**Universidade Estadual de Maringá**
**Ciência da Computação**
**6919 - Gerenciamento de Redes de Computadores**
*Profa. Luciana Andréia Fondazzi Martimiano*

# Trabalho 1 - Kali Linux

Gerenciamento de Redes

Henrique Shiguemoto Felizardo - 115207
Matheus Augusto Schiavon Parise - 107115
Gabriel de Melo Osório - 107862

Maringá
2023

## Resumo

Neste trabalho, pretende-se demonstrar algumas técnicas de teste de invasão com a utilização de algumas ferramentas consolidadas em plataformas que servem para realizar testes de invasão.

São introduzidos os conceitos de testes de invasão e também as ferramentas que foram utilizadas, como as ferramentas Kali Linux, OWASP ZAP, FFUF, entre outras. Com essas ferramentas, testes foram realizados em plataformas como DVWA (*Damn Vulnerable WEB Application*) e o site http://testphp.vulnweb.com/.

São apresentados também os detalhes das análises realizadas com as ferramentas, demonstrando comandos utilizados em cada ferramenta e também quais foram as saídas dessas mesmas ferramentas.

## Introdução

É comum nos dias atuais encontrarmos novos casos de ataques cibernéticos ocorrendo nas mais diversas empresas de diferentes ramos. De acordo com InforChannel (2023), o Brasil é o país que mais sofre ataques cibernéticos na América Latina. A cada semana é possível encontrar novas notícias relacionadas a ataques cibernéticos. Por exemplo: no dia 31 de março de 2023, o site do Superior Tribunal Militar sofreu um ataque para deixar o sistema indisponível com várias tentativas de acesso, conhecido como negação de serviço distribuída (DDoS) (CNN Brasil, 2023).

Um teste de invasão (também chamado de pentest) é uma técnica que visa a segurança de um sistema ou rede de computadores. Isso é feito a partir de uma simulação de ataques cibernéticos a estes sistemas (de preferência realizados por equipes especialistas externas a empresas). Com esses ataques simulados, é possível produzir relatórios informativos sobre as vulnerabilidades que o sistema possui e que podem gerar complicações na rede de computadores caso um ataque real de fato aconteça.

É importante informar a diferença entre profissionais de invasão (pentesters, ethical hackers ou hackers éticos) e hackers. De acordo com Strema (2022), as técnicas de invasão utilizadas pelos dois tipos de indivíduos são as mesmas. No entanto, os objetivos são diferentes. Enquanto o pentester objetiva uma maior segurança para uma empresa ou para uma rede de computadores, hackers nem sempre possuem razões benevolentes para seus ataques.

Em um cenário onde ocorrências de ataques cibernéticos são notícias comuns, testes de invasão são bastante importantes para a segurança de uma rede de computadores, principalmente se essa rede é constituída de usuários com informações valiosas. É

importante ressaltar que testes de invasão aumentam a segurança de uma rede porém é recomendado que essa técnica seja integrada com outras práticas de segurança (Strema, 2022), já que vários tipos de vulnerabilidades podem existir e times de pentesters nem sempre conseguirão averiguar todos os tipos de vulnerabilidades existentes em uma empresa ou redes de computadores.

Profissionais de testes de invasão possuem uma grande quantidade de ferramentas à sua disposição. O sistema operacional Kali Linux é um dos favoritos entre esses profissionais. No entanto, existem outras ferramentas muito utilizadas, como o Nmap, OWASP ZAP, SQLmap, FFUF e Burp Suite. Essas ferramentas mencionadas foram usadas para a execução dos testes deste trabalho.

Este trabalho visa a utilização das ferramentas mencionadas no site http://testphp.vulnweb.com/ e na ferramenta DVWA (*Damn Vulnerable Web Application*) para simular tentativas de invasão. Os resultados da utilização das ferramentas também serão apresentados, explicados e analisados.

## Descrição do Kali Linux e das ferramentas

O Kali Linux é uma distribuição Linux baseada no Debian e projetada especificamente para testes de segurança e penetração (pentests) em sistemas e redes. Ele vem pré-carregado com uma ampla variedade de ferramentas de segurança, incluindo scanners de vulnerabilidades, sniffers de rede, ferramentas de quebra de senhas, programas de engenharia social, entre outros.

