Assignment-5 | Report | cs23mtech14009

Q1. Assume that TLS handshake does not use any nonces. Explain how Trudy can be successful in launching session/connection replay attacks by capturing all the messages exchanged between Alice and Bob a while ago. You can assume Alice and Bob used TLS for securing their communication related to say ordering an item for e-commerce, online payments, secure file transfers, etc. Can Trudy replay Alice's previous messages with Bob for successfully launching a session/connection relay attack with Bob? Explain your answer. Can Trudy replay Bob's previous messages with Alice for successfully launching a session/connection replay attack? Explain your answer.

In the first case, where Trudy is acting as Alice, there is a vulnerability because the TLS handshake does not involve the use of nonces. Because of this, Bob cannot distinguish between Alice and Trudy, as they would share the same random number (Rs). Trudy takes advantage of this situation, can generate the same Master Secret and successfully execute a replay attack with Bob.

However, in the second case, Trudy is acting as Bob. the situation is more secure. Trudy lacks access to Bob's private key, which is required to decrypt the Premaster Secret (PMS) sent by Alice. Since the PMS is encrypted with Bob's public key, Trudy is unable to proceed with the handshake. As a result, Trudy is unsuccessful in launching a replay attack with Alice.

Q2. Explain how nonces employed in TLS help in preventing session/connection replay attacks in Q1.

Regarding the explanation in question 1, when nonces are brought into play, Bob creates different random numbers (new Rs) and conveys them to Trudy. Trudy, in turn, dismisses Bob's messages and exclusively adopts those from Alice, leading Trudy's Master Secret to become MS (Alice's). On the other hand, Bob's Master Secret becomes (new MS*) due to his reliance on Rs(new). This functionality of nonces in TLS functions as a preventive measure against replay attacks by introducing dissimilarity in Master Secrets (MS != MS*), thereby causing a cessation in the Handshaking process.

Q3. How does Alice derive the PreMaster Secret (PMS) which she wants to send to Bob? Refer RFC 5246.

With RSA employed for key exchange, Alice crafts a 48-byte PreMaster Secret, including a 46-byte random number. It also includes a 2-byte TLS version. This PMS is specific to this session and will be used to derive session keys. This PreMaster Secret is subsequently encrypted using Bob's public key from Bob’s digital certificate. The encrypted PMS is sent in the "ClientKeyExchange" message to Bob.

Q4. Why can't Bob derive PMS and share it with Alice?

First We Know that How Alice sends PMS to Bob in this Alice encrypts the PMS with the public key of Bob and where the public key of Bob got from the Digital Certificate of Bob which Alice got from the server hello message. Alice also can send her digital certificate to Bob but it is optional so Bob will not have the public key of Alice every time so Bob can’t derive and share it with Alice. In TLS sessions, the certificate_request message from the server (Bob) to the client (Alice) is optional, this is the main reason.

Q5. Think of a scenario in which it's possible for Bob to derive PMS and share it to Alice. Refer TLS 1.2 handshake message protocol and explain how it can be extended (say, by adding new messages) to achieve this behavior.

In TLS sessions, the certificate_request message from the server (Bob) to the client (Alice) is optional, First we have to mandatory this option. Then Bob will have the public key of Alice and then he can send PMS by using Alice's public key in encrypted form and Alice will decrypt it with her own private key.

Its key exchange step client can send a digital certificate to Bob and Bob verifies it with Certificate Authority's (CA) then generates PMS and sends it to Alice then Alice also end it with sending a digest of all message she receives and encrypt it with Bob’s Private key and to verify the integrity of the messages.

Q6. Note that MS is derived by feeding PMS and nonces of Alice and Bob as inputs to a PRF (that is known to all) by both Alice and Bob independently. Similarly, MS and nonces of Alice and Bob, and key_block size are fed as inputs to a PRF to derive key material which are split into MAC keys, session keys and IVs (IVs for AES-CBC only) by both Alice and Bob independently. To lessen the burden(!) on Bob, Out of her love for Bob, Alice said that she would generate MS from PMS and nonces of Alice and Bob and directly share the MS to Bob by encrypting it with Bob's RSA public key. Trudy captured messages exchanged between Alice and Bob in this modified handshake protocol. Do you think Trudy can succeed in launching session/connection replay attacks on Bob? Justify your answer.

Trudy's attempt to execute replay attacks on Bob will be unsuccessful. This holds true even if Alice generates the Master Secret (MS) and because of extra love and care communicates it to Bob. Because we are using nonce at the Bob side this Nonce will not allow replay attack and it generates a new number every time so this message will be distinct from the Alice message.

