CS 642: Midterm 1 Review Questions and General Study Pointers

March 2020

1 Threat Modeling, Security Mindset

Review the security mindset, and practice threat modeling on a few example websites. Also, review the various types of attacker models and the capabilities of each. E.g., what are the capabilities of a network attacker, web attacker, etc.

2 Cryptography and Authentication

- Modes of operation: Why is ECB not recommended? Because identical blocks of plain text produce identical ciphertext. In many cases, this leaks information about the underlying plaintext.
- Does an encryption scheme provide integrity? No. It only provides confidentiality. To get integrity, we need to use cryptographic hashes, and signatures.
- Is CBC-MAC secure? CBC-MAC is vulnerable to length extension attacks.
- What are the properties of cryptographic hash functions? One way-ness, Collision Resistance, Weak Collision resistance.
- What are recommended hash functions? SHA-256 and higher is good. MD5 has collisions.
- What is an HMAC? It is a hashed message authentication code, See the slides for how it works.
- What is a good way to get authenticated encryption? Encrypt-Then-MAC. Why are other options not good? They break security properties. E.g., Encrypt-AND-MAC violates CPA security.
- How does asymmetric cryptography provide confidentiality and signatures? Each party has a keypair = (public-key, private-key). For confidentiality of a message from Alice to Bob, Alice will encrypt the message

using Bob's public key. That message can only be decrypted using Bob's private key. As only Bob has that private key, no one else can decrypt the message. For signature on a message from Alice to Bob, Alice hashes her message, and then encrypts it using her private key. Anyone can use Alice's public key to decrypt that message, obtain the hash, and then re-hash the plaintext and check if the hashes matched. As only Alice has her private key, only Alice could've generated the signature.

- Review X509 certificates, and how they work/are used.

3 Web Security

- What is the difference between a reflected and stored XSS? Explain with an example. (see slides for answer).
- How can a developer protect against XSS, CSRF, SQL injection?
- Consider the following SQL statement:
  `res = execute("SELECT * FROM Users WHERE id=' & form('user')");`
  What attack can you mount on this statement?
- What is the Same Origin Policy? It is a set of rules that browsers user to isolate content from different origins. Review example domains and see whether the SOP applies or not based on the domain name.
- For a domain, who is allowed to set a cookie? The first-party domain can set its own first-party cookies. If there is any embedded content, then these third-parties can set their own cookies, however, they cannot read first-party cookies.
- Should a developer encrypt cookies? In general no, because a cookie does not contain information that is hidden from a user. It contains user information, such as their shopping cart contents. It does not make sense to encrypt this data in this context.
- What are the ways in which a website can track user activity? (see slides on Web privacy for this).

4 Network Security

- Review Dan Kaminsky's cache poisoning attack in detail.
- Review the document on the website: The first few milliseconds of an HTTPS connection.
- What properties does TLS provide? (review slides for this).

- Will TLS protect you against a phishing attack? Yes, but it requires the user to notice that something is incorrect with the certificate.
- Will TLS protect you from DNS attacks? It again depends on the user inspecting the certificate of the connection and realizing that the cert is not a descendant of a valid chain from a trusted root CA. In general, most people will not check certificates in this way, and will get hacked.