the integrity policy
- System Development (SD): system programs under development, but not yet in production use
- Software Tools (T): programs provided on the production system not related to the sensitive or protected data

Non-Discretionary Controls for Commercial Applications, Steven B. Lipner, IEEE Symposium on Security and Privacy, and Fifth Seminar on the DoD Computer Security Initiative, 1982

Lipners Integrity Matrix Model

Figure 7.

OBJECTS SUBJECTS	Prod. Data	Prod. Code	Dev. App. Prgm.	Dev. Sys. Prgm.	Tools	Sys. Prg.	Audit Trail
System Mgt. & Audit	R	R	R	R	R	R	RW
Production Users	RW	R				R	W
Application Programmers			RW		R	R	W
System Programmers				RW	R	R	W
System Control	RW	RW	RW	RW	RW	RW	W

Figure 7. Effects of the Commercial Lattice

Non-Discretionary Controls for Commercial Applications, Steven B. Lipner, IEEE Symposium on Security and Privacy, and Fifth Seminar on the DoD Computer Security Initiative, 1982

Lipners Integrity Matrix Model

	OBJECTS							
SUBJECTS	**Production Data**	**Production Code**	**Develop. Code & Test Data**	**Develop. Sys. Prog.**	**S/W Tools**	**Sys. Prog.**	**Re-pair Code**	**Audit Data**
System Mgr.	R	R	R	R	R	R	R	RW
Prod. User	RW	R				R		W
App'n. Prog.			RW		R	R		W
Sys. Program				RW	R	R		W
Sys. Control	RW	RW	RW	RW	RW	RW	RW	W
Repair	RW	R				R	R	W

Figure 12. Effects of Commercial Lattice Model with Integrity

Non-Discretionary Controls for Commercial Applications, Steven B. Lipner, IEEE Symposium on Security and Privacy, and Fifth Seminar on the DoD Computer Security Initiative, 1982

Clark-Wilson Integrity Model

A **well-formed transaction** from one consistent state to another consistent state.

- Constrained Data Items: CDIs are the objects whose integrity is protected
- Unconstrained Data Items: UDIs are objects not covered by the integrity policy
- Transformation Procedures: TPs are the only procedures allowed to modify CDIs, or take arbitrary user input and create new CDIs. Designed to take the system from one valid state to another.
- Integrity Verification Procedures: IVPs are procedures meant to verify maintainance of integrity of CDIs.

A Comparison of Commercial and Military Computer Security Policies, David D. Clark and David R. Wilson, 1987

Clark-Wilson Integrity Model

> The model uses a three-part relationship of subject/program/object (where program is interchangeable with transaction) known as a triple or an access control triple. Within this relationship, subjects do not have direct access to objects. Objects can only be accessed through programs

A Comparison of Commercial and Military Computer Security Policies, David D. Clark and David R. Wilson, 1987

See also https://en.wikipedia.org/wiki/Clark%E2%80%93Wilson_model

MLSH Chapter 2: intro

Securing Administrative User Accounts
Managing users is one of the more **challenging** aspects of IT administration. You need to make sure that users can always **access their stuff** and that they can **perform the required tasks** to do their jobs. You also need to ensure that users' stuff is always **secure from unauthorized users** and that users **can't perform any tasks that don't fit their job description**.

Source: *Mastering Linux Security and Hardening* (MLSH), third edition

Can we spot the:

- Confidentiality, Integrity and Availability requirements
- Principle of Least Privilege

MLSH Chapter 2: contents

The specific topics covered in this chapter are as follows:

- The dangers of logging in as the root user
- The advantages of using sudo
- Setting up sudo privileges for full administrative users and for users with only certain dele- gated privileges
- Advanced tips and tricks to use sudo

Source: *Mastering Linux Security and Hardening* (MLSH), third edition

The dangers of logging in as the root user

> A huge advantage that Unix and Linux operating systems have over Windows is that Unix and Linux do a much better job of keeping privileged administrative accounts separated from normal user accounts. Indeed, one reason that older versions of Windows were so susceptible to security issues, such as drive-by virus infections, was the common practice of setting up user accounts with administrative privileges, without having the protection of the User Access Control (UAC) that's in newer versions of Windows.

Source: *Mastering Linux Security and Hardening* (MLSH), third edition

- Agreed, but I may be biased
- Mac OS X made it very simple to run administrative tasks, so you didn't need to run as root
- Modern Linux user interfaces make similar attempts with `pkexec`, `kdesudo`, `gksudo` etc.
- Windows is getting better, many organisations in DK are removing administrative access to regular users, even KEA

The advantages of using sudo

Used properly, the sudo utility can greatly enhance the security of your systems, and it can make an administrator's job much easier. With sudo , you can do the following:

- Assign certain users full administrative privileges, while assigning other users only the privi- leges they need to perform tasks that are directly related to their respective jobs.
- Allow users to perform administrative tasks by entering their own normal user passwords so that you don't have to distribute the root password to everybody and their brother.
- Make it harder for intruders to break into your systems. If you implement sudo and disable the root user account, would-be intruders won't know which account to attack because they won't know which one has admin privileges.
- Create sudo policies that you can deploy across an entire enterprise network, even if that network has a mix of Unix, BSD, and Linux machines.
- Improve your auditing capabilities because you'll be able to see what users are doing with their admin privileges.

