Final exam in

Web Security EITF05

Department of Electrical and Information Technology
Lund University

October 31st, 2014, 8.00–13.00

- You may answer in either Swedish or English.
- If any data is lacking, make (and state) reasonable assumptions.
- Use legible hand writing. If your answers cannot be read, you will receive zero points on that problem.
- Grading is done as follows.
  Grade 3 = 20–29 points,
  Grade 4 = 30–39 points,
  Grade 5 = 40–50 points.

Good luck!

Paul & Martin

Problem 1. There are two principal ways of implementing sessions in PHP, describe each alternative. State one advantage of each method over the other. (3 points)

Problem 2. Does SSL protect against XSS-attacks? Explain why or why not. (3 points)

Problem 3. Digest authentication (RFC2617) calculates the digest according to

$$\text{MD5(}\ \text{MD5}(A1) : \mathit{nonce} : \mathit{nc} : \mathit{cnonce} : \mathit{qop} : \text{MD5}(A2)\ \text{)},$$

with

$$A1 = \mathit{username} : \mathit{realm} : \mathit{password},$$

$$A2 = \begin{cases} \mathit{method} : \mathit{URI} & \text{if } \mathit{qop} = \mathit{auth}, \\ \mathit{method} : \mathit{URI} : \text{MD5}(\mathit{entity\text{-}body}) & \text{if } \mathit{qop} = \mathit{auth\text{-}int}. \end{cases}$$

a) Explain the usage and purpose of the *realm* parameter.
b) Explain the usage and purpose of the *nc* parameter?
c) Explain the usage and purpose of the *cnonce* parameter? (3 points)

Problem 4. You will now consider if it is possible for a Base64 string to encode to itself.

Let $\mathbb{A}$ be the set of all strings composed of 8-bit ASCII characters, and let $\mathbb{B}$ be the set of all strings composed of Base64 characters (printable ASCII characters, $\mathbb{B} \subset \mathbb{A}$). The Base64 encoding procedure can then be viewed as a mapping

$$f : \mathbb{A} \longrightarrow \mathbb{B}.$$

A string s that satisfies $f(s) = s$ is called a fixed point. Base64 has an infinite-length fixed point $s = "Vm\ldots"$.

a) Can a string of finite length be a Base64 fixed point, why or why not?

b) What is the third character of the infinite-length fixed point s above?

c) What is the fourth character of the infinite-length fixed point s above?

Hint: Here are some useful character codes in hex. Character code intervals for A-Z, a-z and 0-9 are contiguous.

symbol	A	Z	a	z	0	9	+	/
Base64	0x00	0x19	0x1A	0x33	0x34	0x3D	0x3E	0x3F
ASCII	0x41	0x5A	0x61	0x7A	0x30	0x39	0x2B	0x2F

(3 points)

Problem 5. A DKIM signature header of an email is given below.

```
DKIM-Signature:
v=1;
a=rsa-sha256;
c=simple/relaxed;
d=gmail.com;
s=gamma;
h=received:message-id:date:from:to:subject:mime-version:content-type;
bh=9gicsZnlcLK7yYh6VIrgyAMMRZiWsSbWqSPIhc78RRk=;
b=k4ofvpHPkaQmvuSoGVhRrnCsPK+JEuv9KUrZO7aiypvf/6Y1N2iIatvLvdzwOnZX
  /W6Kxyx6Z4Ybuk8Dqk/vNTIE7Jpy+GQUUHFvMONFtmZo1CbGRvo8DdHnXRBB/qWw
  lV+Z6wxw/mq71NuJknVprOAaTLws5mwcZ+AWL8KwHgO=
```

a) What is the `s=gamma` part, how is it used?

b) How does the client obtain the public key?

c) Explain the principle idea of DMARC.

(3 points)

Problem 6. Give a regular expression that can be used for illicit mail harvesting. At least the following mail variations should be detected:

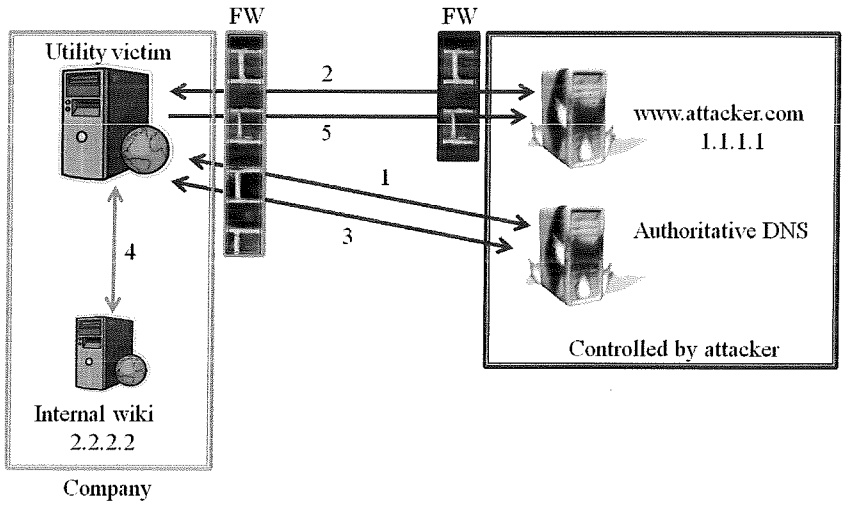
> `first.last@domain.com`
>
> `first` (dot) `last` (at) `domain` (dot) `com`
>
> `first` followed by a dot followed by `last` followed by at followed by `domain` followed by a dot followed by `com`

