Solutions: Final Exam

Network Security Autumn 2019

6 February 2020

Surname, Given Names (*e.g.*, Turing, Alan Mathison): _____

Student Identification Number (*e.g.*, 15-123-456): _____

Student Signature: _____

Rules and guidelines:

- Place your identification card on your desk. An assistant will check your identity during the exam.
- Once the exam starts, make sure you have received **all** pages of the exam. The exam should have **22 pages total**, including a page for extra space. **Do not** separate the exam sheets.
- Do not forget to fill in your **name, student identification number and signature** on this page.
- You **must** answer questions using **black or blue ink**. Illegible answers may not get any credit.
- The use of notes, textbooks or other written materials is **not** allowed. You are allowed to use a **scientific calculator** during the exam. Any other device that provides communication or document storage capabilities is **not** allowed (this includes smart watches).
- You have **120 minutes** to complete this exam. The exam has **100 points**.
- You should write answers that are **clear and concise**. Generally, you do not need to completely fill the space provided for solutions.
- You are **not** required to score all points to get the maximum grade.
- When answering questions, always **explain your reasoning**. If a question asks, for instance, whether A is more secure than B, a plain "yes" or "no" answer will not be awarded any points.
- For questions during the exam, **raise your hand** and an assistant will come to answer your question.
- If you need extra space to answer a question, use the page provided at the back of the exam.
- Please **hand in all exam sheets**: if any sheet is missing, the examination will be marked with grade 1.0 and counts as failed.

Question:	1	2	3	4	5	6	7	8	9	Total
Points:	16	3	7	17	11	14	11	13	8	100
Score:										

1. Securing SMTP (16 points)

You are hired as a system administrator by a new tech startup that plans to provide a mail service. Your first order of business is to set up a Simple Mail Transfer Protocol (SMTP) server that allows your users to submit new messages.

SMTP is a simple, text based, protocol. A transcript of a user submitting an email to an SMTP server is given below.

```
1  S: 220 smtp.server.com Ready      // The server indicates that it is
                                        ready to receive a message
2  C: EHLO client.example.com        // The client identifies itself and
                                        requests the server's abilities
3  S: 250 smtp.server.com
4  S: 250 STARTTLS                   // The server advertises
                                        that it supports STARTTLS (see below)
5  S: 250 AUTH PLAIN                 // The server advertises
                                        that it supports user authentication
6  C: AUTH PLAIN                     // The client requests to authenticate itself
7  S: 334                            // The server accepts the request
8  C: dGVzdAB0ZXN0ADEyMzQ=           // The client sends the base64-encoded
                                        username and password
9  S: 235 2.7.0 Auth. success        // The server accepts the credentials
10 C: MAIL FROM:bob@example.com      // The client starts sending the mail
11 C: .... [truncated] ...
```

Two mechanisms can be used to secure an SMTP session. These two mechanism are usually referred to as `Implicit TLS` and `STARTTLS` and work as follows:

- `Implicit TLS` works similar to HTTPS: the SMTP server listens on a separate, TLS-enabled, port and the entire SMTP session is wrapped in TLS.
- `STARTTLS` reuses the standard SMTP port. It allows servers to advertise that they are TLS capable as shown in the transcript above. When communicating with a TLS-capable server, a TLS-capable client can issue the `STARTTLS` command after the client EHLO (Extended Hello). Once the `STARTTLS` command is issued, the client and server will initiate a TLS session which will be used for the remainder of the SMTP exchange.

We will now compare the security of `Implicit TLS` and `STARTTLS`. You may assume that there are no known vulnerabilities in the TLS protocol and implementation.

(a) (4 points) Eve is a passive network level attacker. That is, she can only eavesdrop on the communication between the SMTP client and server.

i. (2 points) Can Eve compromise the confidentiality of an SMTP connection secured by `Implicit TLS`? If yes, how? If no, why not?

Solution: No. All communication (besides the TLS handshake) is protected by TLS.

Grading scheme:

- +0.5 for no
- +1.5 for correct reasoning

ii. (2 points) Can Eve compromise the confidentiality of an SMTP connection when both client and server are configured to negotiate `STARTTLS`? If yes, how? If no, why not?

Solution: No. She cannot prevent the client or server from sending the STARTTLS message.

