SSAS F2015

Systems Architecture and Security, TRIAL examination set, 2015.

What to hand in

You submit electronically, via LearnIT:

1. A text file with your answers to multiple choice questions, one line per answer: the question identifier, a colon, and the answer digit (format example below). You can submit only one answer for each question; multiple answers will be awarded zero points.
2. A pdf file containing the answers to discussion questions D1 and D2. It may contain text and figures in any form. If there are technical or logistical problems with submitting in this way, please turn to available personnel in order to find a backup solution.

Example multiple choice question:

> **Question C9-47**
>
> This is the question text. It is followed by the possible answers:
>
> 1. The first possible answer.
> 2. The second possible answer.
> 3. The third possible answer.
> 4. The fourth possible answer.

Example answer. You choose "The third possible answer":

```
C9-47:3
```

Questions begin on the next page.

Multiple-choice Questions

Your answers to the multiple choice questions counts 75% towards the final grade.

Security principles

Question C1-02

A company possesses two classes information: publically available information, which should be accessible from the internet, and confidential business plans, which ideally should be accessible only to management, and only when they are on premises. The IT department, in the name of simplicity, puts all information on one machine, making sure the web-server only serves directories with public information. This violates the security principle of:

1. Compartmentalisation
2. Minimum trust and maximum trustworthiness
3. Secure, fail-safe defaults
4. Open design

Question C1-04

Military intelligence services tend to keep highly classified information on machines with no physical connection to the internet. This is an example of adherence to the security principle of:

1. Minimum exposure
2. Traceability
3. Least privilege
4. No single point of failure

Question C1-06

Back when I was a programmer, everybody had root-access to the development server because the IT-guy was tired of being asked to reset people's passwords. This is an example of a violation of the security principle of:

1. Least privilege
2. Minimum exposure
3. No single point of failure
4. Complete mediation

Question C1-08

I write a web-app that I know will be accessible only from inside the ITU wired network. As such, I can't be bothered to encrypt sensitive information (passwords, grades) being sent to and from my app's server—only ITU staff has access to the wires anyway. This is a violation of the security principle of:

1. Usability
2. Minimum exposure
3. No single point of failure
4. Complete mediation

Question C1-10

The computer games SimCity III and Diablo both required and always-on internet connection to be playable, even in single-player mode, where the game otherwise did not need an internet connection. The connection was used to validate with the development companies that the game was a legitimate copy, as opposed to a pirated one. If the internet connection dropped for any reason, the game would be unplayable until it could re-connect to its home server and validate its legitimacy. This is a (very customer unfriendly) example of adherence to the security principle of:

1. Safe, fail-safe defaults
2. Minimum trust and maximum trustworthiness
3. Least privilege
4. Complete mediation

Question C1-12

Edward Snowden and Bradley Manning both accessed enormous amounts of information classified by the US Government. Their accesses were apparently neither logged nor constrained in ways beyond having access to a particular network. This is an example of violation of the security principle of:

1. Complete mediation
2. Traceability
3. Usability
4. No single point of failure

Question C1-14

The Danish "tystys-skandale", in which various celebrities' credit card transactions were leaked to gossip-magazine "Se og Hør"'s staff by insiders at the IBM subsidiary operating the computers behind the Danish credit card system. The inapropriate accesses to credit card information were apparently logged, and so identifying the culprits was fairly easy once the scandal broke. This is an example of adherence to the security principle of:

1. Traceability
2. Usability
3. No single point of failure
4. Least privilege

Question C1-16

Because of a misunderstanding about the semantics of the linux special device "/dev/urandom", a substantial number of consumer-grade routers will generate various cryptographic keys from what they think is a random number but in practice is allways 0. This is a gross violation of the security principle of:

1. Generating secrets
2. Simplicity
3. Single point of failure
4. Minimise trust and maximise trustworthiness

Question C1-18

In early 2007, Microsoft Corporation released Windows Vista. Vista had a security manager, which would ask the user to "cancel or ok" whenever, essentially, any application tried to do anything with disk or network. User's were quickly trained to just hit "ok" while cursing under their breath. This was an example of a violation of the security principle of:

1. Usability
2. Simplicity
3. Single point of failure
4. Minimise trust and maximise trustworthiness

Question C1-20

The "Simplicity" security principle helps because:

1. Simpler systems are easier to analyse and review.
2. Complex systems tends to rely on unsound cryptography.
3. Simpler systems exhibit economy of mechanism.
4. It doesn't; complex systems are actually harder to attack.

