Description

sqlmap_est_un_outil_open_source_qui_permet_d_automatiser_la_detection_et_l_exploitation_de_vulnerabilites_de_type_injection_sql._il_permet_ainsi_de_tester_la_securite_des_bases_de_donnees_utilisees_par_les_applications_web_en_identifiant_les_potentielles_failles_de_securite_sur_celles-ci

installation}

Voici le guide d'installation pour l'outil SQLMap :

-* telechargement_du_paquet_git_afin_de_cloner_le_depot_github_de_sqlmap - **Sous Linux (Debian/Ubuntu) : <code class="bash">sudo apt-get install git</code> -** Sous Linux (Red Hat/Fedora) :

```
sudo yum install git
```

-* clonage_du_depot_github_de_sqlmap_vers_un_dossier_en_local_sqlmap-latest

```
git clone --depth 1 https://github.com/sqlmapproject/sqlmap.git sqlmap-latest
```

-* acces_et_lancement_de_sqlmap - **Se déplacer dans le répertoire précédemment créé avec la commande suivante : <code class="bash">cd sqlmap-latest</code> -** Vérifiez l'installation en lançant SQLmap avec la commande suivante :

```
python3 sqlmap.py --version
```

-** SQLmap devrait afficher sa version, ce qui confirme que l'installation s'est correctement déroulée.

cas_d_utilisation}

-* identification_de_la_structure_de_la_base_de_donnees : SQLmap peut extraire des informations sur la structure de la base de données, notamment les tables, les colonnes et les utilisateurs. Cela peut être utile pour comprendre la structure de la base de données d'une application cible. -* exploitation_de_base_de_donnees : SQLmap permet la détection d'injections SQL dans une application web, il analyse les entrées utilisateur pour rechercher des méthodes d'exploitations appropriées en fonction du type de base de données, des paramètres de celle-ci, etc. -* tests_et_contournement_d_authentification : SQLmap peut être utilisé pour tester l'authentification d'une application en tentant de contourner les mécanismes d'authentification en utilisant des techniques d'injection SQL sur des formulaires ou des accès privés par exemple.

fonctionnalites_principales}

-* detection_automatique : automatise la détection des vulnérabilités SQL en analysant les paramètres de requête HTTP, en identifiant les erreurs SQL, en effectuant des tests d'injection SQL, etc. -* prise_en_charge_de_multiples_bases_de_donnees : compatible avec une large gamme de bases de données, notamment MySQL, PostgreSQL, Microsoft SQL Server, Oracle, SQLite, et bien d'autres. -* support_de_l_injection_basee_sur_les_erreurs : peut identifier les erreurs SQL dans les réponses HTTP afin de déterminer si une vulnérabilité SQL existe et si elle peut être exploitée. -* options_de_personnalisation : offre de nombreuses options pour personnaliser les tests d'injection SQL, y compris la possibilité de spécifier des en-têtes personnalisées, des cookies, des méthodes HTTP, un User-Agent custom, etc. -* gestion_des_sessions : permet de conserver les sessions pour les applications nécessitant une authentification.

manuel_des_options_principales}

```
-h, --help : affiche l'aide et la liste des options disponibles.

-u URL, --url=URL : spécifie l'URL cible pour l'analyse.

-r FILE, --file=FILE : spécifie un fichier contenant une requête HTTP à analyser (plutôt qu'une URL).

--data=DATA : fournit des données POST pour une requête (utilisé avec -u ou -r).

--cookie=COOKIE : fournit des cookies pour l'authentification (utilisé avec -u ou -r).

--level=LEVEL : définit le niveau de tests d'injection SQL (1-5, 1 étant le moins intrusif, 5 le plus
intrusif).

--risk=RISK : définit le niveau de risque pour les tests d'injection SQL (1-3, 1 étant le moins risqué,
3 le plus risqué).

--dbms=DBMS : spécifie le type de base de données à cibler (MySQL, PostgreSQL, Microsoft SQL Server,
etc.).

--technique=TECHNIQUE : spécifie la technique d'injection SQL à utiliser (parmi une liste de techniques
spécifiques).
```

```
--banner : affiche la bannière de SQLmap lors de son démarrage.

--batch : active le mode batch, qui ne nécessite pas d'interaction utilisateur.

--dbs : affiche la liste des bases de données disponibles sur le serveur.

--tables : affiche la liste des tables dans la base de données cible.

--columns : affiche la liste des colonnes dans une table spécifiée.

--dump : extrait les données de la base de données cible.

--os-shell : active un shell interactif sur le système d'exploitation distant (lorsqu'une injection SQL
permet cela).

--tamper=TAMPER : spécifie des scripts de contournement personnalisés pour modifier les données avant
l'injection.

--user-agent=AGENT : spécifie un agent utilisateur personnalisé dans les en-têtes HTTP.

--random-agent : utilise un agent utilisateur aléatoire pour masquer la présence de SQLmap.

--tor : dirige les requêtes SQL via le réseau Tor pour l'anonymat.

--proxy=PROXY : utilise un proxy pour rediriger les requêtes.

--threads=THREADS : définit le nombre de threads à utiliser pour les tests (parallélisation).

--time-sec=TIME_SEC : définit le temps d'attente maximal pour chaque requête.

--flush-session : efface les données de session enregistrées précédemment.

--purge-output : ssupprime les données de sortie de fichiers existants.

--exclude=EXCLUDE : exclut spécifiquement une option de test (par exemple, "--exclude=users").

-v, --verbose : active le mode verbeux, qui affiche davantage d'informations de débogage.

--wizard : active le mode assistant pour guider l'utilisateur dans la configuration des tests.

--eval=CODE : évalue le code Python personnalisé pendant l'exécution de SQLmap.

--sqlshell : active le shell SQL interactif.

--sql-file=SQL_FILE : spécifie un fichier contenant des requêtes SQL personnalisées à exécuter.
```

Retour fiches outils

- Cyber fiches outils

From:
/ - **Les cours du BTS SIO**

Permanent link:
/doku.php/cyber/outils/sqlmap?rev=1751027681

Last update: **2025/06/27 14:34**