Final exam in

Web Security EITF05

Department of Electrical and Information Technology
Lund University

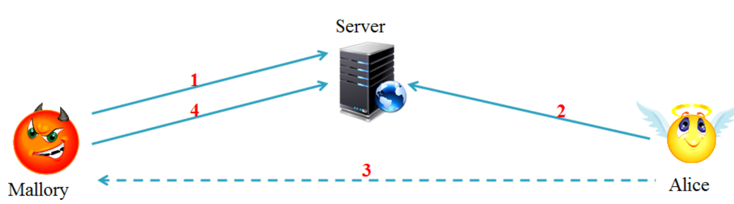
October 30$^{\text{th}}$, 2015, 14.00–19.00

- 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

Problem 1. Consider the following illustration of an XSS attack.



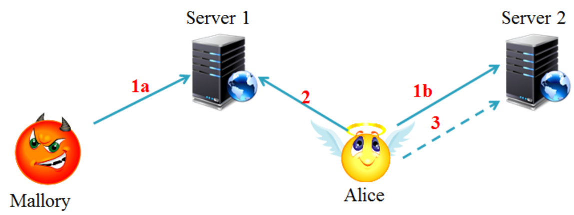
a) Explain how an XSS attack works, detailing where JavaScript(s) reside and are executed. You may refer to the picture.

b) Does TLS protect against XSS attacks? Motivate.

c) Does the same-origin policy protect against XSS attacks? Motivate.

(3 points)

Problem 2. Consider the following illustration of a CSRF attack.



a) Explain how a CSRF attack works. You may refer to the picture.

b) Explain how CSRF protection with synchronizer token pattern works.

c) CSRF attacks are possible even when Alice has disabled JavaScript in her browser. Describe one feasible scenario.

(3 points)

Problem 3. 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/mq7lNuJknVprOAaTLws5mwcZ+AWL8KwHgO=
```

a) How can the client verify that she has obtained the correct public key (for signature verification)?

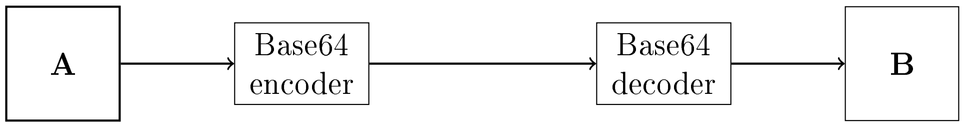
b) Can a domain use different keys for different users or groups of users? Motivate.

c) How does DKIM provide integrity protection for the message part of the email? (What is signed, and how?)

(3 points)

Problem 4. A company has a problem with a communication channel, and you have been hired to resolve the issue. The problem is as follows.

Device A sends single control bytes to device B. A uni-directional channel that requires printable characters to be sent is used, so each byte is Base64-encoded before it is sent from A, and Base64-decoded before it is interpreted at B.



Recently, the Base64 decoder was replaced with a new and improved implementation. However, after this upgrade, only a small fraction of the messages are delivered – most are discarded by the decoder.

One particular message that could be transmitted before but not after the upgrade is "wd==".

a) Explain what the most likely problem is, and how it should be fixed.
b) What exact fraction of the messages does the new decoder discard? Assume uniform distribution where applicable.

Hint 1: Symbol w is 0x30 and d is 0x1D in Base64.

Hint 2: The following section is from RFC4648 titled "The Base16, Base32, and Base64 Data Encodings".

```
3.5.  Canonical Encoding

   The padding step in base 64 and base 32 encoding can, if improperly
   implemented, lead to non-significant alterations of the encoded data.
   For example, if the input is only one octet for a base 64 encoding,
   then all six bits of the first symbol are used, but only the first
   two bits of the next symbol are used.  These pad bits MUST be set to
   zero by conforming encoders, which is described in the descriptions
   on padding below.  If this property do not hold, there is no
   canonical representation of base-encoded data, and multiple base-
   encoded strings can be decoded to the same binary data.  If this
   property (and others discussed in this document) holds, a canonical
   encoding is guaranteed.