Question C1-22

"Open design" helps because:

1. Secrets are difficult to keep.
2. Open source systems historically have higher software-quality.
3. Secrets attract hackers.
4. The OPEN Group's 1987 memo changed the outlook of the industry.

Network Services

Question C3-01

In portscanning, a SYN scan is performed by sending the first packet of the TCP 3-way handshake. The port is closed if the response packet has the following flag(s) set:

1. SYN+ACK
2. ACK
3. SYN
4. RST

Question C3-03

In portscanning, SYN scans are preferable to TCP connect scans because:

1. SYN scans are less likely to be logged
2. SYN scans never trigger cascading RST packets
3. Connect scans are less likely to be logged
4. Connect scans may interrupt the normal operation of the target host

Question C3-05

When portscanning, an ACK scan is used to:

1. Investigate firewall rulesets used by the target host
2. Investigate TCP acknowledgment numbers used by the target host
3. Investigate TCP sequence numbers used by the target host
4. Distinguish between open and closed ports on the target host

Question C3-07

A stealth scan is so-called because

1. The scanned host will not deliver transmitted packets to the process bound to the scanned port
2. They are impossible to detect by firewalls
3. They are impossible to detect by intrusion detection systems
4. Transmitted packets appear to not originate from the adversary's host

Question C3-09

Match the following tools with their purpose:

1. nmap -> port scanner
2. OpenVAS -> vulnerability scanner
3. telnet -> remote login/terminal
4. tcpdump -> packet sniffer

Question C3-11

I use nmap to scan a web-server. This is the output:

```
Not shown: 991 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
80/tcp   open  http
111/tcp  open  rpcbind
443/tcp  open  https
2049/tcp open  nfs
6000/tcp open  X11
```

From this, we learn that:

1. The 991 closed ports indicate the server has been completely hardened
2. The open ports 22, 23, 80, 443 indicate that the server has been hardened with remote administration capabilities
3. The "open" ports all host services with exploitable vulnerabilities
4. The server is either spectacularly multi-purpose or not hardened.

Question C3-13

I run `sudo lsof -i` on my web-server, obtaining the following output:

```
mysqld     871    mysql   12u  IPv4   4844      0t0  TCP localhost:mysql (LISTEN
inetd      930     root    4u  IPv4   4917      0t0  TCP *:telnet (LISTEN)
inetd      930     root    5u  IPv4   4920      0t0  TCP *:ftp (LISTEN)
apache2   1242     root    3u  IPv4   5805      0t0  TCP *:www (LISTEN)
apache2   1242     root    4u  IPv4   5807      0t0  TCP *:https (LISTEN)
apache2   1289 www-data    3u  IPv4   5805      0t0  TCP *:www (LISTEN)
apache2   1289 www-data    4u  IPv4   5807      0t0  TCP *:https (LISTEN)
Xorg      1363     root    1w  IPv6   6252      0t0  TCP *:x11 (LISTEN)
Xorg      1363     root    3u  IPv4   6253      0t0  TCP *:x11 (LISTEN)
ntpd      1621      ntp   16u  IPv4   7769      0t0  UDP *:ntp
ntpd      1621      ntp   17u  IPv6   7770      0t0  UDP *:ntp
```

This output indicates to me that my server is not hardened, because:

1. More than one instance of apache2 is running
2. My webserver needs neither of inetd, Xorg, nor ntpd to function
3. mysqld is exposed to the network
4. The server responds to HTTPS requests

Question C3-15

Port scanning helps the adversary to:

1. Find known, exploitable vulnerabilities
2. Determine which services are running on the target host
3. Deny the functionality of my system to legitimate users
4. It doesn't; port scanning is used pre-emptively by system administrators

Question C3-17

What do I learn with the following command and resulting output:

```
# nc 192.168.56.102 12880
220 192.168.56.102 ESMTP Postfix (Ubuntu)
```

1. The host 192.168.56.102 is running Ubuntu, with an SMTP server on port 12880.
2. The host 192.168.56.102 is a hardened web-server
3. The host 192.168.56 responds to 12880 bytes of random noise on port 102 with a valid ESMTP HELLO-message
4. Nothing; `nc` is a "Network Compiler".

