CHALMERS | GÖTEBORGS UNIVERSITET

Network Security

EDA491 (Chalmers)
DIT071 (GU)

2018-08-31, 14:00 – 18:00

No extra material is allowed during the exam except for an English language dictionary in paper form. **No electronic devices allowed.**

The last page of this exam contains pictures of some protocols and headers that *may* be useful in some questions.

Give clear answers. Your thoughts and ways of reasoning must be clearly understood! Questions must be answered in English.

Teacher:	Tomas Olovsson Dept. of Computer Science and Engineering
Questions during exam:	Tomas Olovsson, 031 - 772 1688

Inspection of exam: See web page for announcement

CTH Grades:	30-38 → 3	39-47 → 4	48-60 → 5
GU Grades:	30-47 → G		48-60 → VG

1. Attacks

a) Ping of death is a well-known attack. It sends IP datagrams with a size > 64 kByte. However, the maximum size of an IP datagram is 64kByte due to its 16-bit length field. Explain how is it still possible send such oversized datagrams! (2p)

A naïve implementation would assume IP datagrams never exceed 65,535 bytes since the length field is 16 bits long.
An oversized IP datagram can be created that exceeds this size by sending a fragment with an offset and a length extending the datagram beyond this limit, for example by setting offset to 65,000 and length to 1,000 bytes.

b) Give an example of a DDoS attack that is hard to trace and stop! Motivate your solution! (Several answers may exist to this question.) (2p)

See for example the slide "DDoS Attacks through Handlers and Zombies".

c) If we have discovered that a system is protected behind a firewall and does not respond to any traffic from us, a third system (a zombie) may be used to scan the "hidden" system for services. One example may be a public web server which is trusted to talk to a backend, for us hidden, server that we are interested in.

How can we scan the backend server and see what services it offers? What IP feature is the attack based on? Try to keep the discussion on a fairly high level but still detailed enough to show that you understand the concept! (3p)

Use dumb/idle scanning: Many implementations increment fragment (packet) ID for each packet sent. A zombie, i.e. another host that may be trusted to talk to the system, can be used if it always increases its fragment numbers by one for each packet sent.

We begin with sending a SYN to the zombie and record the fragment ID. Then we send a SYN to the target system with the zombie's address as the source. The server will send a SYN/ACK to the zombie if it is trusted and the port is open, and the zombie will respond with a RST and increase the fragment number by one. Now we send another SYN to the zombie. If the fragment number was incremented with just one, no response was received, by two it got a reply from the server and the port was open.

d) The allocation of SYN cookies is a possible defense against SYN flooding DoS attacks. The defense is to store a SYN cookie in the initial serial number (ISN) selected by the server:

ISN = hash (source and destination IP-addr. port numbers, secret, client ISN)

Explain what a SYN flood attack is and why SYN cookies can offer protection and why it works! Are there any drawbacks or problems with this method? (3p)

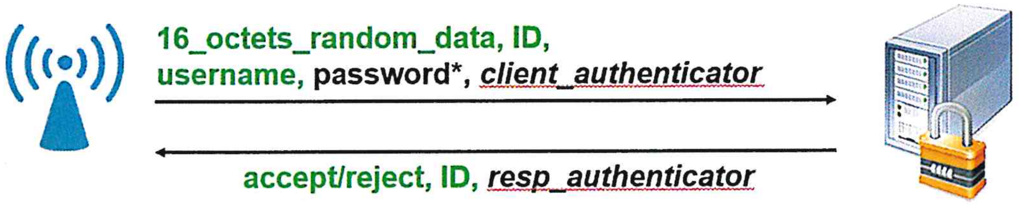
Each SYN makes the receiving host allocate some internal resources (keep state) and return a SYN/ACK message. Internal resources (e.g. memory or data structures) may at some point be exhausted.

Idea with SYN cookies: The server does not save any data when a SYN is received. Instead all state information is contained in the ISN which means that it does not have to allocate any resources until it receives the ACK. Then the ISN is checked with what the client sends back.

Reason it works: The attacker must be using a valid IP address to receive the SYN/ACK with the ISN to be able to respond with a valid ACK. Only if the ACK is valid, resources are allocated. (If one or a few addresses still flood the server, a firewall could easily filter or block this IP address, manually or automatically.)