O objetivo do Kali Linux é fornecer uma plataforma de testes de segurança completa e fácil de usar para profissionais de segurança, estudantes de segurança cibernética e entusiastas. Ele é projetado para ser usado em um ambiente de teste controlado e não deve ser usado para fins maliciosos.

O Kali Linux é atualizado regularmente para incluir as últimas ferramentas de segurança e correções de segurança, e possui uma grande comunidade de desenvolvedores e usuários que contribuem para seu desenvolvimento e suporte. Embora possa ser instalado em um sistema operacional existente, o Kali Linux também pode ser executado a partir de um Live CD ou USB, permitindo que os usuários testem sistemas e redes sem afetar o sistema operacional existente.

O Nmap (Network Mapper) é uma ferramenta de segurança de rede amplamente utilizada para análise e mapeamento de redes. Ele é projetado para detectar hosts e serviços em uma rede, bem como identificar vulnerabilidades e falhas de segurança em sistemas. O Nmap funciona enviando pacotes de rede para os hosts alvo e analisando as respostas

recebidas. Com base nas respostas, o Nmap é capaz de determinar quais hosts estão ativos, quais portas estão abertas e quais serviços estão sendo executados em cada porta. Ele também pode ser usado para identificar o sistema operacional em execução em cada host.

O OWASP ZAP (Zed Attack Proxy) é uma ferramenta de teste de segurança de aplicativos web open-source, que permite identificar vulnerabilidades em aplicativos web, tais como cross-site scripting (XSS), injeção de SQL, entre outras. Ele é projetado para ser fácil de usar, e é muito popular entre profissionais de segurança de aplicativos web, testadores de software, desenvolvedores e outros. A ferramenta pode ser usada tanto para testes manuais quanto para testes automatizados, permitindo que o usuário envie solicitações HTTP e visualize as respostas para identificar possíveis vulnerabilidades de segurança. Ele também possui recursos de geração de relatórios detalhados para ajudar os usuários a entender as vulnerabilidades encontradas e fornecer orientação sobre como corrigi-las.

SQLmap é uma ferramenta de teste de penetração open-source usada para automatizar a detecção e exploração de vulnerabilidades de injeção de SQL em aplicativos web. Ele é projetado para testar a segurança de bancos de dados relacionais, como MySQL, Oracle, Microsoft SQL Server, PostgreSQL e outros. O SQLmap é capaz de detectar vulnerabilidades de injeção de SQL em aplicativos web por meio da exploração de formulários da web, cookies e cabeçalhos HTTP. Uma vez que uma vulnerabilidade é identificada, a ferramenta pode extrair informações sensíveis do banco de dados, como nomes de tabelas, colunas e registros, além de permitir a execução de comandos SQL personalizados.

FFUF é uma ferramenta de teste de penetração de linha de comando para descoberta de diretórios e subdomínios em aplicativos web. Ele permite que os testadores de segurança de aplicativos web descubram rapidamente quaisquer recursos disponíveis em um aplicativo web, como subdomínios, diretórios, arquivos e páginas. O FFUF é uma ferramenta de código aberto e é altamente configurável para atender às necessidades dos usuários. Ele usa um dicionário de palavras-chave para descobrir recursos disponíveis em um aplicativo web, usando uma técnica chamada de "fuzzing", que gera solicitações HTTP personalizadas para cada palavra-chave no dicionário e analisa as respostas recebidas para determinar se um recurso foi encontrado ou não. Além disso, o FFUF pode ser configurado para usar proxies e autenticação, bem como para trabalhar com SSL.

O Burp Suite é uma ferramenta de teste de segurança de aplicativos web que permite identificar e explorar vulnerabilidades em aplicativos web. Ele oferece recursos como um proxy web, scanner de vulnerabilidades, sequenciador de tokens, spider, intruder e recursos personalizáveis, permitindo que os testadores de segurança personalizem suas abordagens de teste e adaptem-se às necessidades de cada projeto. Há uma versão gratuita (Community Edition) e uma versão profissional com recursos adicionais. Neste trabalho usaremos a versão gratuita.