Question C3-19

I look at `/var/log/auth.log` on my server and see the following:

```
Feb  5 13:35:01 bob login[1749]: FAILED LOGIN (1) on 'pts/1' from 192.168.1.7
  FOR `UNKNOWN', User not known to the underlying authentication module
Feb  5 13:35:01 bob login[1749]: FAILED LOGIN (2) on 'pts/1' from 192.168.1.7
  FOR `UNKNOWN', User not known to the underlying authentication module
Feb  5 13:35:01 fred login[1749]: FAILED LOGIN (1) on 'pts/1' from 192.168.1.7
  FOR `UNKNOWN', User not known to the underlying authentication module
Feb  5 13:35:02 fred login[1749]: FAILED LOGIN (2) on 'pts/1' from 192.168.1.7
  FOR `UNKNOWN', User not known to the underlying authentication module
...
```

This goes on for literally thousands of lines, the only variation being the timestamp and the word immediately after the timestamp ("bob" and "fred" above). This log indicates that:

1. An adversary with access to my local network is trying to guess a user/password combination
2. An adversary without access to my local network is trying to guess a user/password combination
3. My host is subject to a TCP ACK portscan
4. My host is receiving TCP packets with flags PSH, RST and SYN simultaneously set.

Question C3-21

I am asked to harden an old multi-purpose server. I intend to do this by replacing old insecure network services with more secure modern variants. Which of the following service is most critical to replace?

1. telnet
2. apache2
3. ntpd
4. X11

Question C3-23

I am asked to secure a multi-purpose server. Which steps should I take?

1. Run services as non-root users where possible
2. Disable all network services
3. Use a firewall to drop all TCP packets to ports in the range 1-1023
4. Remove all server process scripts from `/etc/init.d`

Authentication & Access Control

Question C4-01

"Authentication" is:

1. The process of verifying the identity of a user or other subject
2. A mechanism for determining which resources a user may access
3. A cryptographic primitive
4. An MD5 hash

Question C4-03

"Access control" is:

1. A mechanism for determining which resources a user may access
2. A policy describing which resources a user may access
3. A synonym for "Authorization"
4. A synonym for "Authentication"

Question C4-05

telnet and ftp are considered insecure because:

1. Both protocols transmit all traffic, including passwords, in plaintext
2. Both protocols' original implementations contained multiple exploitable buffer overflows
3. Both protocol's built-in encryption is easily brute-forced
4. They both authenticate using IP addresses

Question C4-07

SSH may use public-key cryptography for authentication, by placing the public keys of authorized users in the special file `~/.ssh/authorized_keys`. The security of this authentication mechanism relies on:

1. OS-support for file-level access control
2. OS-support for changing "effective group id"
3. Network transparency
4. OS-support for dot-file invisibility (`.ssh`)

Question C4-09

Consider this partial output of `ls -l /etc/init.d` on linux.

```
lrwxrwxrwx 1 root root   21 Jul  2  2010 ufw -> /lib/init/upstart-job
-rwxr-xr-x 1 root root 2787 Nov  5  2009 umountfs
-rwx------ 1 root root 2075 Oct 14  2009 umountnfs.sh
-rwxr-x-w- 1 root root 1683 Oct 14  2009 umountroot
```

Which file poses the greater security risk because of poorly chosen file permissions?

1. ufw

2. umountfs
3. umountnfs.sh
4. umountroot

Question C4-11

Consider this partial output of `ls -l /usr/bin`.

```
-rwsr-xr-x 1 root root   18048 Apr  9  2010 /usr/bin/pkexec
-rwtr-xr-x 1 root root   47076 Mar  7  2010 /usr/bin/pkg-config
lrwxrwxrwx 1 root root       5 Jul  2  2010 /usr/bin/pkill -> pgrep
-rwX-w-r-x 1 root root    4531 Apr 23  2010 /usr/bin/pl2pm
```

Which of these files will execute with the highest privilege level, no matter which user executes the file?

1. pkexec
2. pkg-config
3. pkill
4. pl2pm

Question C4-13

I made a script to improve the security of my system:

```
#! /bin/bash
# Kill any running SSH daemon.
LOGFILE=/var/log/ssh-killer.log
DATE=`date`
if [ -e /var/run/sshd.pid ]; then
    kill -9 `cat /var/run/sshd.pid`
    echo "$DATE: Killed unathorized SSH instance." > $LOGFILE
else
    echo "$DATE: No unauthorized SSH instances." > $LOGFILE
fi
```

This script runs setuid root because it otherwise the `kill` command fails. The script is a security risk because:

1. It invokes commands specified by relative pathnames
2. It invokes `kill` which has known vulnerabilities and should never be run `setuid root`
3. It writes to a file in the `/var/log` directory
4. It specifies the `sshd.pid` file with an absolute pathname.