Drawback: Problems to remember if TCP options (such as window size) that were negotiated in the SYN and SYN/ACK packets since they are not saved by the server. Another drawback may be that the computation of the hash may take some time, a fact that may be utilized by the attacker.

2. Authentication


16_octets_random_data, ID,
username, password*, *client_authenticator*
accept/reject, ID, *resp_authenticator*

password* = MD5(shared_secret, 16_octets_random_data) ⊕ password

client_authenticator = MD5(packet_contents, shared_secret)

resp_authenticator = MD5(packet_contents, 16_octets_sent_by_client, shared_secret)

The picture above shows the initial dialog in Radius when an Authenticator wants the Radius server to authenticate a user.

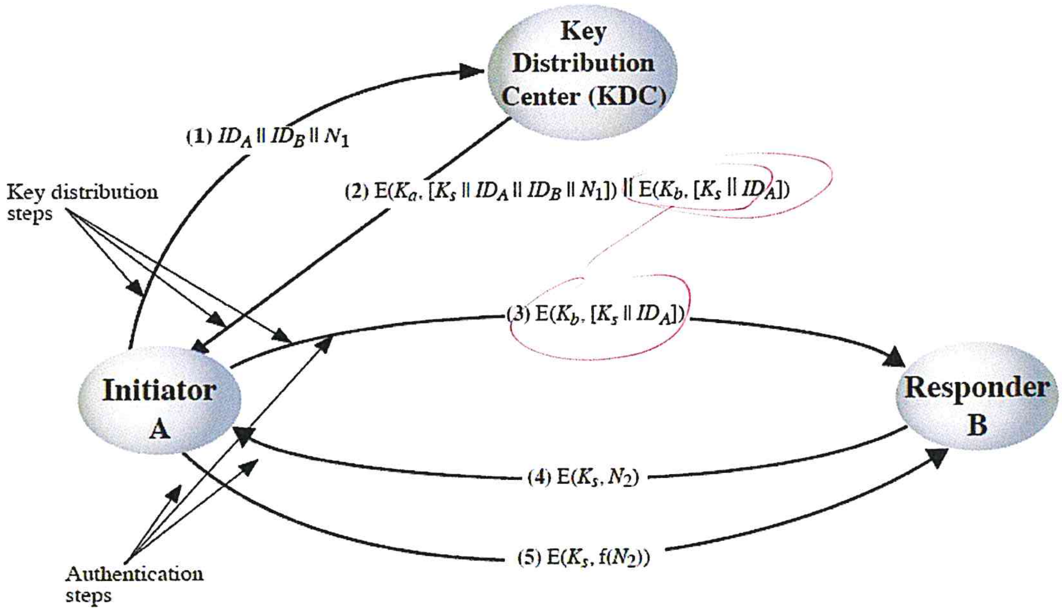
a) If a dishonest supplicant can both send and listen to the traffic between the Authenticator and the Radius server, it is possible to do a number of attacks. Give one example of an attack against the shared secret and one against someone else's password! (4p)

See for example attack 3.1 and 3.4 in the reading material or in the slides.

b) Why is there a *client authenticator* present in the supplicant's message (two purposes)? (2p)

To authorize the client before responding - requests from clients not knowing the correct secret are ignored to prevent guessing attacks. It also prevents message modification.

c) The picture below shows a proposed authentication method using a key distribution center (KDC). In the picture, encrypting something with *Key a* is written E(K_a, …).
Explain on a fairly high level how this protocol works! Don't just repeat what the picture tells but explain with your own words the purpose and outcome of each step! (4p)


Key Distribution Center (KDC)
(1) $ID_A \| ID_B \| N_1$
(2) $E(K_a, [K_s \| ID_A \| ID_B \| N_1]) \| E(K_b, [K_s \| ID_A])$
Key distribution steps
(3) $E(K_b, [K_s \| ID_A])$
Initiator A
Responder B
(4) $E(K_s, N_2)$
(5) $E(K_s, f(N_2))$
Authentication steps

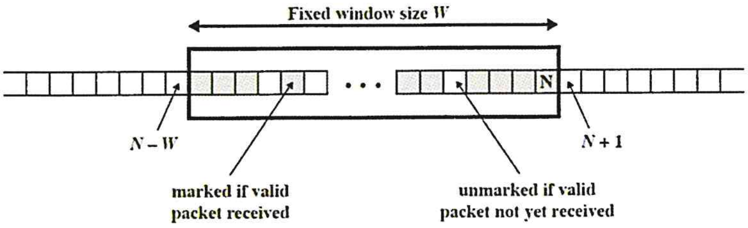
See the book, picture 14.3 and explanation in chapter 15.2.

