Foundations & assessment

Certified Ethical Hacker v12

KNOWLEDGE CHEATSHEET

SECURITY DECISION CYCLE

Scope Authorize → **Assess** Prioritize → **Treat** Verify → **Monitor** Respond

01 Security foundations and assessment

Authorization, risk and attack models

- **Ethical hacking** requires explicit authorization, defined targets, permitted methods, timing and stop conditions. Helpful intent or later disclosure does not replace permission. Bug bounty programs authorize only their published scope.
- **Confidentiality, integrity and availability** describe different security objectives. Authentication establishes identity; authorization controls actions; accountability connects actions to identities through reliable records.
- **Threats** can exploit **vulnerabilities** to create business risk. Prioritize using exposure, exploitability, asset value and impact. Defense in depth combines preventive, detective and response controls; one patched flaw does not eliminate residual risk.
- **Cyber Kill Chain:** reconnaissance, weaponization, delivery, exploitation, installation, command and control, actions on objectives. Creating a malicious attachment is weaponization; sending it is delivery; triggering it is exploitation; exfiltration serves objectives.
- **APTs** pursue sustained objectives through footholds, persistence, internal discovery, lateral movement, staging and exfiltration. They can use stolen credentials and known flaws; a zero-day is not a defining requirement. Insider threats exploit legitimate access or organizational knowledge.

Vulnerability assessment and management

- **Management lifecycle:** inventory and baseline assets, scan, assess risk, remediate, verify fixes, then monitor continuously. Discovering hosts commonly precedes service identification and relevant vulnerability tests; discovery filters can miss reachable services.
- **External assessments** examine exposed assets; internal assessments examine access from inside. Host assessments inspect local settings and permissions. Credentialed scans reveal configurations and patch state unavailable to unauthenticated scans.
- **Active scanning** sends probes; passive assessment observes traffic. Agents inspect their installed hosts; network scanners query reachable services. Inference-based testing selects checks from discovered services; tree-based testing follows predetermined branches.
- **Nessus, Qualys, OpenVAS and GFI LanGuard** support vulnerability assessment. Findings depend on signatures, credentials, reachability and timing. Validate important results; a banner match alone is not proof of exploitability or business impact.

Foundations & assessment / Reconnaissance & enumeration

Certified Ethical Hacker v12

KNOWLEDGE CHEATSHEET

SECURITY DECISION CYCLE

Scope Authorize → **Assess** Prioritize → **Treat** Verify → **Monitor** Respond

- **CVSS v3.x** base metrics describe inherent technical severity; temporal metrics reflect changing conditions; environmental metrics adapt scoring to local context. Ratings: none 0.0; low 0.1-3.9; medium 4.0-6.9; high 7.0-8.9; critical 9.0-10.0.
- A useful **finding** includes affected assets, evidence, vulnerability/CVE identification where applicable, impact and remediation. Reports also summarize overall risk. Retest fixes and track exceptions with owners; patch management includes identifying, deploying and verifying applicable updates.

Response, intelligence and governance

- **Incident response** prepares, detects and analyzes, contains, eradicates, recovers and learns. Triage establishes severity, scope and priority. Preserve relevant evidence while containing harm; restoring service alone does not remove persistence or exposed credentials.
- **Threat intelligence:** strategic informs business decisions; tactical describes adversary techniques; operational explains particular campaigns and intentions; technical supplies indicators such as hashes, domains and IPs. Validate context and age before blocking indicators.
- **HIPAA** concerns protected health information; **PCI DSS** concerns payment-card data environments; **SOX** concerns corporate financial reporting and controls. PII describes a data category. **ISO/IEC 27001** specifies an information security management system.
- Before enabling **audit logging**, assess capacity, privacy, performance, retention and review responsibilities. Centralize and protect logs, synchronize clocks and alert on meaningful events; collecting logs without review gives limited detection value.

02 Reconnaissance and enumeration

Public information and infrastructure mapping

- **Passive reconnaissance** uses already available information; active reconnaissance interacts with target systems. Public DNS queries can still be logged. Website mirroring downloads linked content; footprinting builds a broader technology and infrastructure profile.
- **WHOIS/RDAP** provides registration and allocation information subject to privacy restrictions. A thin registry record can refer to a registrar for further detail. RIR regions: ARIN North America; RIPE NCC Europe/Middle East; APNIC Asia-Pacific; LACNIC Latin America/Caribbean; AFRINIC Africa.
- **DNS records:** A/AAAA map names to IPv4/IPv6; PTR provides reverse mapping; MX identifies mail exchangers; NS identifies authoritative servers; CNAME creates an alias; SOA carries zone administration data. Reverse lookup is not equivalent to listing every hosted domain.

Reconnaissance & enumeration

Certified Ethical Hacker v12

KNOWLEDGE CHEATSHEET

FROM PUBLIC CLUES TO SERVICES

OSINT Public records → **DNS** Names & addresses → **Services** Enumerate → **Evidence** Correlate

- **AXFR** transfers zone data and should be restricted to authorized secondary servers. DNSSEC zone walking enumerates exposed signed denial records under applicable configurations. Cache snooping infers cached names; cache poisoning introduces false DNS data.
- **SecurityTrails** supports DNS-history research; **Sublist3r** discovers subdomains; **theHarvester/Infoga** gather public identity and email information; **Maltego** maps relationships; **Bluto/DNSRecon** support DNS reconnaissance. **Censys** indexes exposed internet services; **FCC ID searches** identify device filings.

