Final exam in

Web Security EITF05

Department of Electrical and Information Technology
Lund University

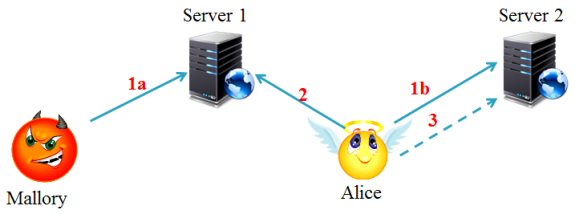
October 28th, 2016

- 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 a CSRF attack.



a) Are CSRF attacks possible if Alice has disabled JavaScript in her browser? Motivate.
b) Are CSRF attacks possible if Alice has disabled third-party cookies in her browser? Motivate.
c) Can an XSS attack disable CSRF protection with the synchronizer token pattern? Motivate.

Answer

a) Yes. Request to Server 2 can be embedded in an image tag.
b) Yes. The attack relies on first-party cookies only, not third-party cookies.
c) Yes. XSS attack can use JavaScript to access entire DOM, including the CSRF token.

(3 points)

Problem 2. You are creating an image harvester, so you need to design a regular expression for your web crawler. Give a regular expression that matches an HTML image tag with a non-empty source attribute. The following variations should match;

`<img src="pic.jpg" any_name="any_value"/>` (any attribute after src is ok)

`<img alt="Me on vacation" src="pic.jpg"/>` (any attribute before src is ok)

but not

`<img src="pi>c.jpg"/>` (no greater than inside tag)

`<img />` (src attribute must exist)

`<img src=""/>` (src attribute must be non-empty)

Answer
One possibility is
`<img[ a-zA-Z0-9=".]*src="[a-zA-Z0-9.]+"[ a-zA-Z0-9=".]*\/>`. (3 points)

Problem 3. Consider Domain Name System Security Extensions (DNSSEC).

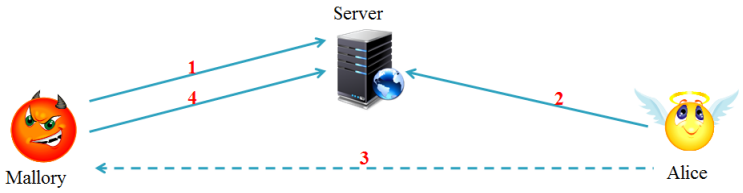
a) What does the DS record contain?
b) All DNSSEC responses are (asymmetrically) signed. How does this affect performance (response time compared to ordinary DNS)? Motivate your answer.
c) Why is zone-walking easy with NSEC but not with NSEC3?

Answer

a) A hash of the public key in DNSKEY, stored at the parent domain.
b) Not at all, all answers are pre-signed.
c) NSEC lists domain names alphabetically and provides neighboring pairs as answers. In NSEC3, ordering is by *hash* of domain name.

(3 points)

Problem 4. Consider the following illustration of an XSS attack with three involved entities; Mallory, Server and Alice.



a) Does TLS protect against XSS attacks? Motivate.

b) What can Alice do to efficiently prevent XSS attacks? Motivate.

c) What can the Server do to efficiently prevent XSS attacks? Motivate.

Answer

a) No. SSL operates on session level in the OSI model, while script injection is applied on application level. SSL protects transport of content, but it does not look at the content itself. Injected scripts reside in (are part of) the website content that is stored on the server. This content is interpreted by the victims browser after transport.

b) She can switch off client-side scripting, but that would break the functionality of many web sites.

c) Apply CSP. Filter user input to avoid script injection.

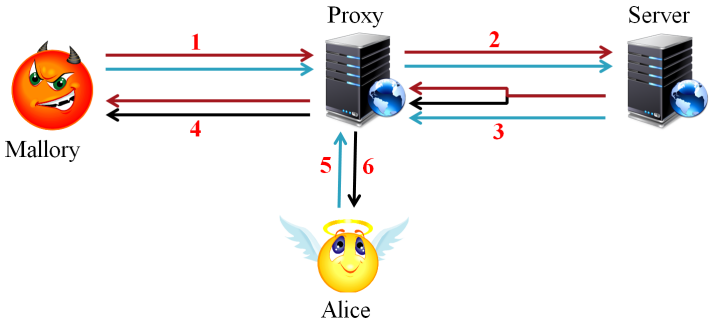
(3 points)

Problem 5. Explain the purpose and principal functionality of Domain-based Message Authentication, Reporting and Conformance (DMARC).

Answer

See lecture notes. (3 points)

Problem 6. Consider the following illustration of an HTTP response splitting attack.



a) Briefly explain how an HTTP response splitting attack works.

b) Does TLS protect Alice against HTTP response splitting attacks? Motivate.

c) Can the first response in step 3 be split into, say, 10 responses (instead of 2)? Motivate.

Answer

a) See lecture notes.

b) No. Proxy caches Mallory's spoofed page before Alice retrieves it, and it does not matter if she uses TLS or not.

c) Yes. Mallory exploits an injection point in the PHP code. The first part of the first page, and the last part of the last page are predetermined. Mallory can decide what to put in the middle part, including splitting this into several (fully customized) responses.