Grading scheme:

- +0.5 for no
- +1.5 for "First messages are visible, later ones not" or "Eve cannot drop the STARTTLS message"
- -1.0 for incorrect statements about authentication
- -0.5 for suggesting all data is encrypted.

(b) (6 points) Mallory is an active network level attacker. She has all the capabilities of Eve, but can also drop, modify, reroute and inject packets.

i. (3 points) Can Mallory compromise the confidentiality of an SMTP connection secured by `Implicit TLS`? If yes, how? If no, why not?

Solution: No. Because the entire SMTP connection is wrapped in TLS, the connection inherits all the standard security properties provided by TLS.

Grading scheme:

- +0.5 for no
- +2.5 for reasoning
- -1.0 for claiming there is no authentication (unless proper attack explained); or for just stating the server is authenticated without explaining what this brings.

OR

- =3 for "yes" plus explanation of how an attack against TLS could be mounted

ii. (3 points) Can Mallory compromise the confidentiality of an SMTP connection when both client and server are configured to negotiate `STARTTLS`? If yes, how? If no, why not?

Solution: Yes. She can drop the STARTTLS message from the server. This will cause the client to think the server is not TLS capable, after which it might authenticate or send the message over the unencrypted connection.

Grading scheme:

- +0.5 for yes
- +2.5 for dropping STARTTLS advertisement message from server
- +1.5 for modifying handshake bot not specifying how
- -1.0 for dropping STARTTLS message from client. Because SMTP is a client driven protocol, the client is unlikely to proceed if it initiated a TLS handshake that failed.

- -1.0 for dropping AUTH PLAIN from the client
- -0.5 for not specifying which STARTTLS message is dropped

(c) (6 points) Because the `STARTTLS` command was not part of the original SMTP specification, old clients might not support it. To prevent clients from sending sensitive data over a clear-text channel, SMTP servers supporting `STARTTLS` can enforce the use of TLS by issuing the error "`530 5.7.0 Must issue a STARTTLS command first`" and then closing the connection when the client does not try to negotiate TLS.

i. (3 points) Is this a foolproof mechanism against passive attackers like Eve? Why (not)? Remember that not all SMTP clients will correctly implement the specification.

Solution: No, this is not foolproof. Some SMTP clients might send all commands, including the authentication information, in one go, without waiting for the server response. Hence, the user's credentials and email content might be transmitted over the internet in clear text.
Alternative reasoning: The error comes as reply to the first command from the client. If this command already contains sensitive data Eve could access that. (For example if the client doesn't try to authenticate but directly issues an "MAIL FROM:bob@example.com" command the email address would be leaked to Eve).

Grading scheme:

- +0.5 for no
- +2.5 for reasoning

ii. (3 points) What would happen if `Implicit TLS` is used? Does the situation change (from a security perspective)? If so, is it improved or worsened? Why?

Solution:
This question can be interpreted in two ways. The first way (and as it was intended) is: "What would happen if the server wants to enforce the use of TLS by only supporting Implicit TLS, and not plain SMTP". The second way is: "What would happen if the server tries to enforce STARTTLS when Implicit TLS is in use. Full points can be achieved for both interpretations. However, it must be clear which interpretation is used. Vague answers do not get points.

Solution and grading scheme for the first case
Implicit TLS significantly improves the situation. Clients that do not support TLS will not be able to open a TCP session to the server (as it is listening on a different port). Therefore, they will also not be able to transmit any (sensitive) information to the server. Key here is that it is made clear that TLS unaware clients will not be able to send data to the server because it is listening on a different port. That is, when the application requests a socket for the plain SMTP port from the OS, the OS will return an error. Claiming that it cannot communicate because it needs to do a TLS handshake is not correct: it is perfectly possible to open a connection to a TLS enabled port and send clear text messages on it. Although the remote endpoint will most likely close the connection as soon as non-TLS content is detected, the data sent by the client will have already traveled over the network.

- +0.5 for improved
- +2.5 for fully correct reasoning
- +1.0 for stating that the client cannot send clear text messages because the port

is TLS enabled

Solution and grading scheme for the second case
In this case, it is correct to say that the situation is improved, because as the entire connection is already tunneled in TLS, no data can be leaked anyway.

2. TLS 1.3 (3 points)

TLS 1.3 does not provide replay protection for 0-RTT data.

