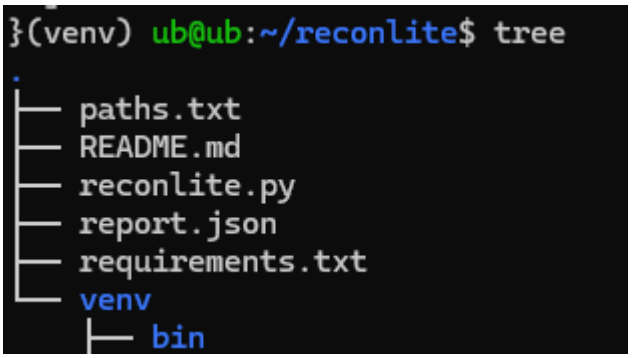
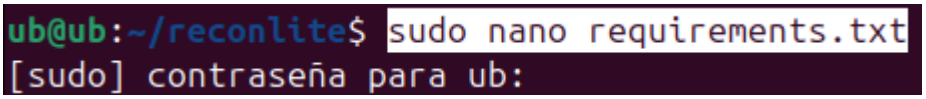
Crea carpeta:



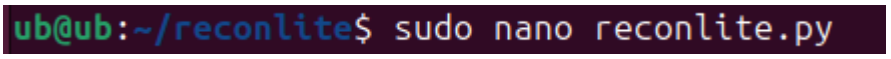
Cree el siguiente archivo:

