Infrastructure Security Using Linux

Computer science / Cybersecurity

Security

27 →

Security

CYBER SECURITY CTF PLAYER

"I'm basically a hacker" -- can't exit vim

"Social engineering is my specialty" -- can't order pizza without panicking

"Time to fire up Kali Linux" -- just to browse Reddit

"This challenge is too easy" -- spent 6 hours on a ROT13 cipher

"I'll crack this in no time" -- still bruteforcing 'admin:admin'

"I'm in" -- accidentally DDoS'd himself

"I could hack the Pentagon" -- uses 'password123' for everything

"Let me try a buffer overflow" -- segfaults his own machine

Security?

- Computer security trails behind other computing fields, with flaws worsening and consequences growing more severe
- 2010: Stuxnet worm sabotaged centrifuges at Iran’s uranium enrichment plant
- 2013: Edward Snowden exposed NSA’s mass-surveillance program and corporate cooperation
- 2013: Emergence of ransomware as a major new attack vector
- 2015: Breach of the U.S. Office of Personnel Management
- 2016: Alleged Russian state-sponsored influence on the U.S. presidential election
- 2017: Global ransomware outbreak on Windows systems in 150+ countries using an NSA-developed exploit

Security Beyond Technology

- Security problems are not purely technical; they cannot be solved by buying a product or service
- Achieving acceptable security requires patience, vigilance, knowledge, and persistence
- Responsibility spans sysadmins, end-users, and management

Balancing Security & Usability

- Every additional security measure adds constraints for you and your users
- Too little security invites risk; too much hinders productivity
- Strike the right balance between protection and ease of use

Inherent OS Vulnerabilities

- No networked OS (UNIX, Linux, Windows, etc.) is inherently secure
- Absolute, unbreachable security demands a physical air gap
- Even air-gapped systems can leak information via side-channels
- Beyond Air Gaps: Acoustic Attacks?
  - Air gaps are not foolproof against advanced attacks
  - Genkin & Shamir & Tromer (2014): RSA keys extracted from laptop audio emissions
  - Security requires defending against both digital and physical side-channels

Elements of Security: CIA Triad

- **Confidentiality**
  - Protect data privacy by limiting access to authorized users
  - Core techniques: Authentication, Access control, Encryption
- **Integrity**
  - Ensure data authenticity and detect unauthorized modifications
  - Verify trustworthiness of sources:
    – TLS certificates (e.g. VeriSign, Equifax) prove server identity
    – PGP signatures guarantee message integrity
- **Availability**
  - Ensure authorized users can access information when needed
  - Covers both malicious and non-malicious outages (admin errors, power failures)
  - Often ignored until service disruption occurs

How security is compromised?

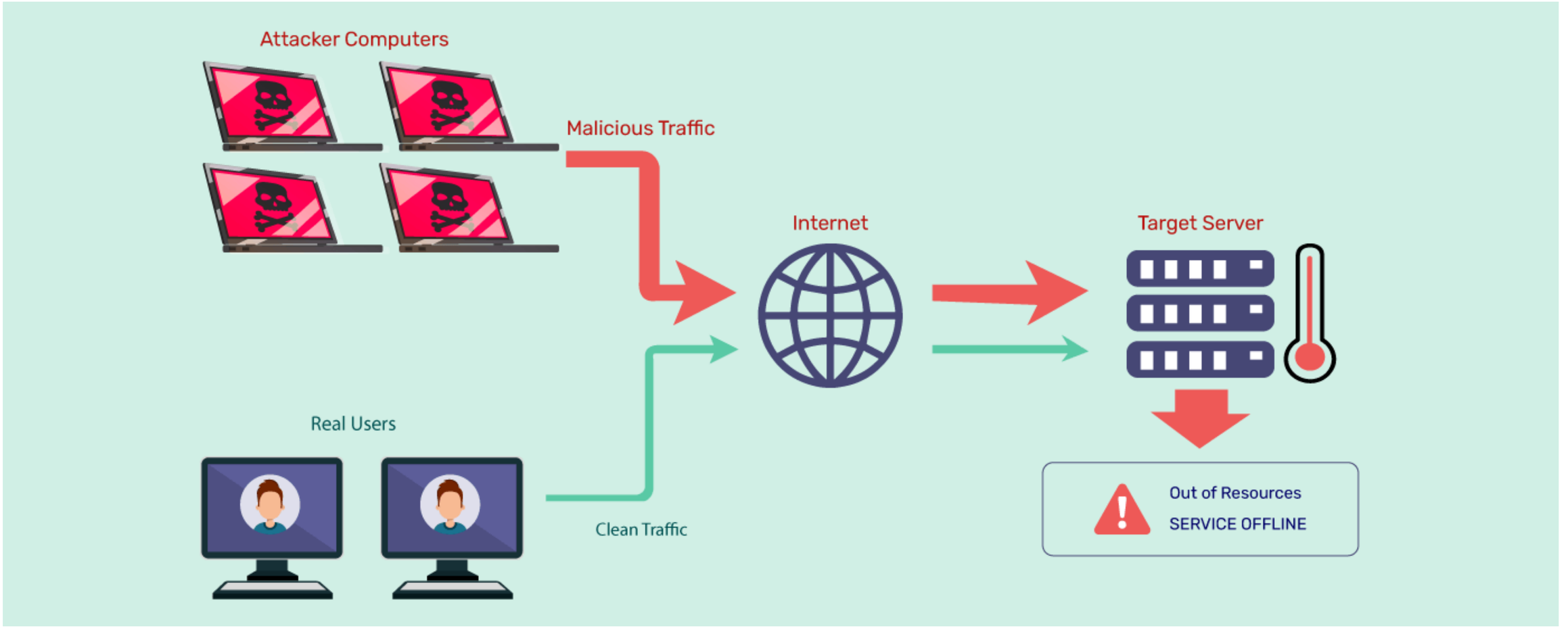
- **Phishing (… it's Social Engineering)**
  - Deceptive attempts to harvest information or trick users into unsafe actions
  - Delivered via email, instant message, SMS or social-media contact
  - Goals include credential theft, malware installation and data exfiltration
- **Social Engineering**
  - Exploits human psychology to bypass technical defenses
  - Phishing is a prime example of social engineering in action
  - One of the hardest threats to neutralize, since it targets people rather than systems

