SSAS F2016

Systems Architecture and Security, Examination set, June 3, 2016.

What to hand in

You submit electronically, via LearnIT:

1. A plain-text file `answers.txt` 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 `discussion.pdf` 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. Feel free to submit a .zip archive containing the above two files. 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 65% towards the final grade.

Security principles

The following questions all describe a situation, then asks which security principle has been either adhered to or violated in that situation. Be sure to note whether the question asks about violation or adherence.

Question C1-1

A Danish hospital requires staff to have passwords that are 32 characters long, contains both numbers and special characters, and does not contain any natural language word as a substring. This policy is a violation of

1. Open design
2. Minimum Exposure
3. Usability
4. Simplicity

Question C1-3

The hospital requires all laptop computers to authorize users using username/password logins before use. However, the hospital never uses encryption as a matter of principle, because that would be annoying to end-users. In particular, laptop disks are not encrypted. This policy is a violation of

1. Compartmentalization
2. Open design
3. Least privilege
4. Complete mediation

Question C1-5

To promote efficiency, all users has access to all medical data---the receptionist can look up medical records if she wants. This is to ensure that vital patient information is always available in a medical emergency. This policy is a violation of

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

Question C1-7

Again to promote access to information in case of medical emergencies, laptops will assume any entered password is valid if they for whatever reason cannot connect to the central authentication server. This implementation violates

1. Least privilege
2. Minimum exposure
3. Safe, fail-safe defaults
4. Complete mediation

Question C1-9

Hospital laptops come pre-equipped with a running web-server, an ftp-server, an ssh-server, a gopher-server, a client for the patient records database, and massive multiplayer online solitaire client. (It was easier for IT to make one image that would fit both IT, medical, and administrative staff.) This setup is a violation of

1. Minimum trust and maximum trustworthiness
2. Minimum exposure
3. No single point of failure
4. Complete mediation

Question C1-11

Hospital server systems are developed in-house by a guy named Werner. Werner decided it was most efficient to have just a single database with a single table containing all information the hospital ever needs to record: payroll information, medical records, accounting, pharmaceutical inventory, everything. This implementation is a violation of

1. Safe, fail-safe defaults
2. Minimum trust and maximum trustworthiness
3. Compartmentalization
4. Usability

Question C1-13

It turns out users have trouble generating passwords conforming to the rules of C1-1. Werner implements a helpful web-service, which generates random passwords by picking a random number between 0 and 1000, then MD5-hashing that number, replacing the second letter with '5' and the third with '!'. This implementation violates

1. Complete mediation
2. Generating secrets
3. Compartmentalization
4. Usability

Question C1-15

Because all hospital data is stored in a single table, Werner decides that logging access to that data is unhelpful. He turns off all the database's access logging features. Werner is violating

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

Question C1-17

Werner realizes that since every hospital can access any piece of hospital data anyway, it doesn't really make sense to distinguish between different users. He switches all users over to the single account "staff" and hands out post-its with the password for this account, admonishing everyone to keep this password a secret. Werner is grossly violating the principle of

1. Generating secrets
2. Simplicity
3. Single point of failure
4. Minimize trust and maximize trustworthiness

Question C1-19

Doctors at the hospital realize they are all sharing the same user and become worried about security. Dr. Mendelssohn comes up with a scheme where, instead of using passwords, users authenticate by submitting an x-ray of their 3rd vertebrae. Log-in now takes 14 minutes and involves non-trivial amounts of radiation. Dr. Mendelssohn's proposal violates the principle of

1. Single point of failure
2. Usability
3. Minimize trust and maximize trustworthiness
4. Simplicity

Network Services

Question C3-01

A SYN scan involves

1. Sending the first packet of the TCP 3-way handshake
2. Sending the second packet of the TCP 3-way handshake
3. Sending the third packet of the TCP 3-way handshake
4. Completing the TCP 3-way handshake

Question C3-03

A CONNECT scan involves

1. Sending the first packet of the TCP 3-way handshake
2. Sending the second packet of the TCP 3-way handshake
3. Sending the third packet of the TCP 3-way handshake
4. Sending all packets of the TCP 3-way handshake

Question C3-05

An ACK scan involves

1. Sending an unsolicited ACK packet, listening for CLR response
2. Sending an unsolicited ACK packet, listening for RST response
3. Sending an unsolicited CLR packet, listening for ACK response
4. Sending an unsolicited RST packet, listening for ACK response

Question C3-07

A stealth scan is preferable to an ACK scan when

1. The target blocks CONNECT scans
2. The target routinely drops RST packets
3. The target is running sophisticated intrusion detection systems
4. The scanner and target are far removed in IP distances

Question C3-09

Which of these tools do not necessarily need network access?