## Análise dos resultados dos testes

A ordem de execução das ferramentas foi escolhida considerando o propósito de cada uma, primeiramente serão executadas as ferramentas Nmap, OWASP ZAP e FFUF, a ordem entre elas não importa, o que realmente importa é que essas três ferramentas sejam executadas antes das outras duas pois elas são ferramentas de varredura, isto significa que elas juntam várias informações e identificam pontos fracos na aplicação web analisada.

Em seguida, após coletar essas informações, ficará mais fácil utilizar o SQLmap e Burp Suite Community Edition. Com exceção do OWASP ZAP e Burp Suite Community Edition, todas as ferramentas são via command-line interface, isto é, precisam ser executadas via linha de comando no terminal.

Primeiramente vamos tentar invadir o domínio http://testphp.vulnweb.com/, vamos começar executando a ferramenta nmap no prompt de comando do Kali linux (esta ferramenta já vem instalada junto com o Kali). É necessário abrir o terminal e digitar o comando nmap e depois a url do alvo sem o http\https e as barras:

- nmap testphp.vulweb.com

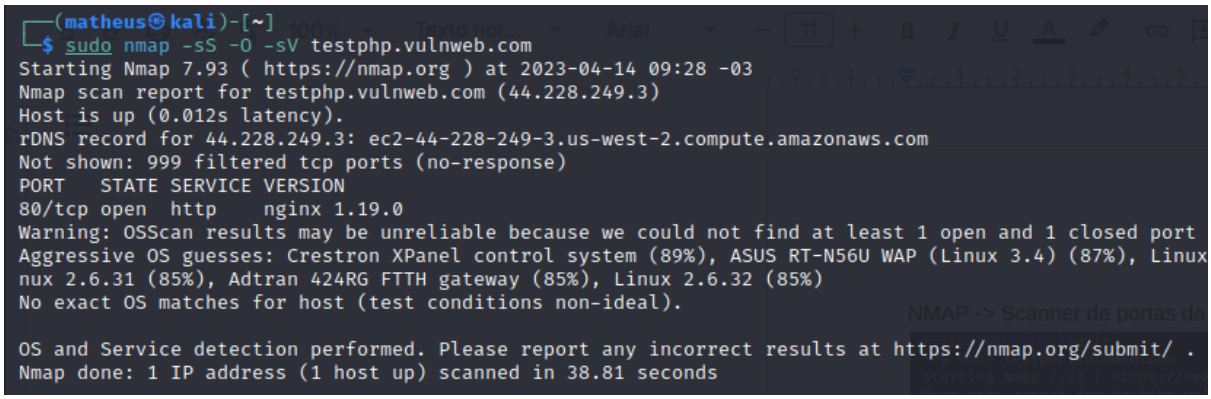
```
┌──(matheus㉿kali)-[~]
└─$ nmap testphp.vulnweb.com
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-14 09:46 -03
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.22s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
Not shown: 999 filtered tcp ports (no-response)
PORT   STATE SERVICE
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 15.39 seconds
```

É possível observar que há um serviço http sendo executado na porta 80 via conexão TCP, isso não nós dá muita informação, então vamos inserir alguns parâmetros para poder extrair mais informações, estes parâmetros são:

-O: Este parâmetro permite a detecção de sistemas operacionais remotos
-sS: Utiliza a varredura SYN, esta varredura é mais silenciosa e furtiva do que a varredura de conexão TCP completa.
-sV: Este parâmetro permite a detecção de versões de software em execução nos serviços detectados.

A utilização dos parâmetros -O e -sS exigem privilégios do administrador, ou seja você precisa conceder uma autorização especial para o nmap poder executar isso, então devemos adicionar o comando “sudo” antes da palavra nmap. Dessa forma o comando final fica dessa maneira (lembrando que “sudo” vem antes do nmap, e os parâmetros do nmap precisam vir depois do comando nmap mas antes da url do alvo, a ordem entre os parâmetros não importa).