Question C4-15

Using `mktemp` to create temporary files is more secure than using a fixed filename, because using `mktemp` adheres to the security principle of:

1. Minimum exposure
2. Generating secrets
3. Minimising trust/Maximising trustworthiness
4. No single point of failure

Question C4-17

I wrote a script which may give extra information when running as root:

```
#! /bin/bash
VERBOSE=0
if [ "$1" = "-v" -a "$USER" == "root" ]; then
```

```
    echo "Verbose mode enabled for user 'root'"
    VERBOSE=1
fi
# ...
```

However, my script is insecure: users other than root may still have the script proceed with `VERBOSE` set to 1, because:

1. The adversary may simply set the `USER` environment variable to "root"
2. The adversary may simple set the `VERBOSE` environment variable to "1"
3. The adversary can perform an injection attack by running the script with first argument `"qux == qux -o foo"`
4. The script elevates itself to running setuid root

Question C4-19

Running a server in a `chroot` jail is an example of adhering to the security principle of:

1. Compartmentalisation
2. Secure, fail-safe defaults
3. Usability
4. Generating secrets

Question C4-21

I'm thinking about whether to run my server in a `chroot` jail or in a virtual machine. Pick the correct argument:

1. The VM introduces a single point of failure because it has its own TCP/IP stack
2. The VM achieves worse compartmentalisation, because once the adversary has `root` on the guest machine, he can easily escape to the host machine
3. The `chroot` jail is practically cumbersome to keep updated with software patches etc.
4. The two are for all intents and purposes the same approach.

Question C4-23

In my C-program, there is a function which checks if the user knows a password:

```
int check_password()
{
    char buf[256] = {0};
    printf("Enter password? ");
    gets(buf);
    return ! strcmp(buf, "secret");
}
```

Does this function contain a potential buffer overflow? Why/why not?

1. No; clearly 256 bytes is large enough for any password anyone might ever pick
2. No; the `strcmp` function will notice the 256 size limit of the buffer
3. Yes; `buf` might overflow and overwrite the return address on the stack frame of `check_password`
4. Yes; `buf` might overflow and overwrite the return address on the stack frame of `strcmp`.

Question C4-25

I'm trying to construct a buffer overflow for a C program, and I've used a debugger to determine that just prior to reading unlimited input into a bounded buffer, my stack looks

like this:

```
0xbffff5d0:     0x00000000      0x0804865b      0x0000000a      0x00000000
0xbffff5e0:     0x42424242      0x00000000      0x00000000      0x00000000
0xbffff5f0:     0x00000000      0x00000000      0x00000000      0x00000000
0xbffff600:     0x00000000      0x00000000      0x00000000      0x00000000
0xbffff610:     0x00000000      0x00000000      0x00000000      0x00000000
0xbffff620:     0x00293ff4      0x00000000      0xbffff748      0x0804855a
0xbffff630:     0xbffff640      0x00118fa6      0x0012bff4      0x00000000
0xbffff640:     0x00000000      0x000000ca      0x00000006      0xbffff68c
```

The debugger tells me that the return address is saved at address 0xbffff62c, with value 0x080455a. It also tells me that the buffer I'm trying to overflow starts at address 0xbffff5e4. I've determined myself that I'd rather have the program return to 0x080485c2. What input should I provide to the program in order to overflow the buffer?

1. 8*4=32 non-zero bytes followed by "\xc2\x85\x04\x08".
2. 16*4=64 non-zero bytes followed by "\xc2\x85\x04\x08".
3. 18*4=64 non-zero bytes followed by "\xc2\x85\x04\x08".
4. 22*4=88 non-zero bytes followed by a random 256-bit pattern.

Logging & Log Analysis

Question C5-01

Network services typically log important and interesting events to files. Why don't they just write on the screen, like, say, Microsoft Word does?

1. Network services run in the background; they may not be connected to a monitor
2. Output on a physical monitor might be monitored by an adversary
3. Log messages on a physical monitor are hard to read
4. File-level access control does not apply to screens

Question C5-03

Logs help provide

1. Complete mediation
2. Secure, fail-safe defaults
3. Traceability
4. Compartmentalisation

Question C5-05

On Linux, there is no authentication for writing messages to the system-level logging service. This gives an adversary the opportunity to:

1. Mislead intrusion detection systems by faking log entries
2. Delete previous log entries
3. Introduce port scanning attacks
4. Subvert the kernel/user-space distinction

Question C5-07

Remote logging systems have the advantage that

1. Compromising the local machine does not give the adversary the ability to tamper with logs
2. Simplifying the overall security architecture

3. Obviating the need for access control for log messages
4. Obviating the nede for a distinction between binary and text-based log messages

Question C5-09

Which of these logging methods is more tamper-proof?