1. nmap
2. syslogd

3. sudo
4. telnet

Question C3-11

I use nmap to scan a the hospital database server. This is the output:

```
Not shown: 497 closed ports, 498 filtered ports
PORT      STATE SERVICE
631/tcp   open  ipp
6000/tcp  open  X11
10000/tcp open  snet-sensor-mgmt
49152/tcp open  unknown
49153/tcp open  unknown
```

How would we proceed with an attack?

1. On one of the 497 closed ports
2. On one of the 498 filtered ports
3. On the custom services running on ports 49152-49153
4. No place; this system is fully hardened.

Question C3-13

Werner notices my scan. He scrambles to harden his system, running `sudo lsof -i` on the Linux box serving as hospital firewall device. He sees the following:

```
mysqld     871    mysql   12u  IPv4   4844      0t0  TCP *:mysql (LISTEN)
sshd      1621      ssh   16u  IPv4   7769      0t0  TCP *:ntp
sshd      1621      ssh   17u  IPv6   7770      0t0  TCP *:ntp
```

What should he do?

1. Stop the database server on 1621 which clearly is not necessary for firewall operations
2. Stop the database server on 871 which clearly is not necessary for firewall operations
3. Stop the telnet server on 4844 which clearly is not necessary for firewall operations
4. Stop the running ftp transfer on 7770 which is probably an attack

Question C3-15

How can port-scanning help Werner protecting hospital servers?

1. By scanning for incoming malicious packets
2. By quickly figuring out which unnecessary services are running
3. By denying access to non-authorized users
4. It cannot; port scanning is only used by the adversary

Question C3-17

I take a closer look at the custom services I found at 49152-3 with telnet:

```
# nc server.hospital.dk 49152
nc: getaddrinfo: nodename nor servname provided, or not known
```

What do I learn?

1. An address (LDAP) server is running at that port
2. Port 49152 is filtered
3. There is no service at port 49152
4. There is no host server.hospital.dk

Question C3-19

Werner looks at `/var/log/auth.log` of his database server and sees this:

```
Feb  5 13:35:01 admin login[1749]: FAILED LOGIN (1) on 'pts/1' from 130.226.142.6
  FOR `UNKNOWN', User not known to the underlying authentication module
Feb  5 13:35:01 admin login[1749]: FAILED LOGIN (2) on 'pts/1' from 130.226.142.6
  FOR `UNKNOWN', User not known to the underlying authentication module
Feb  5 13:35:01 root login[1749]: FAILED LOGIN (1) on 'pts/1' from 130.226.142.6
  FOR `UNKNOWN', User not known to the underlying authentication module
Feb  5 13:35:02 root login[1749]: FAILED LOGIN (2) on 'pts/1' from 130.226.142.6
  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 ("admin" and "root" 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 is trying to guess a user/password combination from a remote host
3. The host is subject to a TCP SYN port scan
4. The host is subject to a TCP ACK port scan

Question C3-21

Werner decides to harden the hospital web server by replacing old insecure network services with more secure modern variants. Which of the following service is most critical to replace?

1. ftp
2. sshd
3. syslogd
4. sudo

Question C3-23

Unsatisfied with replacing services, Werner wants to take additional steps towards hardening the web server. What should he do?

1. Run the server as a VM (Complete mediation)
2. Stop services not related to serving web pages (Minimum exposure).
3. Use a firewall to drop all UDP packets to all ports (SYN FLOOD protection).
4. Stop syslogd (Traceability)

Authentication & Access Control

Question C4-01

Which of these has to do with "Authentication"?

1. ntpd
2. apache2d
3. sshd
4. file-system permissions

Question C4-03

Which of these has to do with "Access control"?

1. ntpd

2. apache2d
3. sshd
4. file-system permissions

Question C4-05

Which of these has to do with "Authorization"?

1. /etc/passwd
2. apache2d
3. file-system permissions
4. ntpd

Question C4-07

SSH may use public-key cryptography for authentication, by placing the keys of authorized users in the special file `~/.ssh/authorized_keys`. Which keys exactly?

1. Private keys
2. Public keys
3. Symmetric keys
4. Shared keys

Question C4-11

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----wx 1 root root 2075 Oct 14  2009 umountnfs.sh
-rwxr-x--- 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 C4-13

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