- sudo nmap -sS -O -sV testphp.vulweb.com

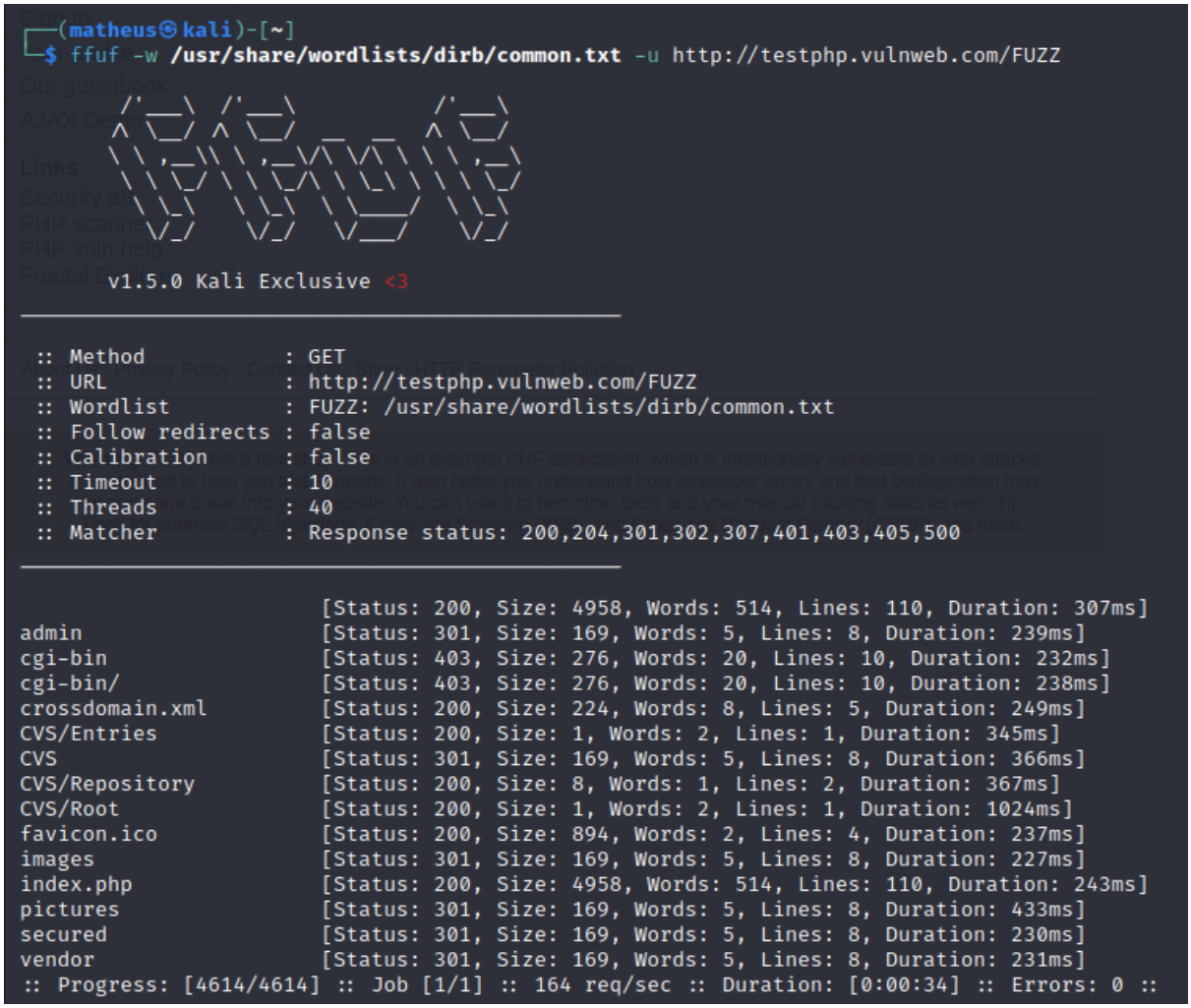



Dessa vez detectamos não apenas o serviço mas sua sua versão nginx 1.19.0, a análise dos sistema operacional não deu certo pois ele não achou uma informação que comprovasse qual sistema operacional era, mas acha que é linux.