You may assume that the name parts of the email address contain only alphanumerical characters. (3 points)

Problem 7. Consider the following illustration of a DNS rebinding attack.

a) Referring to the picture below, explain how an adversary can use DNS rebinding to breach a company firewall.

b) Where does DNS pinning fit in?



(2+1 points)

Problem 8. Consider Domain Name System Security Extensions (DNSSEC).

a) The DS record is used to verify the correctness of the public key in DNSKEY. What does the DS record contain?

b) How does DNS amplification relate to DNSSEC?

c) How does NSEC3 solve the zone enumeration problem?

(3 points)

Problem 9. What is the difference between first and third party cookies? Make sure that you explain what a cookie is. Also, explain how third-party cookies can be used for user tracking or advertising. Why is sniffing an efficient way of retrieving cookies when the client does not use SSL/TLS? (3 points)

Problem 10. Here is some code that may have been used in the 2014 celebrity photo iCloud breach.

```
def TryPass(apple_id,password):
  url = 'https://fmipmobile.icloud.com/fmipservice/device/'+apple_id+'/initClient'
  headers = {'User-Agent': 'FindMyiPhone/376 CFNetwork/672.0.8 Darwin/14.0.0', }

  <some omitted code>

  base64string = base64.encodestring('%s:%s' % (apple_id, password)).replace('\n', '')
  req.add_header("Authorization", "Basic %s" % base64string)
  resp = urllib2.urlopen(req)
```

a) What is a User-Agent?

b) Note that Basic HTTP authentication is used together with SSL. Is that a reasonable way of utilizing Basic HTTP authentication, why or why not?

c) Is it reasonable to use Digest HTTP authentication together with SSL in the same way, why or why not? (3 points)

Problem 11. Consider a website that stores unsalted MD5-hashed passwords in a database. In an on-line attack, passwords are tested against the fully functional website. If the database (together with its credentials) containing the hashed passwords is stolen, an off-line attack can be performed locally on the attackers computer.

Consider the following password attacks:

a) On-line brute-force

b) Off-line brute-force

c) Dictionary/TMTO (Off-line)

Classify the following actions on the website server according to efficiency against each password attack. Categorize as {not, somewhat, very} efficient.

1) Setting strict password criteria; minimum 15 symbols, at least one each from [A-Z], [a-z] and [0-9], disallowing the 10.000 most common passwords,

2) Salting the passwords (unique for each user)

3) Slower hash function

4) A timeout of 3 minutes after 5 failed login attempts

5) Using a CAPTCHA (5 points)

Problem 12. The social networking site Myspace was infected by the XSS worm Samy in 2005. Myspace used a secret validation token for CSRF protection. This protection was bypassed by the worm in order to send friend requests to its creator as it propagated.

a) Explain how CSRF protection with a secret validation token works.

b) Explain how XSS can be used to bypass this type of CSRF protection.

Hint 1: You may assume that Myspace had unfiltered input fields.
Hint 2: JavaScript can access the entire HTML DOM-tree. (2+3 points)

Problem 13. Consider a Hashcash solution in which a string

$$ver : bits : date : resource : rand : counter$$

is hashed using SHA-1, where

ver is version number (currently 1),
bits indicates how costly the function is for sender,
date gives current date,
resource is recipients email address,
rand is a random number (separates different senders).

A spammer wants to send 1000 messages $M_1, \ldots, M_{1000}$ to each and every recipient on his very large mailing list, and he plans on including a Hashcash header with each mail.

a) How many calls to SHA-1 does it take to *generate* a Hashcash header with $bits = 30$? Exactly or on average?
b) How many calls to SHA-1 does it take to *verify* a Hashcash header with $bits = 30$? Exactly or on average?
c) Which values for *bits* are reasonable for normal Hashcash header usage?
d) What prevents the spammer from using the same Hashcash header when sending the message M_1 to *all* recipients on his mailing list?
e) What prevents the spammer from using the same Hashcash header when sending all messages $M_1, \ldots, M_{1000}$ to *one* specific recipient? (5 points)

Problem 14. Briefly explain the following terms.

a) Nolisting
b) File inclusion
c) Same-origin policy
d) Reduction function
e) Cache poisoning (5 points)