3. Secure protocols, IPsec, SSL

a) Many security protocols support sending messages without encryption (encryption=NULL) and can still be protected against packet modification. Describe how this is done! (2p)

The protocols use a cryptographic or keyed checksum (e.g. HMAC) where a key is used together with the plaintext: hash(key | text). It requires the key to be known both to verify and to create hashes. This protects the contents of the message even if it is not encrypted.

b) IPsec uses packet numbering, see the picture. For what purpose? (2p)


Fixed window size W
N
N − W
N + 1
marked if valid packet received
unmarked if valid packet not yet received

It protects against replay attacks by discarding duplicates, but unlike TCP, it allows packets to be missing. and arrive out of order.

c) The ESP header differs depending on what mode is used (see header on last page). Why is it possible to keep the original header in transport mode but not in tunnel mode? (2p)

In tunnel mode, it is not the final receiver of the datagram that should receive the encrypted datagram but a host that decrypts it and forwards it to its final destination. Tunnel mode is often used for transparent site-to-site encryption.

d) The ESP header has a field SPI, security parameter index. How is it used? Describe what its purpose is, i.e. how the sender or receiver uses it! Also describe one use of the padding field! (2p)

The SPI is an index that tells what SA (security association) should be used, i.e. a pointer to a data structure containing info about the type of connection, keys used, etc.

Padding can be used to disguise the real length of the datagram.

e) Is SSL sensitive to NAT (network address translation)? Explain your reasoning! (2p)

No. SSL does not include the sender's or receiver's IP addresses in the protocol (it just secures a TCP session and is independent of both TCP and IP). Transport layer protocol.

4. Firewalls

a) Describe briefly how a stateful firewall works! What information does the firewall (at least) need to save for TCP connections? (2p)

Source and destination IP addresses, source and destination ports, TCP state (according to the state machine), TCP sequence numbers. (Real firewalls store more information but this information is at least needed for a firewall to be called stateful.)

b) If you are given the task to test whether a firewall is stateful or not, how would you do this? Describe what you do (send) and the expected result! Explain clearly why this test would work! (2p)

We could do an ACK or a FIN scan. All normal systems respond with a RST packet to non-matching ACKs (RFC 793). A reply from a system behind the firewall means that the firewall did not keep state and had to forward the ACK to the inside. A stateful firewall would silently drop the packet.

c) Assume you are given the task to test a firewall. It is supposed to be a good deep packet inspection firewall and your task is to give a verdict of it and whether it can be used as a firewall to protect a mid-size company (500 users or so). You can configure it in any way you want but cannot make any assumptions about the firewall's functionality without testing it.

How would you perform the tests? Outline a series of tests you would perform to really stress and test the firewall! In the answer, you need to show that you understand what the tests do. 1p is given for each test described. (6p)

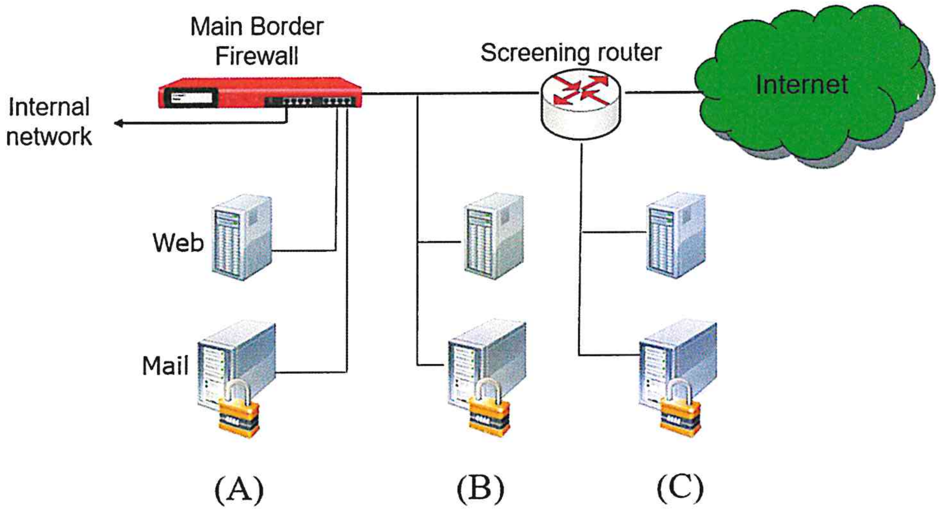
There are many possible answers to this question. One possibility is to set up the firewall in front of a system offering some services, and then test its functionality by:

- Port scanning to see that blocked services are blocked (i.e. test filter rules)

- Test if strange options (source routing, ...)
- Test that clearly invalid packets (IP, TCP, ...) are dropped
- Test if it is stateful (ACK scanning for example)
- Test its handling of fragmentation (ping of death, Teardrop, ...)
- Test fragmentation and if headers can be changed (small fragments)
- Test flooding attacks against firewall and protected systems (SYN flooding)
- Use tools to test attacks against application level protocols (e.g. HTTP, SMTP, ...)
- Test logging abilities
- etc.

5. IDS Systems and network design

a) We normally place external servers on a DMZ, separate from the internal network. In the picture below, we have three suggestions for where to place a web server and a server for incoming mail traffic, A, B and C:



Compare the three suggested locations and explain their advantages and/or disadvantages! (3p)

(A) is the most secure placement (but the firewall may have a limited number of ports if we have many servers).
(B) is the least secure solution. A cracked server (mail or Web) makes it possible for an attacker to listen to all traffic in and out from the corporate network.
(C) is better than (B) but a cracked Web server allows the attacker to listen to all email traffic.

b) We discussed the Jericho Forum during the course. The idea they propose is to move protection closer to the end-points. Does it make sense? Explain your thoughts with some details! (3p)

Services are located both on internal networks and on external (cloud) servers, and users are located both on the inside and on the outside and also bring laptops and other devices from the outside. WLANs also extend the range of the networks outside the physical network boundaries. It is therefore impossible to control security with only a border firewall that only sees a small part of all traffic. The solution is to move protection closer to the end-points: toward the users/client systems and to application servers.

c) Explain why TCP segment reassembly can be problematic for a firewall or IDS system! Give examples of potential problems (the question is NOT about IP fragmentation)! (2p)

It is not well-defined how overlapping TCP segments should be handled. When segments overlap, e.g. in the beginning or in the end with other segments, different systems have different reassembly policies and an IDS system may not understand how the target system will act.

d) TTL can sometimes also be used to fool firewalls and intrusion detection (IDS) systems. How? Mention a possible protection mechanism against this attack. (2p)

Low TTL values may cause some datagrams to be discarded by routers which in turn may result in that the IDS system and the receiving hosts having different meaning of what has been communicated over the network. Border routers could normalize incoming TTL values to a standard value, say 20.

6. Link-level security and WLAN

a) More advanced switches can have protection against several types of link-level problems and attacks. Describe *two* different security mechanisms which may be found in switches, and what they protect against! (2p)

For example; MAC address flooding protection (limit number of addresses per port), DHCP spoofing (trusted ports), block traffic between clients (protected ports), 802.1x port-based access control, lock MAC addresses to ports, functionality to detect IP address spoofing (MAC-IP address monitoring), etc. For details, see slides.

b) What is the major difference between WEP and SSL when considering message integrity during data communication? (2p)

WEP uses a linear checksum (CRC) and SSL hash functions

c) The 802.11i framework (WPA2) offers substantially better security than WEP. Mention two improvements in this protocol! (2p)

AES instead of RC4 encryption, session keys introduced, when all IVs are used new session keys are negotiated, passwords are hashed 4,096 times to make it harder to do off-line searches, session keys were introduced - secret not used directly, etc.

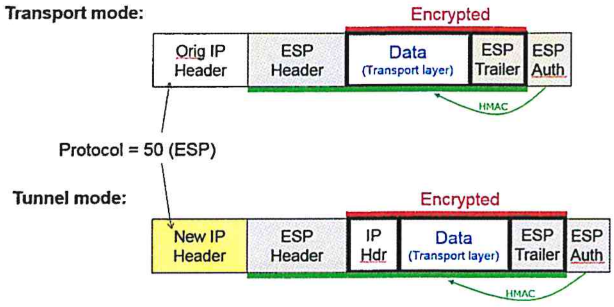
d) Compare WEP and SSL. Describe four differences between the protocols! (4p)

TLS

For example:
- WEP uses shared secrets, SSL supports certificates.
- WEP has no authentication of the AP, only of connecting user, SSL supports authentication of both parties
- SSL: both parties involved in key negotiation, WEP: no negotiation, secret directly used
- WEP key used in both authentication and data encryption, SSL supports DH for key negotiation
- etc.