How security is compromised?

- **Software Vulnerabilities:**
  - ➢ **Ex. Buffer Overflows**
    - Buffer overflow: writing more data into a fixed-size buffer than it can hold
    - Arises from improper input validation
    - Part of a broader class of input-validation vulnerabilities
    - **Mitigation: Least Privilege**
      - Run applications with the minimum privileges needed
      - Unprivileged processes limit the damage of security bugs
      - Combine with regular patching and code audits for defense in depth

How security is compromised?

- **DDoS Attacks**
  - Distributed Denial of Service (DDoS) floods a target to disrupt or degrade service
  - Attackers implant malicious code on external devices to create “bots”
  - Bots are remotely controlled to overwhelm the victim with traffic
- **Ex. IoT Botnets & Mirai Case**
  - Modern botnets leverage unsecured IoT devices (IP cameras, printers, baby monitors)
  - Fall 2016: Mirai botnet launched 620 Gb/s attack on Brian Krebs’s site from tens of thousands of IPs
  - Demonstrates the scale and impact of consumer-device compromises


Attacker Computers
Malicious Traffic
Internet
Target Server
Real Users
Clean Traffic
Out of Resources
SERVICE OFFLINE

How security is compromised?

- **Network, system, or application configuration Errors (Common Pitfalls) :** Software defaults prioritize usability over security
- Exploitable misconfigurations:
  - Accounts without passwords
  - Firewalls with overly relaxed rules
  - Unprotected databases
  - Host example: Linux systems boot without requiring a bootloader password

Mitigation & Trade-Offs

- Apply secure configuration baselines and regular audits
- Principle of least privilege for services and accounts
- Bootloader password adds protection but demands physical admin presence after reboot
- Balance security hardening with operational availability and maintainability

Basic Security measures (Again … but more … Next …)

- **Site Security: High-Level Best Practices**, you can improve your site's security by keeping in mind a few rules of thumb:
- **Least Privilege:** grant only the permissions needed
- **Defense in Depth:** layer controls beyond a single firewall
- **Minimal Attack Surface:** reduce exposed interfaces, services, and systems

Software updates

- Keeping systems updated with the latest patches is an administrator's highest-value security chore.
- **Patch Management Essentials**
  - **Regular Schedule:** Monthly cadence + emergency fixes, planned for minimal user impact
  - **Change Plan:** Document impacts, test & rollback steps, and communicate with stakeholders
  - **Patch Relevance:** Subscribe to vendor advisories and general security feeds
  - **System Inventory:** Maintain an accurate asset list (apps & OS) with coverage reporting tools

• • •

- **unnecessary services**
  - Stock systems come with lots of services running by default. Disable (and possibly remove) those that are unnecessary, especially if they are network daemon.
- **Backups**
  - Regular, tested system backups are an essential part of any site security plan. They fall into the "availability" bucket of the CIA triad.