Web and email footprinting

- Search operators such as **site:**, **inurl:**, **intitle:** and **filetype:** narrow public information searches. Older exam terminology includes related-site and backlink operators; search-engine results and operator support are not complete asset inventories.
- **robots.txt** expresses crawler preferences and can reveal paths; it is not access control. Sitemaps, HTML, scripts, response headers and public archives expose additional structure. User-directed spidering captures paths reached through manual browsing.
- **Banner grabbing** inspects protocol responses to infer services and versions. HTTP headers can be viewed with a header-capable client; wget's **-S** displays server responses while **-q** suppresses routine output. Hidden or modified banners reduce confidence.
- **CeWL** builds wordlists from website text; **Gobuster** enumerates candidate directories, virtual hosts or DNS names using the appropriate mode. Wordlists, extensions, concurrency and authorization determine coverage and load; ignoring certificate errors does not reveal more content.
- **Email headers** reveal routing through Received fields, but untrusted fields can be forged. Tracking pixels have privacy and proxy limitations; they do not reliably prove who read a message or for how long. Reverse image search can help trace publicly reused images.
- **Traffic analytics** measures visits and usage patterns; website change monitors detect content changes; WAF fingerprinting identifies protective technology. Proxies, VPNs and Tor alter network visibility for reconnaissance, including dark-web research, but cannot guarantee anonymity or access authorization.
- **Traceroute** uses increasing TTL/hop limits and returned responses to infer paths. UDP, ICMP and TCP probes encounter different filtering; even UDP/TCP traceroute may rely on ICMP time-exceeded replies. Missing hops do not prove a host is down.

Reconnaissance & enumeration / Network scanning

Certified Ethical Hacker v12

KNOWLEDGE CHEATSHEET

FROM PUBLIC CLUES TO SERVICES

OSINT Public records → **DNS** Names & addresses → **Services** Enumerate → **Evidence** Correlate

Service enumeration

- **SMTP** commonly uses TCP 25. VRFY can validate users, EXPN can expand lists, and RCPT TO responses may reveal recipients. Restrict unnecessary commands and use consistent responses to reduce enumeration without breaking mail delivery.
- **NetBIOS** uses UDP 137/138 and TCP 139; direct-hosted **SMB** uses TCP 445. Legacy suffixes include 00 workstation, 03 messenger, 20 server and 1B domain master browser. **nbtstat -a** queries names by hostname; **-A** uses an IP; **-c** shows the local name cache.
- **IPC$** supports Windows interprocess communication. Historical null sessions could expose users and shares where permitted. Restrict anonymous enumeration and SMB access. NetBIOS over TCP/IP is an IPv4-era mechanism; use appropriate IPv6-capable SMB discovery for IPv6 environments.
- **LDAP** uses TCP 389; TLS may be negotiated with StartTLS there or provided through LDAPS on TCP 636. Validate certificates and require appropriate authenticated binds and directory permissions. Transport encryption alone does not disable anonymous queries.
- **SNMP** uses UDP 161 for queries and usually UDP 162 for traps. MIBs define managed objects addressed by OIDs. Read operations such as GET/WALK enumerate; SET modifies values. Legacy community strings such as public/private are weak shared secrets.
- **SNMPv3 authPriv** supplies authentication and privacy when configured correctly. **NTP** uses UDP 123; ntptrace follows upstream time sources. **NFS** commonly uses 2049; RPC-related enumeration can also involve 111. Restrict management protocols to trusted administration networks.

03 Network scanning

Discovery and TCP behavior

- **Host discovery** asks whether a host responds; port scanning classifies ports; version detection identifies services; vulnerability assessment tests weaknesses. A silent response can result from filtering, loss or a disabled protocol rather than an inactive host.
- **Nmap -sn** performs discovery without a port scan. **-PS** sends TCP SYN discovery probes, **-PA** ACK probes, **-PU** UDP probes and **-PP** ICMP timestamp probes. **-Pn** skips discovery and attempts scans of specified targets.
- **ARP discovery** is useful on a local IPv4 broadcast domain and does not traverse routers. Routed networks need other probes. Combine permitted techniques with inventory data; no ping method guarantees discovery through every firewall.

Network scanning

Certified Ethical Hacker v12

KNOWLEDGE CHEATSHEET

DISCOVERY AND INTERPRETATION

Host Reachability → **Port** Response state → **Service** Version clues → **Limits** Filter & loss

- TCP setup is **SYN, SYN/ACK, ACK**. **FIN** requests orderly closure; **RST** aborts a connection; **PSH** requests prompt delivery to the receiving application. These flags describe transport behavior rather than application authentication.
- **-sS SYN scanning** usually treats SYN/ACK as open and RST as closed, then resets without completing the handshake. **-sT connect scanning** completes an OS-managed connection. Both are detectable; half-open does not mean invisible.

Scan interpretation and fingerprinting

- **-sA ACK scanning** maps filtering: RST suggests unfiltered, while silence or selected ICMP errors suggest filtered. It does not distinguish open from closed ports or conclusively establish firewall architecture by itself.
- **-sN NULL**, **-sF FIN** and **-sX Xmas** probes exploit differing TCP-stack responses. Closed ports often return RST; silence may mean open or filtered. Noncompliant stacks and firewalls can make these scans unreliable.
- **-sM Maimon** sends FIN/ACK and historically distinguished ports on some BSD-derived stacks. **-sU** scans UDP; ICMP port-unreachable commonly indicates closed. A missing UDP reply can mean open or filtered. **-sY** performs SCTP INIT scanning.
- **-sI idle scanning** infers target replies from a suitable zombie's predictable IPv4 ID counter. Additional counter increments can indicate an open port under the scan model; unrelated traffic, random IDs and filtering invalidate simple conclusions.
- **-sV** probes service versions; **-O** performs OS fingerprinting; **-A** enables multiple detection features. TTL, window size, TCP options and responses provide clues. Typical initial TTLs include 64, 128 and 255; hops and configuration prevent definitive OS identification from TTL alone.

