Final Exam

Network Security Autumn 2016

04 February 2017

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

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

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 **15 pages total**, including pages for extra space (see below).
- Do not forget to fill in your **name and student identification number** on this page.
- **Do not** separate the exam sheets.
- You have **90 minutes** to complete this exam.
- You **must** answer questions using **black or blue ink**.
- If you have a question 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 pages provided for you at the back of the exam.
- 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).
- At the end of the exam, please **remain seated** while we collect the exams. You may hand in your exam before the end, except in the last 10 minutes 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.
- You are **not** required to score all points to get the maximum grade.
- As a general guideline, one point should correspond to one minute. Thus you should write answers that are **clear and concise**. Generally, you do not need to fill the provided space for solutions.
- 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.

Question:	1	2	3	4	5	6	Total
Points:	16	15	14	15	19	11	90
Score:							

1. DNS Security (16 points)

Peer-to-peer DNS. Today's DNS is highly centralized and hierarchical. Alternative designs for name-to-address resolution have been proposed which aim for full decentralization instead. We consider one such design here, which we call **P2P-DNS**: in P2P-DNS, all authoritative name servers for all domains participate as equal *peers*. These N peers (assume that $N = 2^l$) are each assigned a unique l-bit identifier. Each peer stores the records (which are exactly the same as DNS records) for all domain names whose hashes start with the same l bits as the peer's identifier. If for instance $l = 4$ (in practice it would be much larger) and `hash`("`www.example.com`") = `10110110`..., then the peer with identifier `1011` would store the records for `www.example.com`.

A consequence of this design is that peers store records for which they are not authoritative: to keep them up to date, each peer regularly receives updates for all stored records from the corresponding authoritative name servers (i.e., from the domain owners). Additionally, popular records are automatically replicated across multiple peers, and if a peer becomes unavailable, other peers can detect this and take over its duties.

The P2P-DNS peers run a distributed protocol among them which allows routing to any peer in an expected number of hops $\mathcal{O}(\log N)$. A query works as follows: (1) a client sends a (plaintext) query over UDP to any peer; (2) the peers forward the query (based on the routing protocol) until the query reaches a peer that stores the corresponding record; (3) this peer replies via UDP directly to the original client providing the record the client requested.

(a) (7 points) You now have to compare P2P-DNS with traditional DNS.

i. (2 points) What advantage does P2P-DNS offer compared to DNS in terms of *resilience*?

Solution: P2P-DNS has the advantage over DNS that it does not have a centralized root nor centralized TLDs, so there is no clear target for DoS attacks aiming to take down large parts of the naming system **(2 points)**.
Alternative: Records are replicated, and peers can take over the duties of failed peers (just what is stated in the exercise description) **(1 point)**.

ii. (3 points) What advantage and disadvantage does P2P-DNS have in terms of *privacy* compared to DNS?

Solution: Queries might traverse a large number of peers, so more entities will see the requests **(2 point)**; on the other hand, clients can send their queries to different peers every time, so that each peer will only see a fraction of the queries **(1 point)**.
Alternative disadvantage: there is no recursive resolver, so queried servers see the exact source address **(1 point)**; peers can do zone enumeration **(1 point)**.
Alternative advantage: no resolver/root server sees all queries **(1 point)**.

iii. (2 points) What additional *integrity* concern does P2P-DNS cause compared to DNS?

Solution: The main *integrity* concern is that the referral system of DNS breaks down in P2P-DNS, since the peers who are responsible for (i.e., who store) a record are not authoritative for the record, and may thus change it, while the authoritative name server has no control over which entity is responsible for its records **(2 points)**.
Alternatives: the query can be altered on the way by intermediate peers **(1 point)**; intermediate peers can directly send a fake reply to the client **(1 point)**.

(b) (4 points) Consider a variant of P2P-DNS based on DNSSEC rather than plain DNS, which we call **P2P-DNSSEC**. In P2P-DNSSEC, in addition to normal DNS records, the peers store DNSSEC cryptographic records (RRSIG, DS, DNSKEY, etc.) *for the entire DNSSEC hierarchy*, but other than that it works exactly like P2P-DNS. What are the main advantages and disadvantages of P2P-DNSSEC compared to P2P-DNS?