Viruses and worms

- **Unix/Linux Virus Resistance**
- Only a few viruses exist (mostly academic)
- No large-scale outbreaks comparable to Windows
- Possible reasons:
  - Lower desktop market share deters attackers
  - Strong access controls limit propagation and damage

Rootkits: Stealth Tools for Persistent Threats

- Conceal system data (processes, disk activity, network traffic)
- Range from user-space app replacements (e.g., modified **ls**, **ps**)
- Advanced variants install as kernel modules—extremely hard to detect
- Notorious example: Sony’s CD copy-protection rootkit deployed on millions of discs

Packet filtering

- **Firewalls & Security Groups**
  - **Network Gateway:** deploy a packet-filtering router or firewall
  - **Host-Based Defense:** add **ipfw** (FreeBSD) or **iptables** (Linux)
  - **Cloud Environments:** use granular security groups with both inbound and outbound rules

Passwords and MFA

- **Password Rules**:
  - Every account must have a password.
  - Passwords should be complex and hard to guess to prevent compromise.
- **Multifactor Authentication (MFA)**:
  - MFA enhances security by requiring two forms of verification:
    - Something you know (e.g., a password).
    - Something you have (e.g., a phone or physical device).
  - MFA is strongly recommended for all internet-facing portals, especially those with administrative access (e.g., VPNs, SSH, web application interfaces).

Passwords and MFA

- **MFA as a Minimum Requirement**:
  - MFA is now considered the **absolute minimum** for securing administrative interfaces.
  - Single-factor (password-only) authentication is increasingly seen as insufficient for any user authentication.
- **Availability of MFA Services**:
  - Cloud-based MFA services like **Google Authenticator** make it easy to implement MFA.

Application penetration testing

- **Why App Penetration Testing Matters**
  - Overall security only as strong as the weakest link
  - Secure network/infrastructure won’t protect a vulnerable application
  - Example: app exposing sensitive data without authentication defeats other security controls
- **OWASP Resources & Testing Approach**
  - OWASP Top Ten: catalog of the most critical web application risks
  - Testing methods:
    - Automated vulnerability scanners
    - Manual probe of input validation, authentication, session management
    - Code review and configuration audit
  - Leverage OWASP guides (Testing Guide, ASVS) for standardized, repeatable assessments

Passwords and user accounts

- **Password Management Best Practices**
  - **Strong Random Passwords:** ≥12 chars (letters + digits + symbols); use a vault for storage
  - **Leverage Length:** security grows exponentially—prefer long passphrases
  - **Extra Complexity:** incorporate deliberate misspellings or symbol substitutions
  - **Regular Rotation:** change root/Admin passwords
    - Every 3–6 months
    - When an authorized user departs
    - If compromise is suspected

Security power tools

Nmap: Network Discovery & Port Scanning

- Probes target hosts to identify listening TCP/UDP ports
- Sends crafted packets and analyzes replies to map devices, services and protocols
- Offers scan modes such as TCP SYN, UDP and OS fingerprinting
- Acts as a remote **netstat** for external reconnaissance

Nessus: next-generation network scanner

- Nessus: Industry-Leading Vulnerability Scanner
- First released in 1998 by Renaud Deraison
  - 31,000 plug-ins for both local and remote security checks
- Scans every port (e.g., web services beyond port 80)Attempts exploits to confirm real-world vulnerabilities
- Widely adopted as the most comprehensive scanner available

Metasploit

- Metasploit is controlled by the U.S.-based security company Rapid7, but its GitHub project has hundreds of contributors. Metasploit includes a database of hundreds of ready-made exploits for known software vulnerabilities.
- Metasploit has the following basic workflow:
  - Scan remote systems to discover information about them.
  - Select and execute exploits according to the information found.
  - If a target is penetrated, use included tools to pivot from the compromised system to other hosts on the remote network.
  - Run reports to document the results.
  - Clean up and revert all changes to the remote system.

Lynis

- Lynis is an open-source security auditing tool designed for Unix-based systems such as Linux, macOS, and BSD. It performs in-depth system scans to assess security configurations, compliance, and vulnerabilities, providing detailed recommendations to improve overall security posture.