Scan controls and detection limits

- **-p** selects ports; **-p-** covers the TCP/UDP port range for the chosen scan type. **-T0 through -T5** change timing; faster scans increase load and may lose accuracy. **--scan-delay** and concurrency controls limit probe pressure.
- **-D** adds decoy sources; **-f** fragments applicable packets; **-g** selects a source port for supported probes. These features do not guarantee IDS evasion. Decoys are distinct from a usable bidirectional connection through a spoofed source address.
- **hping3** crafts network probes; **-a** specifies a spoofed source address. Replies normally return to that address, limiting visibility. Nmap output flags such as **-oN/-oX** select formats; **-n** disables DNS lookup rather than concealing probes.
- Match authorized tests to the environment. Slow or low-parallelism scans may reduce load but do not establish safety for industrial devices. Confirm scope, maintenance windows and recovery options before any active OT testing.

Systems & malware

Certified Ethical Hacker v12

KNOWLEDGE CHEATSHEET

COMPROMISE AND PERSISTENCE

Credentials Access → **Privileges** Escalate → **Movement** Reach hosts → **Persistence** Maintain

04 System hacking and malware

Password and authentication attacks

- **Offline cracking** tests candidates against acquired hashes without server lockouts; Hashcat and John the Ripper support this work. **Online guessing** interacts with an authentication service; Hydra supports many protocols. Netcat provides generic network I/O rather than a dedicated hash-cracking engine.
- **Dictionary attacks** try wordlists; brute force explores combinations; hybrid/rule-based attacks transform candidates with suffixes or substitutions. **Password spraying** tries a few common passwords across many accounts; credential stuffing reuses leaked username/password pairs.
- **Rainbow tables** precompute hash relationships and are tied to the hash scheme. Unique random salts defeat reuse across accounts; a slow password KDF increases guessing cost. Salting and stretching do not make a weak password unpredictable.
- **Windows SAM** stores local credential material; domain credentials are managed separately in AD. Legacy **LM** hashes uppercase and split passwords into short portions and may be disabled or absent. Blank LM fields do not uniquely identify OS version or password length.
- **Pass-the-hash** uses supported NTLM credential material without recovering plaintext. **Pass-the-ticket** reuses Kerberos tickets. **Internal Monologue** abuses local authentication interfaces and security settings to obtain crackable challenge-response material; these are distinct credential techniques.
- **Kerberoasting** requests an SPN's service ticket using authenticated domain access, then attacks service-account key material offline. Use long random managed service-account secrets, suitable encryption and monitoring; rotate exposed account secrets and investigate scope rather than assuming a reboot revokes stolen material.

Privilege escalation and persistence

- **Privilege escalation** gains higher rights on a host; lateral movement reaches other systems. Review service accounts, ACLs, scheduled tasks, registry permissions, unneeded administrators and debug privileges. A vulnerability must be reachable under the attacker's existing access.
- **DLL hijacking** abuses unsafe library search paths or writable library locations. Use fully qualified trusted paths and secure directories. **Named-pipe impersonation** abuses permitted impersonation behavior; **misconfigured services** expose writable binaries, paths or service configuration.
- In **Meterpreter**, **getuid** reports identity, **getsystem** attempts supported elevation techniques, and routing helpers support pivoting. Privilege escalation depends on permissions and target conditions; a command name does not guarantee SYSTEM access.

Systems & malware

Certified Ethical Hacker v12

KNOWLEDGE CHEATSHEET

COMPROMISE AND PERSISTENCE

Credentials Access → **Privileges** Escalate → **Movement** Reach hosts → **Persistence** Maintain

- **Persistence** can involve startup entries, services, scheduled tasks, accounts or firmware. A Linux filename beginning with a period is hidden by common listings, not protected. Shell history can expose typed secrets; prevent secret entry in commands and preserve incident evidence.
- **TOCTOU races** occur when a resource changes between checking and use. Atomic operations, safe file handles and locking reduce the gap. **Spectre** abuses speculative execution side effects; **Meltdown** and **Rowhammer** involve different hardware mechanisms.

Malware types and propagation

- **Viruses** depend on infected hosts/files for replication; **worms** self-propagate. Trojans masquerade as useful software; spyware observes users; keyloggers record input; ransomware extorts through disruption or data exposure; rootkits hide activity or privileged access.
- **Fileless malware** relies heavily on memory, scripts or trusted tools and can still leave logs and persistence artifacts. Application allowlisting alone may not stop abuse of permitted interpreters. Monitor process ancestry, script activity and unusual outbound connections.
- **Polymorphic malware** changes its representation, often through encryption and a varying decoder. **Metamorphic malware** rewrites its body while preserving behavior. Code infection, stealth and persistence are separate traits; no single label implies every capability.
- **Botnets** coordinate compromised devices for objectives such as DDoS. Hit-list propagation starts from known candidates; topological propagation follows discovered relationships; subnet scanning favors local ranges; permutation scanning coordinates address exploration.
- **Emotet** illustrates malicious attachments, credential theft and lateral spread; **Mirai** illustrates vulnerable IoT recruitment and DDoS. Reduce exposure with unique credentials, patching, restricted shares, controlled scripting and egress monitoring rather than relying on one malware-specific port.

Safe analysis and endpoint detection

- A **sheep-dip** system screens suspect media or files in isolation. Dynamic malware analysis needs a controlled testbed, snapshots and restricted networking; simulated services such as INetSim provide observations without exposing production systems.
- **Static analysis** inspects code, strings and metadata without execution; dynamic analysis observes processes, files, registry, network and persistence behavior. Connection tools such as netstat provide clues, but quiet malware or a clean antivirus scan does not prove a system is safe.