- sudo nano requirements.txt



requests>=2.31.0

- sudo nano reconlite.py




```
#!/usr/bin/env python3
import argparse
import json
import re
import sys
import time
from urllib.parse import urljoin, urlparse

import requests


DEFAULT_PATHS = [
    "/", "/robots.txt", "/sitemap.xml",
    "/.git/", "/.env", "/config.php", "/phpinfo.php",
    "/admin", "/admin/", "/login", "/login/",
```

```
    "/wp-admin", "/wp-login.php",
    "/api", "/api/", "/swagger", "/swagger/", "/openapi.json",
    "/server-status", "/actuator", "/actuator/health"
]

SEC_HEADERS = [
    "Strict-Transport-Security",
    "Content-Security-Policy",
    "X-Content-Type-Options",
    "X-Frame-Options",
    "Referrer-Policy",
    "Permissions-Policy",
    "Cross-Origin-Opener-Policy",
    "Cross-Origin-Resource-Policy",
    "Cross-Origin-Embedder-Policy",
]


def normalize_url(raw: str) -> str:
    raw = raw.strip()
    if not raw:
        raise ValueError("URL vacía.")
    if not re.match(r"^https?://", raw, re.IGNORECASE):
        raw = "https://" + raw
    u = urlparse(raw)
    if not u.netloc:
        raise ValueError("URL inválida. Ejemplo: https://example.com")
    clean = f"{u.scheme}://{u.netloc}"
    if u.path and u.path != "/":
        clean += u.path.rstrip("/")
    return clean


def safe_request(session: requests.Session, method: str, url: str, **kwargs):
    t0 = time.time()
    try:
        r = session.request(method, url, **kwargs)
        dt = (time.time() - t0) * 1000.0
        return r, dt, None
    except requests.RequestException as e:
        dt = (time.time() - t0) * 1000.0
        return None, dt, str(e)
```

```
def extract_basic_fingerprint(headers: dict) -> dict:
    server = headers.get("Server")
    powered = headers.get("X-Powered-By")
    via = headers.get("Via")
    return {
        "server": server,
        "x_powered_by": powered,
        "via": via,
    }


def analyze_security_headers(headers: dict) -> dict:
    present = {}
    missing = []
    for h in SEC_HEADERS:
        if h in headers:
            present[h] = headers.get(h)
        else:
            missing.append(h)

    notes = []
    if "Strict-Transport-Security" not in headers:
        notes.append("No HSTS configurado.")
    if headers.get("X-Content-Type-Options", "").lower() != "nosniff":
        notes.append("X-Content-Type-Options no es 'nosniff'.")
    if "Content-Security-Policy" not in headers:
        notes.append("No hay Content-Security-Policy.")

    return {"present": present, "missing": missing, "notes": notes}


def is_interesting_status(code: int) -> bool:
    return code in (200, 201, 202, 204, 301, 302, 307, 308, 401, 403)


def scan_paths(base_url: str, paths: list, timeout: int, verify_tls: bool,
user_agent: str, max_paths: int):
    session = requests.Session()
    session.headers.update({"User-Agent": user_agent})

    results = []
    for p in paths[:max_paths]:
        full = urljoin(base_url + "/", p.lstrip("/"))
        r, dt, err = safe_request(
```

```
            session,
            "GET",
            full,
            timeout=timeout,
            allow_redirects=False,
            verify=verify_tls,
        )

        entry = {
            "path": p,
            "url": full,
            "error": err,
            "ms": round(dt, 2),
        }

        if r is not None:
            entry.update({
                "status": r.status_code,
                "content_type": r.headers.get("Content-Type"),
                "content_length": r.headers.get("Content-Length"),
                "location": r.headers.get("Location"),
            })

            if is_interesting_status(r.status_code):
                results.append(entry)
        else:
            results.append(entry)

    return results


def head_base(base_url: str, timeout: int, verify_tls: bool, user_agent: str):
    session = requests.Session()
    session.headers.update({"User-Agent": user_agent})

    r, dt, err = safe_request(
        session,
        "HEAD",
        base_url,
        timeout=timeout,
        allow_redirects=False,
        verify=verify_tls,
    )
```

```
    if r is None:
        return {"error": err, "ms": round(dt, 2)}

    return {
        "status": r.status_code,
        "ms": round(dt, 2),
        "headers": dict(r.headers),
    }


def main():
    parser = argparse.ArgumentParser(
        description="ReconLite - Recon HTTP/OSINT ligero (solo objetivos
autorizados)."
    )
    parser.add_argument("target", help="URL o dominio (ej:
https://example.com)")
    parser.add_argument("--timeout", type=int, default=8)
    parser.add_argument("--insecure", action="store_true")
    parser.add_argument("--max-paths", type=int, default=40)
    parser.add_argument("--paths-file")
    parser.add_argument("--out", default="report.json")
    parser.add_argument("--ua", default="ReconLite/1.0 (+educational)")

    args = parser.parse_args()

    try:
        base_url = normalize_url(args.target)
    except ValueError as e:
        print(f"[!] {e}", file=sys.stderr)
        sys.exit(1)

    verify_tls = not args.insecure

    paths = DEFAULT_PATHS
    if args.paths_file:
        with open(args.paths_file, "r", encoding="utf-8") as f:
            custom = [line.strip() for line in f if line.strip() and not
line.startswith("#")]
        paths = [p if p.startswith("/") else "/" + p for p in custom]

    report = {
        "target": base_url,
```

```
        "timestamp_utc": time.strftime("%Y-%m-%dT%H:%M:%SZ",
    time.gmtime()),
        "config": {
            "timeout_s": args.timeout,
            "verify_tls": verify_tls,
            "max_paths": args.max_paths,
            "user_agent": args.ua,
        },
        "base_head": {},
        "fingerprint": {},
        "security_headers": {},
        "findings": [],
    }

    base_head = head_base(base_url, args.timeout, verify_tls, args.ua)
    report["base_head"] = base_head

    if "headers" in base_head:
        headers = base_head["headers"]
        report["fingerprint"] = extract_basic_fingerprint(headers)
        report["security_headers"] = analyze_security_headers(headers)

    report["findings"] = scan_paths(base_url, paths, args.timeout, verify_tls,
    args.ua, args.max_paths)

    with open(args.out, "w", encoding="utf-8") as f:
        json.dump(report, f, indent=2, ensure_ascii=False)

    print(f"Reporte generado: {args.out}")


if __name__ == "__main__":
    main()
```

```
 GNU nano 7.2                        reconlite.py *
#!/usr/bin/env python3
import argparse
import json
import re
import sys
import time
from urllib.parse import urljoin, urlparse

import requests


DEFAULT_PATHS = [
    "/", "/robots.txt", "/sitemap.xml",
    "/.git/", "/.env", "/config.php", "/phpinfo.php",
    "/admin", "/admin/", "/login", "/login/",
```

- sudo nano README.md

```
ub@ub:~/reconlite$ sudo nano README.md
```

ReconLite (proyecto resuelto) – Python + requests con mentalidad ciber

Objetivo
Hacer reconocimiento HTTP/OSINT de forma ligera y responsable:
- Cabeceras de seguridad (HSTS, CSP, etc.)
- Fingerprinting básico por headers
- Enumeración “suave” de rutas típicas
- Reporte JSON para evidencias

> Úsalo solo contra objetivos autorizados.