Solution: It has the same advantages that DNSSEC has over DNS, mainly in terms of integrity protection, e.g., no cache poisoning. In particular, this also means that the integrity problem of the previous question is solved, because clients can verify that the records are signed by the correct authority. **(2 points for some details, 1 point for just mentioning authentication/integrity)**. The main disadvantage is in terms of efficiency: since every query can require a large number of hops, performing a properly verified query could take a long time, since unlike for P2P-DNS, a recursive lookup is necessary **(2 points)**.

Alternative disadvantages: it imposes a higher storage burden on peers **(1 point)**; the key management operations introduce additional complexity **(1 point)**.

(c) (3 points) DNS can be leveraged for amplification attacks, a form of DoS attacks. Is P2P-DNS as described above (i.e., using UDP) vulnerable to amplification attacks? What about P2P-DNSSEC?

Solution: P2P-DNS is vulnerable just as DNS: since UDP allows source address spoofing, and since the record are the same as in DNS, amplification attacks through reflection are possible **(1 point)**. P2P-DNSSEC offers no additional protection **(1 point)**, and can actually be worse since the records are typically quite large **(1 point)**. **(Deduct 1 point if it is not mentioned why these systems are vulnerable.)**

Alternative: P2P-DNSSEC might be better in case it relied on TCP, as DNSSEC does **(1 point)**.

(d) (2 points) The design of P2P-DNS offers the opportunity to deviate from the query format of DNS: what would be a clean and simple way to protect P2P-DNS from cache poisoning attacks? (Excluding solutions that rely on authentication—we have P2P-DNSSEC for that!)

Solution: The best way is to have a sufficiently large transaction ID (TXID), which should generated with a proper pseudo-random number generator **(2 points, 1 for alternative solution which is less clean/simple)**.

Alternatives: since the cache is on the client, a cache poisoning attack does not affect multiple clients at once **(1 point)**; multiple requests could be sent to different peers storing the record **(1 point)**.

2. Web-Application Security (15 points)

(a) (6 points) Recall the same origin policy and its exceptions as applied by a web browser. Also recall how these exceptions enable Cross Site Request Forgery (CSRF) attacks. To counter the CSRF attack, one approach is to include a secret validation token in the webpage generated by the server.

A web developer working at Secure Bank (www.securebank.com) decides that he can use timestamps to defend against the CSRF attack. He intends to include a hidden field containing the current time stamp (the time on the server when the page was generated) in every web page served by the server:

```
<input type="hidden" name="auth_token" value="Thu Mar 4 11:00:00 EST 2010"/>
```

Now, whenever any form is submitted to the server, the server code checks that the `auth_token` is included and that its value is within 15 minutes of current time of the server (besides checking the session cookie). If the `auth_token` is missing or its value is more than 15 minutes off, the server rejects the transaction and requests the user to log-in again.

i. (2 points) Is this scheme secure? Explain why or why not?

Solution: This scheme is not secure. The authentication token is predictable by an attacker.

ii. (2 points) Give a technique (different than the one above) to generate authentication tokens to defend against CSRF attacks.

Solution:

- Session independent/dependent nonce
- Keyed hash of session identifier

iii. (2 points) Will the CSRF attack work if the server uses HTTPS? Explain why or why not?

Solution: HTTPS does not prevent CSRF attacks, since they happen on the application layer, whereas SSL is used on the transport layer.

(b) (9 points) The website of Secure Bank contains a search field for transactions. A user can input a transaction ID to get the details about that particular transaction. The transactions are stored in a single table in a SQL database. You discover that the search field is vulnerable to SQL injection and that the developers at Secure Bank left error reporting enabled, i.e., a user can see the errors returned by the SQL database. Assume there exists a second table named "cc_info", containing the credit card information for each customer of Secure Bank and that the website displays all records returned by a query. Describe in detail how you would perform an SQL injection attack to extract credit card information from other customers. Specifically, you should try to answer these questions:

- What type of SQL injection attack would you use?
- You may need to refine your query during the attack, how?
- How would you extract the data?

Solution: Since the credit card information is stored in a different table than the transactions, we have to make use of `UNION` to access its contents. Because the `UNION` operator can only be used if both queries have the exact same structure, we must craft a `SELECT` statement similar to the original query. To do this it is necessary to determine the number of columns in the first query and their data type.