Systems & malware / Network attacks & detection

Certified Ethical Hacker v12

KNOWLEDGE CHEATSHEET

LAYERED NETWORK DEFENSE

- **Link** — ARP & switching
- **Transport** — State & floods
- **Application** — DNS & HTTP
- **Telemetry** — Detect & respond

- **YARA** matches patterns in files or memory; Snort rules primarily inspect network traffic. **yarGen** assists rule creation using malware strings while filtering common goodware strings. Test rules against benign and malicious samples before deployment.
- **EDR** correlates endpoint behavior and supports investigation and containment. Verify unexpected files even when sent by a known contact. Software updates, attachment controls, restricted execution, awareness and recoverable backups address different parts of the attack chain.

05 Network attacks and monitoring

Sniffing and local-network manipulation

- **Passive sniffing** observes traffic visible at an interface. Switched Ethernet normally limits unrelated unicast visibility; authorized SPAN/TAP capture supplies a defined observation point. Active redirection or a compromised endpoint can expose additional traffic.
- **ARP spoofing** supplies false IPv4-to-MAC bindings, enabling interception, modification or disruption. Unexpected gateway MAC changes and conflicting ARP claims are clues; duplicate MAC observations alone are not conclusive. Use dynamic ARP inspection with trusted bindings and encrypted authenticated sessions.
- **MAC flooding** pressures a switch's CAM table and may cause unknown-unicast flooding depending on implementation. Port security limits learned MACs. **STP manipulation** advertises a superior bridge ID to influence root selection; use root guard and BPDU guard where appropriate.
- **DHCP starvation** exhausts leases with forged requests. A **rogue DHCP server** offers attacker-selected gateway or DNS settings. DHCP snooping, trusted uplinks, rate limits and port controls address these distinct problems.
- **VLAN hopping** can exploit switch negotiation or double tagging under suitable configurations. Disable unnecessary trunk negotiation, explicitly configure access ports and control native VLANs. Changing to IPv6 replaces ARP with Neighbor Discovery; it does not eliminate spoofing or sniffing risks.

Hijacking, DNS abuse and evasion

- **On-path attacks** insert an intermediary into communication; forged ARP or routing-related messages may help redirect traffic. Blind TCP hijacking guesses state without observing it. Forged **RST** packets can disrupt a connection when accepted sequence checks match.
- **DNS hijacking/poisoning** redirects names toward attacker-controlled destinations; **pharming** describes redirection toward fraudulent sites. **DNS tunneling** carries commands or data inside DNS traffic. DNSSEC authenticates DNS data; it does not inspect tunneled content for malicious intent.
- Multiple domains can point to one host to provide fallback names. **Fast flux** changes address mappings rapidly; **domain generation algorithms** create candidate names. These behaviors differ from tunneling, staging data and ordinary use of a command-line interface.

Network attacks & detection

Certified Ethical Hacker v12

KNOWLEDGE CHEATSHEET

LAYERED NETWORK DEFENSE

- **Link** — ARP & switching
- **Transport** — State & floods
- **Application** — DNS & HTTP
- **Telemetry** — Detect & respond

- **Obfuscation** changes representation through encoding, case, whitespace or fragmentation. Session splicing divides content across packets. Normalize and reassemble traffic consistently with endpoints; signature-only inspection can miss semantically equivalent requests.
- Prefer **SSH, HTTPS, SFTP or FTPS** over plaintext equivalents. SFTP runs over SSH; FTPS protects FTP using TLS. VPNs protect traffic within their tunnel; endpoint compromise and invalid certificate acceptance remain risks.

Denial of service

- **Volumetric floods** exhaust bandwidth; protocol attacks exhaust network or connection state; application attacks exhaust application resources. Distributed sources complicate blocking. A pulse-wave pattern alternates bursts; IP spoofing does not itself create a distributed botnet.
- **SYN floods** pressure half-open connection state. SYN cookies, backlog controls and upstream filtering help. UDP and ICMP floods stress different resources; reflection/amplification abuses third-party responses directed toward a spoofed victim.
- **Slowloris** holds HTTP connections with incomplete headers; slow HTTP POST sends a request body very slowly. Timeouts, minimum transfer rates, connection limits and capable reverse proxies reduce resource retention.
- **Ping of Death** historically abuses oversized or malformed reassembled IP packets; **Smurf** uses ICMP reflection through broadcast-capable networks. Jamming targets the radio channel; scrambling/corruption disrupts decoding. Ordinary application firewalls do not prevent RF interference.
- Mitigate large **DDoS** attacks with upstream ISP or scrubbing support, edge capacity and layered rate controls. Model the actual bottleneck: requests/second, concurrent connections and bandwidth are different units. Invented arithmetic patterns do not determine real attack severity.

Detection and deception

- **IDS** detects and alerts; **IPS** can block inline. Signature detection recognizes known patterns; anomaly detection compares against expected behavior. WAFs focus on HTTP application traffic; network firewalls enforce network policy with differing inspection depth.
- **True positive:** real malicious activity alerts. **False positive:** benign activity alerts. **False negative:** malicious activity is missed. **True negative:** benign activity causes no alert. Tune using verified events rather than simply suppressing inconvenient alarms.