1. Automatic hard-copy to a printer connected directly to the system
2. Public-key encrypted log-files
3. Symmetric-key encrypted log-files
4. Remote logging

Question C5-11

A network-based IDS

1. Is a headless IDS
2. Is connected to the network it is monitoring
3. Is an IDS implemented as a network of monitors
4. Is a P2P-based IDS implementation

Question C5-13

I suspect my system has been compromised, that one of my users has managed to obtain root privileges. What's the first place I should start looking for confirmation?

1. /var/log/auth.log
2. /var/log/intrusions
3. /var/log/dmesg
4. /var/log/mail.log

Question C5-15

I decide to take a snapshot of "/bin" and "/usr/bin" so that I can later check if an adversary has compromised any of the files there. I don't want to copy them all, so I just md5hash them all. This is insufficient because:

1. Hashes do not account for changes in permission
2. Hashes do not account for changes in size
3. Hashes do not account for changes in content
4. Hashes do not account for perceived usability

Web Application Security (1)

Question C6-01

I'm an IT specialist with Danish Military Intelligence. I'm tasked with producing a report on the security of a server located at IP 198.41.206.163. I'm given a very tight deadline and no other information. Where do I start?

1. A black-box audit
2. A white-box audit
3. A unit test
4. Remote logging

Question C6-03

As part of my black box audit, I try nc at the web-server. This is what I see:

```
HTTP/1.1 200 OK
Date: Tue, 24 Feb 2015 16:04:29 GMT
Server: Microsoft-IIS/7.5
Cache-Control: no-cache, no-store
Pragma: no-cache
Content-Type: text/html; charset=utf-8
Expires: -1
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Content-Length: 41711
Set-Cookie: ASP.NET_SessionId=q5ngm13wi055bgebt10rmjiw; path=/; HttpOnly
Connection: close
```

Which HTTP header gives me the most useful information?

1. Server
2. X-Powered-By
3. Cache-Control
4. Content-Type

Question C6-05

I start looking at the pages served by the server. One of them is a login-page which, when I type in user `foo` and password `bar`, makes my browser output the following rather unusual POST request:

```
POST /login.php HTTP/1.1
Host: 198.41.206.163
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Content-Length: 83

login_query=SELECT+id+FROM+users+WHERE+username%3D%22foo%22+AND+password%3D%22bar%22
```

I think there is an "admin" user I'd like to login as. How should I tamper with this request?

1. Change the first line to `GET /login?user=;admin HTTP/1.1`
2. Insert between lines 1 and 2 `User: admin`
3. Replace the header `Accept: ...` with `Accept: admin/*`.
4. Change the final line to
   `login_query=SELECT+id+FROM+users+WHERE+username%3D%22admin%22`

Question C6-07

Another page has an "upload file" facility, enabled by the following html fragment:

```
...
<form >
  <input type="file" name="file_name" >
  <input type="hidden" name="upload_path" value="/var/www/uploads" >
  <input type="submit"gt;
</form >
...
```

This looks like a remote file upload vulnerability. I'll try to it exploit by:

1. Uploading a file that is setuid root
2. Tamper with the HTTP request replacing the path in the hidden input with `/var/www` when I upload a PHP script of my own.
3. Tamper with the HTTP request, replacing the path in the hidden input with `/etc` when I upload a file `passwd`

4. Uploading a binary containing a buffer overflow I can subsequently exploit.

Question C6-09

Using the remote file upload exploit, I get remote command execution on 198.41.206.163. I use this to inspect the source code for the server. I come across this:

```
$poll->id =$_GET['id'];
$query = 'SELECT a.id, a.text, a.hits, b.voters '
        . ' FROM #__poll_data AS a'
        . ' INNER JOIN #__polls AS b ON b.id = a.pollid'
        . ' WHERE a.pollid = '. $poll->id
        . ' AND a.text <> ""'
        . ' ORDER BY a.hits DESC';
$db->setQuery( $query );
$votes = $db->loadObjectList();
```

I see a vulnerability; specifically an opportunity for:

1. SQL injection
2. Remote command execution
3. Remote file upload
4. Privilege escalation

Question C6-11

Suppose I want to fix the vulnerability I just found in this code, which line should I change?

```
01:   $poll->id =$_GET['id'];
02:   $query = 'SELECT a.id, a.text, a.hits, b.voters '
03:           . ' FROM #__poll_data AS a'
04:           . ' INNER JOIN #__polls AS b ON b.id = a.pollid'
05:           . ' WHERE a.pollid = '. $poll->id
06:           . ' AND a.text <> ""'
07:           . ' ORDER BY a.hits DESC';
09:   $db->setQuery( $query );
10:   $votes = $db->loadObjectList();
```

1. 5:0

Question C6-13

As I work on compromising 192.41.206.163, I suddenly see the following in the log of the web-server on my own machine.

```
198.41.206.163 - - [25/Feb/2015:22:21:08 +0100]
  "GET /index.php?page=';   nc -l -p 4141 -e /bin/sh;'" 404 3995
```

What is this?

1. Someone at 198.41.206.163 is trying to trick my web server into running a shell on port 4141
2. Someone at 198.41.206.163 is attempting to induce a buffer overflow in `/bin/sh`
3. An injection attack was rejected with a HTTP 404 result code.
4. The `nc` web-server executable failed with exit code 3995.

Question C6-15

Now that I have the ability to upload arbitrary files and execute arbitrary commands on 198.41.206.163, I want root access. For starters, I try this command:

```
> whoami
www-data
```

How could I proceed?

1. You're done; you already have remote execution capabilities
2. Attempt privilege escalation through (stealth) portscanning
3. Look for files, especially shell-scripts, that are setuid root.
4. You're done; you already have remote file upload capabilities

Question C6-17

I've so far been unable to achieve root access, so I decide to see if I can guess someone else's password for the web application. This will help me how?

1. I can hope someone uses the web application password also as their login password or maybe even the root password.
2. The root user is a symbolic link to a regular user.
3. It doesn't, passwords for the web application cannot be the same as the passwords of system users.
4. It doesn't, I couldn't possibly guess a password.

Question C6-19

I trick the server into dumping the contents of its "users" table to me. It has two columns "username" and "password". The dump looks like this:

```
admin 83650ab82ceaf1c287f2508aa1afabf9:Ri6i9DocKrNzU7U3ddOQQi03s6Mp2F8N
debois b0ffb76ff62e1a63378821e289d18139:4XtR5R2aEFaJy0fXZXB8rFEeoGcobrRk
trbj 4d488c8d3ea27b351dfd1c336f43828a:NVwooMha1kAPOvDafwpbHOWbABOmSMW8
svensson 5208b5cff4d4fb5a250b22c1b1e0ce46:pKxD82GMlQkI7UpHGYMkVwYubOrU9F6B
```

What is this, and how should I proceed?

1. User-password list; everybody apparently uses hard-to-guess passwords.
2. User-md5hash list; since passwords have been hashed, it'll be impossible for me to guess any.
3. User-md5hash list; if the passwords are weak, an automated tool might be able to guess some.
4. User-user id-salt list; this does not help me guess any passwords.

Discussion Questions

Question D1: Risk analysis

Your answer to this question counts 15% towards the final grade. When a customer applies for a mortgage with a mortgage credit institution, the institution carries out the following steps.

1. Collect budget from the customer.
2. A caseworker pre-screens the budget for missing numbers, requesting a new one if some are missing.
3. The caseworker verifies the budget with the customer's bank.
4. Schedule an on-site visit by a valuation expert.
5. Register the resulting valuation internally.
6. Asses the application

Write an abbreviated risk analysis for this system. Your analysis should consume ~1.5 pages. You may hypothesise details of the system and must stipulate yourself its security

requirements. Be sure to cover System, Stakeholders, Assets, Vulnerabilities, Threats, and Risk. You may find Chapter 8 of the course book helpful.

Question D2: Cryptography

Your answer to this question counts 10% towards the final grade. In the workflow described in the previous question, information is exchanged between various parties. Assume all communication is by e-mail, and so is at-risk for eavesdropping and tampering. Describe how to use public-key cryptography to alleviate those risks; be sure to include (a) the questions of key distribution and non-repudiability where appropriate, (b) a brief description of who uses which keys for what. Your answer should consume ~1 page.

(End of questions.)

Thu May 28 12:26:27 CEST 2015