Instalación
```bash
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

Uso básico
```bash
python reconlite.py https://example.com
```

Cambiar User-Agent
```bash
python reconlite.py example.com --ua "Mozilla/5.0 (ReconLite class)"
```

Añadir tu propio wordlist de rutas
Archivo `paths.txt` (una ruta por línea):
```
admin
admin/
login
login/
robots.txt
api
api/v1
```

Ejecución:
```bash
python reconlite.py https://midominio.com --paths-file paths.txt --max-paths 200
```

TLS
Por defecto verifica TLS.
Si estás en laboratorio con certificados raros (NO recomendado):
```bash
python reconlite.py https://lab.local --insecure
```

Salida
Genera `report.json` con:
- Config usada
- HEAD base con headers
- Seguridad de headers
- Listado de rutas con status, tiempos, redirects y content-type
```

```
 GNU nano 7.2   README.md *
ReconLite (proyecto resuelto) - Python + requests con mentalidad ciber

Objetivo
Hacer reconocimiento HTTP/OSINT de forma ligera y responsable:
- Cabeceras de seguridad (HSTS, CSP, etc.)
- Fingerprinting básico por headers
- Enumeración “suave” de rutas típicas
- Reporte JSON para evidencias

> Úsalo solo contra objetivos autorizados.

Instalación
```bash
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```
```

Crear un entorno virtual

- sudo apt install python3.12-venv

```
ub@ub:~/reconlite$ sudo apt install python3.12-venv
```

- python3 -m venv venv
- source venv/bin/activate
- pip install -r requirements.txt
```

```
ub@ub:~/reconlite$ python3 -m venv venv
ub@ub:~/reconlite$ source venv/bin/activate
(venv) ub@ub:~/reconlite$ pip install -r requirements.txt
```

Ejecución

Uso básico:

- python reconlite.py https://example.com

```
(venv) ub@ub:~/reconlite$ python reconlite.py https://example.com
Reporte generado: report.json
```

- sudo nano report.json

```
 GNU nano 7.2                                        report.json
{
  "target": "https://example.com",
  "timestamp_utc": "2025-12-16T12:23:15Z",
  "config": {
    "timeout_s": 8,
    "verify_tls": true,
    "max_paths": 40,
    "user_agent": "ReconLite/1.0 (+educational)"
  },
  "base_head": {
    "status": 200,
    "ms": 710.7,
    "headers": {
      "Accept-Ranges": "bytes",
      "Content-Length": "513",
      "Content-Type": "text/html",
      "ETag": "\"bc2473a18e003bdb249eba5ce893033f:1760028122.592274\"",
      "Last-Modified": "Thu, 09 Oct 2025 16:42:02 GMT",
      "Cache-Control": "max-age=86000",
      "Date": "Tue, 16 Dec 2025 12:23:15 GMT",
      "Connection": "keep-alive",
      "Alt-Svc": "h3=\":443\"; ma=93600"
    }
  },
  "fingerprint": {
```

Recon rápido:

- python reconlite.py https://testphp.vulnweb.com

```
(venv) ub@ub:~/reconlite$ python reconlite.py https://testphp.vulnweb.com
Reporte generado: report.json
```

Enumeración más grande con tu lista:
Crear paths.txt

- sudo nano paths.txt

```
(venv) ub@ub:~/reconlite$ sudo nano paths.txt
```

admin
admin/
login
login/
robots.txt

api
api/v1

```
  GNU nano 7.2                                  paths.txt
admin
admin/
login
login/
robots.txt
api
api/v1
```

Ejecutar comando

- python reconlite.py https://testphp.vulnweb.com --paths-file paths.txt --max-paths 150

```
(venv) ub@ub:~/reconlite$ python reconlite.py https://testphp.vulnweb.com --paths-file paths.txt --max-paths 150
Reporte generado: report.json
```

Cambiar User-Agent:

- python reconlite.py example.com --ua "Mozilla/5.0 (ReconLite class)"

```
(venv) ub@ub:~/reconlite$ python reconlite.py example.com --ua "Mozilla/5.0 (ReconLite class)"
Reporte generado: report.json
```

Cambiar timeout (objetivos lentos):

- python reconlite.py https://example.com --timeout 15

```
(venv) ub@ub:~/reconlite$ python reconlite.py https://example.com --timeout 15
Reporte generado: report.json
```

TLS:

- python reconlite.py https://lab.local --insecure

```
(venv) ub@ub:~/reconlite$ python reconlite.py https://lab.local --insecure
Reporte generado: report.json
```

Ver el contenido de report.json

- sudo nano report.json

{
  "target": "https://lab.local",
  "timestamp_utc": "2025-12-16T12:56:21Z",
  "config": {
    "timeout_s": 8,
    "verify_tls": false,