Network attacks & detection / Social engineering

Certified Ethical Hacker v12

KNOWLEDGE CHEATSHEET

INTERRUPT THE DECEPTION

Pretext Build trust → **Request** Induce action → **Verify** Trusted channel → **Report** Contain harm

- **Honeypots** are decoys; low-interaction systems emulate services, while high-interaction systems offer fuller environments. Inconsistent banners, timing and protocol behavior can reveal emulation, but virtualization or a particular MAC vendor does not prove a honeypot.
- Combine **endpoint telemetry**, network flows, DNS logs, authentication records and application logs. Unusual login times, data movement or beaconing justify investigation; no single indicator establishes an APT or its initial entry method.

06 Social engineering

Deception techniques

- **Phishing** uses deceptive messages; spear phishing targets selected victims; whaling targets senior figures; vishing uses voice; smishing uses SMS; spimming abuses instant messaging. Brand logos, apparent coworkers and urgent language do not authenticate a sender.
- **Pretexting** creates a believable role or story; a honey trap builds a relationship for disclosure; baiting offers something attractive; quid pro quo offers a benefit for cooperation. Fake support calls often combine authority, urgency and a request to install software.
- **Watering-hole attacks** compromise sites visited by a target group. **Scareware** uses fabricated infection warnings to push software or payment. Shoulder surfing observes activity; dumpster diving recovers discarded information; tailgating bypasses physical entry controls.
- **Synthetic identity theft** combines real or invented details into a new identity. Social-media quizzes and public profiles may reveal recovery-question answers, relationships and locations. Authentic-looking posts from real accounts can still disclose harmful information.

Practical defenses

- Verify sensitive requests through a **separate trusted channel**, especially payment changes, password resets and software installation instructions. Never rely on contact details supplied in the suspicious message itself.
- Combine **awareness training**, simple reporting, phishing-resistant authentication, email filtering, least privilege and approval controls. MFA reduces credential-only compromise but some phishing proxies can relay sessions; known sender accounts can themselves be compromised.
- Restrict public disclosures, shred sensitive records, challenge unauthorized entry and disable former employees' access promptly. A technical control addresses only part of a social engineering scenario; measure whether staff can recognize and report realistic requests.

Web applications & APIs

Certified Ethical Hacker v12

KNOWLEDGE CHEATSHEET

ENFORCE EACH TRUST BOUNDARY

- **Input** — Validate
- **Query** — Parameterize
- **Object** — Authorize
- **Output** — Encode

07 Web applications and APIs

Server hardening and authentication

- **Harden servers** by removing unused services, modules and accounts, patching supported software, limiting administrative access and auditing permissions. Secure actual settings in files such as httpd.conf or php.ini; changing a banner is only a minor information-reduction measure.
- Disable exposed **debug functions**, sample applications, directory listings and detailed production errors. Keep diagnostic detail in protected logs. Review configuration drift and service identities; TLS and backups do not correct excessive permissions.
- **Verbose login errors** can enumerate usernames; use consistent failure responses. Rate limits, step-up controls and monitored lockout policies resist guessing while limiting denial-of-service risk. Client-side checks can be bypassed; enforce validation and authorization on the server.
- **Burp Suite** intercepts authorized browser traffic, modifies requests and supports application testing; Sequencer analyzes token randomness. Web scanners such as Wapiti, Syhunt and Netsparker complement manual testing but do not prove all business logic is correct.

SQL injection and robust defenses

- **SQL injection** occurs when untrusted input changes query structure. Effects may include authentication bypass, data disclosure or modification; command execution and file access depend on database features, privileges and configuration.
- **In-band SQLi** returns results through the application's normal channel. Error-based SQLi exposes useful database errors; **UNION-based** SQLi combines compatible result sets with matching column count and suitable types. Separate stacked statements depend on driver support.
- **Blind SQLi** infers data from behavior: boolean-based tests compare responses; time-based tests observe conditional delays. Generic error pages reduce leakage but do not remove the injection flaw. Control for network noise when interpreting response time.
- **Out-of-band SQLi** uses another channel, such as database-originated DNS or HTTP traffic, where permissions and egress allow it. Restrict outbound database traffic and privileged features as well as fixing unsafe query construction.

Web applications & APIs

Certified Ethical Hacker v12

KNOWLEDGE CHEATSHEET

ENFORCE EACH TRUST BOUNDARY

- **Input** — Validate
- **Query** — Parameterize
- **Object** — Authorize
- **Output** — Encode

- **Parameterized queries** separate values from SQL structure. Allowlist identifiers that cannot be parameters, validate types/ranges/lengths server-side, and use least-privileged database accounts. Escaping and keyword blocklists are fragile substitutes.
- **Encoding, concatenation, comments, case and whitespace variation** can evade naive signatures without changing query meaning. A WAF supplies defense in depth; canonicalization and safe query APIs address the underlying trust boundary.

Browser injection and request abuse

- **XSS** executes attacker-controlled script in a trusted origin. Stored XSS persists in application data; reflected XSS returns in a response; DOM-based XSS flows through unsafe browser-side code. Use context-specific output encoding and safe DOM APIs; sanitize HTML when rich markup is required.
- **HttpOnly** prevents script access to a cookie; **Secure** restricts its transmission to HTTPS; **SameSite** controls certain cross-site cookie sending. HttpOnly does not stop XSS from acting in the user's session, and redirecting to another origin does not expose that site's protected cookies.
- **CSP** restricts executable sources and other browser behavior. A same-origin script allowance can still expose unsafe endpoints or gadgets; it does not prove exploitation. Prefer nonce/hash-based script policies and remove unsafe content paths; CSP complements output safety.
- **CSRF** tricks an authenticated browser into an unwanted request. Use anti-CSRF tokens, origin checks and suitable SameSite settings. **Clickjacking** overlays or frames UI; CSP frame-ancestors and X-Frame-Options address framing rather than SQLi or XSS.
- **SSRF** makes a server fetch attacker-influenced destinations, potentially reaching internal services or metadata. Allowlist expected destinations and schemes, validate resolved addresses and redirects, restrict egress and protect metadata endpoints.

