HTTP - Cookies

Description

Cette vulnérabilité repose sur l'utilisation de cookies HTTP pour stocker des informations d'identification, d'état ou d'autres données critiques sur les systèmes clients. Il existe différentes formes d'exploitation de cette vulnérabilité.

- La première forme consiste à accéder aux cookies HTTP pour extraire les données potentiellement sensibles qu'ils contiennent.
- La seconde forme consiste à intercepter ces données lorsqu'elles sont transmises du client au serveur. Les informations interceptées sont ensuite utilisées par l'adversaire pour se faire passer pour l'utilisateur ou la session à distance.
- La troisième forme est celle où le contenu du cookie est modifié par l'adversaire avant qu'il soit renvoyé au serveur. Dans ce cas, l'adversaire cherche à convaincre le serveur cible de fonctionner sur la base de ces informations falsifiées.

Prérequis d'exploitation

Pour exploiter cette vulnérabilité, il est nécessaire d'avoir accès à une application se basant sur la gestion de cookies de session pour fonctionner. Ces derniers doivent être présents au niveau des requêtes HTTP entre le client et le serveur.

Connaissances nécessaires

- Maîtrise du protocole HTTP (notion d'en-tête Cookie) ;
- Maîtrise de la notion de session utilisateur ;
- Connaissances de base sur des outils de modification et/ou d'interception de requêtes (Burp, Curl).

Outils nécessaires

- Avoir accès à la console développeur du navigateur (Firefox, Chrome) et à des outils de modification et/ou d'interception de requêtes (Burp, Curl).

Flux d'exécution

Explorer

Naviguer sur l'application afin de générer un cookie. Il peut s'agir d'un utilisateur final légitime souhaitant élever ses privilèges ou d'une personne écoutant le réseau pour obtenir un cookie HTTP.

Expérimenter

Tenter d'obtenir des informations sensibles à partir du cookie en essayant de le décoder par diverses méthodes (selon son encodage). Essayer de modifier ou de remplacer les valeurs des attributs que contient le cookie afin de contourner les contrôles de sécurité de l'application.

Exploiter

Conséquences potentielles

Une exploitation réussie de ce type de vulnérabilité peut permettre :

- L'élévation de ses privilèges vers un compte disposant de plus de droits (admin) ;
- L'accès à un espace restreint de l'application.

Contres-mesures

Les contre-mesures suivantes peuvent être mises en œuvre :

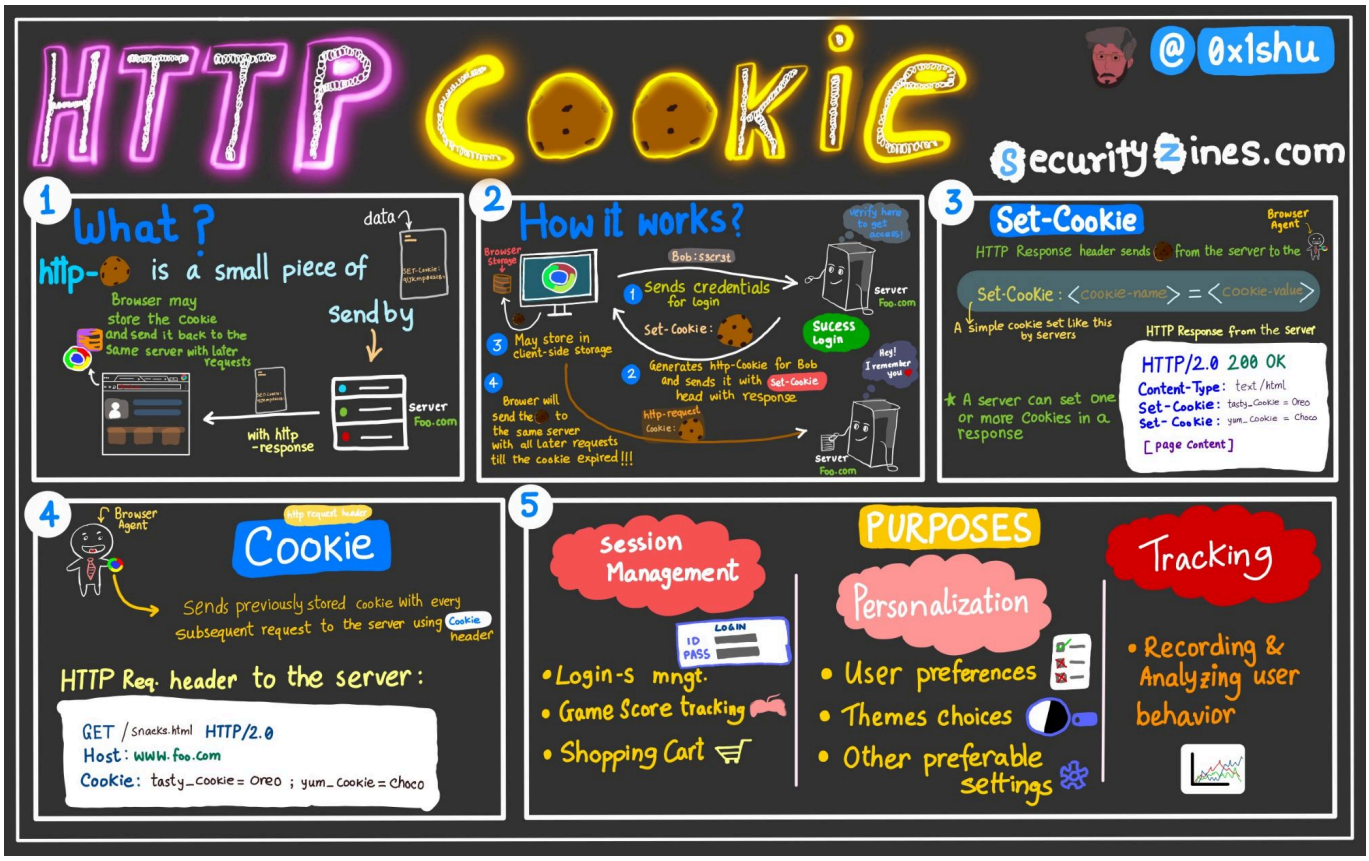
- Utilisez la validation des entrées pour les cookies ;