To find the number of columns that are fetched from the transaction table we can use `ORDER BY` clause followed by a number indicating the numeration of databases column selected:

```
1234 ORDER BY 10;
```

Given the column number specified is greater than the number of columns in the SELECT statement, an error will be returned. Otherwise, the results will be sorted by the column mentioned.

The next step is to find out the type of the columns. Assuming there are 4 columns being in fetched from the transaction table, we can try queries of the form

```
1234 UNION SELECT 'A', 'B', 1, 2 FROM cc_info;
```

until no error message is returned and data is listed. Data types are can be split into two groups: numeric values and strings, thus we only have to test 16 different possibilities (for 4 columns).

Finally, to extract the data, we have to find the column name of the credit card numbers. We can either guess it or use some inherent knowledge about the database system. MySQL databases have a table with metadata containing column names contained in tables.

```
1234 UNION SELECT table_name, column_name, 1, 2
FROM information_schema.columns;
```

Assume, we found the column name to be `cc_number`. The credit card numbers can then be extracted with:

```
1234 UNION SELECT cc_number, 'B', 1, 2 FROM cc_info;
```

Grading Notes: *The following things will award points:*

- UNION based SQL injection (1)
- Mentioning correct number of columns and correct data types are needed (1 + 1)
- Describing how to find number of columns and data types (2 + 2)

- Correctly describe how to extract the data (inherent knowledge about MySQL isn't required for full points) (2)
- SQL code is not strictly needed, but then the explanation has to include what the SQL code would say.

3. Botnets and DDoS Attacks (14 points)

(a) (5 points) On October 21, 2016, several waves of DDoS attacks against the DNS provider *Dyn* knocked dozens of popular websites offline, among them Twitter, SoundCloud, Spotify, and Reddit. More precisely, the authoritative name servers of certain domain names were attacked by a large number of *conventional* DNS queries (i.e., following the DNS protocol, no invalid messages, etc) from vulnerable devices ("bots") attempting to saturate the network resources that are close to the locations of the authoritative name servers. These bots are assumed to be hacked customer devices that reside in domestic networks behind correctly operating routers with NAT functionality.

Under this assumption, state whether preventing *source address spoofing* can help mitigate this attack. If so, explain how. If not, explain why not. Mention at least two different arguments.

Solution: In contrast to sophisticated attacks that exploit potential zero-day exploits, in this case only conventional DNS queries were sent; in other words, the large traffic volume was the main cause for that attack to work out.

Preventing source address spoofing **does not help** in this case since [1 point]

- the attack traffic was "legitimate" in the sense that the attack traffic was conventional DNS and its volume was not amplified (a potential amplification can be avoided when source addresses are not spoofed), [2 points]
- a vulnerable device behind a NAT cannot spoof the source address of outgoing packets (and the routers were operating correctly), and [2 points]
- (optional) as the bots were used as proxies, the attacker has no interest in concealing his/her identity at this point.

(b) (3 points) Dyn reported "We observed 10s of millions of discrete IP addresses associated with the Mirai botnet that were part of the attack." This lets us conclude that the attack was not against the regular DNS infrastructure (i.e., not against the recursive resolvers), but directly against the authoritative name servers which were evidently overwhelmed and started dropping (also legitimate) queries. The reason for this conjecture is that the authoritative name servers would otherwise have seen the usual set of recursive resolvers only.

If this assumption is correct, how could the Dyn infrastructure be better protected? (Adding more power, more capacity, more servers, etc is not part of a correct answer).

Solution: Assuming that no source addresses are spoofed, then the authoritative servers could be configured to accept only requests from the set of recursive resolvers (whitelisting). [There are roughly 10K IP addresses that correspond to the collection of visible recursive resolvers, which serve some 95 % of the entire user base on the Internet.] Using two processing queues, one could drop requests not belonging to the recursive resolvers when under attack. [3 points]

(c) (6 points) The impact of the attack with respect to the public highly depends on the caching behavior of the recursive resolvers (which are typically used by the legitimate public).

i. (3 points) Explain in two sentences why caching and the time-to-live (TTL) play an important role here. (Hint: what happens if a cache entry expires? What impact does the TTL have on the effectiveness of the attack?)