(a) (3 points) Which of the following mechanisms are sufficient methods to mitigate replay attacks when using TLS 1.3? You may assume that you only have one server.

true ☐ false ☒ Only allowing the use of cipher suites which offer perfect forward secrecy (PFS).

Solution: No, the cipher suite has no influence on the feasibility of replay attacks.

true ☐ false ☒ Verifying that the timestamp in the `ClientHello` message lays less than one RTT in the past.

Solution: No, also also within a small time window a message can be replayed.

true ☐ false ☒ Verifying that the timestamp in the `ClientHello` message lays less than two RTTs in the past.

Solution: No, also also within a small time window a message can be replayed.

true ☒ false ☐ Keeping track of the nonce value in the `ClientHello`, and ensuring that each nonce is used only once.

Solution: Yes, this way only TCP session can be opened per ClientHello sent by the client.

true ☒ false ☐ Serving a fully static website.

Solution: Yes, this effectively means that all requests are idempotent.

true ☒ false ☐ Disabling 0-RTT on the server.

Solution: Yes, this way the server will ignore all 0-RTT data.

3. Certificates and Trust (7 points)

(a) (3 points) Online Certificate Status Protocol (OCSP) stapling allows web servers to pre-fetch OCSP responses and attach ("staple") them to their HTTPS responses. Name two major issues with standard OCSP that are resolved by OCSP stapling.

Solution:

- OCSP-stapling eliminates the additional round trip required by the client to obtain the OCSP status from the OCSP server.
- OCSP-stapling eliminates the leakage of the user's browsing behaviour to the OCSP server.
- OCSP-stapling increases availability as the OSCP server might not be available when the client requests the website.

(b) (4 points) Explain the main difference between the trust model used in DNSSEC and that (typically) used by HTTPS and give a disadvantage of each approach.

Solution:

- DNSSEC uses a single root of trust (i.e. it uses the "monopoly" model)
- HTTPS uses many different roots of trust (i.e. it uses the "oligarchy" model)
- The main disadvantage of the DNSSEC / monopoly model is that the single root of trust if effectively a kill switch.
- The main disadvantage of the HTTPS / oligarchy model is that because of it has many roots of trust it is much more likely that one of them will be compromised.

4. Probabilistic Counting (17 points)

As part of the network monitoring in the core of a large backbone ISP, you would like to estimate the number of flows on all the routers. Due to the limited storage capacity, you decide to use *probabilistic counting* to obtain an estimate:

Each router hashes the flow tuple of each packet, interpretes the result as a value in the interval $[0, 1)$, and keeps track of the smallest $k = 16$ values (you can assume for this problem that the number of flows, n, is much larger than k).

(a) (2 points) What properties (in addition to being efficiently computable) does a hash function need to have in order to be usable for probabilistic counting?

Solution:

- 1 point each for the following (or similar):
  - The result of the hash function needs to be unpredictable for an adversary.
  - The hash function needs to output uniformly distributed values in the interval $[0, 1)$.
- 0.5 points for
  - High entropy
  - Cryptographic hash function
  - Memory-efficient storage
- No points for
  - Low collision probability
  - Collision resistant
- Remove 0.5 points if too many wrong properties mentioned.

(b) (2 points) Assume that you have chosen an appropriate hash function, what is the expected value of the kth smallest hash value x_k as a function of the number of flows n? (You can assume $n \gg 1$.)

Solution: The expectation value of x_k is $\frac{k}{n+1} \approx \frac{k}{n}$.

Grading scheme:

- 0.5 points for $1/n$ or $1/(n+1)$ or k/x_k
- 1 point for reverse equation, i.e., $n = k/x_k(-1)$
- 1 point for $(k/n) + / - 1$, $(k-1)/n$

(c) (2 points) As a hash function, your boss suggests to use the efficient and widely implemented MD5 algorithm. Why is this a bad idea? What could an attacker do to compromise the system?

Solution: The hash function is widely known. An attacker can craft traffic (e.g., by modifying source and destination ports) that produces specific hash values and thus influences the result.

The shortcomings of MD5 as a cryptographic hash function are irrelevant for this application as the result of the hash function is never communicated.

Grading scheme:

- 1 point each for shortcoming and attacker actions.
- Well-argued issues with collisions get 0.5-1 points.
- 0.5 points for "MD5 is broken" instead of "MD5 is not secret".

