(5th November 2025) - JAGUAR COHORT
Made by Chris Cownden
I'm creating recap guides as revision notes for future use. Please use this document as you see fit.

7th Lab Recap Guide: Security+ (Session Cookies, ACRONYMS)

Recording is here

Timestamps

00:00 "Breaking News: Wormable Malware Targeting GitHub Repos"

19:38 "Kerberos Explained Through Fairground Analogy"

22:31 "Vulnerability Protocols and Authentication"

38:19 "Cookie-Based Authentication Explanation"

42:41 "Understanding JSON Web Tokens"

01:13:00 "Cryptography: A Classified Weapon"

01:15:26 "Exam Practice: Understanding Hashing Algorithms"

01:24:12 "Salting & Password Hashing"

01:41:47 "Secure Key Transfer Basics"

01:55:27 "Root CA and Cybersecurity Risks"

01:57:51 "Key Management and Rotation"

Recommended Resources

- Burp Suite - Download Community version for free
- OWASP Juice Shop
- GitHub
- Jon's GitHub Security Architecture Course
- Exam Compass = Practice questions

(5th November 2025) - JAGUAR COHORT

Made by Chris Cownden

I'm creating recap guides as revision notes for future use. Please use this document as you see fit.

- Cyber Chef
- JWT.io – Decode, inspect, and debug JSON Web Tokens online
- NVD (National Vulnerability Database)
- Reverse MD5 Hashing
- Contact Student Support to see if you can get access to the new Cyber Academy.

Other Recordings To Date

Date	Order	Recordings	Lab Recap Guides
26th November 2025	Lab 7	RECORDING	THIS ONE
19th November 2025	NO SESSION AS JON WAS ABSENT		
12th November 2025	Lab 6	RECORDING	RECAP GUIDE
5th November 2025	Lab 5	RECORDING	RECAP GUIDE
2nd November 2025	3rd Sunday	RECORDING	
29th October 2025	Lab 4	RECORDING	RECAP GUIDE
22nd October 2025	Lab 3	RECORDING	RECAP GUIDE
19th October 2025	2nd Sunday	RECORDING	
15th October 2025	Lab 2	RECORDING	RECAP GUIDE
8th October 2025	Lab 1	RECORDING	RECAP GUIDE
5th October 2025	1st Sunday	RECORDING	

Core Concepts

Repository

- **Definition:** A central location for code (GitHub Repository, NPM Repository).
- **Why it matters:** Malware can spread via infected repositories; understanding secure code storage is essential.
-

(5th November 2025) - JAGUAR COHORT
Made by Chris Cownden
I'm creating recap guides as revision notes for future use. Please use this document as you see fit.

Security Architecture

- **Secure by Design:** Build security into products from the project's start, not as an afterthought.
- **Role:** Security Architecture is highly in-demand; often follows Penetration Testing.

Digital Certificates

- **Purpose:** Prove identity and secure communication.
- **Acronyms to remember:**
    - **CRL (Certificate Revocation List):** List of invalid certificates.
    - **OCSP (Online Certificate Status Protocol):** Fastest way to check certificate validity.
    - **CSR (Certificate Signing Request):** Submitted to get a certificate signed.

Encryption Types

Symmetric Encryption

- **Uses the same key for encryption and decryption (e.g., AES, DES).**
- **Pro:** Fast for bulk data (used for disk volumes).
- **Con:** Key distribution problem (how to share the key safely).

Asymmetric Encryption

- **Uses a public/private key pair (e.g., RSA, ECC).**
- **Pro:** Enables digital signatures, solves key distribution issue.
- **Con:** Slower for bulk data.

Hybrid Encryption Order (TLS/HTTPS Example)

1. Generate a symmetric session key.
2. Encrypt session key with the recipient's public key.
3. Recipient decrypts with private key.
4. Encrypt/decrypt data with the symmetric key for efficiency.

Major Algorithms

- **AES (Advanced Encryption Standard):** 128, 192, 256 bits
- **DES/Triple DES:** Older, broken, smaller key space.
- **RSA:** Asymmetric; good for public key cryptography.
- **ECC:** Lightweight public key cryptography (higher efficiency).

- **SHA, MD5:** Hashing algorithms for integrity. SHA2/SHA3 are considered secure; MD5/SHA-1 are collidable and no longer reliable.

Authentication Protocols

- **Kerberos:** Uses port 88, issues Ticket Granting Tickets (TGTs) for single sign-on.
- **CHAP:** Challenge Handshake Authentication Protocols; know vulnerabilities (replay attacks, session hijacking).
- **PEAP/LEAP:** Extensible authentication protocols, mainly for wireless networks.