   In some environments, the alteration is critical and therefore
   decoders MAY chose to reject an encoding if the pad bits have not
   been set to zero.  The specification referring to this may mandate a
   specific behaviour.
```

(2+1 points)

Problem 5. 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.

a) Explain the principles of Hashcash, and detail the relationship between the *counter* and *bits* parameters.
b) How can HashCash be misused if the *rand* parameter is removed from the protocol?

(2+1 points)

Problem 6. An IPv4 address is a group of four numbers from 0 to 255 (inclusive) separated by dots, as in 127.0.0.1, 130.235.202.25, and so on.

Write regular expressions for matching the following.

a) All IPv4 addresses, with no additional restriction on the numbers.
b) All IPv4 addresses, restricting all four numbers to $0, \ldots, 255$.

Hint 1: You may disregard leading zeros.

(1.5+1.5 points)

Problem 7. Consider Domain Name System Security Extensions (DNSSEC).

a) Explain how the correctness of the public key in DNSKEY is verified. Compare the procedure to that of verifying a certificate.
b) How can NSEC and NSEC3 provide precomputed answers for *all* requests? (What trick is used?)
c) Describe one very significant problem with DNSSEC.

(3 points)

Problem 8. Explain how Content Security Policy (CSP) works.

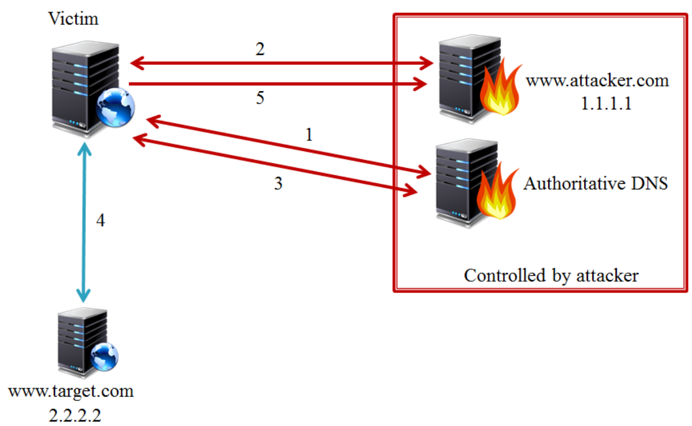
(3 points)

Problem 9. Consider the following illustration of a DNS rebinding attack.

a) Referring to the picture below, explain how JavaScript is used in a DNS rebinding attack.

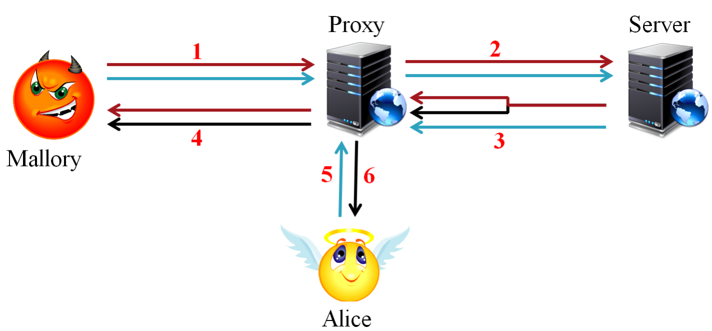
b) Why will the attack fail if the attacker instructs the script to go directly to www.target.com? Motivate. (Which protection mechanism is triggered in this case?)

c) What can the target server (www.target.com) do to avoid serving content in DNS rebinding attacks? (How can it detect DNS rebinding attacks?)



(3 points)

Problem 10. Consider the following illustration of an HTTP response splitting attack.



a) Referring to the picture, briefly explain how an HTTP response splitting attack works.

b) Write some PHP code that enables the attack. Your code does not need to be syntactically correct, but it should show a feasible case.

c) What would happen if the proxy was not a *caching* proxy?

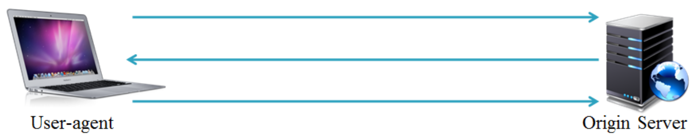
(3 points)

Problem 11. HTTP digest authentication (RFC2617) is a challenge response protocol in which the client calculates the digest (the response) according to

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

with