Bro (now Zeek)

- **Bro (now Zeek): the programmable network intrusion detection system**
- Bro inspects all traffic flowing into and out of a network. It can operate in passive mode, in which it generates alerts for suspicious activity, or in active mode, in which it injects traffic to disrupt malicious activity. Both modes likely require modification of your site's network configuration.
- More on zeek.org

Bro (now Zeek)

- Example: Suspicious Domain Detection: Using Zeek's scripting capabilities, you can write a script to flag connections to known malicious domains:

```
event http_request(c: connection, method: string, host: string, uri: string) {
        local suspicious_domains = set("malicious.com", "badguy.org");
        if (host in suspicious_domains)  {
                print fmt("ALERT: Suspicious domain accessed: %s by %s", host, c$id$orig_h);
        }
}
```

- This script triggers an alert whenever a client connects to one of the listed malicious domains.

Snort: the popular network intrusion detection system

- Snort (snort.org) is an **Open-Source Network IDS/IPS** originally written by Marty Roesch and now maintained by Cisco.
- **How Snort works:**
  - **Traffic Inspection:** Snort examines packets as they traverse the network, dissecting their headers and payloads.
  - **Signature-based pattern matching against rule database:** Traffic is compared against a database of predefined signatures and rules for known attack patterns.
  - **Alerts:** If a match is found, Snort generates an alert, which can be logged, sent to a monitoring system, or used to trigger a defensive action (like dropping the traffic).

OSSEC: host-based intrusion detection

- OSSEC (Open-Source HIDS) serves up the following:
  - Root kit detection
  - Filesystem integrity checks
  - Log file analysis
  - Time-based alerting
  - Active responses
- **Agent–Manager Architecture**
  - Agents collect and forward event logs
  - Manager analyzes logs against rule sets
- **Automated Response**
  - On rule match, Manager instructs Agents to block/unblock IPs

Cryptography primer

- **Cipher:** mathematical algorithm securing messages
- **Encryption:** converts plaintext → ciphertext
- **Decryption:** converts ciphertext → plaintext
- **Key Properties:**
  - **Confidentiality:** only intended recipients can read it
  - **Integrity:** tampering is detectable
  - **Non-repudiation:** sender’s authenticity is verifiable

Symmetric Key Cryptography

- Shared secret key for both encryption & decryption
- Secure initial key exchange; key reused indefinitely
- Only parties with the key can decrypt or tamper
- Highly efficient: fast processing & compact ciphertext
- AES (NIST standard) is the predominant algorithm

Public Key Cryptography

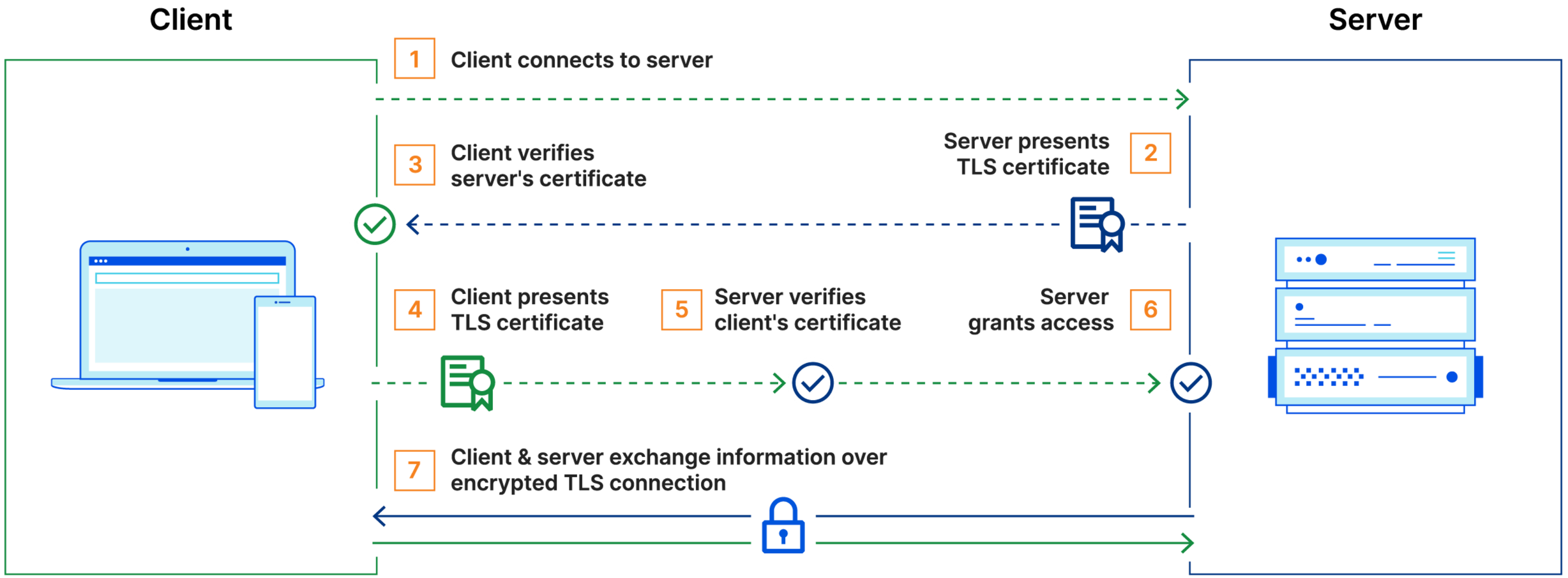
- Solves secure key-exchange problem of symmetric schemes
- Each user has a **key pair**:
  - **Public key:** openly shared
  - **Private key:** kept secret
- To send a message:
  - Encrypt with recipient's public key
  - Only their private key can decrypt
- Computationally heavy for large data → typically used to:
  - Establish a session
  - Securely share a symmetric key
- Hybrid use: asymmetric for key exchange + symmetric for bulk encryption