Network & Web Attacks

CSRF (Cross-Site Request Forgery)

- **What:** Attacker tricks an authenticated victim into performing unauthorised actions.
- **How to prevent:** CSRF tokens, browser security improvements.
- **Difference from XSS:** CSRF exploits authentication/session, XSS injects malicious code.

ARP (Address Resolution Protocol)

- **Purpose:** Maps IP address to MAC address at Layer 2/3 of the OSI model.

Hashing & Salting

- **Hashing**
  - Hashing is: It is a One-way cryptographic function for integrity.
  - Hashing is NOT: It is not Encryption. It doesn't maintain confidentiality.

- **Salting:**
  - Adds random data to hashes to combat rainbow tables and avoid identical hashes for identical data.
- **Rainbow Tables:**
  Precomputed hash tables for cracking passwords salting breaks their efficiency.

PKI (Public Key Infrastructure)

- **Definition:** The infrastructure underlying digital certificates (Root CA, Intermediate CA).
- **Exam Tip:** The root CA is your "anchor of trust"—compromise here breaks all trust.

CIA Triad

- **Confidentiality:** Who can read the data (encryption).
- **Integrity:** Is data unmodified (hashes).
- **Availability:** Can authorised users access the data (backups, redundancy).

Digital Signatures

- **Process:** Hash data, sign with your private key. Provides integrity & non-repudiation.

Steganography

- **Use:** Hide data within other files/media.

Tokenization/Data Masking

- **Tokenization:** Replace sensitive data with reversible tokens.
- **Data Masking:** Obscure sections of sensitive data (e.g., credit card numbers).

Terminology

Repository: A central location where code is stored and managed (e.g., GitHub, NPM).
Node Package Manager (NPM): Online tool for managing and sharing JavaScript code packages.
Malware: Malicious software designed to infect, damage, or exploit systems.
API Key: A code passed in by computer programs to identify the calling program, its developer, or its user.
Secrets: Sensitive data (passwords, keys, etc.) used in code or environments, requiring protection.
Forking: Copying a repository from one account to another for independent development.
Secure by Design / Shift Left Security: Building security into a product from the start of the project.
CRL (Certificate Revocation List): A published list of revoked digital certificates.
OCSP (Online Certificate Status Protocol): Protocol for checking real-time status of a digital certificate.

(5th November 2025) - JAGUAR COHORT

Made by Chris Cownden

I'm creating recap guides as revision notes for future use. Please use this document as you see fit.

Packet Capture (PCAP): The process of intercepting and recording network traffic for analysis.
Kerberos: Network authentication protocol using tickets for secure access.
Ticket Granting Ticket (TGT): The credential provided to a user for accessing multiple Kerberos services.
One Time Password (OTP): Password valid for a single authentication session or transaction.
Hashing: Converting data to a fixed-length output to ensure data integrity.
Symmetric Encryption: Uses the same key for both encryption and decryption.
Asymmetric Encryption: Uses a public and private key pair for secure communications.
Salt: Random data added to inputs before hashing, to prevent precomputed attacks (rainbow tables).
Rainbow Table: A large precomputed table for reversing cryptographic hash functions, used in password cracking.
Public Key Infrastructure (PKI): A system governing the creation, management, and validation of digital certificates.
Digital Signature: Cryptographic value that validates the origin and integrity of a digital message or document.
Data Masking: Hiding parts of sensitive data fields for privacy (e.g., showing only last four digits of a credit card).
Tokenization: Replacing sensitive data with non-sensitive substitutes, often reversible.
Steganography: Concealing data within another file, image, or medium.
Availability: Ensuring data and services are accessible when needed, often maintained through redundancy or backups.
Collision: When two different inputs produce the same hash output, a major vulnerability in weak hashing algorithms.
Perfect Forward Secrecy (PFS): Ensures session keys are not compromised even if the private key is.

Acronyms

CRL = Certificate Revocation List
OCSP = Online Certificate Status Protocol
OSPF = Open Shortest Path First
CSR = Certificate Signing Request
AIS = Automatic Indicator Sharing (or Automatic Identification System

PCAP = Packet Capture
EDR = Endpoint Detection and Response
OTP = One Time Password
AS = Autonomous System
SAML = Security Assertion Markup Language
XML = Extensible Markup Language
SOAP = Simple Object Access Protocol

(5th November 2025) - JAGUAR COHORT

Made by Chris Cownden

I'm creating recap guides as revision notes for future use. Please use this document as you see fit.