- Utilisez SSL/TLS pour protéger le cookie en transit (protocole HTTP) ;
- Assurez-vous que le serveur web met en œuvre tous les correctifs de sécurité pertinents.

Comment cela fonctionne

Le scénario suivant peut être joué via l'exploitation de cette vulnérabilité :

- Un attaquant parvient à intercepter les requêtes HTTP d'une application à l'aide d'une attaque de type MiTM (Man in The Middle). Il parvient alors à récupérer les cookies d'un utilisateur et se connecte avec ce cookie. Il peut alors usurper son identité et lancer une attaque de type phishing afin de récupérer ou pivoter sur d'autres comptes utilisateurs ou administrateurs de l'application en se faisant passer pour l'utilisateur en question.



Exemple 1

Voici un exemple de requête HTTP simple avec un cookie contenant une donnée sensible :

```
GET /page.php HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:80.0) Gecko/20100101 Firefox/80.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Cookie: sessionID=abc1234; username=john.doe; password=MySensitivePassword
```

Dans cette requête, nous envoyons un cookie nommé "password" qui contient une donnée sensible, en l'occurrence le mot de passe "MySensitivePassword".

Il est important de noter que l'envoi de données sensibles (par exemple des mots de passe) dans des cookies est à proscrire car ils peuvent être facilement interceptés par de potentiels attaquants.

Exemple 2

L'extrait de code suivant récupère la valeur d'un cookie de navigateur pour déterminer le rôle du visiteur. Il est facile pour un attaquant de modifier la valeur “role” trouvée dans le cookie stocké localement, permettant ainsi une potentielle escalade des privilèges.

```
Cookie[] cookies = request.getCookies();
for (int i =0; i< cookies.length; i++) {
    Cookie c = cookies[i];
    if (c.getName().equals("role")) {
        userRole = c.getValue();
    }
}
```

Exemple 3

Dans l'exemple suivant, un cookie nommé “authenticated” est utilisé pour déterminer si un utilisateur doit être autorisé ou non à accéder à un système.

La modification de la valeur d'un cookie du côté client est triviale, mais de nombreux développeurs supposent que les cookies sont essentiellement immuables.

```
boolean authenticated = new Boolean(getCookieValue("authenticated")).booleanValue();
if (authenticated) {
    ...
}
```

CWEs

- CWE-565 : Reliance on Cookies without Validation and Integrity Checking
  - The application relies on the existence or values of cookies when performing security-critical operations, but it does not properly ensure that the setting is valid for the associated user.
- CWE-302 : Authentication Bypass by Assumed-Immutable Data
  - The authentication scheme or implementation uses key data elements that are assumed to be immutable, but can be controlled or modified by the attacker.
- CWE-311 : Missing Encryption of Sensitive Data
  - The software does not encrypt sensitive or critical information before storage or transmission.
- CWE-113 : Improper Neutralization of CRLF Sequences in HTTP Headers ('HTTP Request/Response Splitting')
  - The software receives data from an HTTP agent/component (e.g., web server, proxy, browser, etc.), but it does not neutralize or incorrectly neutralizes CR and LF characters before the data is included in outgoing HTTP headers.
- CWE-539 : Use of Persistent Cookies Containing Sensitive Information
  - The web application uses persistent cookies, but the cookies contain sensitive information.
    - CWE-20 : Improper Input Validation
      - The product receives input or data, but it does not validate or incorrectly validates that the input has the properties that are required to process the data safely and correctly.
- CWE-315 : Cleartext Storage of Sensitive Information in a Cookie
  - The application stores sensitive information in cleartext in a cookie.
- CWE-384 : Session Fixation
  - Authenticating a user, or otherwise establishing a new user session, without invalidating any existing session identifier gives an attacker the opportunity to steal authenticated sessions.
- CWE-472 : External Control of Assumed-Immutable Web Parameter
  - The web application does not sufficiently verify inputs that are assumed to be immutable but are actually externally controllable, such as hidden form fields.
- CWE-602 : Client-Side Enforcement of Server-Side Security
  - The software is composed of a server that relies on the client to implement a mechanism that is intended to protect the server.
- CWE-642 : External Control of Critical State Data
  - The software stores security-critical state information about its users, or the software itself, in a location that is accessible to unauthorized actors.

References

URL :

- https://repository.root-me.org/Exploitation%20-%20Web/EN%20-%20Cookies,%20sessions,%20and%20persistence.pdf
- https://www.invicti.com/blog/web-security/understanding-cookie-poisoning-attacks/
- https://developer.mozilla.org/fr/docs/Web/HTTP/Cookies

- https://sulimanalruz.com/en/Ultimate-Guide-to-HTTP-Cookie-Security-Attacks-Prevention-and-Best-Practises/
- https://www.sitedetout.com/les-cookies-http/

Retour fiches vulnérabilités

- Cyber fiches vulnérabilités