Q7. More love from Alice. Extension to Q6. Alice said that she would generate key material from MS and nonces of Alice and Bob, and key_block size and share the key material directly to Bob by encrypting it with Bob's public key. Trudy captured messages exchanged between Alice and Bob in this modified handshake protocol. Do you think

Trudy can succeed in launching session/connection replay attacks on Bob? Justify your answer.

No,
If Alice is generating both random numbers and sending PMS directly to Bob. One way to think about this is if Trudy stores all the messages for the whole session when session is close Trudy can act as Alice and replay all messages and communicate with Bob. Bob can distinguish by finishing the message digest. Bob will generate a new random number so the digest of all messages at Bob's side will not be the same as Trudy sends it .

Q8. Sequence number counter (initially set to 0) is used by Alice to input the current value of the sequence number counter while calculating MAC for inclusion into TLS records for integrity protection. Assume that Alice has been sending 10 TLS records carrying application data (each of size 500 Bytes) to Bob. Trudy being Woman-in-the-Middle between Alice and Bob, deletes record numbered 7th. She wants to fool TCP's in-sequence delivery mechanism so that the TCP receiver at Bob thinks everything is perfect and forwards the received TLS records to the TLS layer. How could she get away and pass through TCP checks? Hint: Trudy has to manipulate TCP segments numbered 8th, 9th and 10th. How?

Yes, Trudy has the capacity to manipulate TCP's in-sequence delivery mechanism by tampering with specific TCP header fields, including ACK number, and sequence number also checksum and more,. In her role as a Woman-in-the-Middle, Trudy strategically removes the 7th TLS record from Alice to Bob. To accomplish this, she modifies the sequence numbers of records 8, 9, and 10 to 7, 8, and 9, ensuring a seamless transition from the manipulated sequence. Trudy also adjusts the checksum to mislead Bob's TCP into perceiving the records' integrity. To further mislead the system, Trudy transmits a falsified ACK of the 7th record to Alice, successfully evading TCP checks.

Q9. Having successfully fooled the TCP receiver of Bob in Q8, do you think Trudy can fool the TLS receiver of Bob? Explain.

No, I think Trudy can not fool the TLS receiver of Bob because we know that there is modification in sequence number in TLS that is verified by the TLS receiver and this counter of sequence number is used for the keyed MAC.

Q10. Assume that Trudy captured application data messages exchanged between Alice and Bob using TLS 1.2. Alice is a web browser whereas Bob is a web server with Digital Certificate signed by a CA using RSA. After a year from this correspondence between Alice and Bob, Trudy hacked into the webserver and stole Bob's private key. Explain how Trudy can decrypt all of the old application data exchanged between Alice and Bob? This means there is no forward secrecy. It's indeed possible when TLS_RSA_WITH_AES_256_CBC_SHA256 is used as the cipher suite.

Trudy intercepted messages between Alice and Bob in a secure conversation. A year later, she hacked into Bob's web server, getting hold of his secret key. There's no extra security, especially in a setup like TLS_RSA_WITH_AES_256_CBC_SHA256. Trudy decrypts all the old messages by having copies of the initial messages from a year ago, including ServerHello and ClientHello. These messages have a special number. After hacking the server and reading Alice's secret message (Premaster Secret or PMS), Trudy uses that, along with the special number, to create another secret (Master Secret or MS). With this new secret, Trudy can read all the old messages sent between Alice and Bob.

Q11. You are tasked with providing perfect forward secrecy by fixing the issue described in Q10. What tweaks do you make to TLS_RSA_WITH_AES_256_CBC_SHA256 for that? Hint: You can't replace RSA with any other algorithm in the ciphersuite.

To ensure perfect forward secrecy and address the problem mentioned in Q10, I think I can implement the following adjustments to TLS_RSA_WITH_AES_256_CBC_SHA256:
The issue from Q10 can be effectively mitigated by generating a new Diffie-Hellman (DH) private key for each handshake. Opt for one of the DHE (Ephemeral Diffie-Hellman) cipher suites, as they offer perfect forward secrecy. Replace the existing cipher suite with ECDH_RSA, a variant that incorporates Ephemeral Diffie-Hellman and ensures perfect forward secrecy. This modification strengthens the security of the TLS communication, making it more resilient to compromises.

Q12. Compare and contrast TLS_ECDH_RSA_WITH_AES_256_GCM_SHA and TLS_RSA_WITH_AES_256_CBC_SHA ciphersuites? Does TLS_ECDH_RSA_WITH_AES_256_GCM_SHA offer perfect forward secrecy? Explain.