(d) (3 points) Can an adversary cause a significant *underestimation* of the number of flows (assuming he only contributes a small share to the total number of flows)? If yes, how can he achieve this? If not, why is it not possible?

Solution: No. Even if the attacker succeeds in having all their flows being hashed to large values $\sim$1, the estimation would simply correspond to the estimation without their flows (he cannot influence the hash values of other flows). As we assumed that he only contributes a small share of flows, this does not significantly alter the expected result.

Grading scheme:

- -0.5 points if not mentioned or implied that attacker cannot influence honest flow's lowest hash values.
- -0.5 points if not mentioned or implied that attacker can only hide his own (insignificant) traffic.
- 1–2 points if no with incomplete or partially incorrect explanation
- 0.5 points for no without explanation
- 0 point if no with incorrect reasoning
- 1 point if yes and citing hash collisions or large hash values
- 2 points if arguing correctly for a network-level attacker

(e) (3 points) Can an adversary cause a significant *overestimation* of the number of flows (assuming he only contributes a small share to the total number of flows)? If yes, how can he achieve this? If not, why is it not possible?

Solution: Yes, by crafting 16 flows whose flow tuples map to very small hash values and thus determine the overall estimate.

Grading scheme:

- -1.5 point if smallest hash values are not mentioned.
- -.5 points if only one flow considered
- -1 point if not explained how hash values are obtained.
- -1 point for miscellaneous misunderstandings

(f) (2 points) If you absolutely have to use MD5, how could you modify the computation such that it can be used for probabilistic counting?

Solution: You can salt the flow tuple, i.e., prepend it with a secret value that is not shared outside of your ISP. Another possibility is to XOR the flow tuple with a secret value. In principle, you could even use a different secret value for each router. Note that you *should not* use a different secret value (nonce) for each packet as this would completely mess up the probabilistic counting.

Grading scheme:

- 2 points also for using HMAC
- 1.5 points for using "key" without further explanation.
- 1.5 points for applying hash a random number of times (high overhead and little entropy).
- 1 point if only "randomness" mentioned but not indicated how it would be used.
- 1 point for using something out of the source's control.
- -0.5 points for confusing "nonce" with "key" or "salt"; -1 point if scheme actually uses nonces (would not work any more).

(g) (3 points) Remember that the variance of the kth smallest value is approximately $\frac{k}{n^2}$ (for large n). What can you conclude about the precision of the estimation when using x_{16} compared to x_1? Which result is more precise and by which factor?

Solution: The relative standard deviation of the kth smallest value is $\sim\frac{1}{\sqrt{k}}$, which translates to the relative standard deviation of the estimation of the number of flows. Using x_{16} instead of x_1 thus leads to an estimate which is about 4 times more precise.

Grading scheme:

- 0.5 points for remembering/concluding that k=16 is more precise by incorrectly noting that x_{16} has lower variance
- 1 point for remembering that x_{16} is more precise by a factor of 4 (without explanation).
- 1 point for mentioning that variance of x_k is higher and reaching conclusion that x_1 is more precise.
- +0.5 points for looking at standard deviation
- 3 points for correct reasoning including relative standard deviation
- -0.5 points if obtaining factor k instead of $\sqrt{(}k)$
- 0 points for completely wrong argumentation

5. VPNs and Anonymous Communication (11 points)

In 2013, Eldo Kim, a student at Harvard University, sent a bomb threat during the exam session in order to delay one of his exams. He was connected to the university's wireless network; wanting to remain anonymous, he used Tor to send the threat email.

We will use this incident to illustrate important concepts of anonymous communication.

(a) (3 points) Even though he was using Tor correctly and sent the email via an anonymous email server, Eldo Kim was very quickly identified as the most likely culprit by the police. He was arrested and confessed to sending the threat email (even though the police did not have hard evidence).

Explain the concept of an *anonymity set* and use this to explain why the student was immediately identified as the culprit.

Solution: An anonymity set describes the set of users from which a particular user is indistinguishable. The larger this set is, the more anonymous the user is.In this case, the anonymity set was very small: There were very few (or even only a single) Tor users inside the university network that were online around the time the email was sent. The police thus only had to investigate these few people.