$$\begin{aligned} A1 &= username : realm : password, \\ A2 &= \begin{cases} method : URI & \text{if } qop = auth, \\ method : URI : \text{MD5}(entity\text{-}body) & \text{if } qop = auth\text{-}int. \end{cases} \end{aligned}$$



a) Explain the usage and purpose of the *qop* parameter?
b) Explain the usage and purpose of the *nonce* parameter.
c) In which form are the credentials stored on the server?
d) Both *nonce* and *cnonce* are randomly selected strings, so why is it that the *cnonce* parameter protects against a MITM that can modify messages and has TMTO/Rainbow capabilities, when the *nonce* parameter does not?

(1+1+1+2 points)

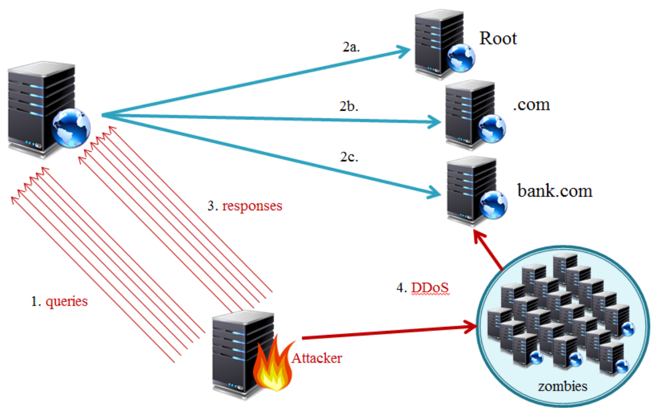
Problem 12. You have successfully stolen a database from a popular website using an SQL injection. Passwords in this database have been salted with a site-wide salt, same for all passwords, and then hashed with SHA3-512. You have full access to the database, and the salt is known to you. You want to recover as many passwords as possible, so you are considering a TMTO attack with parameters t, n and ℓ. That is, the TMTO attack uses t tables, each with n start-/endpoint pairs (SP/EP-pairs), with chains of length ℓ.

a) Why will you not find pre-built tables online that you can use?
b) Explain how chains are constructed using the hash- and reduction functions.
c) Explain why and how hash table(s) are used during password recovery.
d) In terms of t, n and ℓ, what is the expected time-complexity for recovery of *one* password? (How many hash table lookups?)

Hint 1: Alternatively, you may answer the same question for a rainbow attack setting with parameters n and ℓ as above (one table). If you do so, please state this clearly.
Hint 2: Hash tables can provide $O(1)$ lookup time. That is, lookups into a hash table can be performed in constant time, regardless of how many entries that are stored.

(1+1+2+1 points)

Problem 13. Consider the following illustration of a DNS cache poisoning attack.



a) Why is DNS spoofing (DNS cache poisoning) trivial for an adversary that can observe outgoing traffic (2a, 2b and 2c above) from the DNS server?

b) Modern DNS implementations do not only randomize transaction IDs. What else do they randomize?

c) How would usage of TCP (instead of UDP) protect against DNS cache poisoning attacks?

d) Make a reasonable estimate of how long it would take for an attacker to succeed with a DNS cache poisoning attack for a moderately popular website. State your assumptions, estimates and approximations along with your calculation.

(1+1+1+2 points)

Problem 14. Briefly explain the following terms.

a) Birthday paradox

b) SPF

c) CORS

d) Greylisting

e) Idempotent HTTP method

(5 points)