Public Key Infrastructure (PKI)

- **Trust Challenge:** verifying that a public key truly belongs to its claimed owner
- **PGP Web of Trust:** decentralized key-signing model with limited adoption
- **PKI Solution:** hierarchical trust via Certificate Authorities (CAs)
- **Certificate Workflow:**
  - CA signs user public-key certificate with its private key
  - Clients verify certificates against CA public keys in their trust store
- **CA Ecosystem:**
  - Major CAs (GeoTrust, VeriSign) bundled in OS distributions
  - Issuance fees reflect CA reputation and certificate features
- **Let's Encrypt:** free, automated CA sponsored by EFF, Mozilla, Cisco, etc.
- **Private CA Option:** use OpenSSL to create your own CA and deploy its certificate across your systems for internal signing.

TLS (Transport Layer Security)

- Wraps TCP connections to secure any Layer 4 protocol (HTTP, SMTP, FTP, etc.)
- Leverages public-key cryptography and PKI for key exchange
- Encrypts all message content (URL, headers); only host/port remain visible
- Supports one-way TLS (server authenticated by client)
- Supports mutual TLS (mTLS): both client & server present certificates


Client
Server
1
Client connects to server
2
Server presents
TLS certificate
3
Client verifies
server's certificate
4
Client presents
TLS certificate
5
Server verifies
client's certificate
6
Server
grants access
7
Client & server exchange information over
encrypted TLS connection

Cryptographic Hash Functions

- Map any-length input → fixed-length digest
- Avalanche effect: small input change → ~50% bits flipped
- Pseudorandom & one-way: infeasible to invert or engineer collisions
- Collision risk ↓ with longer digests; use SHA-2/3 (e.g., SHA3-512); MD5 is deprecated
- Common use: file integrity checks (sha256sum), PGP signatures
- HMAC: keyed hash for combined integrity & authenticity

Cryptographic Software Selection

- Trust open-source libraries (e.g., **OpenSSL**) over closed-source
- Open transparency: public disclosure & rapid patching of vulnerabilities
- Community review by thousands enhances security
- Avoid home-grown or bespoke cryptosystems — correct library use is hard; custom designs invite flaws

The openssl Command

- TLS Multitool: key pair generation, file encryption/decryption, cert management, format conversion
- Key & Cert Prep: create CSRs, act as CA, inspect remote TLS endpoints
- Start by creating a 2048-bit private key:

  openssl genrsa -out admin.com.key 2048

- Use the private key to create a certificate signing request.

