香港中文大學
The Chinese University of Hong Kong

WEB SECURITY

CSCI2720 2022-23 Term 1

Building Web Applications

Dr. Chuck-jee Chau
chuckjee@cse.cuhk.edu.hk

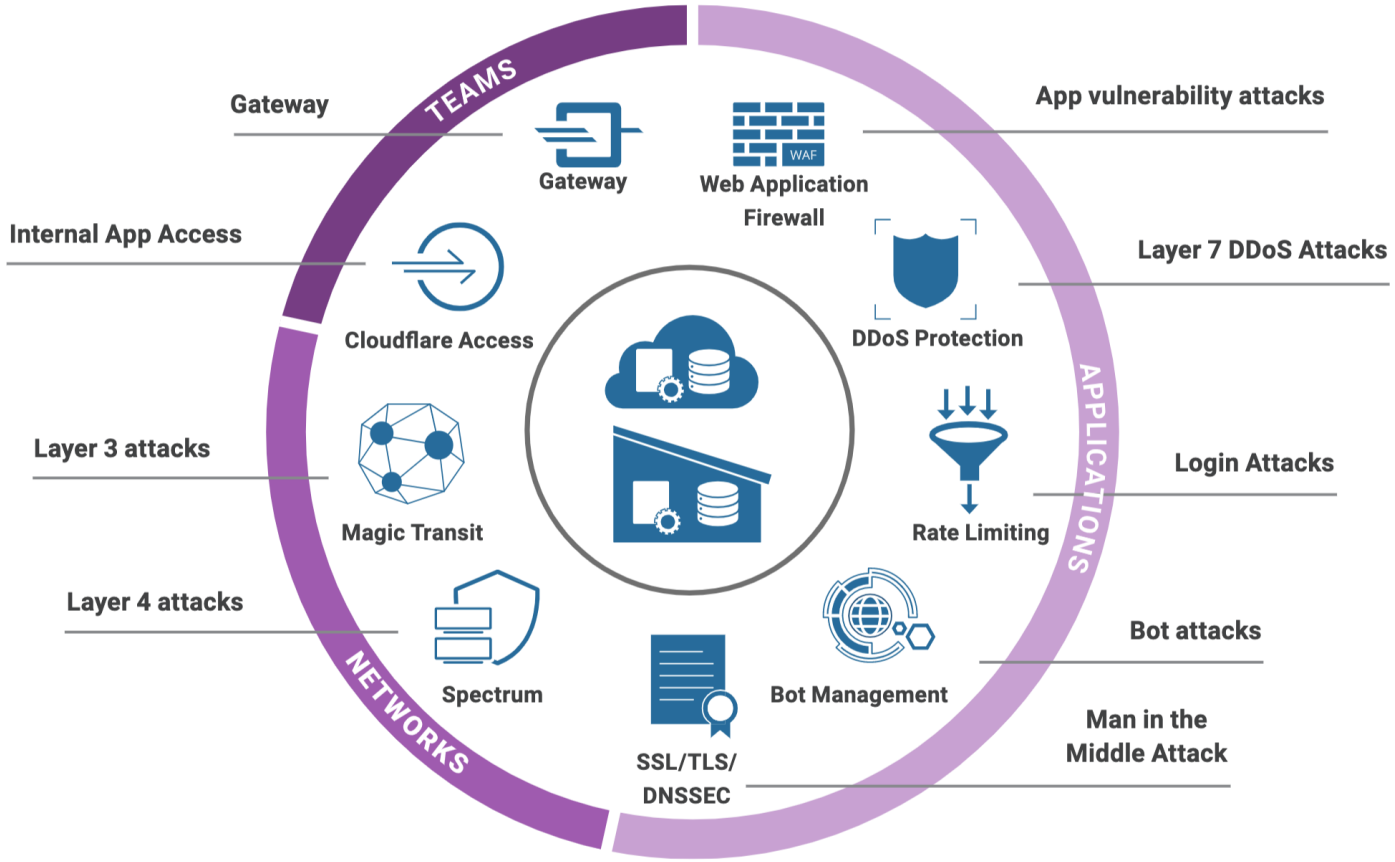
OUTLINE

- Important terms in cyber security
- Top 10 Web Application Security Risks
- Mitigation to threats
- Authentication methods
- HTTPS
- DDoS