(3 points)

Problem 7. 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+GQUUHFvM0NFtmZo1CbGRvo8DdHnXRBB/qWw
  lV+Z6wxw/mq7lNuJknVprOAaTLws5mwcZ+AWL8KwHg0=
```

a) How does DKIM provide confidentiality of the message part of the email? (What is encrypted, and how?)

b) How does DKIM provide integrity protection, and for which parts of the email? (What is signed, and how?)

c) How is the DKIM header itself protected?

Answer

a) It does not.

b) `bh` is a hash of the body. The DKIM header (including `bh`) is included in `h`, even though it is not explicitly listed. Thus, the signature in `b` covers `bh` as well. All signed headers and the message part are covered.

c) Integrity protected according to b). where the signature itself (in `b`) is omitted when including the DKIM header in the message specified by `h`. This message is first hashed and then signed. The signature is Base64-encoded and inserted into the DKIM header.

(3 points)

Problem 8. If the new international version of the Bible were more Base64-inspired, then a paragraph from 1 Samuel 17 might have read:

A champion named `R29saWF0aA==`, who was from `R2F0aA==`, came out of the Philistine camp. His height was six cubits and a span. He had a bronze helmet on his head and wore a coat of scale armor of bronze weighing five thousand shekels; on his legs he wore bronze greaves, and a bronze javelin was slung on his back. His spear shaft was like a weaver's rod, and its iron point weighed six hundred shekels. His shield bearer went ahead of him.

a) Who was this warrior? (decoded name)
b) Where was he from? (decoded name of city)
c) What is his opponent's name in Base64?

Hint: Decimal representation of ASCII characters is given by:

$$\text{A} = 65, \text{B} = 66, \ldots, \text{Z} = 90, \text{a} = 97, \text{b} = 98, \ldots, \text{z} = 122$$

The Base64 alphabet is:

$$0 = \text{A}, \ldots, 25 = \text{Z}, 26 = \text{a}, \ldots, 51 = \text{z}, 52 = 0, 53 = 1, \ldots, 61 = 9, 62 = +, 63 = /$$

(3 points)

Answer

a) Goliath
b) Gath
c) RGF2aWQ=

Problem 9. 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) How is a proper *counter* value determined?
b) How many times must the hash function be invoked to *generate* a valid HashCash header with $bits = 30$? Exactly or on average?
c) How many times must the hash function be invoked to *verify* a valid HashCash header with $bits = 30$? Exactly or on average?

Answer

a) The HashCash header is valid if the hashed string has *bits* initial zeros. You can create a valid HashCash header as follows. Initially construct the string with the *counter* part set to zero. While that string does not hash to a value with *bits* initial zeros (while it is not a valid HashCash string), increase *counter*.
b) About 2^{30}.
c) Exactly once.

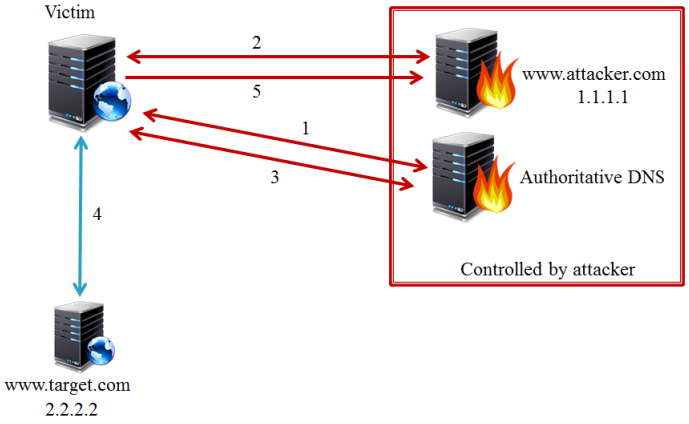
(3 points)

Problem 10. Consider the following illustration of a DNS rebinding attack.

a) Sitting behind a company firewall does not provide the Victim with any protection from DNS rebinding attacks, why is that?

b) Can an XSS exploit be used to trigger a DNS rebinding attack? Motivate.

c) What can the target server (`www.target.com`) do to prevent involvement in DNS rebinding attacks? Motivate.



Answer

a) The victim initiates the attack from behind the firewall, for example by clicking on a link to Mallory's web page.

b) Yes. JavaScript can trigger page retrieval on `www.attacker.com`.

c) Check the HTTP Host header, even if the server does not support virtual hosts.

(3 points)

Problem 11. On October 21st 2016, we saw one of the largest DDoS attacks to date. The attack targeted a major DNS host (Dyn), thus affecting a vast number of sites and services.

On September 30th 2016, the source code for the IoT botnet 'Mirai', that allegedly was used in the attack, was released. The code reveals that IoT devices (IP cameras, routers, ...) with default or hard-coded passwords were harvested for the botnet.

Reports claim traffic loads reaching 1.2Tbps, tens of millions of unique IP addresses being involved and 100,000 physical IoT devices. For comparison, a total of about 6.4 billion IoT devices have been forecast for 2016, and 20.8 billion for 2020.

a) How can DNS amplification be used to maximize the impact of a botnet? (Explain how DNSSEC is involved.)

b) Assuming a worst-case scenario (companies do not care about security, IoT grows exponentially,...), what should you expect in terms of traffic loads from IoT botnets 30 years from now (when you are the seasoned security expert)? Make and state reasonable assumptions and explain your calculations/estimate.