Em seguida vamos executar o FFUF, esta ferramenta não é instalada por padrão no kali linux, então é necessário fazer o download dela primeiro. O funcionamento do FFUF é diferente do nmap, ele precisa de mais informações além da url do alvo, ele precisa dos nomes de diretórios que ele irá buscar, o kali pode nos ajudar com isso. Por padrão o kali possui uma lista dos nomes de diretórios mais utilizados, podemos usar essa lista a nosso favor, a localização padrão dele é “/usr/share/wordlists/dirb/common.txt”.

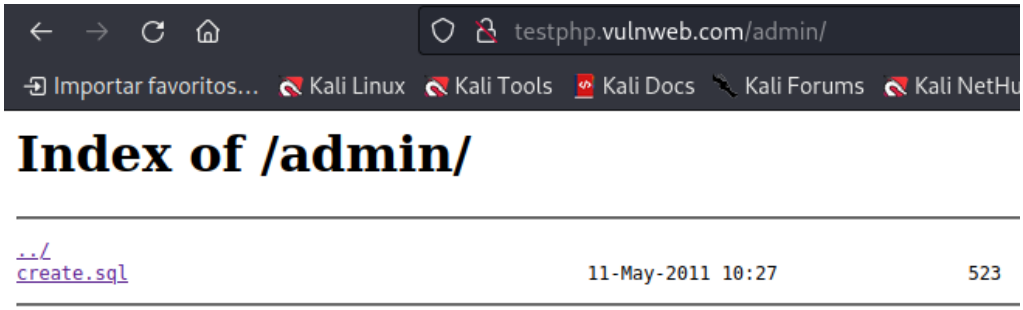
Para utilizar o FFUF abra o terminal e digite o seguinte comando FFUF, depois coloque o parâmetro -w para especificar que existe uma lista de nomes que será utilizada, em seguida digite a localização da lista de nomes sem aspas, após isso precisamos especificar o alvo, podemos fazer isso com o parâmetro -u, digite o parâmetro e coloque o site alvo com http/https e as barras. Por fim um último detalhe de suma importância é necessário colocar a palavra “FUZZ” em seguida da última barra pois especifica o começo da varredura, o comando fica dessa forma:

- ffuf -w /usr/share/wordlists/dirb/common.txt -u http://testphp.vulnweb.com/FUZZ




As informações descobertas a partir desta análise são extremamente interessantes, cada um dos nomes mostrados nesta lista é um diretório ou página do site em questão, além do nome temos status do acesso, tamanho, entre outras informações. Neste caso as informações mais importantes são nome e status, se você substituir um nome da lista na localização FUZZ da url e tentar acessar determinada página via navegador, dependendo do status da resposta será possível acessar essa página, o que é tão incrível nisso ? nem todas essas páginas deveriam poder ser acessadas. Como nesse caso são poucas páginas podemos tentar acessar todas elas e tentar descobrir algo útil, mas caso sejam muitas é interessante escolher as páginas com nomes interessantes como “admin” que é uma abreviação para administrador. Segue abaixo os resultados interessantes descobertos:

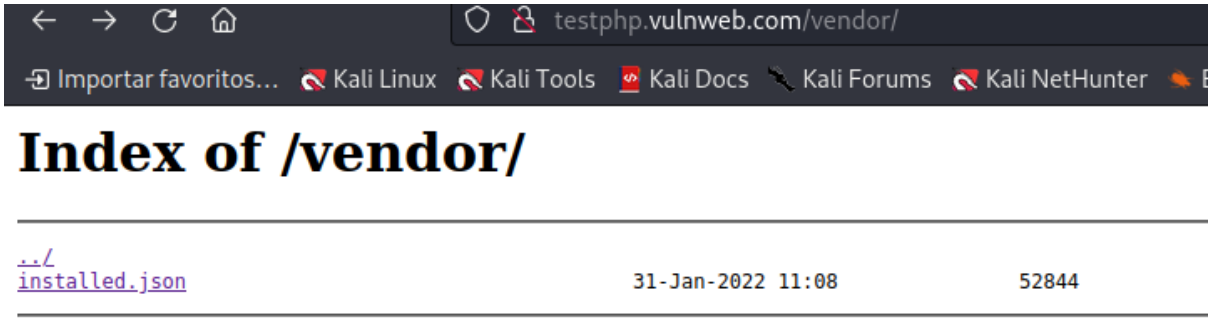
Ao acessar http://testphp.vulnweb.com/admin/ foi possível descobrir uma pagina que possuía o arquivo “create.sql”