CYBER SECURITY

- ***Application*** security
  - Assets at the software level
  - Database, documents
- ***Network*** security
  - Infrastructure of network level
  - Connection, hardware

See: https://www.cloudflare.com/en-gb/security/



IMPORTANT TERMS

- ***Validation***
  - To check if something (e.g., an account, an email) is valid or existing
- ***Verification***
  - To check if something (e.g., account ownership) is real
- ***Authentication***
  - To verify a user with credentials (e.g., username/password) as the correct person
- ***Authorization***
  - To determine the permission on what a user can access (e.g., change a file, or to remove some data)

APPLICATION SECURITY

TOP 10 OF OWASP

- ***Open Web Application Security Project***
- Top 10 Web Application Security Risks (2021)
  1. Broken access control
  2. Cryptographic failures
  3. Injection
  4. Insecure design
  5. Security misconfiguration
  6. Vulnerable and outdated components
  7. Identification and authentication failures
  8. Software and data integrity failures
  9. Security logging and monitoring failures
  10. Server-side request forgery
- *See more: https://owasp.org/Top10/*

COMMON ACCESS CONTROL VULNERABILITIES

- Access granted to ***more than necessary*** capabilities, roles, or users
- ***Bypassing*** access control checks possible with URL/state modification
  - E.g., access with an admin link without proper authentication
- Permitting access to someone else's account with ***unique identifier***
- ***Metadata manipulation*** with cookies or security tokens
- ***CORS misconfiguration*** giving rise to access from unauthorized origins
- *See more: https://owasp.org/Top10/A01_2021-Broken_Access_Control/*

BAD IMPLEMENTATION

- There may be ***carelessness*** or ***ignorance to threats***
  - Including sensitive data in URL
  - Password not encrypted in storage or transit
  - Storing credentials in public code repositories
  - Permitting brute force attacks
  - Running application in development/debug mode for production
  - Session timeout unhandled
  - Missing access control to functions
  - Using components with known vulnerabilities
- ***Using security frameworks instead might be helpful!***
- Test the application thoroughly and rigorously

MITIGATION TO ATTACKS

- Plan carefully for authentication and authorization
- Combination of multiple layers of security measures
- Sanitize all untrusted data
  - All user input should be considered untrusted, and should go through:
    - ***Validation***: check if the string format is as expected
    - ***Escaping***: special characters such as **<** or **>** should be changed to **`<`** and **`>`** to prevent injection of HTML code
    - ***Sanitization***: if needed, only allow certain code in a whitelist
- Enforce same-site requirements
  - Allow cross-site ***only if needed***, with only ***minimal possibilities***

AUTHENTICATION FOR WEB APPS

- Membership is one important feature in apps and services, but how to check the ***identity of users***?

HTTP Authentication	Session/token based	Delegating/Decentralizing
• HTTP Basic/Bearer/Digest authentication • User/password pairs to be checked • Stateless: resending all data in every request	• Authenticated with user/password pairs • Stateful: user info stored on server or client	• OpenID Connect / OAuth 2.0 • User identity being checked by a ***third party***, e.g., "Sign in with Google" • More robust if set up properly
Well supported, not preferred	***Currently most preferred***	Outsourcing – is it good?