```
-rwsr-xr-x 1 pkg  pkg    18048 Apr  9  2010 /usr/bin/pkexec
-rwtr-xr-x 1 root root   47076 Mar  7  2010 /usr/bin/pkg-config
-rwX-w-r-x 1 root root    4531 Apr 23  2010 /usr/bin/pl2pm
```

Which of these files will execute with the privileges of root, no matter which user executes the file?

1. pkexec
2. pkg-config
3. pl2pm
4. none of the above

Question C4-15

Let's use a fixed filename to create temporary files rather than using `mktemp`. This would:

1. Follow the principle of Minimum exposure

2. Violate the principle of Minimum exposure
3. Violate the principle of Generating secrets
4. Follow the principle of Generating secrets

Question C4-17

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

```
#! /bin/bash
if [ "$1" = "-v" -a `id -u` == "0" ]; then
    echo "Verbose mode enabled for user 'root'"
    FLAG=1
fi
# ...
```

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

1. The adversary may simply set the `USER` environment variable to "root"
2. The adversary may simply set the `$1` environment variable to "-v"
3. The adversary may simple set the `FLAG` environment variable to "1"
4. The adversary can perform an injection attack by running the script with first argument `"SELECT * FROM TABLE USERS; --"`

Question C4-19

Running a server on a virtual machine is an example of adhering to the security principle of:

1. Compartmentalization
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. Pick the correct argument:

1. chroot introduces a single point of failure because it has its own TCP/IP stack
2. chroot achieves worse compartmentalization than a virtual machine, because it controls access at the file-system level
3. chroot achieves better compartmentalization than a virtual machine, because it controls access at the file-system level
4. chroot achieves better compartmentalization than doing nothing

Question C4-23

In my C-program, there is a function `receive_packet` which reads at most 64 bytes from the network and puts them into its argument buffer. I use it like this:

```
int get_next()
{
    char buf[256] = {0};
    receive_packet(buf);
    return atoi(buf);
}
```

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

1. No; the `buf` is large enough to that `receive_packet` will never overrun it
2. No; the `receive_packet` function will notice the 256 size limit of `buf`

3. Yes; `buf` might overflow and overwrite the return address on the stack frame of `get_next`
4. Yes; `buf` might overflow and overwrite the return address on the stack frame of `atoi`.

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:

```
0xbffad5d0:     0x00000000      0x0804865b      0x0000000a      0x00000000
0xbffad5e0:     0x42424242      0x00000000      0x00000000      0x00000000
0xbffad5f0:     0x00000000      0x00000000      0x00000000      0x00000000
0xbffad600:     0x00000000      0x00000000      0x00000000      0x00000000
0xbffad610:     0x00000000      0x00000000      0x00000000      0x00000000
0xbffad620:     0x00293ff4      0x00000000      0xbffff748      0x0804858a
0xbffad630:     0xbffff640      0x00118fa6      0x0012bff4      0x00000000
0xbffad640:     0x00000000      0x000000ca      0x00000006      0xbffff68c
```

The debugger tells me that the return address is saved at address 0xbffad62c, with value 0x080458a. It also tells me that the buffer I'm trying to overflow starts at address 0xbffad5ec. I've determined myself that I'd rather have the program return to 0x080485c5. What input should I provide to the program in order to overflow the buffer?

1. 8*4=32 non-zero bytes followed by "\xc5\x85\x04\x08".
2. 16*4=64 non-zero bytes followed by "\xc5\x85\x04\x08".
3. 18*4=64 non-zero bytes followed by "\xc5\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 to the network, like, say, a web-server does?

1. Output on a physical network link might be monitored by an adversary
2. They might
3. Network traffic overhead
4. File-level access control does not apply to the network

Question C5-03

Which of the following is correct?

1. syslogd is cryptographically secure
2. Logging in modern operating systems is predicated on the identity ln(e)=1
3. Logs are an important part of intrusion-detection systems
4. Network logging is practically unusable because of network overhead

Question C5-05

How can the adversary defeat logging?

1. By re-routing network logging over FTP
2. By deleting entries indicative of his activity
3. By port scanning (DDOS/LOG)
4. By subverting the kernel/user-space distinction

Question C5-07

Local logging systems have the advantage over remote ones that

1. Compromising the local machine does not give the adversary the ability to tamper with logs
2. Obviating the need for access control for log messages
3. Obviating the need for a distinction between binary and text-based log messages
4. They are simpler

Question C5-09

Which of these logging methods is more tamper-proof?

1. syslogd
2. mesgd
3. Remote logging
4. Local logging

Question C5-11

The hospital IDS is a headless server attached to the internal hospital network. This is an example of

1. A P2P IDS
2. A network IDS
3. An Internet Delay Signal
4. A beheaded monitor

Question C5-13

Werner is suspicious that the hospital database server has been compromised, and that the adversary (me) has achieved root access on that server. (He is right.) He wants to remove the adversary's access. What should he do?

1. Consult /var/log/auth.log, look for root logins
2. Consult /var/log/dmesg, look for port scans
3. Restore the database server from backup
4. Harden the server

Question C5-15

Werner previously copied the contents of "/bin" and "/usr/bin" of the database server onto a USB drive, so that he can check if an adversary has compromised any of the files there simply by seeing if any of the files changes. He notices the following differences; which one is most likely to be due to the adversary?

1. /var/log/system.log: Update time and contents differ
2. /etc/passwd: Last-access time differ
3. /var/log/auth.log: Update time and contents differ
4. /bin/sh: Permissions differ

Web Application Security

Question C6-01

I want to break into the hospital web server, so I do a black-box audit. What isn't included

by a black-box audit?

1. Port scanning
2. Accessing the pages served by the server
3. Reviewing hospital IT design documents for the server
4. Monitoring IP traffic to and from the server

Question C6-03

As part of my black box audit, I try nc at the web-server, trying to find out what version it is. This is what I see:

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2016 11:01:04 GMT
Vary: Host
Last-Modified: Mon, 23 May 2016 10:55:08 GMT
ETag: "3201af-81b4-53380461418ff"
Accept-Ranges: bytes
Content-Length: 33204
Content-Type: text/html; charset=utf-8
Connection: close
```

