????? ? ????????

Обход файловых путей (англ. Path Traversal или Directory Traversal) – это уязвимость веб-безопасности, позволяющая злоумышленнику читать произвольные файлы на сервере, на котором запущено приложение. Сюда могут входить код приложения и данные, учетные данные для внутренних систем и конфиденциальные файлы операционной системы. В некоторых случаях злоумышленник может записать в произвольные файлы на сервере, что позволит ему изменить данные или поведение приложения, и, в конечном счете, получить полный контроль над сервером.

Появляются из-за отсутствия валидации входных данных пользователя, а также из-за отсутствия разграничения доступа приложения или функции на возможность обращаться к файловым ресурсам всего сервера.

Пример уязвимого кода:

```
<?php
$template = 'red.php';
if (isset($_COOKIE['TEMPLATE'])) {
    $template = $_COOKIE['TEMPLATE'];
}
include "/home/users/phpguru/templates/" . $template;
```

В данном примере функция include подключает файлы из директории /home/users/phpguru/templates/ , имена которых отправил пользователь в своих Cookie данных на сервер. В таком случае пользователь может изменить отправляемые им данные и добавить в данные отправляемые на сервер строку ../../../../../etc/passwd . В таком случае, сервер выполнит запрос к файлу по адресу /home/users/phpguru/templates/../../../../../etc/passwd , что, в конечном итоге, превратится в обращение к файлу /etc/passwd, и предоставит возможность чтения пользователю системных файлов ОС, к которым он не должен был иметь доступа.

Скрипт сохранения файлов из открытых директории

```
#!/bin/bash
FILE="${1}"
OUTPUT_FOLDER="${2}"

if [[ ! -s "${FILE}" ]]; then
  echo "You must provide a non-empty hosts file as an argument."
  exit 1
```

```
fi

if [[ -z "${OUTPUT_FOLDER}" ]]; then
  OUTPUT_FOLDER="data"
fi

while read -r line; do
  url=$(echo "${line}" | xargs)
  if [[ -n "${url}" ]]; then
    echo "Testing ${url} for Directory indexing..."
    if curl -L -s "${url}" | grep -q -e "Index of /" -e "[PARENTDIR]"; then
      echo -e "\t -!- Found Directory Indexing page at ${url}"
      echo -e "\t -!- Downloading to the \"${OUTPUT_FOLDER}\" folder..."
      mkdir -p "${OUTPUT_FOLDER}"
      wget -q -r -np -R "index.html*" "${url}" -P "${OUTPUT_FOLDER}"
    fi
  fi
done < <(cat "${FILE}")
```

Сканирование директорий, закрытых от индексирования в robots.txt

Можно просканировать закрытые директории, просмотрев коды ответов при обращении:

```
#!/bin/bash

TARGET_URL="http://172.16.10.12"
ROBOTS_FILE="robots.txt"

while read -r line; do
  path=$(echo "${line}" | awk -F'Disallow: ' '{print $2}')
  if [[ -n "${path}" ]] then
    url="${TARGET_URL}/${path}"
    status_code=$(curl -s -o /dev/null -w "%{http_code}" "${url}")
    echo "URL: ${url} returned a status code of: ${status_code}"
  fi
done < <(curl -s "${TARGET_URL}/${ROBOTS_FILE}")
```

Инструмент dirsearch

Поиск скрытых путей и файлов на web сервере. Отдельная база.

```
dirsearch -u http://172.16.10.10:8081/
```

--max-rate=REQUESTS Ограничение числа запросов.

Охрененный инструмент.

MSF backup_file

Ищет резервные файлы, случайно сохраненные на сервере. auxiliary/scanner/http/backup_file

MSF dir_listing

Список возможных путей. Похоже нужно указывать словарь для проверки.

Получение git репозитория (GitJacking)

Можно при возможности качнуть git проекта.

```
gitjacker http://172.16.10.11/backup/acme-impact-alliance/ -o acme-impact-alliance-git
```

Потом git log и другие инструменты поиска например файлов авторизации.

Revision #3
Created 27 September 2025 18:08:58 by Admin
Updated 6 October 2025 06:40:47 by Admin