Também era possível fazer download do mesmo, após o download acessei o arquivo e foi descoberto que o mesmo possuía informações sobre o banco de dados do alvo.

```
create database waspart;
use waspart;

CREATE TABLE IF NOT EXISTS forum(
    sender      CHAR(150),
    mesaj       TEXT,
    senttime    INTEGER(32));

CREATE TABLE IF NOT EXISTS artists(
    artist_id   INTEGER(5) PRIMARY KEY AUTO_INCREMENT,
    aname       CHAR(50),
    adesc       BLOB);

CREATE TABLE IF NOT EXISTS categ(
    cat_id      INTEGER(5) PRIMARY KEY AUTO_INCREMENT,
    cname       CHAR(50),
    cdesc       BLOB);

CREATE TABLE IF NOT EXISTS pictures(
    pic_id      INTEGER(5) PRIMARY KEY AUTO_INCREMENT,
    pshort      BLOB,
    plong       TEXT,
    price       INTEGER,
    img         CHAR(50));
```

O mesmo foi feito para http://testphp.vulnweb.com/vendor/, nesta pagina há um arquivo installed.json.



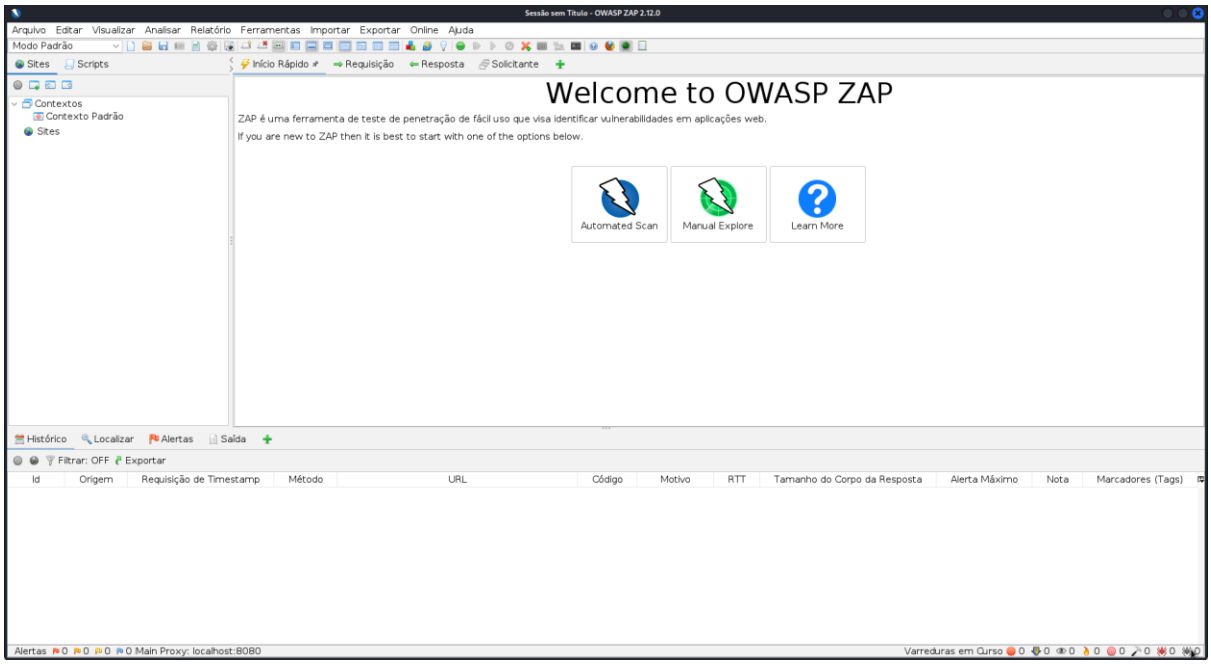

Foi possível realizar o download do mesmo e nele continha as informações das bibliotecas json utilizadas no site.


```
{
    "packages":[
    {
        "name": "doctrine/instantiator",
        "version": "1.4.0",
        "version_normalized": "1.4.0.0",
        "source": {
            "type": "git",
            "url": "https://github.com/doctrine/instantiator.git",
            "reference": "d56bf6102915de5702778fe20f2de3b2fe570b5b"
        },
        "dist": {