Source: *Mastering Linux Security and Hardening* (MLSH), third edition

Why use Sudo conclusion

Main thing about sudo is that you do NOT give out the root password to anybody! They will use their own credentials and can be limited to single commands, scripts and even parameters. You could have a single sudoers file for your own organisation, that includes groups of servers, user groups etc.

Sidenote: sudo also has a number of CVEs unfortunately

Setting up sudo privileges

Chapter has multiple example of configuration. Mostly we add users to a predefined admin group, but for my personal systems I add named users with privileges - user *hlk*

As an exercise, if you haven't done before – make sure sudo works on your Kali and Debian. If it is pre-configured, check the settings, how did this happen?

Debian usually does NOT come with sudo installed, so -verb+apt install sudo+

- Lets do a few of the commands from chapter 2, check sudo on your systems
- Always use the command `visudo` to edit your sudoers file! Checks syntax, and you avoid locking yourself out!
- I recommend the book: *Sudo Mastery*, 2nd Edition by Michael W Lucas, https://www.tiltedwindmillpress.com/product/sudo-mastery-2nd-edition/

MLSH chapter 3

Enforcing strong password criteria You wouldn't think that a benign-sounding topic such as strong password criteria would be so contro- versial, but it is. The conventional wisdom that you've undoubtedly heard for your entire computer career says:

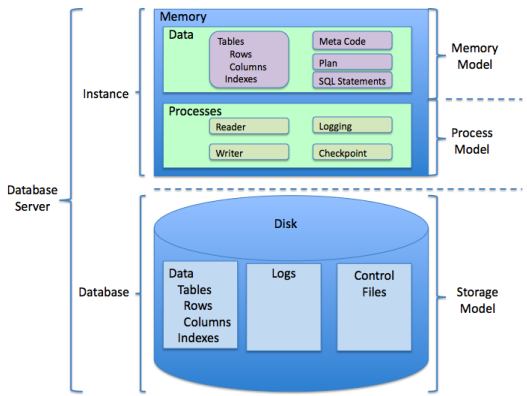
- Make passwords of a certain minimum length.
- Make passwords that consist of a combination of uppercase letters, lowercase letters, numbers, and special characters.
- Ensure that passwords don't contain any words that are found in the dictionary or that are based on the users' own personal data.
- Force users to change their passwords on a regular basis.

Source: *Mastering Linux Security and Hardening* (MLSH), third edition

Unix systems are mostly single-user today

- Fewer and fewer Unix systems, including Linux are multi-user systems today
- Chapter is still important, but less so for now, use it as a reference later
- Password policies are relevant for all systems, single or multi user
- Pro-tip use Ansible or similar to configure all systems

Relational Database Management System RDBMS



Relational Database Management System RDBMS is a common database architecture

Common examples MS-SQL, MySQL/MariaDB, PostgreSQL

Picture: By Scifipete - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=11506013

https://en.wikipedia.org/wiki/Relational_database#RDBMS

PostgreSQL security

	11	10	9.6	9.5	9.4
Channel binding for SCRAM authentication	Yes	No	No	No	No
Column level permissions	Yes	Yes	Yes	Yes	Yes
Default permissions	Yes	Yes	Yes	Yes	Yes
GRANT/REVOKE ON ALL TABLES/SEQUENCES/FUNCTIONS	Yes	Yes	Yes	Yes	Yes
GSSAPI support	Yes	Yes	Yes	Yes	Yes
Large object access controls	Yes	Yes	Yes	Yes	Yes
Native LDAP authentication	Yes	Yes	Yes	Yes	Yes
Native RADIUS authentication	Yes	Yes	Yes	Yes	Yes
Per user/database connection limits	Yes	Yes	Yes	Yes	Yes
ROLES	Yes	Yes	Yes	Yes	Yes
Row-Level Security	Yes	Yes	Yes	Yes	No
SCRAM-SHA-256 Authentication	Yes	Yes	No	No	No
Search+bind mode operation for LDAP authentication	Yes	Yes	Yes	Yes	Yes
security_barrier option on views	Yes	Yes	Yes	Yes	Yes
Security Service Provider Interface (SSPI)	Yes	Yes	Yes	Yes	Yes
SSL certificate validation in libpq	Yes	Yes	Yes	Yes	Yes
SSL client certificate authentication	Yes	Yes	Yes	Yes	Yes
SSPI authentication via GSSAPI	Yes	Yes	Yes	Yes	Yes

Feature overview security features in PostgreSQL
https://www.postgresql.org/about/featurematrix/#security

Deadlocks

Definition 7-1 A *deadlock* is a state in which some set of processes block, each waiting for another process in the set to take some action.