Files, parsers and web services

- **Path traversal** uses sequences such as ../ or encoded equivalents to escape an intended directory. Resolve and validate canonical paths against an allowed root; avoid direct user-controlled filesystem paths. Directory listings and traversal are related but distinct exposures.
- **SSI injection** abuses server-side include directives; **SSTI** abuses template evaluation. Extensions such as .shtml, .stm or .shtm may suggest SSI handling but do not prove injection. Disable unused execution features and keep user data outside executable templates.
- **XXE** abuses unsafe XML external-entity processing to read resources or cause server requests. Disable unnecessary DTD/external-entity resolution. XML injection changes document structure; **CRLF injection** inserts line delimiters into contexts such as headers or logs.

Web applications & APIs / Wireless & mobile

Certified Ethical Hacker v12

KNOWLEDGE CHEATSHEET

ENFORCE EACH TRUST BOUNDARY

- **Input** — Validate
- **Query** — Parameterize
- **Object** — Authorize
- **Output** — Encode

- **REST** is an architectural style, often using HTTP GET, POST, PUT and DELETE; SOAP uses structured XML messages. **WSDL** describes services; **WS-Security** supports message-level protection; **WS-Addressing** supplies message addressing and routing information.
- Validate **SOAP routing headers, actions and XML structure** and authorize the requested operation. Spoofed WS-Addressing or SOAPAction values can cause misrouting or unauthorized behavior when trusted blindly. **Webhooks** are event-triggered HTTP callbacks; authenticate them and reject replayed deliveries.

API authorization and sessions

- **Broken object-level authorization (BOLA/IDOR)** lets a caller access another user's object by changing an identifier. Check authorization for every object and action on the server; unpredictable IDs and successful authentication are insufficient.
- **ABAC** evaluates attributes of subject, object, action and environment; RBAC uses assigned roles. Choose controls that enforce ownership and tenant boundaries. A missing ABAC implementation is not the universal name for every API authorization flaw.
- **CORS** controls browser access to cross-origin responses; it is not API authentication or authorization. Avoid reflecting arbitrary origins with credentials. Check business rules, quotas and function-level permissions separately from input validation.
- **Session hijacking** takes over an established session; fixation makes a victim authenticate using an attacker-known identifier. Session donation can cause a victim to enter data into an attacker's already authenticated account. Rotate IDs after authentication and privilege changes and bind sensitive actions to verified intent.
- Protect **session tokens** with strong randomness, secure transport, suitable cookie flags, expiry, logout invalidation and reauthentication for sensitive actions. Web cache poisoning manipulates a cached response shared with other users; include security-relevant request variation in cache design.

08 Wireless and mobile security

Wi-Fi authentication and attacks

- **WEP** uses obsolete RC4-based protection with serious IV weaknesses. WPA/TKIP is also legacy. WPA2 with AES-CCMP and WPA3 provide stronger protection; a strong unique passphrase matters. Hiding SSIDs or filtering MACs does not supply encryption or strong identity proof.
- **WPA2-Personal** uses a shared secret and four-way handshake. Captured handshake material can support offline password guessing; knowing an SSID or seeing encrypted traffic does not reveal the passphrase. Strong passphrases make guessing expensive rather than impossible in principle.
- **WPA3-Personal** uses SAE/Dragonfly instead of the traditional PSK exchange, resisting passive offline dictionary attacks. **WPA3-Enterprise 192-bit mode** uses a stronger cryptographic suite, including GCMP-256 and SHA-384-based mechanisms; it is distinct from Personal mode.

Wireless & mobile

Certified Ethical Hacker v12

KNOWLEDGE CHEATSHEET

SECURE THE CONNECTION AND DEVICE

- **Radio** — Authenticate
- **Transport** — Validate TLS
- **Device** — Patch & manage
- **Data** — Encrypt & restrict

- **802.1X/EAP** supports individual enterprise authentication through an authenticator and authentication server. Validate the server certificate and use suitable EAP methods. Shared passwords, even on an encrypted WLAN, provide limited individual accountability.
- **KRACK** exploits key reinstallation behavior in affected WPA2 implementations. **Dragonblood** names SAE/Dragonfly weaknesses in affected implementations. Transition modes can expose downgrade paths; patch clients and APs and configure supported protections rather than assuming every device is vulnerable.
- **Evil twins** imitate legitimate WLANs; rogue APs are unauthorized access points. Wardriving discovers wireless networks from a moving vehicle. Wireless monitoring, trusted enterprise authentication and HTTPS help; a familiar SSID or captive portal does not authenticate an AP.

Bluetooth, WPS and cellular exposure

- **WPS** can expose weak PIN enrollment even when the Wi-Fi passphrase is strong. Disable unneeded WPS; **wash** discovers WPS-enabled APs and reported lock status. Lockout, firmware and configuration determine actual susceptibility.
- **Bluejacking** sends unsolicited Bluetooth messages; **bluesnarfing** steals data; **bluebugging** gains unauthorized control; **bluesmacking** disrupts service. Patch devices, limit discoverability and reject unexpected pairing requests.
- **Btlejacking** targets affected Bluetooth Low Energy connections and can involve sniffing, disruption or hijacking using tools such as Btlejack. Susceptibility depends on protocol and implementation; Bluetooth proximity alone does not grant access.
- **Rogue cellular base stations** can impersonate infrastructure or manipulate connections under applicable conditions. **aLTEr** concerns LTE user-plane manipulation under a specific on-path model; **KRACK** is a Wi-Fi issue. Strong application TLS remains important across radio technologies.