            "type": "zip",
            "url": "https://api.github.com/repos/doctrine/instantiat
            "reference": "d56bf6102915de5702778fe20f2de3b2fe570b5b",
            "shasum": ""
        },
        "require": {
            "php": "^7.1 || ^8.0"
        },
        "require-dev": {
            "doctrine/coding-standard": "^8.0",
            "ext-pdo": "*",
            "ext-phar": "*",
            "phpbench/phpbench": "^0.13 || 1.0.0-alpha2",
            "phpstan/phpstan": "^0.12",
            "phpstan/phpstan-phpunit": "^0.12",
            "phpunit/phpunit": "^7.0 || ^8.0 || ^9.0"
        },
        "time": "2020-11-10T18:47:58+00:00",
        "type": "library",
```

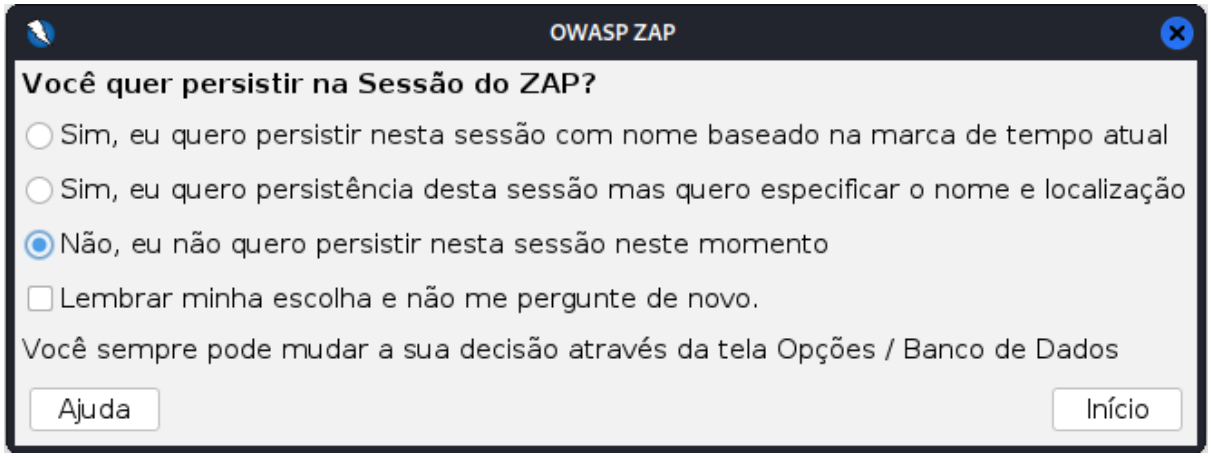

Com essas informações é possível saber que o alvo utiliza banco de dados SQL, temos também informações sobre essa estrutura deste banco o que nos permite arquitetar um vetor de ataque. Além disso, existem várias bibliotecas json utilizadas, algumas delas podem ser vulneráveis ou estar desatualizadas, permitindo outro vetor de ataque.

Vamos para a próxima ferramenta Owasp zap, esta ferramenta não é instalada com o kali linux então é necessário fazer a instalação da mesma, é válido relembrar que esta é a

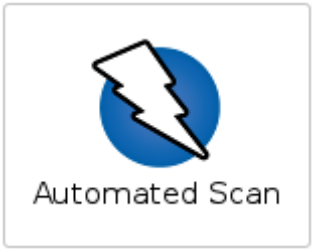
única ferramenta com interface gráfica própria. Após a instalação execute o owasp zap, a seguir uma imagem da interface gráfica da página inicial.