  openssl req -new -sha256 -key admin.com.key -out admin.com.csr

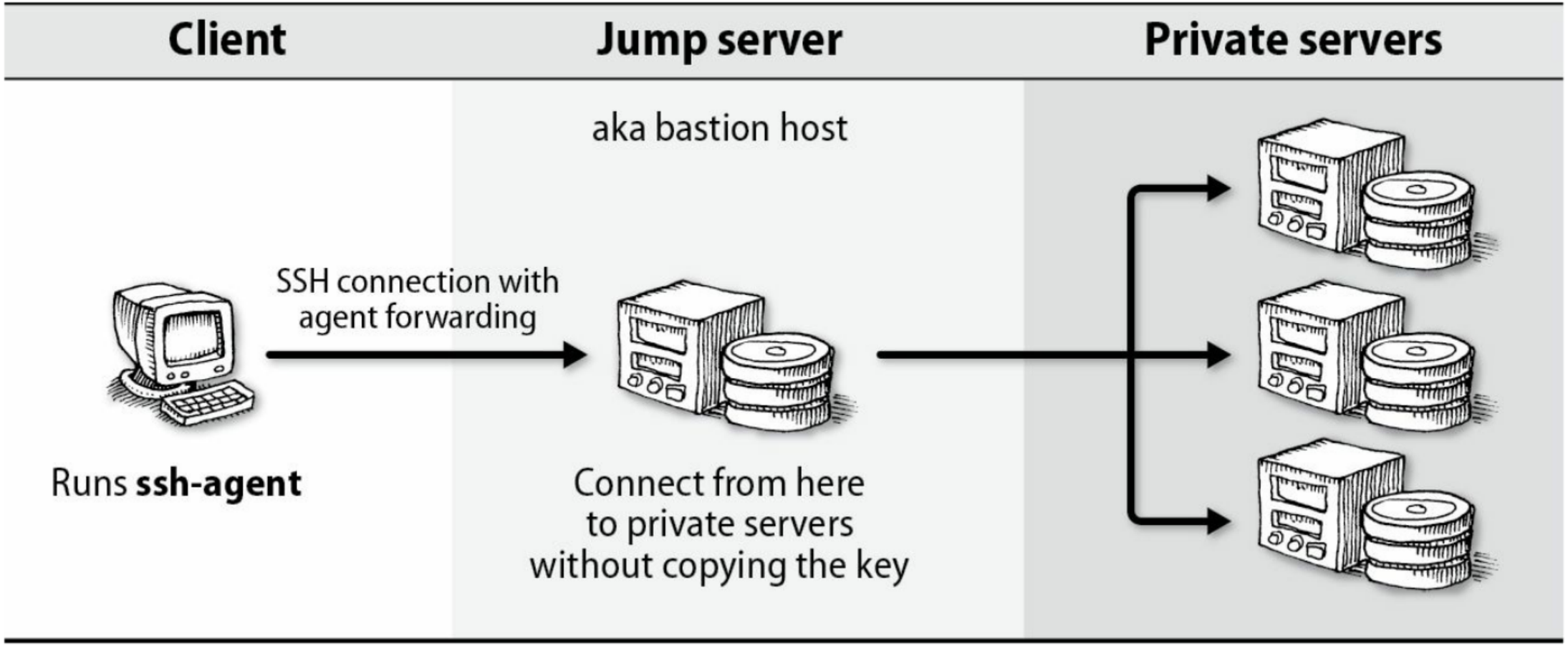
- To check certificates:

  openssl x509 -noout -text -in google.com.pem

SSH (Secure Shell)

- Client/server protocol for encrypted authentication, confidentiality & integrity
- **OpenSSH tools:** ssh, sshd, ssh-keygen, ssh-add/ssh-agent, ssh-keyscan, sftp-server, sftp, scp
- Connects on TCP port 22; verifies server via public-key fingerprint
- Supports multiple auth methods: passwords, public keys, GSSAPI, PAM/OTP
- Tunable via **/etc/ssh** configuration files

