CS974-Final-Exam

● Graded

Student

Mohammed Jawed

Total Points

50 / 50 pts

Question 1

DNS

10 / 10 pts

Part (a)

✔ **+ 5 pts** Corect

+ 3 pts Partially correct

+ 0 pts Incorrect / Not attempted

Part (b)

✔ **+ 5 pts** Correct

+ 3 pts Partially correct

+ 0 pts Incorrect / Not attempted

Question 2

Authenticity, Integrity, Confidentiality

10 / 10 pts

Part (a)

✔ **+ 5 pts** Corect

+ 3 pts Partially correct

+ 0 pts Incorrect / Not attempted

Part (b)

✔ **+ 5 pts** Correct

+ 3 pts Partially correct

+ 0 pts Incorrect / Not attempted

Question 3

DDoS

10 / 10 pts

Part (a)

✔ **+ 5 pts** Corect

+ 3 pts Partially correct

+ 0 pts Incorrect / Not attempted

Part (b)

✔ **+ 5 pts** Correct

+ 3 pts Partially correct

+ 0 pts Incorrect / Not attempted

Question 4

Local vs Internet

10 / 10 pts

Part (a)

✔ **+ 5 pts** Correct

+ 0 pts Incorrect

Part (b)

✔ **+ 5 pts** Correct

+ 0 pts Incorrect

Question 5

Digital Certificates

10 / 10 pts

✔ **+ 10 pts** Correct

+ 8 pts Minor mistakes

+ 5 pts Partially correct

+ 0 pts Incorrect / Not attempted

Q1 DNS
10 Points

DNS queries are resolved either by looking up the cached translation of domain names to IP addresses in the local DNS resolvers, or by going to one of the root level DNS servers which then directs the query to top level domain (TLD) name servers, which again directs the query to organization specific name servers etc.

(a) In standard DNS, the query message in the DNS protocol is neither authenticated nor encrypted. List the top 3 cyber threats that arise out of this lack of cryptographic protection in DNS protocol.

following are Top 3 cyber threats :
1. Eavesdropping: DNS queries are not encrypted, it's easy for attackers to intercept these and gather sensitive information.
2. Data Manipulation: Without authentication, attackers can alter DNS responses, directing users to malicious sites.
3. Replay Attacks: the absence of sequence validation in DNS allows attackers to replay previously captured queries. This can redirect users repeatedly to malicious sites or cause disruptions in service, further compromising security.

(b) Suppose that we change the DNS protocol such that the queries are encrypted and responses are also encrypted. What will be some of the major problems in making that work? Even if we can make it work, will that be enough to ensure that the translation from URL to IP addresses are trustworthy?

Part-1
Some of the major problems that will add as an overhead are:
1.Key Management: Managing encryption keys for billions of DNS servers and clients would be a significant challenge
2. Interoperability: Ensuring that all DNS servers, resolvers, and clients can understand and implement the new encrypted protocol would be a complex task.
3. Performance Impact: Encrypting and decrypting DNS queries and responses can significantly slow down DNS resolution due to added computational work.
4. Scalability: As we know DNS s/m handles a massive volume of queries, Implementing encryption could load current infrastructure.

Part-2:
Even with encryption, the trustworthiness of URL-to-IP mappings isn't

assured because encryption alone doesn't ensure the data's integrity or authenticity; compromised DNS servers could still return false information. in short: it does not ensure the integrity of DNS data; compromised servers could still return incorrect mappings, necessitating supplementary measures like DNSSEC for validation and trustworthiness.

Q2 Authenticity, Integrity, Confidentiality
10 Points

Most application layer protocols such as HTTPS, SSH, SFTP etc., already use encryption and authentication.

(a) What cryptographic techniques are used at the application layer to ensure confidentiality of the messages at the application level, integrity of the messages, and the authenticity of messages?

Following are cryptographic techniques can be used at the application layer:

1. Encryption ensures confidentiality by transforming readable data into an unreadable format using algorithms such as AES , only decipherable by those possessing the correct key.
2. Hash functions, like SHA-256, are used to maintain integrity by generating a fixed-size hash value from data, which is checked on both ends to detect alterations.
3. Digital signatures and certificates verify authenticity, proving the origin and identity of the data source. Protocols like SSL/TLS use these signatures alongside public key infrastructure (PKI) to establish secure and trusted connections.