Solution: Once the recursive resolvers had their local cache expiry timers expired, they try to fetch fresh entries from the authoritative name servers. These, however, will not be able to respond, which causes an unavailability that is visible to the client. [2 points]
The attack must last at least for the TTL period to ensure that recursive resolvers' cache expires. [1 point]

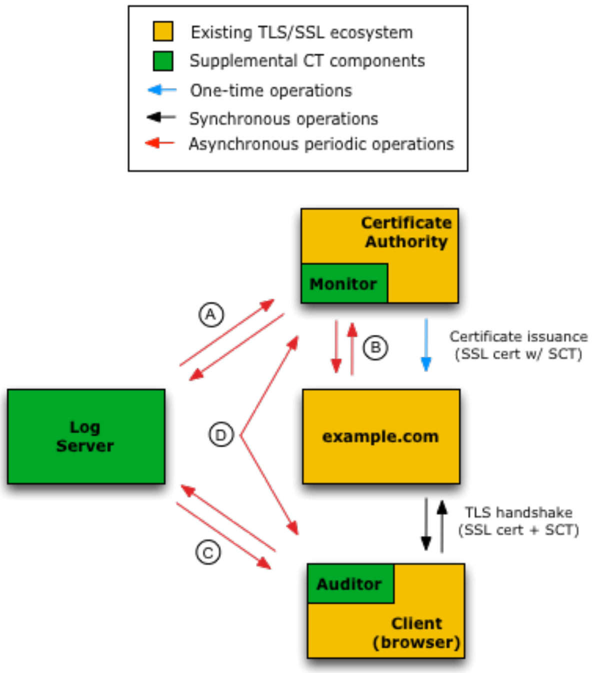
ii. (3 points) Explain how a different TTL and a different caching strategy could mitigate the attack. Think of how the existing hierarchical structure of DNS could be leveraged better?

Solution:

- a change in the expiration time: a longer expiration time would naturally give the distributed recursive resolvers more freedom to defer their queries to the authoritative name servers. [1 points]
- a change in the caching strategy – independent of the TTL: whenever an authoritative name server is not responding within a certain time bound, the recursive resolver could ignore that, possibly assume an attack, and answer with the previously cached entry. [2 points]

4. Public Key Infrastructure (15 points)

Recall that Certificate Transparency (CT) is a mechanism that makes issuance and existence of SSL/TLS certificates public for the examination by domain owners, Certificate Authorities (CA), and domain users. Figure 1 briefly explains how CT works.



🟨 Existing TLS/SSL ecosystem
🟩 Supplemental CT components
← One-time operations
← Synchronous operations
← Asynchronous periodic operations
Certificate Authority
Monitor
A
B
Certificate issuance (SSL cert w/ SCT)
Log Server
D
example.com
TLS handshake (SSL cert + SCT)
C
Auditor
Client (browser)

Figure 1: Certificate transparency. A: Monitors watch logs for suspicious certificates and verify that all logged certificates are visible. B: Certificate owners query monitors to verify that nobody uses illegitimate certificates for their domains. C: Auditors verify that a particular certificate has been logged. D: Monitors and auditors exchange information about logs to help detect malicious logs.

(a) (7 points) Signed Certificate Timestamps (SCT). During the certificate issuance, together with the SSL/TLS certificate, the CA also sends an SCT to the domain owner.

i. (3 points) What is the purpose of SCT? Which entity creates SCT? (Hint: The log server can only periodically update its log.)

Solution: Because the log server can only periodically update its log, a valid certificate submitted by the CA cannot be immediately added to the log. The log server creates SCT as a promise to add the certificate to the log within some time period. [1 point] The domain owner can use SCT to show the validity of the certificate to the user before the certificate is added to the log. [1 point]
The log server creates SCTs. [1 point]

ii. (4 points) What is the problem of issuing SCT if a malicious log server colludes with a malicious CA? How can this be detected?