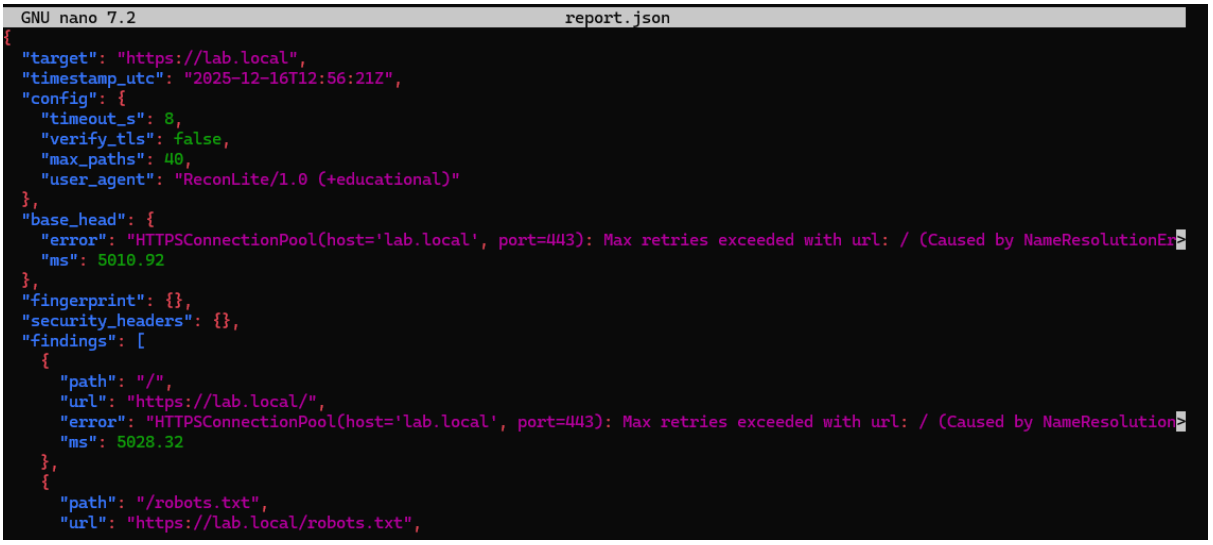
```
    "max_paths": 40,
    "user_agent": "ReconLite/1.0 (+educational)"
  },
  "base_head": {
    "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries
exceeded with url: / (Caused by
NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed
to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
    "ms": 5010.92
  },
  "fingerprint": {},
  "security_headers": {},
  "findings": [
    {
      "path": "/",
      "url": "https://lab.local/",
      "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries
exceeded with url: / (Caused by
NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed
to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
      "ms": 5028.32
    },
    {
      "path": "/robots.txt",
      "url": "https://lab.local/robots.txt",
      "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries
exceeded with url: /robots.txt (Caused by
NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed
to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
      "ms": 5007.68
    },
    {
      "path": "/sitemap.xml",
      "url": "https://lab.local/sitemap.xml",
      "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries
exceeded with url: /sitemap.xml (Caused by
NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed
to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
      "ms": 5009.4
    },
    {
      "path": "/.git/",
      "url": "https://lab.local/.git/",
```

```
    "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /.git/ (Caused by NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
    "ms": 5027.56
   },
   {
    "path": "/.env",
    "url": "https://lab.local/.env",
    "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /.env (Caused by NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
    "ms": 5008.27
   },
   {
    "path": "/config.php",
    "url": "https://lab.local/config.php",
    "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /config.php (Caused by NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
    "ms": 5011.32
   },
   {
    "path": "/phpinfo.php",
    "url": "https://lab.local/phpinfo.php",
    "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /phpinfo.php (Caused by NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
    "ms": 5042.38
   },
   {
    "path": "/admin",
    "url": "https://lab.local/admin",
    "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /admin (Caused by NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
    "ms": 5014.95
   },
   {
    "path": "/admin/",
```

```
    "url": "https://lab.local/admin/",
    "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /admin/ (Caused by NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
    "ms": 5010.96
  },
  {
    "path": "/login",
    "url": "https://lab.local/login",
    "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /login (Caused by NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
    "ms": 5010.9
  },
  {
    "path": "/login/",
    "url": "https://lab.local/login/",
    "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /login/ (Caused by NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
    "ms": 5008.88
  },
  {
    "path": "/wp-admin",
    "url": "https://lab.local/wp-admin",
    "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /wp-admin (Caused by NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
    "ms": 5010.2
  },
  {
    "path": "/wp-login.php",
    "url": "https://lab.local/wp-login.php",
    "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /wp-login.php (Caused by NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
    "ms": 5010.67
  },
  {
```

```
    "path": "/api",
    "url": "https://lab.local/api",
    "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /api (Caused by NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
    "ms": 5008.09
  },
  {
    "path": "/api/",
    "url": "https://lab.local/api/",
    "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /api/ (Caused by NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
    "ms": 5009.32
  },
  {
    "path": "/swagger",
    "url": "https://lab.local/swagger",
    "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /swagger (Caused by NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
    "ms": 5014.39
  },
  {
    "path": "/swagger/",
    "url": "https://lab.local/swagger/",
    "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /swagger/ (Caused by NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
    "ms": 5007.84
  },
  {
    "path": "/openapi.json",
    "url": "https://lab.local/openapi.json",
    "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /openapi.json (Caused by NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