1. The resource is not shared (mutual exclusion)
2. An entity must hold the resource and block, waiting until another resource becomes available (hold and wait)
3. A resource being help cannot be released (no preemption)
4. A set of entities must be holding resources such that each entity is waiting for a resource held by another entity in the set (circular wait)

Often found in Relational Database Systems, if two processes want to update two tables, and each one has a write lock on one table, waiting for the write lock on the other

See also https://en.wikipedia.org/wiki/Deadlock

Common Discusssion

Databases - discussion about Relational Database Management System RDBMS Model and NoSQL databases, which ones do you and your company use?

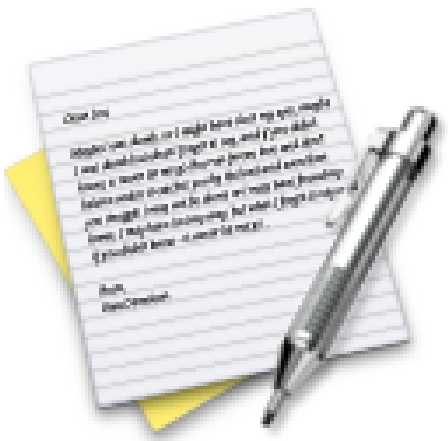
Exercise

Now lets do the exercise

⚠ Configure a Database - 20 min

which is number **19** in the exercise PDF.

Exercise



Now lets do the exercise

⚠ RBAC Access permissions on GitHub 30-45min

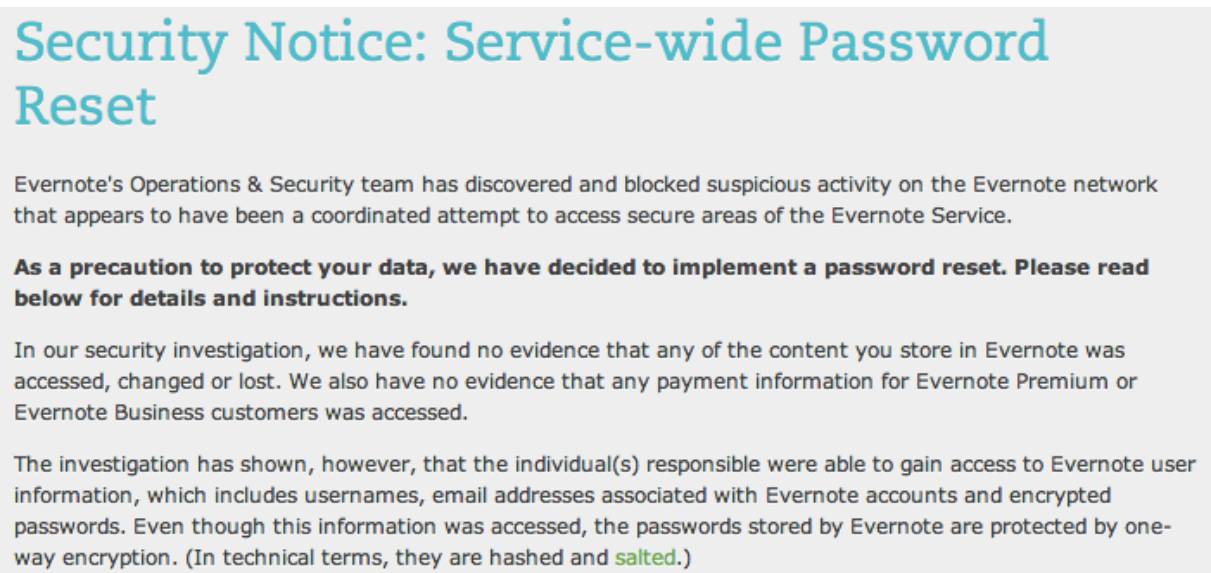
which is number **20** in the exercise PDF.

Passwords vælges ikke tilfældigt

Source: https://wpengine.com/unmasked/

Evernote password reset

What happens when security breaks?



Security Notice: Service-wide Password Reset

Evernote's Operations & Security team has discovered and blocked suspicious activity on the Evernote network that appears to have been a coordinated attempt to access secure areas of the Evernote Service.

