**Lernzettel 9: Websicherheit**

**1. Grundlagen der Websicherheit**

* **Komponenten:** Websicherheit umfasst die Sicherheit von Clients/Browsers und Webservern. Sie deckt Aspekte der Netzwerksicherheit und Softwaresicherheit ab, insbesondere zum Schutz vor Code-Injection-Angriffen.
* **Informationsfluss:** Web-Interaktionen laufen über den Browser, der Anfragen an den Webserver sendet. Dieser kommuniziert gegebenenfalls mit einem Back-End-Server, um Anfragen zu verarbeiten und Ergebnisse zurückzusenden.

**Lernbrücke:** Websicherheit erfordert eine nahtlose Integration von Sicherheitsmaßnahmen auf allen Ebenen, von der Netzwerkschicht bis hin zur Anwendungsschicht.

**2. Authentifizierte Sitzungen**

* **Sitzungsverwaltung:** Sitzungen basieren in der Regel auf TCP, jedoch sind HTTP-Sitzungen von Natur aus nicht authentifiziert. Authentifizierte Sitzungen können auf verschiedenen Schichten implementiert werden, z.B. mit SSL/TLS auf der Transportschicht.
* **Session-Kennungen (SID):** Server generieren eine SID, die über Cookies, URI-Query-Strings oder POST-Parameter an den Client übermittelt wird. Der Client sendet diese SID bei nachfolgenden Anfragen, um die Sitzung aufrechtzuerhalten.
* **Cookie Vergiftung:** Angriffe wie das Erraten oder Stehlen von Cookies zielen darauf ab, Berechtigungen zu erhöhen. SIDs müssen daher unvorhersehbar und sicher gespeichert werden.

**Lernbrücke:** Authentifizierte Sitzungen sind ein kritischer Bestandteil der Websicherheit, da sie sicherstellen, dass nur legitime Benutzer Zugriff auf geschützte Ressourcen haben.

**3. Code-Ursprungs Politik**

* **Ziel:** Schutz von Anwendungen und Sitzungs-Kennungen vor externen Angreifern. Skripte dürfen nur mit der Domain interagieren, von der sie stammen (same-origin policy).
* **Gleicher Ursprung:** Zwei URLs haben denselben Ursprung, wenn sie das gleiche Protokoll, den gleichen Hostnamen und die gleiche Portnummer teilen.

**Lernbrücke:** Die Code-Ursprungs-Politik schützt vor Angriffen, die darauf abzielen, die Interaktionen zwischen verschiedenen Domains zu manipulieren.

**4. Cross-Site Scripting (XSS)**

* **Grundlagen:** XSS-Angriffe nutzen das Vertrauen eines Clients in einen Server aus, indem sie schädlichen Code in eine vertrauenswürdige Webseite einschleusen.
* **Typen:**
  + **Reflektiertes XSS:** Das Skript wird auf der Seite eines Angreifers gehostet und an den vertrauenswürdigen Server gesendet.
  + **Gespeichertes XSS:** Das Skript wird direkt auf dem vertrauenswürdigen Server gespeichert und bei einem Aufruf durch den Client ausgeführt.
* **Cookie-Diebstahl:** XSS kann verwendet werden, um Cookies auszulesen und an den Angreifer zu senden.

**Lernbrücke:** XSS-Angriffe können durch strikte Eingabevalidierung und die Anwendung von Sicherheitsrichtlinien verhindert werden, die sicherstellen, dass nur vertrauenswürdiger Code ausgeführt wird.

**5. Cross-Site Request Forgery (CSRF)**

* **Grundlagen:** Bei CSRF-Angriffen stellt ein Angreifer Anfragen auf einer Zielwebseite mit den Privilegien eines authentifizierten Nutzers.
* **Angriffstypen:**
  + **Reflektierter Angriff:** Ein Angreifer erstellt eine Webseite, die versteckte Aktionen auf einer Zielseite durchführt, während der Benutzer in einer aktiven Sitzung ist.
  + **Gespeicherter Angriff:** Eine bösartige Seite auf dem Server speichert schädlichen Code, der beim Aufruf durch den Client ausgeführt wird.
* **Verteidigung:** Aktionen müssen richtig authentifiziert werden, z.B. durch temporäre Geheimnisse oder einzigartige Authentifikatoren.

**Lernbrücke:** CSRF-Angriffe umgehen die ursprungsbasierte Sicherheit und erfordern daher spezifische Gegenmaßnahmen, um sicherzustellen, dass nur legitime Anfragen ausgeführt werden.