Mobile platforms and data protection

- **AndroidManifest.xml** declares application components and permissions; classes.dex contains executable bytecode; resources hold packaged assets. Review exported activities/services/receivers, permissions, data storage and network behavior during authorized reverse engineering.
- **Rooting/jailbreaking** relaxes platform protections. Tethered jailbreaks require computer assistance to boot jailbroken; untethered ones persist across boots; semi-untethered variants need on-device reactivation; semi-tethered variants require computer assistance to re-enable the jailbreak.
- **MDM/MAM** enforces approved apps, work profiles, encryption, patch compliance and access policy. BYOD designs balance work-data controls with personal privacy. App-store restrictions and training reduce malicious-app risk; VPNs do not make installed apps trustworthy.

Wireless & mobile / IoT & operational technology

Certified Ethical Hacker v12

KNOWLEDGE CHEATSHEET

CONTROL DEVICE EXPOSURE

Inventory Identify → **Segment** Limit paths → **Monitor** Observe safely → **Recover** Protect process

- Minimize sensitive **local mobile data**, encrypt required storage and protect keys with platform keystores. Biometric unlocking controls access but does not replace data encryption. Authenticate TLS peers, reject certificate errors and securely manage any certificate-pinning lifecycle.
- **Agent Smith** illustrates malicious Android app replacement; **Trident** refers to an iOS exploit chain associated with spyware. **Spearphone** research uses motion-sensor data to infer loudspeaker audio; **man-in-the-disk** abuses exposed external storage. These names describe different attack mechanisms.

09 IoT and operational technology

Device exposure and industrial networks

- **IoT** devices connect sensing or actuation to services; **IoMT** handles medical use cases; **IIoT/OT** supports industrial processes. Device inventories should record model, firmware, owner, connectivity and criticality before selecting controls or tests.
- **Default credentials**, exposed management services, obsolete firmware and weak update validation enable compromise. IoTSeeker checks selected devices for default credentials; exposure indexes and device filings provide clues, not proof of vulnerability.
- Use **zones and controlled conduits** to separate enterprise IT, industrial control, medical devices and remote access. A separate IoT network limits wider exposure; device-to-device policies or microsegmentation are needed to constrain movement within that network.
- **Zigbee** builds on IEEE 802.15.4 for low-rate local wireless networks. **MQTT** is a publish/subscribe application protocol; LPWAN and NB-IoT address different connectivity needs. Protocol names alone do not imply authenticated or encrypted operation.
- Industrial protocol clues include **EtherNet/IP 44818**, **Siemens S7 102** and **Modbus/TCP 502**. Nmap's enip-info and s7-info scripts query supported device information. Prefer passive inventory and vendor-approved assessment in fragile or safety-critical environments.

Replay, physical attacks and resilience

- **Replay attacks** retransmit captured valid commands. Fresh nonces, counters, authenticated messages and replay windows help; encryption without freshness can leave replay possible. Radio analysis tools such as URH inspect signals but do not establish the trustworthiness of a command.
- **Timing side channels** infer secrets from computation duration; constant-time comparisons reduce leakage. **Power/clock/reset glitching** and voltage/frequency tampering inject faults that may skip instructions. Optical and electromagnetic fault injection use different physical channels.

IoT & operational technology / Cloud & containers

Certified Ethical Hacker v12

KNOWLEDGE CHEATSHEET

CLOUD RESPONSIBILITY LAYERS

- **Identity** — Least privilege
- **Data** — Protect keys
- **Images** — Verify supply
- **Runtime** — Restrict access

- Protect **firmware and boot chains** with signed updates, integrity checks, secure boot and restricted debug interfaces. Isolate compromised devices using their actual traffic and role; there is no single universal port whose blocking resolves every IoT infection.
- OT defenses combine **segmentation, controlled remote access, unique credentials, approved patching, monitoring and tested recovery**. Prioritize safety and availability when planning changes; indiscriminate reboots, active exploitation and consumer antivirus assumptions may disrupt the process.

10 Cloud and container security

Cloud models and responsibilities

- **IaaS** supplies infrastructure while customers manage more of the OS and application stack; **PaaS** adds a managed runtime; **SaaS** provides an application. Customers retain responsibilities such as identity, data handling and secure configuration according to the service contract.
- **Public clouds** serve multiple customers; private clouds serve one organization; community clouds serve a community with shared concerns; hybrid clouds combine distinct environments with integration. Multicloud uses multiple providers and does not itself define a security boundary.
- NIST cloud actors include **consumer**, **provider**, **broker**, **auditor** and **carrier**. A carrier transports connectivity; a broker intermediates services; an auditor independently assesses. These roles differ from the cloud deployment model.
- **Zero trust** verifies access using identity, device and contextual signals and grants least privilege. Network location alone supplies no implicit trust. MFA, segmentation and monitoring support the model; none individually implements the entire approach.

Cloud threats and controls

- **CASB** provides visibility and policy enforcement for supported cloud-service use, often including access controls and DLP. **CSPM** focuses on cloud configuration posture; workload protection covers runtime threats. Select integrations that actually cover the resources and actions required.
- **Cloud Hopper** describes campaigns compromising managed service providers to reach customers. **Cloudborne** concerns bare-metal firmware persistence across tenants. Cryptojacking abuses compute for mining; man-in-the-cloud attacks can abuse synchronization credentials or tokens.