Grading scheme:
1 point each for

- Definition/explanation of (sender) anonymity set [Only 0.5 points for imprecise definitions of anonymity set (e.g., the set of people using TOR, could have done something, potential senders, etc.)]
- Statement/implication that larger set means better anonymity
- Small anonymity set in this case with explanation [Only 0.5 points if no reason is given why AS was small in this case]

(b) (2 points) Explain how it is possible for the police and university to identify Tor users based on their network traffic.

Solution: Without additional mechanisms, a client needs to contact one entry guard to start setting up a Tor connection. The set of entry guards is publicly available. Therefore, the university only has to check its logs for connections towards these well-known IP addresses. In addition, the initial communication with the entry guard is not encrypted and could be identified as Tor traffic through packet inspection.

Grading scheme:

- 2 points for either of these explanations.
- -0.5 point for writing "exit node" instead of "entry guard"
- -0.5 point for not mentioning/implying that entry guards are publicly known
- Only 0.5 points for unspecific statements about "analyzing traffic" or 1 point for "packet inspection"
- Up to 1 point for reasonable techniques for *deanonymizing* Tor users.

For the remainder of this question, let us assume that the student wanted to remain anonymous for less illegal purposes, e.g., for anonymously reporting illegal activity, etc.

(c) (3 points) Name and explain at least two ways in which a Tor user intending to send an anonymous email can achieve better anonymity compared to Eldo Kim.

Solution: There are two things that lead to the small anonymity set in the case of Eldo Kim: temporal and spatial relationship with the message, i.e., investigators will first look at Tor users that were online around the time when the email was sent and from locations that are related to the university. This leads to two simple mechanisms to increase anonymity:

1. Remain online in the Tor network at all times. While you may still be considered a suspect in this case, you are much less suspicious and could argue that you generally use Tor to stay anonymous.
2. Access Tor from a location that is unrelated to the university. The number of Tor users at any particular time in Boston is much larger as that of Tor users in the Harvard network.

Other correct solutions:

- Obfuscate the use of Tor through pluggable transports or bridges.
- Use a proxy or VPN through which to connect to Tor (if argued properly).
- 1 point for sending email with time delay; the server still knows and can log when the sender connected

Incorrect answers:

- Hidden services; these only protect the anonymity of the services but not that of the sender.
- PGP
- Cover traffic (0.5 points)
- Avoid fingerprinting
- Mix cascades (0.5 points)

Grading scheme:

- 1.5 points each
- 0.5 points w/o explanation.
- Remove 0.5–1 points if two solutions are too similar.

(d) (3 points) Using Tor can itself make someone look suspicious. One of your friends thus suggests to simply use a VPN instead. How do you respond, i.e., how do the anonymity guarantees of VPNs compare to Tor? (Name at least two differences.)

Solution: Possible answers (1.5 points each):

- The VPN provider can observe and log all connections and is thus a single point of failure.
- The VPN provider has your credentials/payment info/identity (1 point for only "client authentication").
- Assuming an honest provider, the anonymity set contains all users of that VPN, which may be fewer than Tor users.
- Traffic analysis by a network attacker is easier as a single point of presence is sufficient for timing-based traffic analysis.

- Tor browser bundle solving fingerprinting.
- VPN usage may be considered less suspicious (1 point)

Grading scheme:

- -0.5-1 points if two differences are too similar
- +0.5 points for detailed argument

6. DNS Security (14 points)

(a) (6 points) For every attack on DNS below, state whether the deployment of DNSSEC would make the attack less effective, equally effective or more effective. Moreover, justify your answer. Only answers that are correctly justified will be awarded points.

i. (1½ points) Reflection/Amplification Attack

> **Solution:** More effective, as responses get longer due to signatures. [1.5]

ii. (1½ points) Botnet control over DNS

> **Solution:** Equally effective (no change), since DNSSec does not affect domain registration. [1.5]
> More effective, because nowadays, botnets are often taken down by law-enforcement agencies spoofing DNS records to bots, which would not be possible anymore with DNSSEC. [1.5] (That said, law enforcement could probably easily take control of the authoritative DNS servers and simply change the respective entries instead of spoofing them.)
> 0 points for:
>
> - Botnets would not be possible because malicious actors would not be able to get their DNS records signed by the authoritative name server. However, DNSSEC contains no such 'benevolence' checks.
> - Botnets would work less efficiently because the bots would need to query large signature records and expensively verify the other records. However, bots do not *need* to do either of these things.