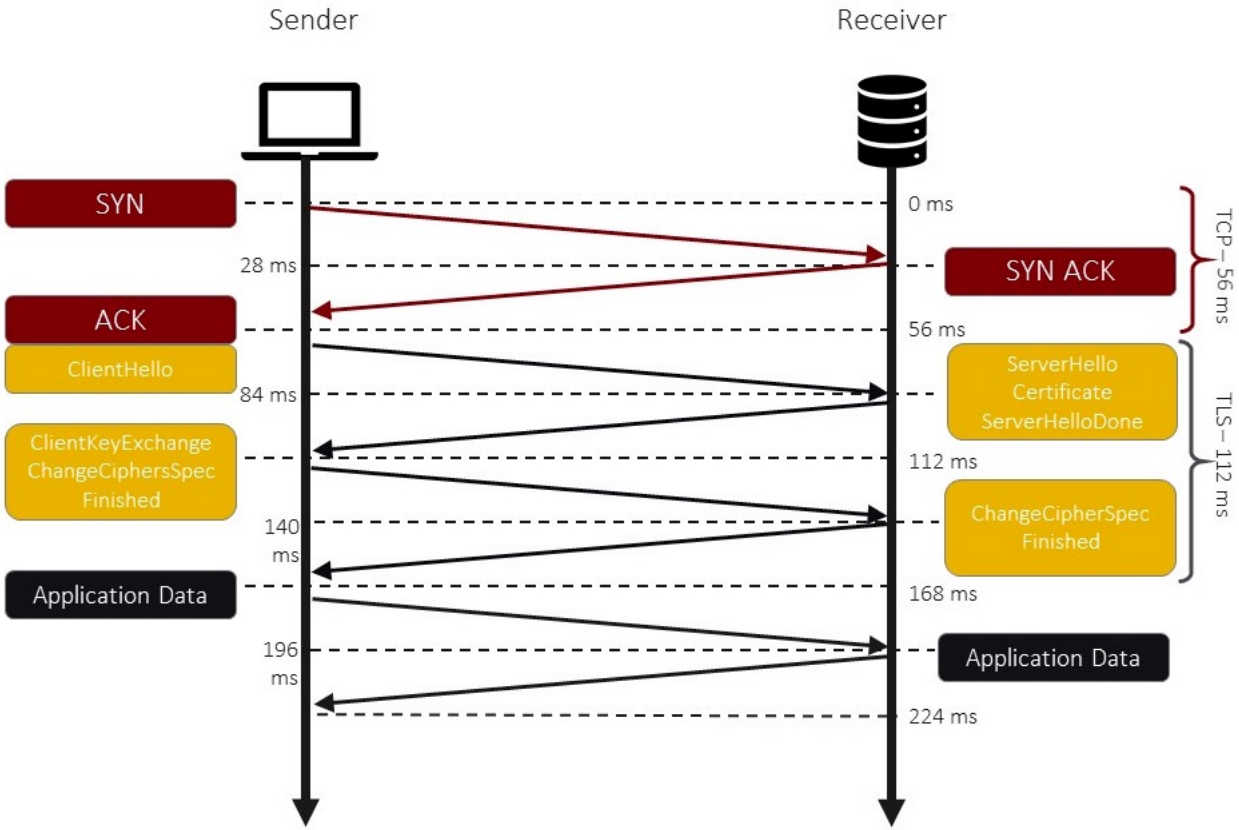
See: https://testdriven.io/blog/web-authentication-methods/