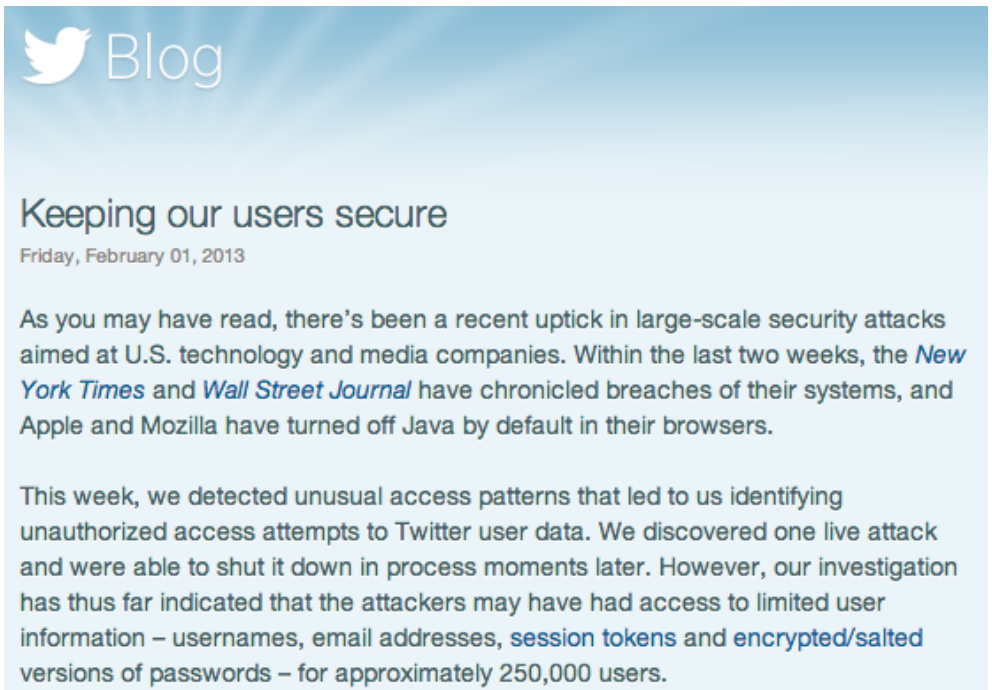
As a precaution to protect your data, we have decided to implement a password reset. Please read below for details and instructions.

In our security investigation, we have found no evidence that any of the content you store in Evernote was accessed, changed or lost. We also have no evidence that any payment information for Evernote Premium or Evernote Business customers was accessed.

The investigation has shown, however, that the individual(s) responsible were able to gain access to Evernote user information, which includes usernames, email addresses associated with Evernote accounts and encrypted passwords. Even though this information was accessed, the passwords stored by Evernote are protected by one-way encryption. (In technical terms, they are hashed and salted.)

Sources:
http://evernote.com/corp/news/password_reset.php

Twitter password reset



Blog

Keeping our users secure

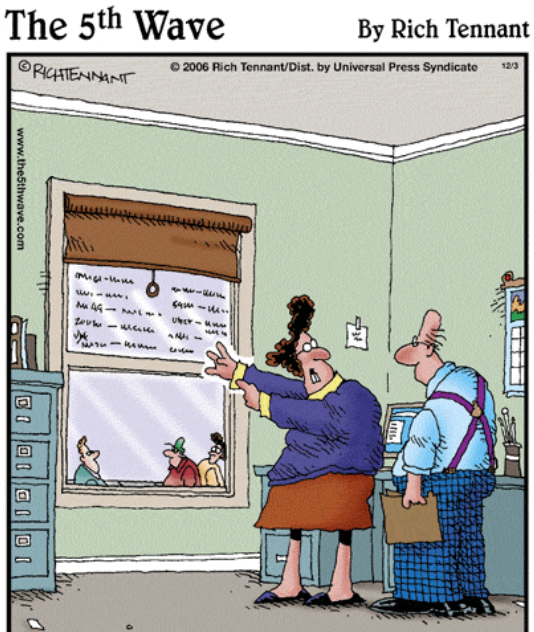
Friday, February 01, 2013

As you may have read, there's been a recent uptick in large-scale security attacks aimed at U.S. technology and media companies. Within the last two weeks, the *New York Times* and *Wall Street Journal* have chronicled breaches of their systems, and Apple and Mozilla have turned off Java by default in their browsers.

This week, we detected unusual access patterns that led to us identifying unauthorized access attempts to Twitter user data. We discovered one live attack and were able to shut it down in process moments later. However, our investigation has thus far indicated that the attackers may have had access to limited user information – usernames, email addresses, session tokens and encrypted/salted versions of passwords – for approximately 250,000 users.

