Technische Universität Berlin
SoSe 2026

Prof. Stefan Schmid
Berthold Blatt Lorke

Internet and Network Security

Musterlösung der Klausur vom 21.07.2026

Aufgabe 1

Mark for each statement if it is true or false. Every correct answer gives points, and a false answer does not cause any penalty.

(a) [2 points] In RSA, a message is encrypted with the recipient's private key.

☐ True. ☒ False.

(b) [2 points] The Same-Origin Policy prevents a website from setting any cookies.

☐ True. ☒ False.

(c) [2 points] In a Tor circuit, only the exit relay sees the unencrypted payload.

☒ True. ☐ False.

(d) [2 points] WEP is considered a secure protocol for protecting Wi-Fi traffic.

☐ True. ☒ False.

(e) [2 points] A reflection-based DoS attack uses a spoofed source IP address so that responses are sent to the victim.

☒ True. ☐ False.

(f) [2 points] PGP provides non-repudiability.

☒ True. ☐ False.

Aufgabe 2

A block cipher such as AES is used in **Cipher Block Chaining (CBC)** mode.

a) [3 points] What is the purpose of the Initialization Vector (IV) in CBC mode? Explain your answer!

b) [3 points] What can go wrong, if the same IV (and key) is used to encrypt two messages that start with the same content?

Solution:

a) The IV randomizes the encryption so that encrypting the same plaintext (under the same key) twice yields different ciphertexts. It is XORed with the first plaintext block before encryption, which removes deterministic patterns across messages.

b) If the IV (and key) are reused, identical message prefixes produce identical ciphertext blocks. An attacker observing the ciphertext can tell that two messages begin with the same content (loss of confidentiality / leakage of plaintext equality), which can leak information about the data.

Aufgabe 3

The unprivileged user `student` on a server `server01` is restricted by these outbound firewall rules:

```
iptables -A OUTPUT -p all -m owner --uid-owner student -d 127.0.0.1 -j ACCEPT
iptables -A OUTPUT -p all -m owner --uid-owner student -d 10.7.0.0/24 -j ACCEPT
iptables -A OUTPUT -p all -m owner --uid-owner student -j DROP
```

The user may still log in via SSH.

(a) [2 points] `student` logs in on `server01` and runs `curl https://thepiratebay.org`. Explain why the website cannot be reached.

(b) [2 points] Can `10.7.1.67` be reached?

(c) [2 points] Describe why a reverse tunnel such as

```
ssh -R 127.0.0.1:8080:thepiratebay.org:443 student@server01
```

can circumvent the restriction from (a). That is, a malicious actor with ssh access to `server01` as user `student` can get the contents from `https://thepiratebay.org` via `server01`.

Solution:

a) The final rule drops any outbound packet by a process owned by `student`, and `thepiratebay.org` is not in the two allowed destination ranges.

b) No, `10.7.1.67` is not in the `10.7.0.0/24` range, and also is not `127.0.0.1`.

c) The user's `curl` only talks to the allowed `127.0.0.1`, while the actual outbound connection is made by `sshd` running as root, so the `--uid-owner student` rule does not match and the traffic is dropped.

Aufgabe 4

Let H be a cryptographic hash function with an output of m bits.

a) [3 points] Briefly explain what a cryptographic hash function is. Name two essentials properties.

b) [3 points] A **collision** is a pair of distinct messages $x \neq y$ with $H(x) = H(y)$. Roughly how many hash evaluations does an attacker need to find a collision with probability$\geq$ 50% by brute force, and what is this attack called?

Solution:

a) A cryptographic hash function takes an arbitrary length input and computes a fixed sized output (of size m bits). Properties: Many-to-one, easy to compute, fixed size output, collision resistance.

b) About $2^{m/2}$ evaluations, because of the **birthday paradox**. The attack is called the **birthday attack**.

Aufgabe 5

A server wants to authenticate a client. The client proves its identity by sending its password encrypted with a shared secret key. The server accepts if the decryption yields the correct password.

a) [3 points] An attacker who can eavesdrop on the network defeats this protocol without ever learning the password. Describe the attack.

b) [3 points] How does adding a **nonce** (a fresh, one-time random value chosen by the server) fix the problem?

