Interview Preparation

<https://geektastic.com/blog/five-killer-interview-questions-for-developers?&ref=quuu&utm_content=buffer16252&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer>

What happens between typing a URL into your browser address bar, hitting enter and seeing a web page?

DNS, etc, Protocol, network knowledge

What are the things you should consider if you were writing your own database server?

ACID

Atomicity, Consistency, Independence, Durability

Computer security considerations

CIA

Confidentiality, Integrity, Availability

OWASP

Top 10 current threats

1. Injection (SQL Injection)
2. Broken authentication and session management: leaks in the authentication system, poor passwords management in all the areas, such as resets, etc, session management – not timing out, leaking data in session state info
3. Cross site scripting (XSS): Injecting scripts, e.g. javascript because input is not properly escaped, or validated
4. Broken Access Control: e.g. horizontal or vertical privilege escalation
5. Security misconfiguration: leaving default accounts around, unpatched flaws, unprotected files or directories
6. Sensitive data exposure: hackers stealing data, man-in-the-middle, stealing keys. Not encrypting sensitive data.
7. Insufficient attack protection: detecting and blocking attacks, rather than ignoring them, e.g. incorrect password lockout.
8. Cross site request forgery
9. Using components with known vulnerabilities: teams not keeping libraries up to date
10. Under protected APIs
11. Click-jacking
12. Denial of service
13. Information leakage, incorrect error handling
14. Malicious file execution
15. Unvalidated redirects and forwards

Some encryption knowledge in .NET – what is currently the best practice?

**Symmetric algorithms**

**AES** – Advanced Encryption Standard (Replaces DES). A subset of Rijndael but with a fixed block size of 128bits, the jkey can be 128, 192 or 256 bits.

DES – Data Encryption Standard (old – not very secure – can be brute forced)

RC2 – again old don’t use

Triple DES or 3DES – is a more secure version of DES, but cascading 3 instances of DES. It is slow.

Rijndael – block size can be specified in size o 32 bits (from 128 – 256), AES is based on it but with a fixed block size of 128 bit.

**Asymmetric algorithms**

Public and private keys are inverses of each other: public key enc – private key dec & private key end – public key dec. Asymmetric encryption is quite slow, so typically if encrypting a lot of data, you would use symmetric encryption and then use asymmetric encryption to encrypt the key that you provide for decryption.

**RSA**

**Hash algorithms**

MD5 – susceptible to collisions where 2 messages can have the same hash

SHA1 – still susceptible but not as much as MD5

**SHA2** – the most resistant – use this – in .Net this is SHA256, SHA384, SHA512

SHA3 – this is the newest standard but question marks over performance

<https://www.cryptologie.net/article/400/maybe-dont-skip-sha-3/>

BLAKE, SHAKE, BLAKE2

Consider how secure the data needs to be and also how quickly it needs to hash, because the less secure versions are faster.

<https://stackoverflow.com/questions/5554526/comparison-of-des-triple-des-aes-blowfish-encryption-for-data>

Block cipher: encryption is applied to blocks of data rather than one bit at a time.

Initialization Vector: this should be used with AES to prevent the same data being encrypted exactly the same way. So it add some random initial block which will change how the whole encryption array looks.

Public key to encrypt – private key to decript

Symmetric algorithms

Uses the same key to encrypt and decrypt

Hashing

Also, know more about secure cookies, and SSL certificates, etc.

Microservices – patterns

Bitwise operations

Algorithms

Code quality

SOLID

Design Patterns

C#

* Streams
* Async
* Delegates

Javascript

Domain Driven Design

Enterprise Design Patterns