iii. (1½ points) DNS Spoofing

> **Solution:** Less effective, as DNS responses are authenticated. Preventing DNS spoofing is the whole point of DNSSEC. [1.5]

iv. (1½ points) Stefan Frei's TTL Attack
Background: *Before your lecturer Dr. Stefan Frei had to transfer the domains schweiz.ch, suisse.ch and swizzera.ch to the Swiss state, he set the TTL on the old DNS records, which were still pointing to his servers, to 136 years. This information was cached by DNS servers around the world, some of which were still serving the information for weeks after the transfer.*

> **Solution:** Equally effective, because data is authenticated, albeit obsolete. There is nothing like a revocation mechanism in DNSSEC. [1.5]
> One answer was: More effective, since there are more obsolete records out there that would have to be revoked. We gave [1] points for this. It is true that a higher number of records has to be revoked, but the main negative impact of this attack is directing client traffic to the wrong servers. This impact is not changed by DNSSEC.
> 0 points for the frequent answer that the authentication inherent in DNSSEC would enable a DNS server to detect that a cached record is obsolete. The problem here is that Stefan Frei's hypothetical DNSSEC records were authentic at the time of cache insertion; the DNSSEC validation is only done when looking up a record for insertion into the cache (for scalability reasons). As long as the TTL of the cached entry has not run out, the DNS server would not look up the cached record again and thus would not discover that the record has changed.

(b) (3 points) While DNSSEC provides authentication, the communication with a DNSSEC-enabled server is not encrypted and thus not confidential. This lack of confidentiality leads to a range of privacy issues, especially Internet usage monitoring of customers by their ISP.

Given that public keys (DNSKEY records) are already present on a DNS server for verifying record signatures, a privacy-sensitive colleague of yours suggests to use these public keys also to achieve confidentiality.

Your colleague proposes to use TLS with the DNSKEY as the server's public key. After key agreement, the user's DNS request would then be communicated over the TLS-secured connection.

Would this DNSSEC-assisted DNS-over-TLS be a good protocol to enhance DNS privacy? Explain why/why not.

Solution:

- No other DNS server than the authoritative name server could serve DNSSEC records of a zone, because the responding DNS server would need to have access to the private key corresponding to the signing key. Even for authoritative name servers, keeping signing keys online is a security risk. [1.5]
- Since DNSSec validation is mostly done on the recursive resolver of the ISP (and not on the client), this protocol would only secure the communication between the recursive resolver and the authoritative name servers, which does not address the privacy concerns. [1.5]
- TLS would put additional load on DNS servers, harming the scalability of DNS. [1]

(c) (5 points) The following two questions relate to DNS root name servers, the basic footing of the global DNS infrastructure.

i. (3 points) Name three reasons why root name servers are replicated across the globe.

> **Solution:** Load balancing, latency minimization, failure tolerance, reduced attack surface by short paths to name servers, political reasons [1 each, but max. 3]

ii. (2 points) A recent proposal suggests to eliminate DNS root servers altogether and distribute the data that would be on those root servers (i.e., the root zone file) to recursive resolvers.
This proposal has a security benefit for DNS (not DNSSEC). What is the security benefit of this proposal for DNS?

> **Solution:** DNS queries have to traverse less of the network, thus the attack surface for spoofing is reduced. [2]

7. Intrusion Detection (11 points)

You are the network engineer of an important stock exchange receiving transactions over the Internet. Since the users of the stock exchange engage in high-frequency trading with large volumes, low latency and high availability are important requirements. However, this stock exchange is also an attractive attack target, thus skilled cybercriminals are constantly attempting to break in and shut down the stock exchange.

(a) (9 points) Given this threat scenario, you have to evaluate different IDS architectures. Name one advantage and disadvantage (each with justification) of each architecture presented below.

i. (3 points) Single signature-based IDS

Solution: **(+):** Low latency (important for HFT), low false positives (important because investigation would take too long for HFT) [1.5]
(-): Reactive, thus attack must be known beforehand, which is unrealistic given dedicated cybercriminals [1.5]

ii. (3 points) Single sandbox-based IDS

Solution: **(+):** Can detect unknown threats (important given dedicated cybercriminals) [1.5]
(-): High latency (not suitable for HFT), execution in sandbox is prone to DDoS attacks [1.5]

iii. (3 points) Conditional Sequence of IDSs: Signature-based IDS that forwards suspicious traffic to a sandbox-based IDS and bypasses the sandbox-based IDS for unsuspicious traffic

