Cheat Sheet for comprehensive CIW Web Security Associate

Network Security Fundamentals

Common Network Threats

- **Malware**: Viruses, worms, Trojans, ransomware
- **Phishing**: Fake emails or websites to steal credentials
- **Man-in-the-Middle (MitM)**: Intercepting communication between two parties
- **Denial of Service (DoS)**: Overloading resources to make them unavailable
- **Distributed DoS (DDoS)**: Coordinated DoS attacks from multiple sources

Network Security Protocols

- **SSL/TLS**: Secure Sockets Layer/Transport Layer Security
  - Encrypts data between web browsers and servers
- **IPsec**: Internet Protocol Security
  - Provides secure communication over IP networks
- **SSH**: Secure Shell
  - Encrypted network protocol for secure data communication
- **HTTPS**: HTTP Secure
  - Uses SSL/TLS to secure HTTP communications

Firewalls

- **Types**:
  - **Packet-Filtering**: Filters packets based on source/destination IP, port, and protocol
  - **Stateful Inspection**: Tracks connection state and filters packets based on context
  - **Proxy Firewalls**: Acts as an intermediary between internal and external networks
- **Configuration**:
  - Define rules for inbound and outbound traffic
  - Regularly update rules and monitor logs

Web Application Security

Common Web Vulnerabilities

- **SQL Injection**: Injecting malicious SQL queries
- **Cross-Site Scripting (XSS)**: Injecting client-side scripts into web pages
- **Cross-Site Request Forgery (CSRF)**: Forcing users to execute unwanted actions
- **Session Hijacking**: Stealing session IDs to impersonate users
- **Insecure Direct Object References (IDOR)**: Direct access to objects based on user input

Secure Coding Practices

- **Input Validation**: Sanitize and validate all user inputs
- **Parameterized Queries**: Use prepared statements to prevent SQL injection
- **Output Encoding**: Encode output to prevent XSS
- **Least Privilege**: Grant minimum permissions necessary
- **Error Handling**: Avoid exposing detailed error messages to users

Web Application Firewalls (WAF)

- **Functionality**:

  - Filters, monitors, and blocks HTTP traffic to and from a web application

- **Configuration**:

  - Define rules to block common attacks (e.g., SQL injection, XSS)
  - Regularly update rules and monitor logs

Authentication and Authorization

Authentication Methods

- **Multi-Factor Authentication (MFA)**: Combines two or more verification factors
- **Single Sign-On (SSO)**: Single authentication for multiple applications
- **Biometric Authentication**: Uses physical traits (e.g., fingerprint, facial recognition)
- **Token-Based Authentication**: Uses tokens (e.g., JWT) for session management

Authorization Models

- **Role-Based Access Control (RBAC)**: Access based on roles

- **Attribute-Based Access Control (ABAC)**: Access based on attributes (e.g., user, resource, environment)

- **Access Control Lists (ACL)**: List of permissions attached to an object

Password Management

- **Password Policies**:

- Minimum length, complexity requirements
- Regular password changes

- **Password Hashing**:

- Use strong hashing algorithms (e.g., bcrypt, Argon2)
- Salt passwords to prevent rainbow table attacks

- **Password Managers**:

- Store and generate strong passwords

Data Security and Privacy

Data Encryption

- **Symmetric Encryption**: Single key for encryption and decryption

- Examples: AES, DES

- **Asymmetric Encryption**: Public and private keys

- Examples: RSA, ECC

- **Hybrid Encryption**: Combines symmetric and asymmetric methods

Data Integrity

- **Hash Functions**:

- Generate fixed-size hash values (e.g., SHA-256)
- Use for data integrity checks

- **Digital Signatures**:

- Verify authenticity and integrity of data
- Use asymmetric encryption

Data Privacy Regulations

- **GDPR**: General Data Protection Regulation (EU)

- Rights of individuals, data protection principles

- **CCPA**: California Consumer Privacy Act (USA)
  - Consumer rights, data protection requirements
- **HIPAA**: Health Insurance Portability and Accountability Act (USA)
  - Privacy and security of health information