Sources:
http://blog.twitter.com/2013/02/keeping-our-users-secure.html

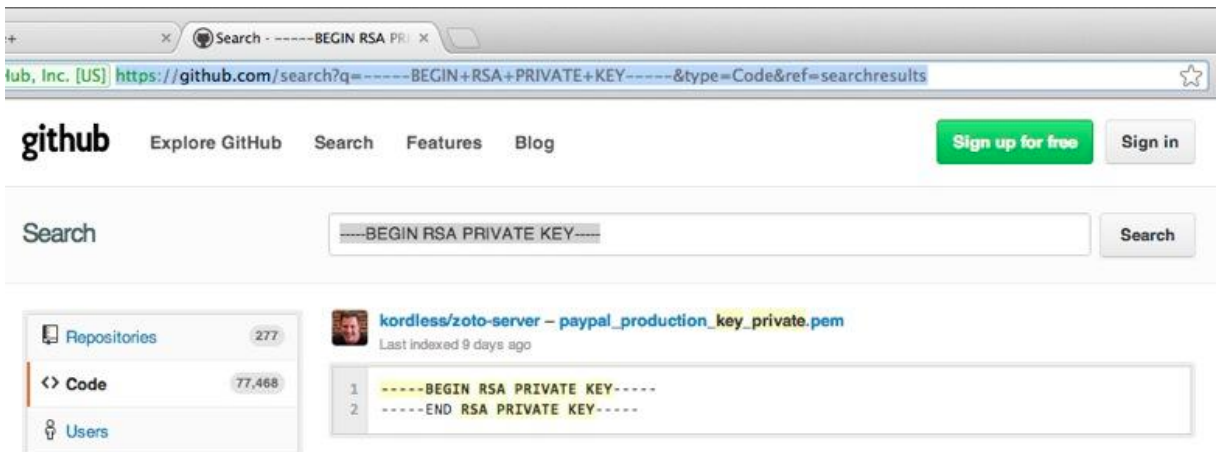
Saving passwords



"Don't be silly — of course my passwords are safe. I keep them written on my window, but then I pull the shade if anyone walks in the room."

Use some kind of Password Safe program

January 2013: Github Public passwords?



Sources:

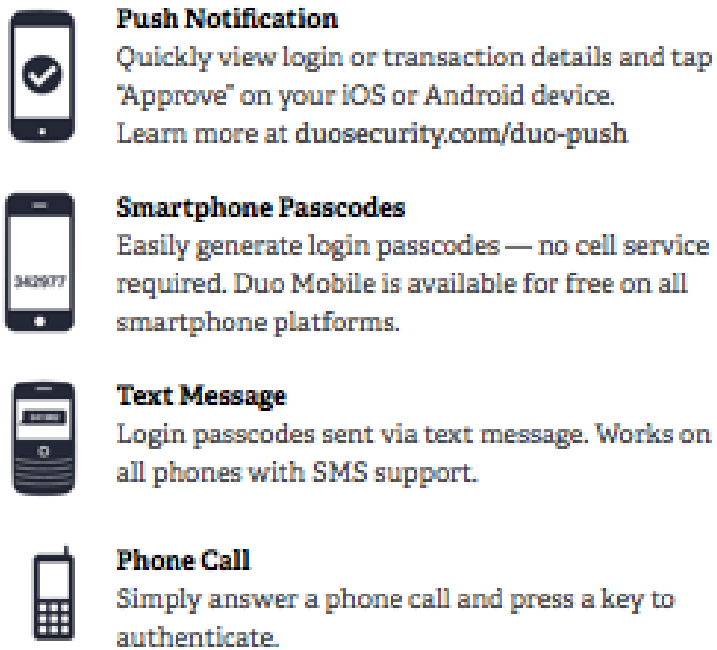
https://twitter.com/brianaker/status/294228373377515522

http://www.webmonkey.com/2013/01/users-scramble-as-github-search-exposes-passwords-security-details/

http://www.leakedin.com/

http://www.offensive-security.com/community-projects/google-hacking-database/

TwoFactor authentication: Example Duosecurity



Push Notification
Quickly view login or transaction details and tap "Approve" on your iOS or Android device.
Learn more at duosecurity.com/duo-push

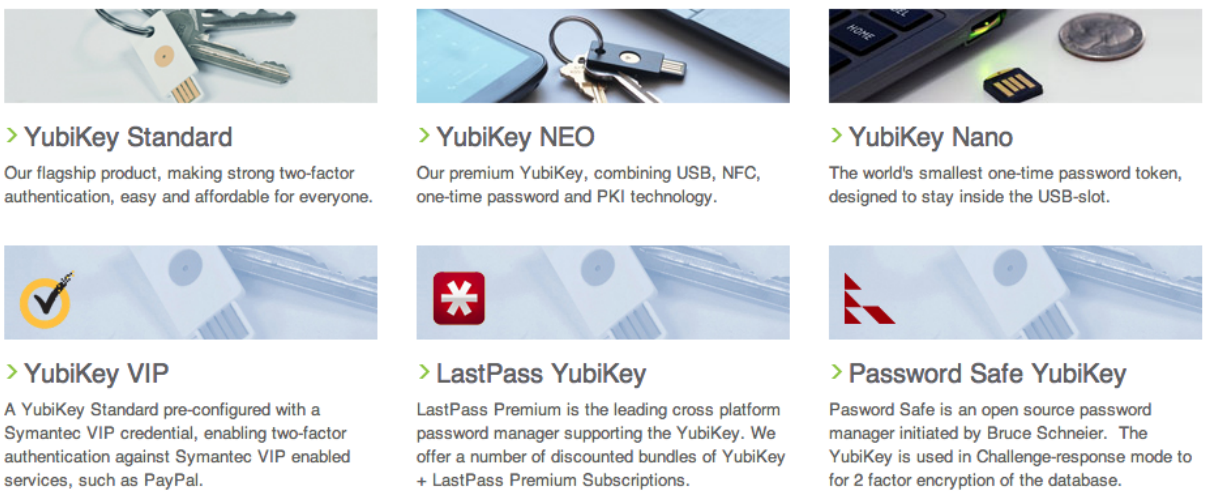
Smartphone Passcodes
Easily generate login passcodes — no cell service required. Duo Mobile is available for free on all smartphone platforms.