Headers and pictures that may be useful

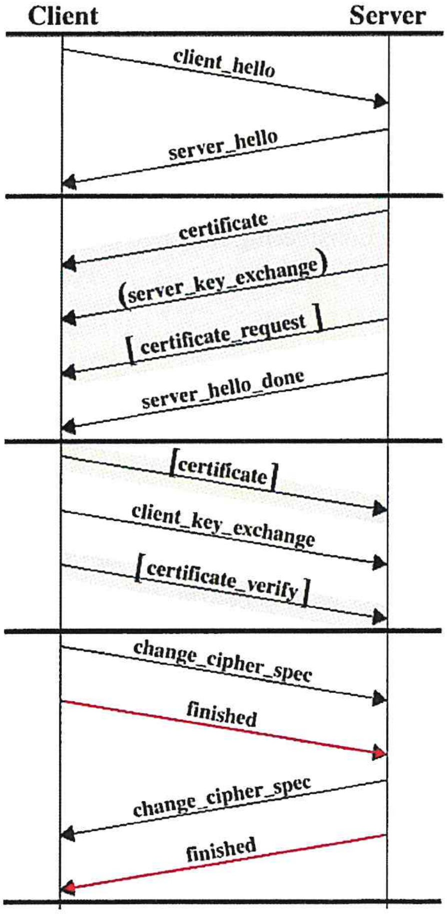
<table>
<tr><th>t: 0</th><th>16</th><th>24 31</th></tr>
<tr><td colspan="3">Security Parameters Index (SPI)</td></tr>
<tr><td colspan="3">Sequence Number</td></tr>
<tr><td colspan="3">Payload Data (variable)</td></tr>
<tr><td colspan="3">Padding (0 - 255 bytes)</td></tr>
<tr><td></td><td>Pad Length</td><td>Next Header</td></tr>
<tr><td colspan="3">Authentication Data (variable)</td></tr>
</table>



<table>
<tr><td colspan="4">Application</td></tr>
<tr><td>Change cipher proto (20)</td><td>Alert protocol (21)</td><td>Handshake protocol (22)</td><td>Appl. Protocol (23)</td></tr>
<tr><td colspan="4">TLS record layer protocol</td></tr>
<tr><td colspan="4">TCP</td></tr>
<tr><td colspan="4">IP</td></tr>
</table>

Bit 0 Bit 31

<table>
<tr><td colspan="8">Source Port Number (16 bits)</td><td>Destination Port Number (16 bits)</td></tr>
<tr><td colspan="9">Sequence number (32 bits)</td></tr>
<tr><td colspan="9">ACK number (32 bits)</td></tr>
<tr><td>Header Length (4 bits)</td><td>Reserved (6 bits)</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td>Window Size (16 bits)</td></tr>
<tr><td colspan="8">TCP Checksum (16 bits)</td><td>Urgent Pointer (16 bits)</td></tr>
<tr><td colspan="8">Options</td><td>Padding</td></tr>
<tr><td colspan="9">Data....</td></tr>
</table>



Bit 0 Bit 31

<table>
<tr><td>Version (4 bits)</td><td>Header Length (4 bits)</td><td>QoS (8 bits)</td><td colspan="2">Total Length (16 bits)</td></tr>
<tr><td colspan="3">Fragment identification (16 bits)</td><td>Flags (DF, MF)</td><td>Fragment Offset (13 bits)</td></tr>
<tr><td colspan="2">Time to Live (8 bits)</td><td>Protocol (8 bits)
1=ICMP, 6=TCP, 17=UDP, 50=ESP, ...</td><td colspan="2">Header Checksum (16 bits)</td></tr>
<tr><td colspan="5">Source IP Address (32 bits)</td></tr>
<tr><td colspan="5">Destination IP Address (32 bits)</td></tr>
<tr><td colspan="4">Options (if any)</td><td>Padding</td></tr>
<tr><td colspan="5">Data....</td></tr>
</table>