Security Policies and Procedures

Security Policies

- **Acceptable Use Policy (AUP)**: Defines acceptable use of IT resources
- **Password Policy**: Guidelines for creating and managing passwords
- **Incident Response Policy**: Procedures for handling security incidents
- **Data Classification Policy**: Classifies data based on sensitivity

Incident Response

- **Incident Identification**: Detect and identify security incidents
- **Incident Containment**: Limit the impact of an incident
- **Incident Eradication**: Remove the root cause of the incident
- **Incident Recovery**: Restore systems and data
- **Incident Lessons Learned**: Review and document lessons from incidents

Security Awareness Training

- **Phishing Simulations**: Train users to recognize phishing attempts
- **Secure Coding Practices**: Educate developers on secure coding
- **Incident Response Drills**: Practice incident response procedures
- **Regular Updates**: Keep training materials current with latest threats

Tools and Technologies

Security Tools

- **Vulnerability Scanners**: Identify security weaknesses (e.g., Nessus, OpenVAS)
- **Penetration Testing Tools**: Simulate attacks (e.g., Metasploit, Burp Suite)
- **SIEM**: Security Information and Event Management (e.g., Splunk, ELK Stack)

- **IDS/IPS**: Intrusion Detection/Prevention Systems (e.g., Snort, Suricata)

Encryption Tools

- **OpenSSL**: Command-line tool for SSL/TLS encryption
- **GPG**: GNU Privacy Guard for encryption and signing
- **VeraCrypt**: Disk encryption software

Monitoring and Logging

- **Log Management**: Centralize and analyze logs (e.g., ELK Stack, Splunk)
- **Real-Time Monitoring**: Monitor network and system activity in real-time (e.g., Nagios, Zabbix)
- **Alerting**: Set up alerts for suspicious activities (e.g., email, SMS)

Best Practices

Regular Updates

- **Patch Management**: Regularly update software and systems
- **Security Patches**: Apply security patches promptly

Backup and Recovery

- **Regular Backups**: Schedule regular backups of critical data
- **Offsite Storage**: Store backups in a secure, offsite location
- **Recovery Testing**: Regularly test backup recovery procedures

Continuous Monitoring

- **Log Analysis**: Regularly review and analyze logs
- **Threat Intelligence**: Stay updated on latest threats and vulnerabilities
- **Incident Response**: Be prepared to respond to incidents quickly

Examples

SQL Injection Prevention

```
-- Bad: Vulnerable to SQL Injection
SELECT * FROM users WHERE username = '$username' AND password =
'$password';

-- Good: Using Parameterized Queries
SELECT * FROM users WHERE username = ? AND password = ?;
```

XSS Prevention

```
<!-- Bad: Vulnerable to XSS -->
<div>Welcome, <?php echo $_GET['username']; ?></div>

<!-- Good: Encode Output -->
<div>Welcome, <?php echo htmlspecialchars($_GET['username'],
ENT_QUOTES, 'UTF-8'); ?></div>
```

Password Hashing

```
import bcrypt

password = b"mysecretpassword"
hashed = bcrypt.hashpw(password, bcrypt.gensalt())

if bcrypt.checkpw(password, hashed):
    print("Password matches!")
else:
    print("Password does not match.")
```

Summary

- **Network Security**: Protect networks from threats using firewalls, protocols, and monitoring.

- **Web Application Security**: Secure web applications by preventing common vulnerabilities.

- **Authentication and Authorization**: Implement strong authentication and authorization mechanisms.

- **Data Security and Privacy**: Protect data through encryption, integrity checks, and compliance with regulations.

- **Security Policies and Procedures**: Establish and enforce security policies and incident response procedures.

- **Tools and Technologies**: Utilize security tools for vulnerability scanning, monitoring, and encryption.

- **Best Practices**: Follow best practices for updates, backups, and continuous monitoring.

By Ahmed Baheeg Khorshid

ver 1.0