Text Message
Login passcodes sent via text message. Works on all phones with SMS support.

Phone Call
Simply answer a phone call and press a key to authenticate.

Source: https://www.duo.com/

Yubico Yubikey



› YubiKey Standard

Our flagship product, making strong two-factor authentication, easy and affordable for everyone.

› YubiKey NEO

Our premium YubiKey, combining USB, NFC, one-time password and PKI technology.

› YubiKey Nano

The world's smallest one-time password token, designed to stay inside the USB-slot.

› YubiKey VIP

A YubiKey Standard pre-configured with a Symantec VIP credential, enabling two-factor authentication against Symantec VIP enabled services, such as PayPal.

› LastPass YubiKey

LastPass Premium is the leading cross platform password manager supporting the YubiKey. We offer a number of discounted bundles of YubiKey + LastPass Premium Subscriptions.

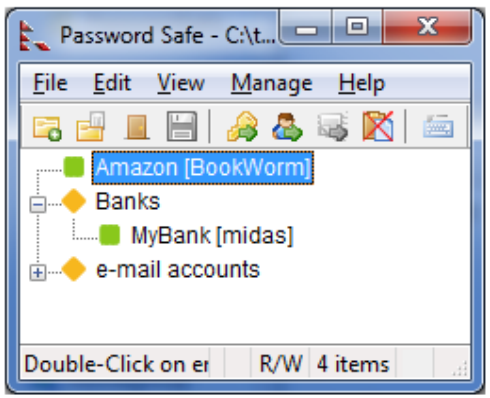
› Password Safe YubiKey

Pasword Safe is an open source password manager initiated by Bruce Schneier. The YubiKey is used in Challenge-response mode to for 2 factor encryption of the database.

A Yubico OTP is unique sequence of characters generated every time the YubiKey button is touched. The Yubico OTP is comprised of a sequence of 32 Modhex characters representing information encrypted with a 128 bit AES-128 key

http://www.yubico.com/products/yubikey-hardware/ and also https://wiki.debian.org/Smartcards/YubiKey4

Storing passwords



Use password managers – even though they also have security issues the overall improvement is great

PasswordSafe https://pwsafe.org/ – Note: research for yourself which password manager to use!

Apple Keychain provides an encrypted storage

Browsere, Firefox Master Password, Chrome passwords, ... who do YOU trust

Google looks to ditch passwords for good 2013

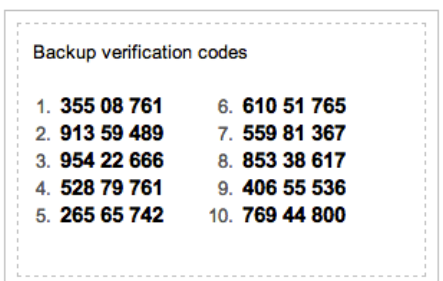
"Google is currently running a pilot that uses a YubiKey cryptographic card developed by Yubico

The YubiKey NEO can be tapped on an NFC-enabled smartphone, which reads an encrypted one-time password emitted from the key fob."

Source: http://www.zdnet.com/google-looks-to-ditch-passwords-for-good-with-nfc-based-replacement-7000010073/

Low tech 2-step verification

Printing code on paper, low level pragmatic


Backup verification codes

1. 355 08 761
2. 913 59 489
3. 954 22 666
4. 528 79 761
5. 265 65 742
6. 610 51 765
7. 559 81 367
8. 853 38 617
9. 406 55 536
10. 769 44 800

Login from new devices today often requires two-factor - email sent to user

Google 2-factor auth. SMS with backup codes

Also read about S/KEY developed at Bellcore **in the late 1980s**
http://en.wikipedia.org/wiki/S/KEY

Conclusion passwords: integrate with authentication, not reinvent

Integrate or develop?

From previous slide:

Conclusion passwords: integrate with authentication, not reinvent

Dont:

- Reinvent the wheel - too many times, unless you can maintain it afterwards
- Never invent cryptography yourself
- No copy paste of functionality, harder to maintain in the future

Do:

- Integrate with existing solutions
- Use existing well-tested code: cryptography, authentication, hashing
- Centralize security in your code
- Fine to hide which authentication framework is being used, easy to replace later

Cisco IOS password 2013

Title: Cisco's new password hashing scheme easily cracked

Description: In an astonishing decision that has left crytographic experts scratching their heads, engineer's for Cisco's IOS operating system chose to switch to a **one-time SHA256 encoding - without salt** - for storing passwords on the device. This decision leaves password hashes vulnerable to high-speed cracking - modern graphics cards can compute over **2 billion SHA256 hashes in a second - and is actually considerably less secure than Cisco's previous implementation.** As users cannot downgrade their version of IOS without a complete reinstall, and no fix is yet available, security experts are urging users to avoid upgrades to IOS version 15 at this time.