TLS_ECDH_RSA_WITH_AES_256_GCM_SHA: > TLS_RSA_WITH_AES_256_CBC_SHA

TLS_ECDH_RSA_WITH_AES_256_GCM_SHA gives you Perfect Forward Secrecy, because it use each time use new key to encrypt PMS so that if one key compromise it can decrypt only one session messages and it is also difficult because it don’t store the key in server so to encrypt messages to get the private key you have limited amount of time for it.

The TLS_RSA_WITH_AES_256_CBC_SHA doesn't give you Perfect Forward Secrecy because it uses the same key for all messages (more in question 10).

TLS_ECDH_RSA_WITH_AES_256_GCM_SHA:

- This uses a good method called Ephemeral Elliptic Curve Diffie-Hellman (ECDH) for sharing secret keys.
- Adds extra security with Galois/Counter Mode (GCM) for encryption, ensuring a strong and authenticated lock on messages.

Were as,

TLS_RSA_WITH_AES_256_CBC_SHA:

- It has a traditional method called RSA for key exchange.
- Cipher Block Chaining (CBC) for encryption, a reliable method but without some of the good features of GCM.

Q13. Refer RFC 5246 on Cipher Suites of TLS 1.2 and list down the ones that offer perfect forward secrecy

Cipher Suite	Key Exchange	Cipher	Mac
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA	DHE_DSS	3DES_EDE_CBC	SHA
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA	DHE_RSA	3DES_EDE_CBC	SHA
TLS_DHE_DSS_WITH_AES_128_CBC_SHA	DHE_DSS	AES_128_CBC	SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA	DHE_RSA	AES_128_CBC	SHA
TLS_DHE_DSS_WITH_AES_256_CBC_SHA	DHE_DSS	AES_256_CBC	SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA	DHE_RSA	AES_256_CBC	SHA
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256	DHE_DSS	AES_128_CBC	SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	DHE_RSA	AES_128_CBC	SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	ECDHE	AES_128_CBC	SHA
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	ECDHE	AES_128_CBC	SHA

Q14. What measures are taken in TLS 1.2 with respect to TLS_ECDH_RSA_WITH_AES_256_CBC_SHA cipher suite to guard against various attacks?

Using the TLS 1.2 protocol with the cipher suite TLS_ECDH_RSA_WITH_AES_256_CBC_SHA:

ECDH_RSA Key Exchange: This intelligent and brilliant method not only facilitates key exchange but also ensures Perfect Forward Secrecy, effectively guarding against replay attacks.

AES_256_CBC Encryption: Data confidentiality is maintained through the application of the Advanced Encryption Standard (AES) in Cipher Block Chaining (CBC) mode, featuring a 256-bit key. The Initialization Vector in CBC injects variability, even when the plaintext remains consistent.

SHA for Integrity: Employing the Secure Hash Algorithm (SHA) guarantees the overall authenticity and integrity of the session.

Collectively, these measures create a robust defense, providing formidable protection against various forms of attacks.

Q15. Refer RFC 8446 on Cipher Suites of TLS 1.3 and list down the ones that offer perfect forward secrecy

Perfect Forward Secrecy (PFS) stands as the central feature of TLS 1.3, inherently provided by all employed ciphers. Key exchange techniques in TLS 1.3 cipher suites primarily revolve around DHE (Ephemeral Diffie-Hellman) and ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). Notable examples of these suites include TLS_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384, and TLS_AES_128_CCM_SHA256. TLS 1.3's unwavering commitment to PFS ensures robust security across the spectrum of cryptographic operations.

Q16. Privacy issues with TLS 1.2: Does any 3rd party like ISPs/enterprises profile their users (i.e., browsing patterns) even though their application data is encrypted? Explain!

Encrypting application data doesn't completely address privacy concerns, even with TLS 1.2. Despite content protection, metadata vulnerability exists, enabling organizations and ISPs to identify browsing trends, traffic patterns, DNS queries, destination IPs, and more. Additional steps are required to lessen the exposure of metadata and allay any worries in order to solve these concerns.

References:

- Slide deck on TLS
- https://tools.ietf.org/html/rfc5246
- https://www.coursera.org/learn/crypto/lecture/WZUsh/case-study-tls-1-2

PLAGIARISM STATEMENT :

I certify that this assignment/report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment/report has not previously been submitted for assessment in any other course, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that I have not copied in part or whole or otherwise plagiarized the work of other students and/or persons. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, I understand my responsibility to report honor violations by other students if I become aware of it.

Name: Popat Raj Rameshkumar
Date: 1/2/2024
Signature: cs23mtech14009