Cheat Sheet for comprehensive CIW Web Security Professional

Network Security Fundamentals

Common Network Threats

- **Malware**: Viruses, worms, Trojans, ransomware
- **Phishing**: Email scams to steal sensitive information
- **Man-in-the-Middle (MitM)**: Intercepting communication between two parties
- **Denial of Service (DoS)**: Overloading a system to make it unavailable
- **Distributed DoS (DDoS)**: Coordinated DoS attack from multiple sources

Network Security Protocols

- **SSL/TLS**: Secure Sockets Layer/Transport Layer Security
  - **Ports**: 443 (HTTPS)
- **IPSec**: Internet Protocol Security
  - **Modes**: Transport, Tunnel
- **SSH**: Secure Shell
  - **Ports**: 22
- **VPN**: Virtual Private Network
  - **Types**: Remote Access, Site-to-Site

Web Application Security

Common Web Vulnerabilities

- **SQL Injection**: Injecting SQL commands into input fields
- **Cross-Site Scripting (XSS)**: Injecting malicious scripts into web pages
- **Cross-Site Request Forgery (CSRF)**: Forcing users to execute unwanted actions
- **Session Hijacking**: Stealing session IDs to impersonate users
- **Clickjacking**: Trick users into clicking on hidden elements

Security Best Practices

- **Input Validation**: Sanitize and validate all user inputs
- **Output Encoding**: Encode data to prevent injection attacks
- **Use Prepared Statements**: For SQL queries to prevent injection
- **Implement CSRF Tokens**: Protect against CSRF attacks
- **Use HTTPS**: Encrypt data in transit

Authentication and Authorization

Authentication Methods

- **Multi-Factor Authentication (MFA)**: Combines two or more factors (e.g., password + OTP)
- **Single Sign-On (SSO)**: One login for multiple systems
- **Biometric Authentication**: Fingerprint, facial recognition, etc.
- **Certificate-Based Authentication**: Digital certificates for identity verification

Authorization Models

- **Role-Based Access Control (RBAC)**: Access based on roles (e.g., admin, user)
- **Attribute-Based Access Control (ABAC)**: Access based on attributes (e.g., department, location)
- **Mandatory Access Control (MAC)**: Strict access control by the system
- **Discretionary Access Control (DAC)**: Access control by the resource owner

Cryptography and Data Protection

Encryption Types

- **Symmetric Encryption**: Same key for encryption and decryption
  - **Algorithms**: AES, DES, 3DES
- **Asymmetric Encryption**: Different keys for encryption and decryption
  - **Algorithms**: RSA, ECC
- **Hash Functions**: One-way encryption
  - **Algorithms**: SHA-256, MD5

Key Management

- **Key Generation**: Securely generate keys
- **Key Distribution**: Safely share keys
- **Key Storage**: Securely store keys
- **Key Rotation**: Regularly change keys

Security Policies and Compliance

Security Policies

- **Acceptable Use Policy (AUP)**: Defines acceptable use of IT resources
- **Password Policy**: Guidelines for creating strong passwords
- **Data Classification**: Classify data based on sensitivity (e.g., public, confidential)
- **Incident Response Plan**: Plan for handling security incidents

Compliance Standards

- **GDPR**: General Data Protection Regulation (EU)
- **HIPAA**: Health Insurance Portability and Accountability Act (USA)
- **PCI DSS**: Payment Card Industry Data Security Standard
- **SOC 2**: Service Organization Control 2 (Trust Services Criteria)

Incident Response and Forensics

Incident Response Steps

1. **Preparation**: Develop an incident response plan
2. **Identification**: Detect and identify security incidents
3. **Containment**: Limit the scope of the incident
4. **Eradication**: Remove the root cause of the incident
5. **Recovery**: Restore affected systems
6. **Lessons Learned**: Review and improve the response process

Digital Forensics

- **Data Acquisition**: Collect evidence without altering it
- **Data Analysis**: Analyze data to determine the cause of the incident

- **Reporting**: Document findings and create a report
- **Chain of Custody**: Maintain a record of evidence handling

Security Tools and Technologies

Firewalls

- **Types**: Packet-filtering, Stateful Inspection, Next-Generation
- **Configuration**: Define rules to allow/deny traffic
- **Monitoring**: Log and analyze traffic

Intrusion Detection Systems (IDS)

- **Types**: Network-based, Host-based
- **Alerts**: Notify of suspicious activities
- **Logging**: Record detected incidents

Intrusion Prevention Systems (IPS)

- **Types**: Network-based, Host-based
- **Blocking**: Automatically block detected threats
- **Monitoring**: Continuous monitoring of network traffic

Security Information and Event Management (SIEM)

- **Data Aggregation**: Collect logs from various sources
- **Correlation**: Analyze logs to detect patterns
- **Reporting**: Generate reports on security incidents

Practical Tips and Tricks

Password Management

- **Use Strong Passwords**: Combination of letters, numbers, and symbols
- **Password Managers**: Tools like LastPass, 1Password
- **Regularly Change Passwords**: Periodically update passwords

Secure Coding Practices

- **Code Reviews**: Peer review code for vulnerabilities
- **Static Analysis Tools**: Analyze code for security issues

- **Dynamic Analysis Tools**: Test running applications for vulnerabilities

Regular Updates and Patching

- **Operating Systems**: Regularly update OS for security patches
- **Applications**: Keep software up-to-date
- **Firewalls and Antivirus**: Update definitions and rules

Examples

SQL Injection Prevention

```
-- Bad Practice
SELECT * FROM users WHERE username = '$_POST['username']';

-- Good Practice
SELECT * FROM users WHERE username = ?;
```

XSS Prevention

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

<!-- Good Practice -->
<div>Welcome, <?php echo htmlspecialchars($_GET['username']); ?></div>
```

CSRF Token Implementation

```
<form action="submit.php" method="POST">
  <input type="hidden" name="csrf_token" value="<?php echo
$_SESSION['csrf_token']; ?>">
  <!-- Other form fields -->
</form>
```

Summary

- **Network Security**: Protect networks from threats using protocols like SSL/TLS, IPSec, and SSH.
- **Web Application Security**: Prevent vulnerabilities like SQL Injection, XSS, and CSRF.
- **Authentication and Authorization**: Implement strong authentication methods and access control models.
- **Cryptography**: Use encryption and secure key management practices.

- **Security Policies**: Develop and enforce security policies and comply with standards.

- **Incident Response**: Have a plan for responding to and recovering from security incidents.

- **Tools**: Utilize firewalls, IDS/IPS, and SIEM for monitoring and protection.

- **Practical Tips**: Follow best practices for password management, secure coding, and regular updates.

By Ahmed Baheeg Khorshid

ver 1.0