c) How could and should DDoS attacks from IoT botnets be mitigated? (Open question.)

Answer

a) Since answers from a DNSSEC-enabled DNS server are signed, the size of the answers increase. For small requests, responses can be about 50 times larger. UDP is used, so bots can query DNS and write the victim's IP address as return address in the request. This increases the traffic load (efficiency of the botnet) by a factor of about 50, compared to not using DNS amplification.

b) No given answer.

c) No given answer.

(1+2+2 points)

Problem 12. HTTP digest authentication (RFC2617) is a challenge response protocol in which the client calculates the digest (the response) according to

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

with

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

a) Explain the usage and purpose of the *realm* parameter.

b) In which form are the credentials stored on the server?

c) Follow-up question to b): In what way is this worse than the way operating systems typically store credentials?

d) Both *nonce* and *cnonce* are random strings. While *cnonce* protects against TMTO attacks, *nonce* does not. Why is that?

Hint for d: How can you attack the authentication system if there is no *cnonce*?

Answer

a) A text string telling the user which password to enter (same site can use different ones for access to different resources).

b) $MD5(A1)$ is stored on the server.

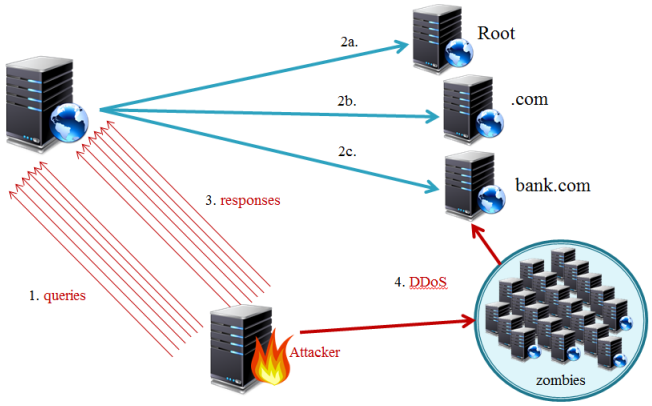
c) $MD5(A1)$ is essentially the password, and this value can be read by anyone with server access.

d) A MITM could perform a TMTO attack if the *cnonce* parameter is not present. She replaces all randomness generated by the server with her own pre-determined random looking string that she has pre-built her TMTO tables for. This does not work if the *cnonce* parameter is present, because it is generated by the client and used when preparing the digest, and it is revealed to the MITM *afterwards*. Pre-generate tables therefore becomes meaningless, as the MITM would need to use different tables for every request, leaving the MITM with brute-force as a better but infeasible option.

(1+1+1+2 points)

More on next page!

Problem 13. Consider the following illustration of a DNS cache poisoning attack. The success rate of the attack depends on how many queries and responses that can be sent in steps 1 and 3 (before the first response in step 2c has been delivered).



a) Explain why the birthday paradox applies to DNS cache poisoning.

Roughly how many queries and responses need to be sent in steps 1 and 3 if the DNS server randomizes

b) only transaction IDs?

c) both transaction IDs and port numbers?

How is the success probability of the attack affected if

d) transaction IDs in step 3 are generated sequentially rather than uniformly at random?

e) the spoofed responses in step 3 are all sent from different computers (botnet)?

Answer

a) The primary DNS (P) queries the `bank.com` DNS (B) for the `www.bank.com` subdomain. The queries from P include randomized transaction IDs, generated by P. P also stores these so that they can be matched with the response from B. P simply discards any non-matching responses. Since several requests are sent more or less at the same time from P to B, P will store many different transaction IDs, and the Attacker needs to match the transaction ID of one of her spoofed responses with any one of the outstanding requests from P to B.

b) Transaction IDs are 16 bits, so roughly $2^8 = 256$.

c) Transaction IDs and port numbers together are 32 bits, so roughly $2^{16} = 65536$.

d) Not affected (or a little better actually since we avoid collisions among the spoofed responses).

e) Not affected.

(5 points)

Problem 14. Briefly explain the following terms and acronyms.

a) Remote file inclusion
b) PTR record
c) CSP
d) Reflected (non-persistent) XSS attack
e) First-party cookies

Answer

a) An attack that involves including and interpreting PHP code from a secondary (remote) server.
b) DNS record used for reverse lookup.
c) Content Security Policy, standard for preventing XSS, lets domain choose which content sources that are trustworthy.
d) An XSS attack in which (part of) the request is mirrored on the web page, like repeating the query on a search page.
e) Text string stored in the user's computer, created by the web site the user is visiting (main document).

(5 points)