Solution:

a) **Replay (playback) attack**: the attacker records the encrypted password sent by the legitimate client and later resends (replays) the exact same ciphtertext to the server. The server decrypts it to the correct password and authenticates the attacker, even though the attacker never learned the password.

b) The server first sends a fresh random nonce R to the client. The client responds with $E_K(\text{password}, R)$. The server checks the response against the R it just issued. Because a new R is used every time, a recorded old response is invalid for the next session, so replays fail.

Aufgabe 6

a) [3 points] Explain how a **SYN flooding** attack works, and why it denies service to legitimate clients

b) [3 points] Explain how **SYN cookies** defend against this attack.

Solution:

a) The attacker sends many TCP `SYN` packets (typically with spoofed source addresses) but never completes the three-way handshake (no final `ACK`). For each `SYN` the server allocates space for a half-open connection. The table of half-open connections fills up, so the server can no longer accept connections from legitimate clients.

b) With SYN cookies, the server does **not** store space on receiving a `SYN`. Instead it encodes the connection state into the initial sequence number of the `SYN-ACK` (computed from the connection paramters and a secret). Space is only created when the client returns a valid `ACK`, whoe acknowledgement number lets the server reconstruct / verify the cookie. Thus, spoofed `SYN`s that are never acknowledged consume no memory.

Aufgabe 7

In Tor, a client builds a circuit through three relays (entry, middle, exit) and shares a seperate symmetric key K_1, K_2, K_3 with each of them.

a) [3 points] The client wants to send the payload D to the exit relay. In which order does it apply the keys, i.e. what does it put on the wire towards the entry relay? Explain why this is called **onion** routing.

b) [3 points] Why can no single relay link the client to the destination server?

Solution:

a) The client wraps the payload in layers, encrypting with the exit key first and the entry key last: $K_1(K_2(K_3(D)))$. Each relay removes (decrypts) exactly one layer with its own key before forwarding, like peeling an onion – hence onion routing.

b) Each relay only knows its immediate predecessor and successor (local view). The entry relay sees the client but not the destination; the exit relay sees the destination (and the payload) but not the client; the middle relay sees neither endpoint's data. So no single relay simultaneously knows both who the client is and what it is communicating with.

Aufgabe 8

The following PHP code reflects a search term back to the user:

```
<?php
echo "Results for: " . $_GET['term'];
?>
```

(a) [2 points] Give a concrete value for the `term` parameter that triggers a Cross-Site Scripting (XSS) attack, and state what type of XSS this is.

(b) [2 points] Why is this dangerous? Name one concrete thing the injected script can do.

(c) [2 points] How should the developer fix the code to prevent the attack?

Solution:

a) E.g. `term=<script>alert('XSS')</script>` (any payload that injects executing JavaScript is accepted). Because the payload comes from the request and is not stored, this is **reflected XSS**.

b) The script runs in the victim's browser in the site's origin, so it can e.g. steal the session cookie (`document.cookie`) and send it to the attacker, deface the page, or perform actions as the victim.

c) Apply context-correct **output encoding** of the user input before placing it in the HTML, e.g. `htmlspecialchars($_GET['term'], ENT_QUOTES)`, so that `<`, `>`, etc. are rendered as text instead of being interpreted as HTML/script. (Input validation / a CSP are also acceptable as additional measures).

Aufgabe 9

Smartphones discover Wi-Fi networks by broadcasting unencrypted **probe requests** on all channels.

a) [3 points] Why can an attacker **track** the device before it ever connects? Name the two identifying pieces of information a probe request leaks.

b) [3 points] What does MAC address randomisation do, and why can tracking still sometimes succeed despite it?

Solution:

a) Probe requests are cleartext broadcasts, so any passive sniffer can receive them without association. They leak (i) the device's persistant **MAC address**, allowing the attacker to correlate proves across time / location, and (ii) the **preferred network list (PNL)**, i.e. SSIDs of previously used networks, which can reveal where the user has been.

b) Randomisation uses a periodically rotated, random MAC (per network) instead of the real hardware address, breaking the persistant identifier. Tracking can still succees through physical-layer **fingerprinting**, timing / content features, asynchronous rotation with another identifier such as BLE, or distinctive PNL contents.