LDAP - Injection

Description

Les injections LDAP sont une forme d'attaque qui exploitent des vulnérabilités dans l'implémentation du protocole LDAP. Elles consistent à insérer du code malveillant dans les requêtes LDAP à destination de l'annuaire. L'objectif peut être de contourner les contrôles de sécurité et d'accéder à des informations sensibles. Ces injections peuvent être effectuées via des formulaires présents sur des applications web ou des interfaces de gestion d'annuaire.

LDAP (Lightweight Directory Access Protocol) est un protocole utilisé pour interroger une base d'annuaire. Ce dernier définit également un langage de requêtes permettant de communiquer avec les données présentes dans l'annuaire. Ces requêtes peuvent par exemple être utilisées dans des systèmes d'authentification utilisateurs (login + mot de passe) ou bien dans des applications afin de récupérer des informations.

De la même manière qu'il est parfois possible d'effectuer des injections SQL sur une application vulnérable pour lire le contenu d'une base de données ou bypasser une authentification, des applications peuvent être vulnérables aux injections LDAP.

Les principales différences entre les requêtes LDAP et SQL sont les suivantes :

- **La syntaxe** : les requêtes LDAP et SQL utilisent une syntaxe différente pour interroger leurs sources de données respectives ;
- **Le type de source de données** : LDAP est généralement utilisé pour interroger des annuaires de services, tandis que SQL est utilisé pour interroger des bases de données relationnelles ;
- **Les données renvoyées** : les requêtes LDAP renvoient des objets qui représentent des entités dans l'annuaire, tandis que les requêtes SQL renvoient des enregistrements de données ;
- **Les opérations possibles** : LDAP est principalement utilisé pour la recherche et la récupération de données, tandis que SQL permet également la modification et la suppression de données.

Pré-requis d'exploitation

Pour exploiter cette vulnérabilité, il est nécessaire d’avoir accès à une application qui permet de soumettre des requêtes LDAP.

Connaissances nécessaires

- Connaissance approfondie du protocole LDAP.

Outils nécessaires

- Utilisation d’outils de type Burp Suite ou curl pour la création/modification de requêtes HTTP ;
- Outils de fuzzing avec des wordlists spécialisées aux requêtes LDAP.

Flux d'exécution

Explorer

Naviguer sur l’application pour identifier les points d'entrées potentiellement vulnérables à des injections LDAP.

Expérimenter

Envoyer des requêtes LDAP pour tenter de schématiser la structure de l'annuaire LDAP cible. Cette étape permettra à l'attaquant de pouvoir écrire des requêtes qui contourneront les contrôles de sécurité mis en place.

Exploiter

Conséquences potentielles

Une exploitation réussie de ce type de vulnérabilité peut permettre :

- L'accès à des informations sensibles stockées dans un annuaire LDAP, telles que les mots de passe et les détails de connexion d'un

utilisateur ;
- La modification ou la suppression des informations dans l'annuaire LDAP, ce qui peut entraîner des perturbations importantes dans le fonctionnement de l'organisation ;
- La création de nouvelles entrées dans l'annuaire LDAP, ce qui peut être utilisé pour diffuser de l'information trompeuse ou pour accéder à des ressources sensibles de l'organisation.

Contre-mesures

Les contre-mesures suivantes peuvent être mises en œuvre :

- Filtrer les entrées de l'utilisateur avant de les utiliser pour construire des requêtes LDAP ;
- Utiliser des mots de passe forts et uniques pour chaque compte d'utilisateur ;
- Configurer les pare-feux et autres dispositifs de sécurité pour bloquer les connexions non autorisées à l'annuaire LDAP ;
- Appliquer des contrôles d'accès pour limiter l'accès aux ressources sensibles aux seuls utilisateurs autorisés.

Comment cela fonctionne

Les scénarios suivants peuvent être joués via l'exploitation de cette vulnérabilité :

- Un attaquant utilise une injection LDAP pour accéder à des informations sensibles, telles que les mots de passe et les détails de connexion d'un utilisateur ;
- Un attaquant utilise une injection LDAP pour modifier ou supprimer des informations dans l'annuaire LDAP.

Exemple 1

Voici l'exemple d'une application Python qui réalise une requête LDAP à destination d'un annuaire :

```python
import ldap

# Connexion au serveur LDAP
ldap_conn = ldap.initialize("ldap://ldap.example.com")

# Recherche d'utilisateur
search_filter = "(uid=%s)" % user_input
result = ldap_conn.search_s("dc=example,dc=com", ldap.SCOPE_SUBTREE, search_filter)

# Affichage des résultats
for dn, entry in result:
    print(entry)

# Fermeture de la connexion
ldap_conn.unbind()
```

La variable user_input est utilisée pour construire le filtre de recherche LDAP. Si un utilisateur malveillant fournit une entrée malicieuse dans cette variable, il peut alors être en mesure de lister tous les utilisateurs de l'annuaire ou accéder à des informations sensibles.

Exemple 2

Voici un code vulnérable qui effectue une authentification via un annuaire LDAP :

```php
$ldap_server = "ldap.example.com";
$ldap_dn = "ou=users,dc=example,dc=com";
$username = $_POST['username'];
$password = $_POST['password'];

$ldap_conn = ldap_connect($ldap_server);
ldap_set_option($ldap_conn, LDAP_OPT_PROTOCOL_VERSION, 3);

$bind = ldap_bind($ldap_conn, $username, $password);
$filter = "(&(objectclass=user)(uid=$username))";
$search = ldap_search($ldap_conn, $ldap_dn, $filter);
```

```
$entries = ldap_get_entries($ldap_conn, $search);
```

Dans le cas ou un utilisateur s'authentifierait de manière légitime, la requête LDAP générée par le code ci dessus serait :

```
(&(objectclass=user)(uid=admin)(userPassword=password123))
```

Si un utilisateur malveillant insère le payload “admin)(&))” dans le champ username, et “toto” dans le champ password, alors la requête LDAP générée sera :

```
(&(objectclass=user)(uid=admin)(&))(userPassword=toto))
```

L'insertion permet de bypasser le contrôle sur le mot de passe car le serveur LDAP va interpréter seulement le premier filtre et retourner les informations de l'utilisateur admin, sans évaluer la condition sur le mot de passe.

Exemple 3

Voici un exemple de requête LDAP pour rechercher tous les utilisateurs d'un domaine avec un nom de famille spécifique :

```
(&(objectCategory=user)(sn=Smith))
```

Cela renverra tous les utilisateurs dont le nom de famille est “Smith”.

En comparaison, voici un exemple de requête SQL réalisant une recherche similaire :

```
SELECT * FROM users WHERE last_name = 'Smith';
```

CWEs

- CWE-116 : Improper Encoding or Escaping of Output
- The software prepares a structured message for communication with another component, but encoding or escaping of the data is either missing or done incorrectly. As a result, the intended structure of the message is not preserved.
- CWE-90 : Improper Neutralization of Special Elements used in an LDAP Query ('LDAP Injection')
- The software constructs all or part of an LDAP query using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended LDAP query when it is sent to a downstream component.

References

URL :

- https://repository.root-me.org/Exploitation%20-%20Web/EN%20-%20Blackhat%20Europe%202008%20%20-%20LDAP%20Injection%20&%20Blind%20LDAP%20Injection.pdf
- https://cheatsheetseries.owasp.org/cheatsheets/LDAP_Injection_Prevention_Cheat_Sheet.html

Retour fiches vulnérabilités

- Cyber fiches vulnérabilités