JSON = JavaScript Object Notation
PEAP = Protected Extensible Authentication Protocol
MSCHAP = Microsoft Challenge Handshake Authentication Protocol
LEAP = Lightweight Extensible Authentication Protocol
CHAP = Challenge Handshake Authentication Protocol
SOW = Statement of Work
MOU = Memorandum of Understanding
MSA = Master Service Agreement
MOA = Memorandum of Agreement
CSRF = Cross Site Request Forgery
ARP = Address Resolution Protocol
SAN = Subject Alternative Name
CN = Canonical Name (Common Name)
PKI = Public Key Infrastructure
HSM = Hardware Security Module
TPM = Trusted Platform Module
PFS = Perfect Forward Secrecy
ECC = Elliptic Curve Cryptography
DES = Data Encryption Standard
AES = Advanced Encryption Standard
SHA = Secure Hash Algorithm
MD5 = Message Digest 5
PGP = Pretty Good Privacy
GPG = GNU Privacy Guard

Analogies

Kerberos

Imagine you show up at the gates of the local fairground, and you want to enjoy ALL the rides—the roller coaster, the waltzers, and the ghost train. You don't want to buy a separate ticket at each ride (that would be a pain).

Instead, you visit the main ticket booth (the Kerberos Ticket Granting Service, or TGS). You prove who you are (maybe with ID and cash—think of this as your credentials). If everything checks out, they give you a special wristband—let's call it your Ticket Granting Ticket (TGT).

Now, as you walk around the fairground, whenever you want to hop on a ride, you just flash your wristband. The ride operator knows it's legit, and you're good to go—no need to show your ID or pay again each time!

(5th November 2025) - JAGUAR COHORT

Made by Chris Cownden

I'm creating recap guides as revision notes for future use. Please use this document as you see fit.

The magic:

- The ticket booth is the Kerberos Ticket Granting Service (TGS).
- The wristband is the Ticket Granting Ticket (TGT).
- The process saves you the hassle of authenticating again and again—once you're in, you're in!

So, Kerberos is just like getting a wristband at a fairground. Once you're trusted, you can access all the cool stuff without wasting time (or money) at every stop!

Repository

A repository is like a central code library (think GitHub), but don't get it confused with a suppository! One is for storing code, the other… well, let's just say it's a pain in the backside in a whole different way. Both can be a headache, but only one helps build websites!

Hashing

- You start with an egg (plain text data).
- You crack it, scramble it, cook it, eat it, and, well, eventually pass it on…
- The point? Once you've made an omelette from an egg (created a hash from the data), there's no way to turn it back into the original egg. It's a one-way process—just like cryptographic hashes (MD5, SHA)—you can't un-hash back to the original data.

Cookies

Imagine you're Neo, the hero from The Matrix, and you're trying to get some wisdom from the Oracle. But before she'll talk to you, she asks you to hand her a cookie. Why a cookie? Well, in the context of web security, that "cookie" is exactly what websites use to identify and authenticate you.

When you log in to a website, your browser receives a little piece of data a "cookie" from the web server. This cookie is proof that you've passed the test (entered a valid username and password), so future interactions are smooth. You don't have to keep proving yourself every time you browse a new page; you just present the cookie automatically.

It's a lot like the Oracle requiring Neo to give her a cookie before he can get any answers. The web server is the Oracle; the cookie in your browser is—well—a "magic cookie." No cookie means no special access. With a cookie, you're recognized, welcomed, and given what you need.

In summary:

- **Neo:** You
- **The Oracle:** The web server.
- **Handing over a cookie:** Presenting proof (the authentication token) you're trusted.
- **The result:** You are given personalised access (wisdom from the Oracle, or in web context, your data, account, etc).

General Q&A from the session

1. **What is the fastest way for checking the validity of a digital certificate?**
    a. **Answer:** OCSP
        i. **Note:**
            - CRL is a list of revoked (bad) certificates.
            - OCSP checks certificate validity more quickly/efficiently than just consulting the CRL.

2. **Which of the terms listed below refers to a process of intercepting network traffic data for analysis and troubleshooting processes or purposes?**
    a. **Answer:** PCAP (Packet Capture file)
        i. **Notes:**
            - Wireshark, TCPDump, Tshark are examples of packet capture tools
            - PCAP is a file format used to store network traffic that's been recorded (captured) from a device or network

3. **Who knows what port Kerberos uses?**
    a. **Answer:** Port 88
        i. Notes:
            - Uses the mnemonic: "88 looks like a two-headed dog … Cerberus is the multi-headed dog of the underworld."

4. **Which of the following answers refers to a language primarily used for automating the assessment of security vulnerabilities and configuration issues on computer systems?**
    a. **Answer:** OVAL (Open Vulnerability Assessment Language)
        i. Notes:
            - SCAP (Security Content Automation Protocol) relates to automating security compliance and vulnerability scanning.
            - SAML (Security Assertion Markup Language) is used in authentication/authorization, not security scanning.