NETWORK SECURITY

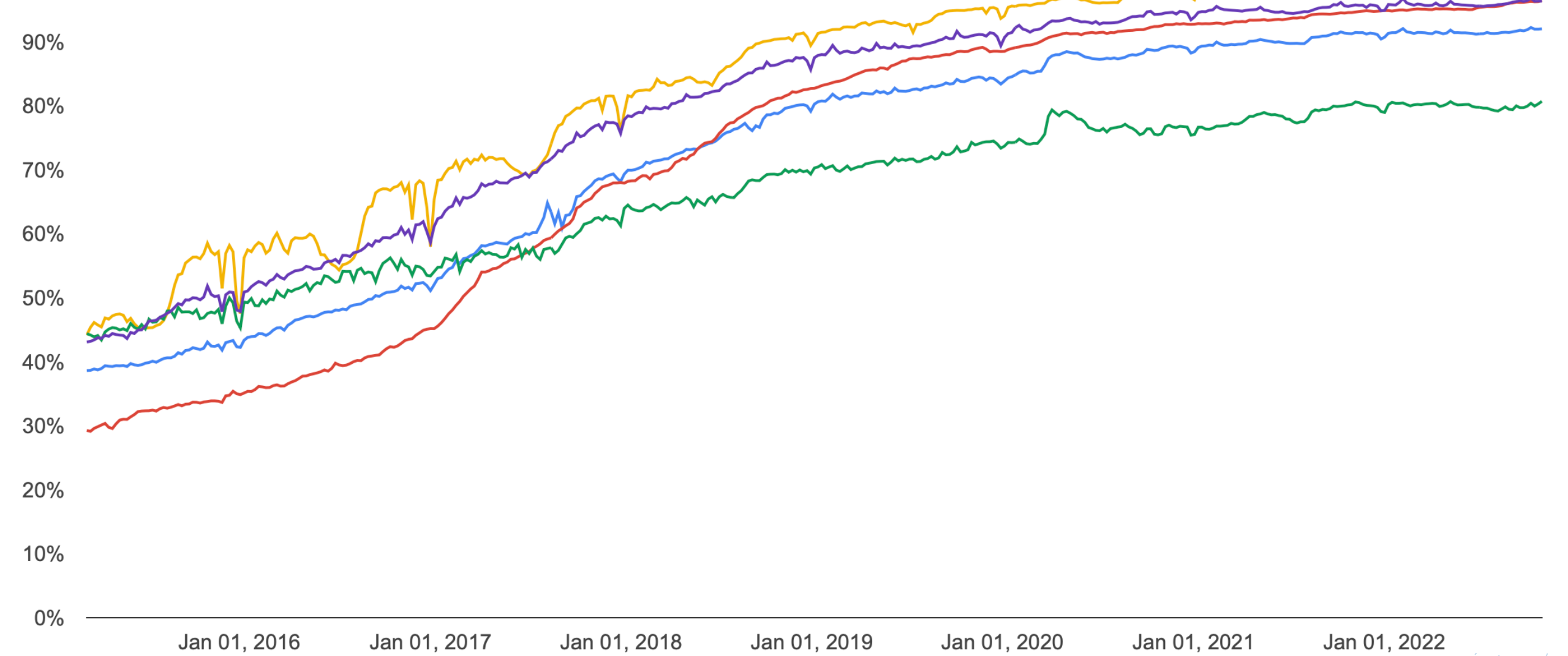
HTTPS

- By design, HTTP transfers everything in plain text!
- ***HTTP Secure*** is an extension to HTTP
  - ***Authentication***: to prove its identity, visited website must present a valid digital certificate signed by an authority
  - ***Encryption***: HTTP requests and responses are transmitted over an added layer of SSL/TLS, so all messages are transferred in ***ciphertext***
- Transport Layer Security (TLS)
  - Private connection with symmetric cryptography
    - A key is used for encryption of *plain text* and decryption of *ciphertext*
    - A unique session key are generated at the beginning of each connection during handshake

THE HTTPS CONNECTION



See: https://love2dev.com/blog/how-https-works/



TREND OF HTTPS PAGE LOADS IN GOOGLE CHROME

See: https://transparencyreport.google.com/https

CERTIFICATES

- To verify identity, signed by a Certificate Authority (CA)
- Server certificates
  - ***Domain Verification***: owning the domain name with DNS records
  - ***Organization Verification***: company name and public address
  - ***Extended Verification***: existence and location of a legal entity
- Browsers and OSes maintain trusted list of CAs
  - If a cert is issued by these CAs, the cert is trusted