> **Solution:** A malicious log server can create SCTs for crafted illegitimate certificates, but not add the certificates into the log. [2 points]
> (Open question) For each SCT, the auditor on the client side can check whether the corresponding certificate is added to the log in retrospect. [2 points]

(b) (6 points) A malicious log server can also launch a "branched log" attack (or "split world" attack). In this attack, a malicious log server maintains two versions of the log. For example, the log of version 1 contains a certificate *cert*, but the log of version 2 does not. The log server can then distribute logs of different versions to different requesters (monitor/auditor).

i. (3 points) Describe how a malicious log server can use the branched log attack to trick an auditor on the client side to accept an invalid certificate (e.g., a certificate from another compromised CA) without being detected by monitor on the benign CA.

> **Solution:** Assume the log server adds the invalid certificate into the log of version 1, but not the log of version 2. The log server then replies the log of version 1 to the client and the log of version 2 to the monitor on the benign CA.

ii. (2 points) Describe how step D in Figure 1 prevents the "branched log" attack.

> **Solution:** By exchanging information about the log, the auditor and monitor can check the log they retrieved is of the same version, and prevents the above attacks.

iii. (1 point) Recall that the log server maintains the log as a Merkle hash tree. What information about the log does the monitor and auditor exchange?

> **Solution:** Root hash value/Root node.

(c) (2 points) What is the privacy problem of step C in Figure 1?

> **Solution:** The log server will know all domains (if the domain implements CT) a client wants to access.

5. TLS (19 points)

(a) (5 points) We consider an enterprise network that connects to the Internet via a gateway G, as shown in Figure 2. The gateway G is deployed by the administrator of the enterprise network who wants to stealthily intercept TLS communications between an internal host (e.g., H_1 or H_2) and an external server S. G can decrypt TLS traffic and re-encrypt it after inspection. Essentially what G does is create a fake certificate of the server S and perform a man-in-the-middle (MitM) attack between the host and the server.

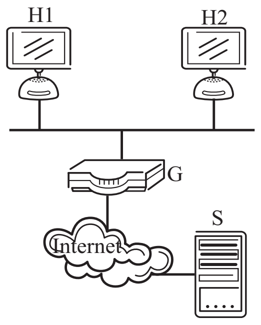


Figure 2: Enterprise network.

i. (2 points) Now consider H_1 initiates a TLS session to S. What modification has to be done on H_1 so that the user will not realize that an invalid certificate was created by G (i.e., the browser will not pop out a warning message)?

> **Solution:** The company needs to install an additional root key.

ii. (1 point) Can the server S detect such a MitM attack? Briefly justify your answer.

> **Solution:** No, because the server commonly does not verify any clients' certificates.

iii. (2 points) You are a student intern at this company, and you suspect that the company is intercepting your TLS traffic. What can you do to detect such an attack when you are using H_1?

> **Solution:** This type of attack can be detected by checking the server certificate manually.

(b) (10 points) The TAs and the instructors for the NetSec class have long term public-private key pairs. While preparing the exam, having to use the open campus wireless network, the TAs use a TLS-protected channel to discuss the questions and solutions to this exam. Shortly before the exam, the instructors and the TAs find out that their long-term private keys may have been learned by one of the students who came to their office hours. While they are sure that no communication about the exam took place since the compromise, they know that the students might have been actively listening on the communication channel in the past. For the following, comment on whether the instructors should be worried that the exam solutions could have been leaked. Briefly describe why or why not they are secure.

i. (2 points) Anonymous Diffie-Hellman was used as a key exchange method, 128-bit AES, and 128-bit SHA-1 based MAC.

Solution: Yes, they should be worried. This approach is not secure as it makes man-in-the-Middle attack possible.

ii. (2 points) Ephemeral Diffie-Hellman was used as a key exchange method, 128-bit AES, and 128-bit SHA-1 based MAC.

Solution: This approach is secure as using Ephemeral Diffie-Hellman with 128-bit AES and 128-bit SHA-1 based MAC provides Perfect Forward Secrecy and is secure against MitM attacks.

iii. (2 points) Ephemeral Diffie-Hellman was used as a key exchange method, 40-bit DES, no MAC.

Solution: This approach is not secure. One can break 40-bit DES and read, modify, and inject the messages.

iv. (2 points) Fixed Diffie-Hellman was used as a key exchange method, 128-bit AES, and 128-bit SHA-1 based MAC.

