Shell scripting - IP scanner project

Introduction

This **IP Scanner** script is designed to scan an IP address in various ways, providing useful information about the IP through *`whois`*, *`reverse DNS`*, *`HTTP/HTTPS queries`*, and *`geolocation lookup`*. The script takes an IP address as input from the user and performs these scans sequentially. This documentation outlines the core components of the script and provides details on how the script operates.

Key Components of the Script

- Whois Lookup

- The script performs a `whois` lookup for the given IP address to retrieve network-related information.
- Fields retrieved include **netname**, **descr** (description), **country**, **role**, and **abuse-mailbox** (for reporting abuse).

- Reverse DNS Lookup

- The script uses the *`host`* command to perform a reverse DNS lookup on the IP address.
- If a reverse DNS record exists, the associated **domain name** is displayed. If no record is found, the script informs the user that no reverse DNS record exists.
- This lookup helps determine if the IP address is linked to a domain name.

- HTTP and HTTPS Query

- The script sends **HTTP** and **HTTPS** requests to the IP address using `curl`, and it retrieves the **HTTP response codes**.
- **HTTP response codes** such as *`200 OK`* or *`404 Not Found`* provide insight into the availability and response status of the server at the IP address.
- Both HTTP and HTTPS queries are performed separately, and the results are displayed.

- Geolocation Lookup

- The script performs a **geolocation lookup** using the **ip-api.com** API to retrieve details about the IP's geographical location.
- The information retrieved includes **city**, **region**, **country**, **ISP**, **latitude**, and **longitude**.
- This feature provides insights into where the IP address is physically located, and which ISP is responsible for it.

Script Breakdown

1. IP Validation

- Function: `*check_ip_octets*`
  - Before running the scan, the script validates the IP format by ensuring that each octet of the IP is between `0` and `255`.
  - If the IP is invalid, an error message is displayed, and the script exits.

2. Main Process

- Function: `*process_scan*`
  - This function orchestrates the entire scanning process by calling the other functions in the following order:
    1. Whois Lookup
    2. Reverse DNS Lookup
    3. HTTP Query
    4. HTTPS Query
    5. Geolocation Lookup

3. User Input and Loop

- The script continuously prompts the user for an IP address to scan until the user types `*q*` to quit the program.
- If the inputted IP address is valid, the scan is performed, and the results are displayed. If the IP is invalid, the user is notified.

Example Usage

```
Please enter an IP address to scan (or type 'q' to quit): 8.8.8.8
Scanning IP address: 8.8.8.8

## Whois lookup ##
NetName:        GOOGLE
descr:          Google LLC
Country:        US
role:           Google Inc Network Operations
abuse-mailbox:  network-abuse@google.com

## Reverse DNS Lookup ##
dns.google.

## HTTP Query ##
HTTP response code: 301

## HTTPS Query ##
```

```
HTTPS response code: 200

## Geolocation info ##
Country: US
Region: California
City: Mountain View
ISP: Google LLC
Lat: 37.422
Lon: -122.084
```

Dependencies

- **jq**: This script uses `jq` for parsing JSON data returned from the geolocation API.
- **curl**: Required for making HTTP and HTTPS requests, as well as for interacting with the geolocation API.
- **host**: Utilized for performing reverse DNS lookups.
- **whois**: Necessary for performing whois lookups.

To install these dependencies on a Linux-based system:

```
sudo apt-get install jq curl dnsutils whois
```

Conclusion

This **IP Scanner** script is a useful tool for gathering various pieces of information about an IP address. It's modular in nature, with separate functions for each type of scan. By inputting an IP, users can quickly retrieve details such as network information, domain names, HTTP/HTTPS status codes, and geolocation data.