Solution: **(+):** Can reduce false positives while rarely triggering expensive sandbox execution (important for HFT) [1.5] Latency is typically as good as signature-based solution ([1], because this is not really a unique advantage of this construction)
(-): No additional security from sandbox, as unknown threats are still not detected by signature-based IDS (and hence bypass the sandbox IDS that would maybe detect them) [1.5]

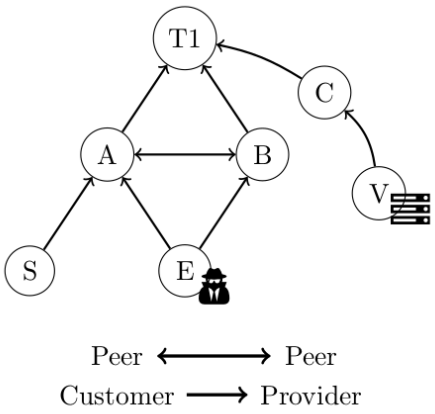
(b) (2 points) While discussing the IDS architecture, you learn that most transactions for the stock exchange come from a handful of trusted stock exchanges and banks.

What kind of IDS solution does this fact enable?

Solution: Sources can be authenticated [1] to bypass the IDS [1]
We can use a firewall/a signature-based IDS/a whitelist that *only has to check the source IP* [1] (Missing point: The origin must be authenticated, or else it could simply be spoofed!)
0 points for: Simply use a signature-based IDS (not further specified) for the traffic from trusted sources. We must also authenticate the source of the packet, otherwise it could simply be spoofed. Once the source is authenticated, we do not need an IDS (which is more than a simple source whitelist, see above) anymore, because the source is trusted.

8. BGP Hijacking (13 points)

Consider the inter-domain topology in the Figure. AS E is compromised, and operated by a malicious entity. Assume the standard BGP forwarding rules apply. A refresher of BGP routing is provided in the following.



BGP routing reference
BGP route propagation and forwarding follows business relationships:

1. the announcement for a prefix coming from a customer is propagated to all customers, peers and providers;
2. the announcement for a prefix coming from a provider or a peer is only propagated to customers.

Additionally, if the announcement for two paths to the same prefix are received, they are ranked by their preference. The announcement for the path with the highest preference is then propagated. Preferences are computed in the following way:

1. forwarding to customers is preferred to forwarding to peers. Forwarding to peers is preferred to forwarding to providers (customers > peers > providers);
2. to break ties, the number of hops in the path is used. Shortest paths have higher preference;
3. further ties are broken using local preferences, that depend on the AS.

(a) (2 points) **Attack A:** The compromised AS E wants to attract all traffic destined to the victim AS V, and originating from ASes A, B and S. The address block announced by AS V is `10.0.0.0/16`. What is the simplest announcement that AS E can make to achieve this goal?

Solution: Full points were awarded for:

- Announce the prefix 10.0.0.0/16 as originating from AS E.
- Announce a set of subprefixes that covers the full 10.0.0.0/16 address space.

1.0 points deducted if the announced subprefix does not cover the address space. The question explicitly asks to attract all traffic. Points were also removed for incorrect supbprefixes.

(b) (3 points) **Attack B:** What can you say about the forwarding policy of the Tier-1 AS $T1$? The malicious AS E now wants to attract all traffic destined to a specific server in AS V. The address of the server is `10.0.1.1`. How can AS E modify the announcement from **Attack A** to make sure to attract all the traffic *originating from* AS $T1$ and *destined to* the server?

Solution: AS T1 policies: if the announced prefix in Attack A was 10.0.0.0/16, then the forwarding policy of AS T1 will depend only on local policies (the path length is the same). If Attack A announced a subprefix, then AS T1 will prefer to route through A or B (1 point).