(b) Suppose IPSEC becomes ubiquitous --i.e, all IPv4 is replaced by IPSEC, in other words, all network later traffic is encrypted, message authentication code added, digital signature added etc. Does that mean that we will no longer need HTTPS, SSH etc? Does that mean that HTTP and TELNET will be secure?

Even if IPSEC encrypts all network traffic, application layer protocols like HTTPS and SSH are still necessary. IPSEC doesn't provide end-to-end encryption directly between applications, leaving potential endpoint vulnerabilities. Additionally, HTTPS and SSH handle user authentication and manage digital certificates, essential for confirming identities and maintaining secure communications.
Hence, IPSEC would not replace the need for these protocols, nor would it make HTTP and TELNET inherently secure at the application level.

Q3 DDoS
10 Points

(a) Explain why only UDP-based protocols are used for amplifying DDoS attacks.

UDP-based protocols are commonly used for amplifying DDoS attacks because UDP is connectionless, allowing attackers to easily spoof IP addresses and send requests that elicit large responses to a target's IP. This efficiency in sending large volumes of data without connection verification or integrity checks makes UDP ideal for executing high-impact DDoS attacks.

(b) Many DDoS attacks use misconfigured UDP services to amplify traffic, but require a network without egress filtering to launch the attack. Explain why we do not know which networks serve as the source of these DDoS amplification attacks

DDoS amplification attacks using misconfigured UDP services are hard to trace because attackers use spoofed IP addresses, making it appear as though the attack originates from the victim's network. Without proper egress filtering, many networks allow these spoofed requests to go through, making it difficult to track down the true source of the attack and complicating efforts to trace the originating network.

Q4 Local vs Internet
10 Points

What attack can be mounted on a local network to intercept and eavesdrop on network traffic that cannot be mounted over the Internet? What can you do if you are building an Internet service to protect against this potential eavesdropping?

In a local network, ARP spoofing is a common attack where an attacker redirects traffic to their machine by associating their MAC address with the IP address of another device, typically the gateway. This lets them intercept and eavesdrop on network traffic.

To protect an Internet service against such eavesdropping, I would ensure all data transmitted over the Internet is encrypted using SSL/TLS. This encryption prevents anyone who might intercept the data from being able to read it. Additionally, implementing VPNs for remote access can provide an extra layer of security by encrypting all traffic entering and leaving the network.

Q5 Digital Certificates
10 Points

A digital certificate is issued by a CA and is used to bind a public key to an entity such as a website. In practice, a CA provides the website with a certificate chain, say A → B → C, which means that a trusted CA called A issued a certificate to another trusted CA called B; later B issued a certificate to website C. The website presents its certificate chain (A → B → C) to the browser. The browser checks that the A →B certificate and the B → C certificate are both valid. Then, if it trusts the CA called A, then it also trusts the CA called B, and therefore can trust the public key of C. Here A is called the top-level CA and B is called an intermediate CA.

Let's see a potential problem with this mechanism. John Smith generates a public/private key pair (sk,pk), and obtains a certificate from a trusted CA binding his public key pk to the domain johnsmith.com which he owns. He then generates a second public/private key pair (sk′,pk′), and uses his first private key sk to sign a certificate that binds pk′ to the domain www.amazon.com. In effect John is acting as a CA. Now John has a certificate chain for www.amazon.com, where johnsmith.com plays the role of the intermediate CA. Browsers might incorrectly accept this fake Amazon certificate because there is a valid certification path to a trusted top-level CA.

Why does this scenario present an attack? How can it be exploited?

In this scenario,
John Smith generates a fake certificate chain for www.amazon.com, where his legitimately certified domain johnsmith.com is incorrectly acting as an intermediate CA. This misuse could lead browsers to accept the fraudulent certificate if they trust the root CA that certified John's domain.

As a result, this could facilitate man-in-the-middle attacks where John intercepts data meant for Amazon, exploiting the trust users place in the seemingly secure connection. To prevent such misuse, it's crucial for Certificate Authorities to enforce strict verification processes and for systems like Certificate Transparency to monitor and expose unauthorized certificates.