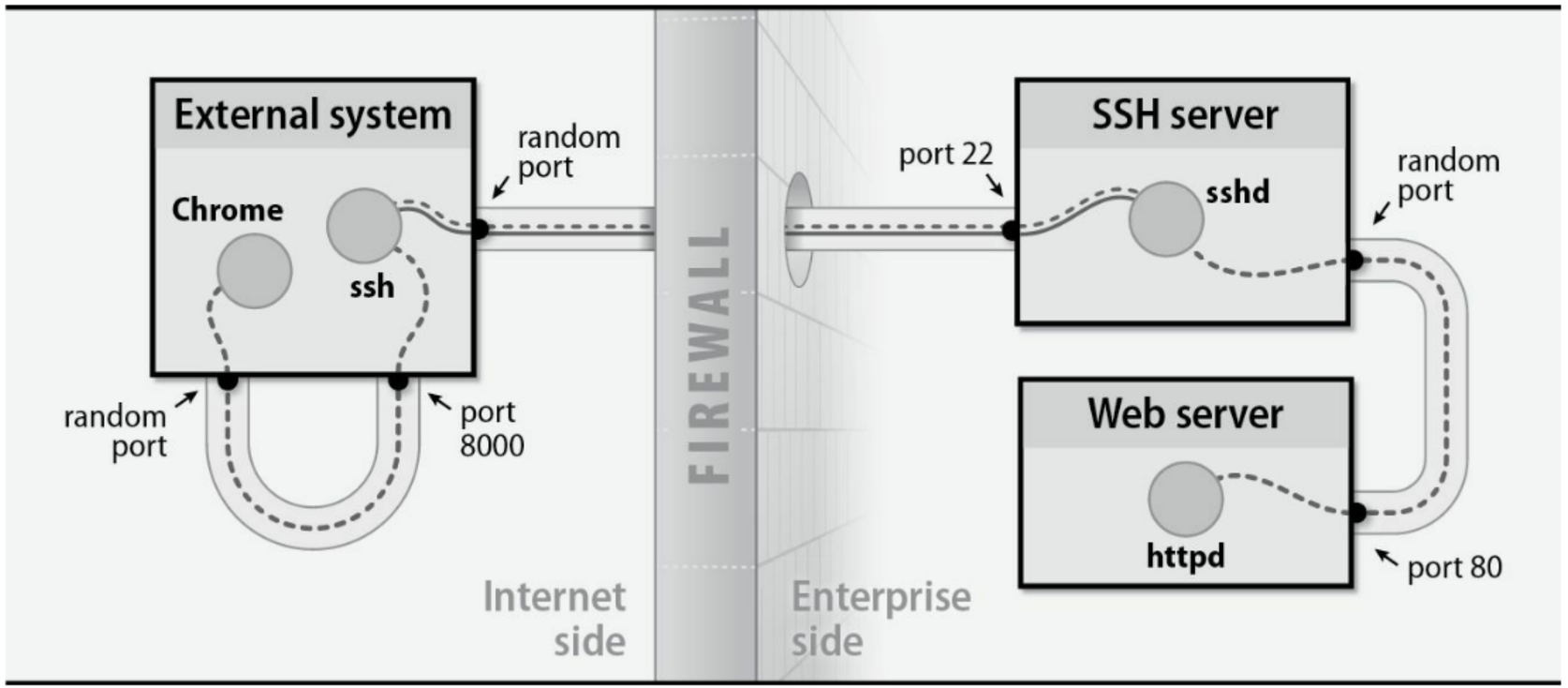
Remote systems



Port forwarding

- Another useful ancillary feature of SSH is its ability to tunnel TCP connections securely through an encrypted channel, thereby allowing connectivity to insecure or firewalled services at remote sites.

Exhibit D: An SSH tunnel for HTTP



Firewalls: Core Concepts

- **Packet-Filtering:**
  - Controls traffic by IP, port & protocol
  - Built-in tools: Linux iptables/**ufw**, FreeBSD **ipfw**
  - Dedicated appliances (Check Point, Palo Alto, etc.) offer greater reliability
- **Service Filtering:**
  - Allow only specific service ports (e.g., TCP 80 for HTTP)
  - Deploy filters on gateway or internal segments

Firewalls: Core Concepts

- **Defense-in-Depth & DMZ:**
  - Two-stage filtering: Internet gateway → DMZ → internal network
  - Expose Internet-facing services in DMZ for isolation
- **Stateful Inspection:**
  - Tracks connection state (TCP streams, UDP dialogs)
  - Makes context-aware decisions beyond single-packet rules

Firewall's safe?

- **Firewalls Alone Won’t Secure Your Network**
- Don’t rely solely on perimeter defenses
- Patch, harden & monitor every host individually
- **Use host-level tools**: Zeek (Bro), Snort, Nmap, Nessus, OSSEC, Wazuh
- Layer controls—defense-in-depth beyond the firewall
- Administrator vigilance is the ultimate safeguard