Reference: via SANS @RISK newsletter
http://arstechnica.com/security/2013/03/cisco-switches-to-weaker-hashing-scheme-passwords-cracked-wide-open/

NT hashes

NT LAN manager hash værdier er noget man typisk kan samle op i netværk

det er en hash værdi af et password som man ikke burde kunne bruge til noget - hash algoritmer er envejs

opbygningen gør at man kan forsøge brute-force på 7 tegn ad gangen!

en moderne pc med l0phtcrack kan nemt knække de fleste password på få dage!

og sikkert 25-30% indenfor den første dag - hvis der ingen politik er omkring kodeord!

ved at generere store tabeller, eksempelvis 100GB kan man dække mange hashværdier af passwords med almindelige bogstaver, tal og tegn - og derved knække passwordshashes på sekunder. Søg efter rainbowcrack med google

Pass the hash

Lots of tools in pentesting pass the hash, reuse existing credentials and tokens *Still Passing the Hash 15 Years Later* http://passing-the-hash.blogspot.dk/2013/04/pth-toolkit-for-kali-interim-status.html

> If a domain is built using only modern Windows OSs and COTS products (which know how to operate within these new constraints), and configured correctly with no shortcuts taken, then these protections represent a big step forward.

Source:
http://www.harmj0y.net/blog/penetesting/pass-the-hash-is-dead-long-live-pass-the-hash/ https://samsclass.info/lulz/pth-8.1.htm

John the ripper

> John the Ripper is a fast password cracker, currently available for many flavors of Unix (11 are officially supported, not counting different architectures), 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 flavors, supported out of the box are Kerberos AFS and Windows NT/2000/XP/2003 LM hashes, plus several more with contributed patches.

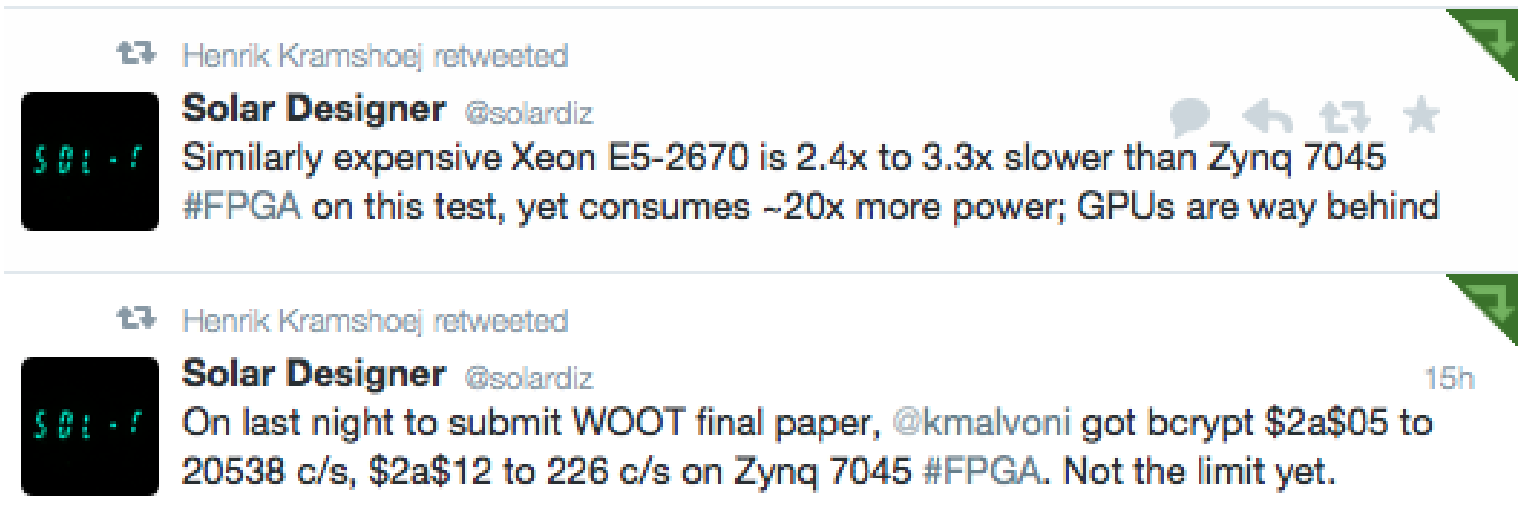
UNIX passwords kan knækkes med alec Muffets kendte Crack program eller eksempelvis John The Ripper http://www.openwall.com/john/

Cracking passwords

- Hashcat is the world's fastest CPU-based password recovery tool.
- oclHashcat-plus is a GPGPU-based multi-hash cracker using a brute-force attack (implemented as mask attack), combinator attack, dictionary attack, hybrid attack, mask attack, and rule-based attack.
- oclHashcat-lite is a GPGPU cracker that is optimized for cracking performance. Therefore, it is limited to only doing single-hash cracking using Markov attack, Brute-Force attack and Mask attack.
- John the Ripper password cracker old skool men stadig nyttig