Cloud & containers / Cryptography & keys

Certified Ethical Hacker v12

KNOWLEDGE CHEATSHEET

CLOUD RESPONSIBILITY LAYERS

- **Identity** — Least privilege
- **Data** — Protect keys
- **Images** — Verify supply
- **Runtime** — Restrict access

- For privacy from the **provider**, encrypt client-side and manage keys independently where application functionality allows. Provider-managed encryption and customer-managed server-side keys have different trust models; control over a key's lifecycle does not always prevent the service from using it.
- Secure **cloud identities**, APIs, storage permissions and metadata access; log administrative and data activity, restrict egress and segment workloads. Encryption at rest does not stop an authorized-but-compromised account from reading data through the application.

Containers and orchestration

- **Containers** package applications and dependencies while sharing the host kernel; VMs include separate guest operating systems. Container isolation is not a guarantee against host compromise. Remove unnecessary privileges, constrain capabilities and protect the runtime.
- The **Docker client** sends API requests; the **daemon** manages images, containers, networks and volumes; registries distribute images. Restrict daemon/socket access because control of the runtime can confer powerful host access.
- A secure image pipeline separates **development, testing/accreditation, registries, orchestration and runtime**. Verify dependencies, scan and sign images before trusted distribution; admission policies and runtime monitoring address later stages.
- In **Kubernetes**, the API server exposes the control plane, **etcd** stores cluster state, the **scheduler** assigns unscheduled pods to nodes, and controllers reconcile desired state. Protect secrets, RBAC, service accounts, network policies and node privileges.

11 Cryptography and key management

Algorithms and primitives

- **Symmetric encryption** shares a secret key; asymmetric cryptography uses paired public/private keys. **Hashing** produces a digest and is not reversible encryption. **HMAC** adds a shared key for integrity and authenticity; an unkeyed hash alone does not prove the sender.
- **AES** has a 128-bit block and 128-, 192- or 256-bit keys. **DES** has a 64-bit block and 56-bit effective key; three-key **3DES** applies DES stages with 168 nominal key bits but lower effective security. Legacy small-block ciphers should not be chosen merely for large key sizes.
- Cipher recognition: **Twofish** uses a 128-bit block and keys up to 256 bits; **Serpent** uses a 128-bit block and 32 rounds; **Blowfish** uses a 64-bit block and keys up to 448 bits; **CAST-128** uses a 64-bit block and 12 or 16 rounds.

Cryptography & keys

Certified Ethical Hacker v12

KNOWLEDGE CHEATSHEET

MATCH PRIMITIVE TO PURPOSE

- **Encrypt** — Confidentiality
- **MAC** — Shared-key integrity
- **Sign** — Public verification
- **Keys** — Trust & recovery

- **RSA** relies on integer-factorization difficulty; **ECC** uses elliptic-curve problems. Key sizes are not directly comparable across algorithms. Stronger symmetric encryption does not repair a broken public-key exchange; simplistic runtime formulas are not cryptographic security estimates.
- **Authenticated encryption**, such as AES-GCM, protects confidentiality and detects modification when used correctly. Respect nonce requirements. Block size, key size, mode, randomness and key management are separate properties; encryption alone need not provide integrity.

Key exchange, signatures and trust

- For **confidentiality**, protect the data key for the recipient using the recipient's public key; the recipient uses their private key. For a **digital signature**, the sender signs with their private key and others verify with the sender's public key.
- **Hybrid encryption** uses fast symmetric encryption for data and public-key mechanisms to establish or protect the data key. **Diffie-Hellman/ECDH** derives a shared secret; authenticate the exchange to prevent an on-path impersonator. Key agreement is not itself bulk data encryption.
- **TLS** authenticates peers as configured and provides confidentiality and record integrity; validate certificates, hostnames and trust chains. **IPsec** protects IP traffic; **SSH** protects remote sessions. Adding a second protocol is not inherently required for tamper detection already supplied by authenticated TLS.

Cryptography & keys

Certified Ethical Hacker v12

KNOWLEDGE CHEATSHEET

MATCH PRIMITIVE TO PURPOSE

- **Encrypt** — Confidentiality
- **MAC** — Shared-key integrity
- **Sign** — Public verification
- **Keys** — Trust & recovery

- **OpenPGP/GnuPG** supports hybrid encryption and signatures; its web of trust uses certifications between keys. **S/MIME** commonly uses certificate-authority-based trust. Verify whose key or certificate is trusted before relying on an encrypted or signed message.
- A signature supports **integrity, origin authentication and non-repudiation** when identity, key custody and validation are sound. It does not conceal the message. Symmetric MAC participants share a key, so a MAC does not provide the same independent signer attribution.

Keys, passwords and legacy weaknesses

- **KDFs** derive keys; **key stretching** deliberately raises password-guessing cost using suitable work factors. Unique salts prevent identical-password hash reuse; they are not secrets. Store passwords with a dedicated slow password-hashing scheme rather than reversible encryption for login comparison.
- A **TPM** protects key operations and can seal secrets to platform state. BitLocker can use TPM-backed protection and separate recovery material. Recovery-key escrow in an approved directory supports recovery; protect and audit escrow access as carefully as privileged credentials.
- **Key escrow** gives an approved recovery party access to recovery material; archival retains keys for later recovery. Plan rotation, revocation, backup and destruction. Losing an encryption key can make data permanently unrecoverable despite intact ciphertext backups.
- **DROWN** exploits SSLv2 support and related RSA key reuse across services; remove insecure protocol exposure, including other services sharing the key. **Padding oracles** leak validity information; **CRIME** concerns compression leakage. These are different failure mechanisms.
- **Replay, side-channel and downgrade attacks** target freshness, implementation leakage or negotiation rather than simply key length. Disable obsolete protocols, patch affected implementations and use authenticated modern configurations; do not assume a named algorithm alone secures a system.