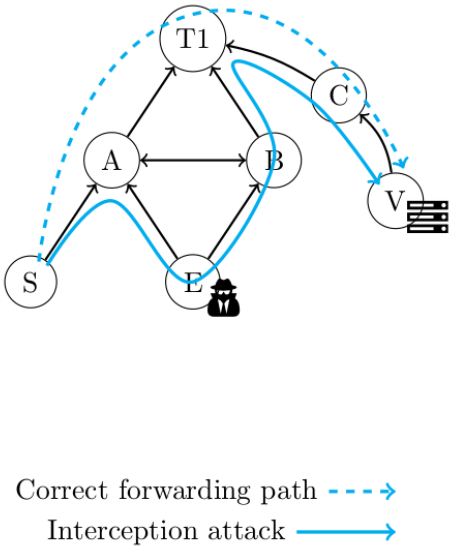
Attack B: if the announced prefix in Attack A was 10.0.0.0/16, then Attack B requires to announce a sub-prefix that contains the server's IP. If Attack A announced a subprefix, the the attack already achieves the goal (2 points).

Points were deducted for wrong subprefix splits.

(c) (3 points) Is origin authentication sufficient to prevent **Attack A**? What about **Attack B**?

> **Solution:** Attack A: In general, no. AS E can still capture the announcement from V, append itself and propagate the announcement. Since it is a customer, its providers will still prefer this path (1.5 points). In the particular case of attack A, the attack can be prevented (1 point).
> Attack B: It is always sufficient, as AS E cannot generate an announcement with the subprefix that will pass authentication.

This particular type of hijacking is called a *redirection* attack: the hijacked traffic terminates in AS E, and never reaches the correct destination (AS V). The malicious operator, however, now wants to perform an *interception* attack on AS S. The goal is to capture **all the traffic originating from** AS S to the address block `10.0.0.0/16` in AS V, while still being able to forward this traffic through AS B, as in the following figure:



(d) (3 points) Name one use case for redirection attacks, and one for interception attacks.

> **Solution:** Redirection attacks: DNS spoofing, phishing, impersonation, DoS; (1.5 points for one)
> Interception attacks: Correlation attack on anonymous networks, man in the middle against CAs, traffic analysis (1.5 points for one)
> For interception attacks, 0.5 points were awarded for answers regarding "spying traffic", "inspecting" and "sniffing unencrypted traffic". Although correct in principle, these answers are generic and do not reach the required level of precision.
> No points were awarded for attacks on economy: although one could see the point in theory, such an attack requires the malicious entity to redirect large amounts of traffic. To do so, the attacker needs to build and provision a network as an ISP. At this point, the attacker is just doing business as a normal ISP! Even more, if the attacker is in a similar situation as in the example topology (i.e., a leaf AS), it has to pay for both incoming and outgoing traffic, making this a very bad business model!

(e) (2 points) **Attack C:** AS E tries to achieve the interception attack (as in the Figure) by announcing the hijacked prefix only to AS A. AS E is hoping that AS B will use the correct announcement originating in AS V, and received from AS $T1$. Will this work? Why/why not?

Solution: Full points for:

- No. B will receive the announcement from A, and since peer is preferred over provider (T1), the announcement from T1 will discarded.
- Poisoning B and T1

1 point for:

- Poisoning only B
- B sends traffic to E
- Update sequence A→T1→B instead of A→B

9. SCION and Hijacking (8 points)

In the following, consider the SCION Internet architecture *without extensions*.

(a) (2 points) SCION addresses differ from IP addresses. Specifically, which are the parts that compose a SCION address?

Solution: The address is composed of ISD number, AS number and local address. This means that the address is always tied to a specific AS, unlike in IP.

1 point awarded if only 2 of the above were specified.

(b) (2 points) How is path discovery secured in SCION?

Solution: Beacons contain AS signatures on hop-fields. By verifying the signatures, an AS is able to verify that the path segment is authentic.

(c) (2 points) How does SCION prevent hijacking attacks from an off-path attacker? (hint: use your answers from the previous 2 questions)

Solution: Full points: Since addresses are tied to AS numbers, and the hop fields are authenticated with signatures, an off path attacker has no way to tamper with the routing process. In addition, forwarding is only based on the path in the packet header which cannot be influenced by off-path attackers.

1.5 points: Data plane and control plane are separated, and, since beaconing is secured, off path attackers cannot influence the routing behaviour.

1.0 points: Beaconing is authenticated and/or adversaries cannot forge signatures.

(d) (2 points) On-path attackers can always intercept or drop traffic. What other mechanism provided in SCION can mitigate the effect of an on-path attacker?

Solution: On-path attackers can always intercept or drop traffic. What other mechanism provided in SCION can mitigate the effect of an on-path attacker?