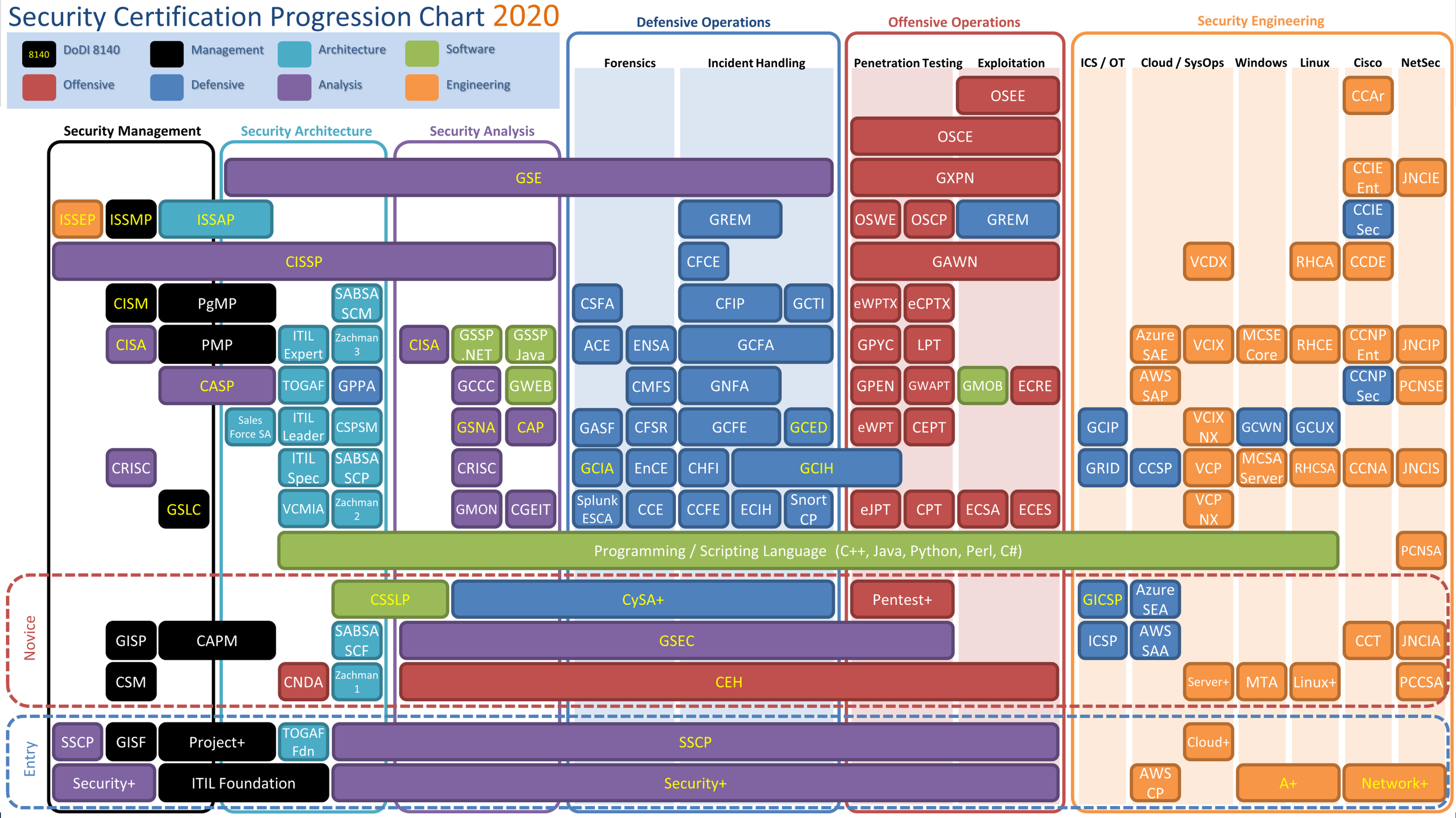
Virtual Private Networks

- Creates a “secure tunnel” to make a remote network appear local
- Authenticated via shared secret (e.g., passphrase)
- Encrypts all end-to-end traffic over untrusted networks
- No-budget alternative: use SSH for quick secure tunneling

IPsec Tunnels

- Suite for authenticating & encrypting IP traffic (IPv4 & IPv6)
- **Tunnel Mode**
  - Encrypts entire original IP packet (header + payload)
  - Encapsulated in a new IP header (gateway-to-gateway)
  - Foundation of VPNs & network-to-network security
  - Firewall challenge: port/header info hidden
- **Transport Mode**
  - Encrypts only packet payload; original header remains clear
  - Maintains compatibility with existing firewall rules
- Default implementations favor Transport Mode for filtering ease; use Tunnel Mode when full encapsulation is required.

Security Certification Progression Chart 2020



STandards


ISO
HIPAA
FINRA
Patent Law
Cyber Security Standards
PCI DSS
IT Act
GDPR
Copyright Act
IPR
NIST CSF

Sources of sec infos

- BROWSE, READ AND **SUBSCRIBE TO MAILING LIST!**

When your site has been attacked

- **Incident Response: Stay Calm & Act Deliberately**
- **Panicking** wastes time; assume breach occurred earlier
- Focus on **containment over blame**
- Maintain regular **backups** as **routine practice**
- Follow your incident response plan:
  - Identify & assess impact
  - Contain & eradicate threat
  - Recover systems
  - Review & improve procedures