    "ms": 5009.79
  },
```

```
    {
      "path": "/server-status",
      "url": "https://lab.local/server-status",
      "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries
exceeded with url: /server-status (Caused by
NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed
to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
      "ms": 5010.58
    },
    {
      "path": "/actuator",
      "url": "https://lab.local/actuator",
      "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries
exceeded with url: /actuator (Caused by
NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed
to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
      "ms": 5033.1
    },
    {
      "path": "/actuator/health",
      "url": "https://lab.local/actuator/health",
      "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries
exceeded with url: /actuator/health (Caused by
NameResolutionError(\"HTTPSConnection(host='lab.local', port=443): Failed
to resolve 'lab.local' ([Errno -2] Name or service not known)\"))",
      "ms": 5010.15
    }
  ]
}
```

```
{
  "path": "/robots.txt",
  "url": "https://lab.local/robots.txt",
  "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /robots.txt (Caused by Name>
  "ms": 5007.68
},
{
  "path": "/sitemap.xml",
  "url": "https://lab.local/sitemap.xml",
  "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /sitemap.xml (Caused by Nam>
  "ms": 5009.4
},
{
  "path": "/.git/",
  "url": "https://lab.local/.git/",
  "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /.git/ (Caused by NameResol>
  "ms": 5027.56
},
{
  "path": "/.env",
  "url": "https://lab.local/.env",
  "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /.env (Caused by NameResolu>
  "ms": 5008.27
},
{
  "path": "/config.php",
  "url": "https://lab.local/config.php",
  "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /config.php (Caused by Name>
  "ms": 5011.32
},
{
  "path": "/phpinfo.php",
  "url": "https://lab.local/phpinfo.php",
  "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /phpinfo.php (Caused by Nam>
  "ms": 5042.38
},
{
  "path": "/admin",
  "url": "https://lab.local/admin",
  "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /admin (Caused by NameResol>
  "ms": 5014.95
},
{
  "path": "/admin/",
```

```
{
  "path": "/admin/",
  "url": "https://lab.local/admin/",
  "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /admin/ (Caused by NameReso>
  "ms": 5010.96
},
{
  "path": "/login",
  "url": "https://lab.local/login",
  "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /login (Caused by NameResol>
  "ms": 5010.9
},
{
  "path": "/login/",
  "url": "https://lab.local/login/",
  "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /login/ (Caused by NameReso>
  "ms": 5008.88
},
{
  "path": "/wp-admin",
  "url": "https://lab.local/wp-admin",
  "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /wp-admin (Caused by NameRe>
  "ms": 5010.2
},
{
  "path": "/wp-login.php",
  "url": "https://lab.local/wp-login.php",
  "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /wp-login.php (Caused by Na>
  "ms": 5010.67
},
{
  "path": "/api",
  "url": "https://lab.local/api",
  "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /api (Caused by NameResolut>
  "ms": 5008.09
},
{
  "path": "/api/",
  "url": "https://lab.local/api/",
  "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /api/ (Caused by NameResolu>
  "ms": 5009.32
},
{
  "path": "/swagger",
```

```
    },
    {
      "path": "/swagger",
      "url": "https://lab.local/swagger",
      "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /swagger (Caused by NameRes>
      "ms": 5014.39
    },
    {
      "path": "/swagger/",
      "url": "https://lab.local/swagger/",
      "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /swagger/ (Caused by NameRe>
      "ms": 5007.84
    },
    {
      "path": "/openapi.json",
      "url": "https://lab.local/openapi.json",
      "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /openapi.json (Caused by Na>
      "ms": 5009.79
    },
    {
      "path": "/server-status",
      "url": "https://lab.local/server-status",
      "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /server-status (Caused by N>
      "ms": 5010.58
    },
    {
      "path": "/actuator",
      "url": "https://lab.local/actuator",
      "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /actuator (Caused by NameRe>
      "ms": 5033.1
    },
    {
      "path": "/actuator/health",
      "url": "https://lab.local/actuator/health",
      "error": "HTTPSConnectionPool(host='lab.local', port=443): Max retries exceeded with url: /actuator/health (Caused by>
      "ms": 5010.15
    }
  ]
}
```