Source:
http://hashcat.net/wiki/
http://www.openwall.com/john/

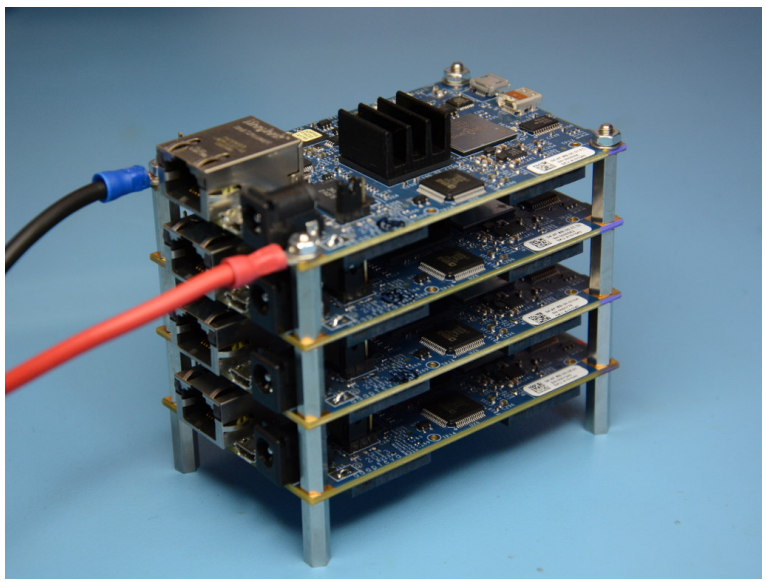
Parallella John



https://twitter.com/solardiz/status/492037995080712192

Expect specialized hardware to be used by NSA, GCHQ, and perhaps even organised crime

Stacking Parallella boards



http://www.parallella.org/power-supply/

Encryption key length

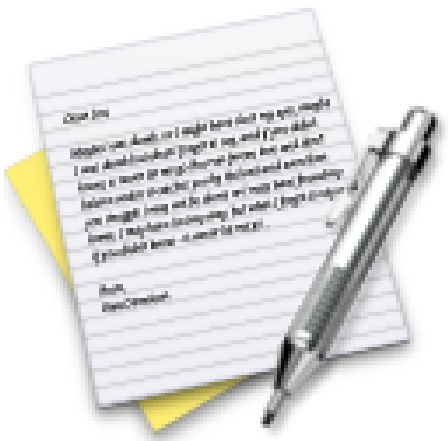
Encryption key lengths & hacking feasibility

Type of Attacker	*Budget*	Tool	*Time & Cost/Key 40 bit*	*Time & Cost/Key 56 bit*
Regular User	Minimal $400	Scavenged computer time FPGA	1 week 5 hours ($.08)	Not feasible 38 years ($5,000)
Small Business	$10,000	FPGA [1]	12 min.($.08)	556 days ($5,000)
Corporate Department	$300,000	FPGA ASIC [2]	24 sec. ($.08) 0.18 sec. ($.001)	19 days ($5,000) 3 hours ($38)
Large Corporation	$10M	ASIC	0.005 sec.($0.001)	6 min. ($38)
Intelligence Agency	$300M	ASIC	0.0002 sec.($0.001)	12 sec. ($38)

Source: http://www.mycrypto.net/encryption/encryption_crack.html

More up to date:
In 1998, the EFF built Deep Crack for less than $250,000
https://en.wikipedia.org/wiki/EFF_DES_cracker
FPGA Based UNIX Crypt Hardware Password Cracker - 100 EUR in 2006
http://www.sump.org/projects/password/

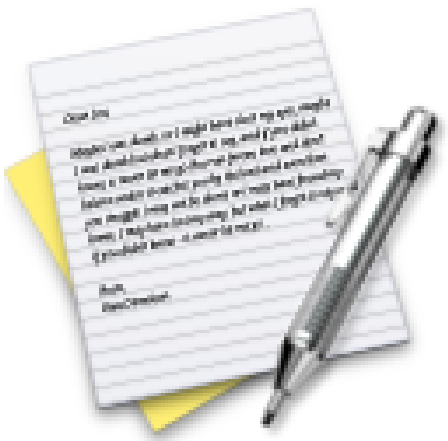
Exercise



Now lets do the exercise

⚠ Password Cracking 15min

which is number 27 in the exercise PDF.

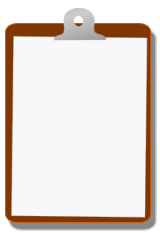


Now lets do the exercise

Configure SSH keys for more secure access

which is number **28** in the exercise PDF.

For Next Time



Think about the subjects from this time, write down questions

Check the plan for chapters to read in the books

Visit web sites and download papers if needed

Retry the exercises to get more confident using the tools