Which HTTP header gives a lead on the server version?

1. ETag
2. Last-Modified
3. Accept-Ranges
4. None of them

Question C6-05

I start looking at the pages served by the server. One of them has a "post a comment" form, with a list of previously posted comments below. I try entering SQL into that form, but nothing happens. I'll try an XSS attack. Should I:

1. Add a comment `]] char buf[256]; gets(buf); return buf;`
2. Add a comment `DROP TABLE USERS; --`
3. Add a comment `<button onClick="console.log('pwned!')">`
4. Add a comment `)>) [ "user": "root", "login": "immediately" ]`

Question C6-07

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

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

Maybe I can get a remote file upload vulnerability. Which is more likely to work?

1. Upload a .php-file to /var/www/index.html
2. Upload a .php-file to ../index.html
3. Upload a .php-file to ./index.html
4. Upload a .php-file to ../scripts

Question C6-09

Using the remote file upload exploit, I get remote command execution on web.hospital.dk. I

want to get the patient records, but there is no local database. I can see the web-server has open connections to port 3306 (mysql) on 10.0.0.3, though. What should I do?

1. Attempt privilege escalation at web.hospital.dk
2. Attempt a black-box audit of 10.0.0.3 from web.hospital.dk
3. Attempt to brute force public keys of web.hospital.dk
4. Attempt XSS at web.hospital.dk

Question C6-11

I discover that 10.0.0.3 is running mysql 5.1.61. How should I proceed?

1. Look in the CVE register for 10.0.0.3 vulnerabilities
2. Look in the CVE register for mysql 5.1.61 vulnerabilities
3. Attempt a SQL injection attack
4. Attempt a remote file upload attack

Question C6-13

I trick the database server into dumping the contents of its "auth" table to me. It has two columns "id" and "hash". The dump looks like this:

```
83650ab82ceaf1c287f2508aa1afabf9;admin
b0ffb76ff62e1a63378821e289d18139;foo
4d488c8d3ea27b351dfd1c336f43828a;id
5208b5cff4d4fb5a250b22c1b1e0ce46;攻殻機動隊
```

What is this?

1. Possibly an md5hash-user table, excluding salts
2. Possibly an md5hash-user table, including salts
3. SSH DSA keys for password-less remote access
4. SSH RSA keys for password-less remote access

Discussion Questions

Question D1: Risk analysis

Your answer to this question counts 20% towards the final grade. A hospital's central administration is reponsible for filing and digitizing paper-based patient records. The record of a particular patient takes the following course:

1. A practicing physician refers a patient to the hospital by e-mail.
2. The hospital secretariat fills out a physical form based on that e-mail, and registers the patient in the hospital database.
3. While the patient is admitted, the secretariat receives copies of the paper record weekly, scanning the records into the database.
4. Patients can at any time request a copy of their (digital) records using a web-page.
5. The secretariat forwards records to the referring physician every 3 weeks or upon patient release.
6. Physicians may submit requests for Special Medication directly to various external institutions, usually by e-mail, occasionally by physical mail.
7. Requests for Special Medication must be approved by the chief surgeon or chief administrative physician before being administered to patients. Approval is nearly automatic and usually communicated by e-mail or post-it notes.

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 15% towards the final grade. In the workflow of the previous question, information is exchanged between various parties; this exchange is typically carried out on paper or by unencrypted 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 Jun 16 13:15:44 CEST 2016