Solution: This approach is not secure as Fixed Diffie-Hellman does not guarantee Perfect Forward Secrecy.

v. (2 points) Fixed Diffie Hellman was used as a key exchange method, 40-bit DES, and 128-bit SHA-1 based MAC.

Solution: This approach is also not secure. In addition to the points mentioned above, one can break 40-bit DES and read messages, but not modify or inject messages.

(c) (4 points) Bank x.com has a secure web site for its customers to log into their accounts. The bank wants to make sure that the username/password dialog is only shown on a TLS-protected page.

A customer accessing `http://www.x.com` is immediately redirected to the TLS-protected page `https://www.x.com`.

i. (2 points) Is there a difference in security between (a) typing `https://www.x.com` directly and (b) typing `http://www.x.com` and then being redirected to `https://www.x.com`? Briefly explain.

Solution: Using HTTP and being redirected to a TLS page is not a secure approach. An attacker can forge the HTTP page and redirect you somewhere else. Having the user go directly to https page would prevent such an attack.

ii. (2 points) Present at least one reason why a bank would still want to redirect the login from an `http` site to a TLS protected page, instead of simply using only TLS-protected web sites?

Solution: Using HTTPS for busy web sites can cause too much overhead, and therefore affect the performance of the system.

6. Identity and Authentication (11 points)

Captive Portal. A *captive portal* is a popular solution to authenticate users to (insecure) public wireless networks in public places (e.g., airport, coffee shops). In this problem, we investigate possible security vulnerabilities of captive portal systems.

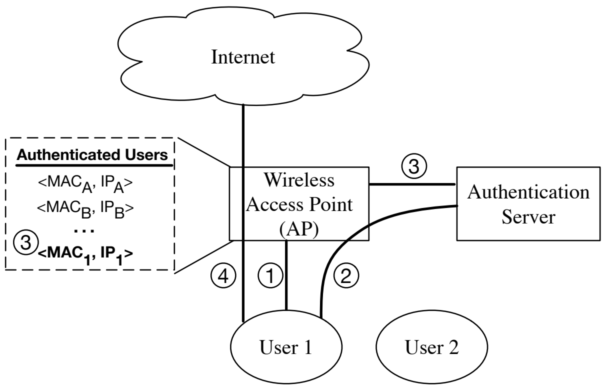


Figure 3: A simplified description of Captive Portal Systems

Figure 3 shows a simplified protocol description of a captive portal system. The following actions take place at the correspondingly enumerated steps:

1. *User 1* joins the wireless access point (AP). The AP assigns an IP address (IP_1) to *User 1*.
2. *User 1* is redirected by the wireless AP to the authentication server. The authentication server authenticates the user using information such as e-mail address, mobile phone number.
3. Once authenticated, the authentication server instructs the AP to whitelist the user. The AP adds the MAC address of the user's wireless interface (MAC_1) and the assigned IP address (IP_1) to the *Authenticated Users* table.
4. Upon receiving *User 1*'s traffic, the AP consults its list of authenticated users to verify the user and forwards his/her traffic to the Internet.

(a) (3 points) Assume that *User 1* has already authenticated to the public wireless network and is connecting to the Internet. Then *User 2* comes within the wireless coverage area of the AP. Can *User 2* see the packets between the AP and *User 1*? Why or Why not? If so, what information can *User 2* learn from the packets to/from *User 1*?

Solution: Yes. MAC Header and the IP header.Also, content if not encrypted.

(b) (4 points) Is this system secure? That is, can *User 2* access the Internet without authenticating itself to the wireless network? If not, why is this impossible? If so, describe your mechanism step-by-step.

Solution:

1. Learn MAC and IP address of *User 1*
2. Spoof the learned addresses
3. Send packets using the spoofed addresses

(c) (4 points) Which additional measure would you implement to further increase the security of this system? Discuss advantages and disadvantages of your proposal.

Solution:

- Negotiate Cryptographic Keys between the user and the AP during authentication process. This requires a separate protocol, which may require changes to the users' devices.
- Expire whitelist more frequently (Not perfect and may annoy the user)
- There can be many other answers.

Extra Page

Please use this page in case you run out of space elsewhere in the exam.

Extra Page

Please use this page in case you run out of space elsewhere in the exam.