- XML and SOAP are data/configuration or legacy API languages (not vulnerability assessment languages).

5. **Which of the acronyms listed below refers to a formal and legally binding document that specifies detailed terms, obligations, responsibilities of all parties involved?**
    a. **Answer:** MOU (Memorandum of Understanding)

6. **Which of the following answers refers to CSRF?**
    a. **Answer:** CSRF = Cross-Site Request Forgery
        i. **Notes:**
            - Targets an authenticated user.
            - Attacker crafts a malicious HTTP request, tricks the user into clicking it, and the user's browser sends it with their valid session cookie, causing actions the attacker desires (example: sending money to a different account).

7. **How is that different from Cross-Site Scripting (XSS)?**
    a. **Answer:** CSRF: Leverages social engineering, targets an authenticated user, uses the user's browser and session cookie. XSS: Involves injecting JavaScript into a webpage, rendered by the browser, typically exploits input validation/sanitization failures.

8. **What does ARP mean?**
    a. **Answer:** Address Resolution Protocol
        i. **Notes:**
            - ARP maps IP addresses to MAC addresses (Layer 3 to Layer 2), not MAC to IP addresses

9. **In context of cryptography, what are the three things we're trying to ensure with security always?**
    a. **Answer:** CIA: Confidentiality, Integrity, Availability
        i. **Notes:**
            - How do we ensure Confidentiality? Encryption, Access controls
            - How do we ensure Integrity? Hashing (with Cryptography)
            - How do we ensure Availability? Redundancy, Backups, Failover systems

(5th November 2025) - JAGUAR COHORT
Made by Chris Cownden
I'm creating recap guides as revision notes for future use. Please use this document as you see fit.

10. What is plain text?

a. **Answer:** Unencrypted, human-readable data

11. What is ciphertext?

a. **Answer:** Encrypted text; not human-readable

12. What is symmetric encryption?

a. **Answer:** Uses the same key for both encryption and decryption. Is fast, suitable for large data (e.g., disk encryption, HTTPS traffic) Downside: "Key distribution problem" (How to securely share the key?)

13. What is asymmetric encryption?

a. **Answer:**
   i. Uses a key pair: public and private keys
   ii. Public key: available to everyone
   iii. Private key: only to owner
   iv. Slower than symmetric; used for secure key exchange, digital signatures
   v. Example algorithms: RSA, ECC

14. Examples of symmetric encryption algorithms?

a. **Answer:**
   - AES (Advanced Encryption Standard):
     - AES-128, AES-192, AES-256
   - DES (Data Encryption Standard):
     - 56-bit (weak, obsolete)
   - 3DES (Triple DES):
     - 168-bit (applies DES three times)

15. Examples of asymmetric encryption algorithms?

a. **Answer:** RSA, ECC

16. What is hashing NOT?

a. **Answer:**
   i. Hashing is NOT encryption
   ii. It's a one-way function—cannot be reversed
   iii. Used to guarantee integrity, not confidentiality

17. What is a hash collision?

a. **Answer:**
   i. When two different inputs produce the same hash output (bad in cryptography—look for algorithms like SHA-2, SHA-3 that minimize collisions)

18. How do we protect passwords beyond hashing?

a. **Answer:** Salting (add randomness to input) Slows down brute-force and rainbow table attacks

19. Examples of hashing algorithms?

a. **Answer:**
   i. MD5 (now obsolete/broken; 128-bit output)
   ii. SHA-1 (compromised, collision-prone; 160-bit output)
   iii. SHA-256 (part of SHA-2 family; more secure)
   iv. Bcrypt (common for password hashing, especially in web applications)

20. What is PKI?

a. **Answer:** The certificate infrastructure that supports asymmetric encryption: Includes root CAs, intermediate CAs, certificate signing, CRL & OCSP checking, certificate management

21. What is a digital signature?

a. **Answer:** Hash a piece of data and sign it with your private key. Provides integrity and non-repudiation (only the owner of the private key could have signed it)

22. What is salting (in cryptography)?

a. **Answer:** Add random characters to a password before hashing to ensure no two hashes for the same password are the same. Implemented by OS or applications automatically

23. What are rainbow tables?

a. **Answer:** Precomputed tables of hash outputs for rapid password cracking. Salting defeats rainbow tables because each hash is unique even for identical passwords

24. What does increasing key length achieve in encryption?

a. **Answer:** Expands keyspace, resists brute-force attacks

25. What is perfect forward secrecy (PFS)?

a. **Answer:** Utilises ephemeral (unique, single-use) keys, so past communications remain secure even if the server's private key is later compromised