CERTIFICATES

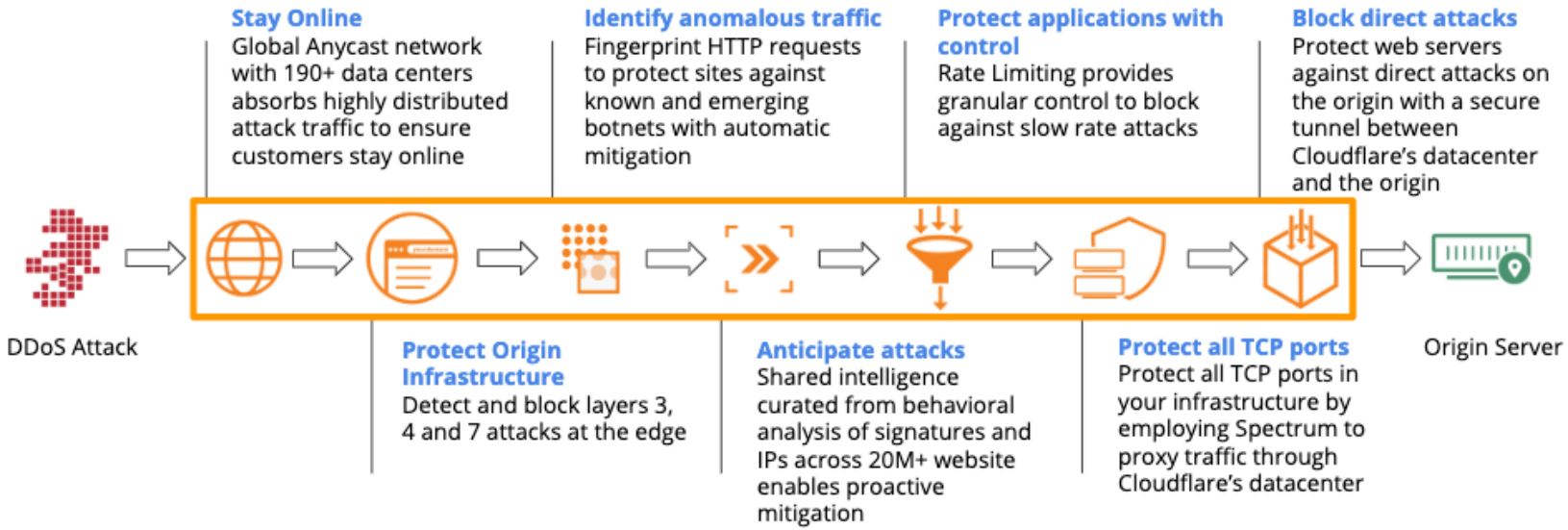
- Commercial CAs
  - Paid service for verification
  - Recognized CAs in Hong Kong: *https://www.ogcio.gov.hk/en/our_work/regulation/eto/ordinance/ca_in_hk/*
- Let's Encrypt
  - Free of charge, supported by sponsors
  - DV only – Fully automated
- Self-signed certificates / private CA
  - Browsers need to trust the certificate manually
- *Read more: https://www.digitalocean.com/community/tutorials/a-comparison-of-let-s-encrypt-commercial-and-private-certificate-authorities-and-self-signed-ssl-certificates*

DDOS

- Distributed Denial-of-Service attack
  - Exhausting the resource of the target, e.g., consuming all the available bandwidth, or computation power
  - Distributed: not a single source of attack, usually using botnets
- Layer 7 DDoS
  - Flooding with application requests (e.g., HTTP)
- Layer 3 or 4 DDoS
  - Protocol attacks (e.g., SYN flood)
  - Volumetric attacks (e.g., DNS amplification)
- *See: https://www.cloudflare.com/en-gb/learning/ddos/what-is-a-ddos-attack/*

CLOUD SOLUTIONS FOR DDOS

- Distributed and intelligent systems to mitigate attacks
  - *e.g.*, Cloudflare, AWS Shield, Nexusguard



See: https://www.cloudflare.com/ddos/

A LOT OF HARD WORK AHEAD…

- The Internet evolves with improving concern on security
- Cyber security depends heavily on
  - *The developers*
  - *The system administrators*
  - *The users*
- Wish you good luck!
- Check out OWASP Cheatsheets: *https://cheatsheetseries.owasp.org*

READ FURTHER…

All links on “OWASP Top Ten”

https://owasp.org/Top10/

10 Most Common Web Security Vulnerabilities

https://www.toptal.com/security/10-most-common-web-security-vulnerabilities

Web Security on MDN

https://developer.mozilla.org/en-US/docs/Web/Security