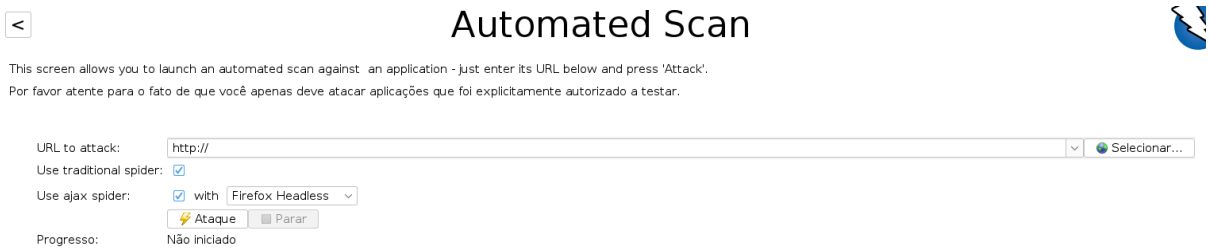

Quando uma nova sessão do owasp zap é criada, o programa pergunta se o usuário deseja salvar previamente ou não, se sim ele irá escrever os arquivos gerados durante a sessão no local desejado, caso contrário os arquivos serão escritos na pasta “/tmp” (pasta de arquivos temporários do Kali Linux).



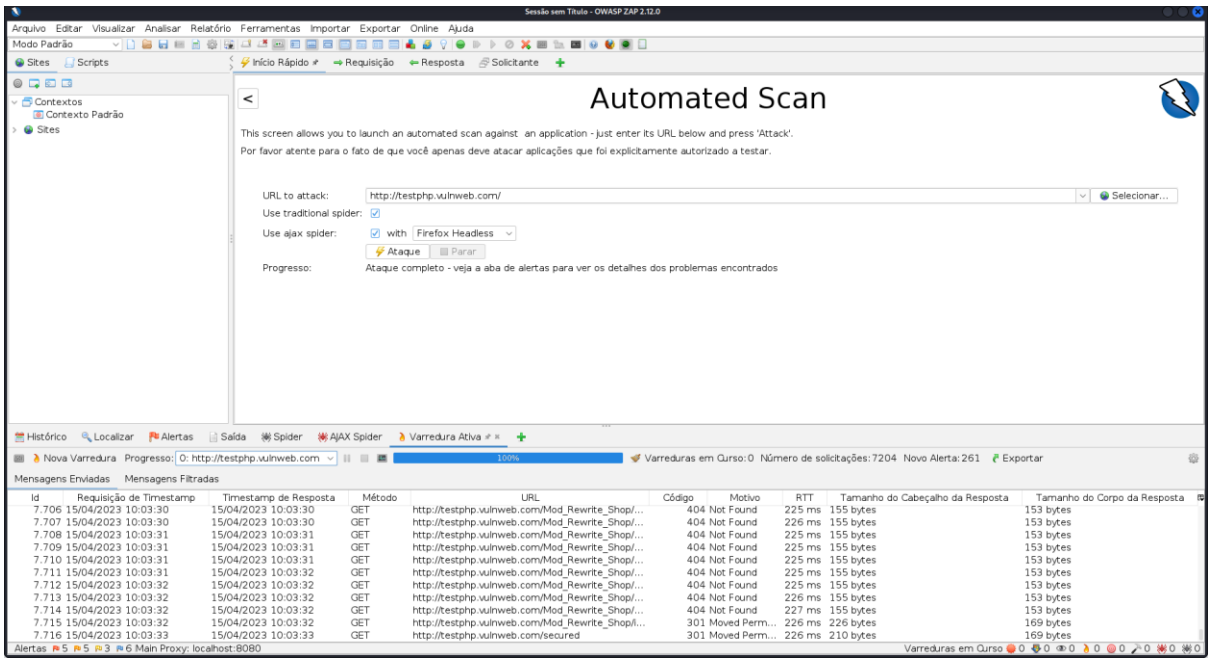

É possível fazer dois tipos principais de teste, o teste automatizado onde é realizado um scan automático por parte da ferramenta, neste caso não há a necessidade de interação com o usuário. Outro tipo de teste é o manual, onde o usuário explora manualmente o site alvo e a ferramenta fica sendo executada em segundo plano fazendo varredura da página explorada. Vamos executar o teste automático, para isso clique no botão “Automated Scan”.



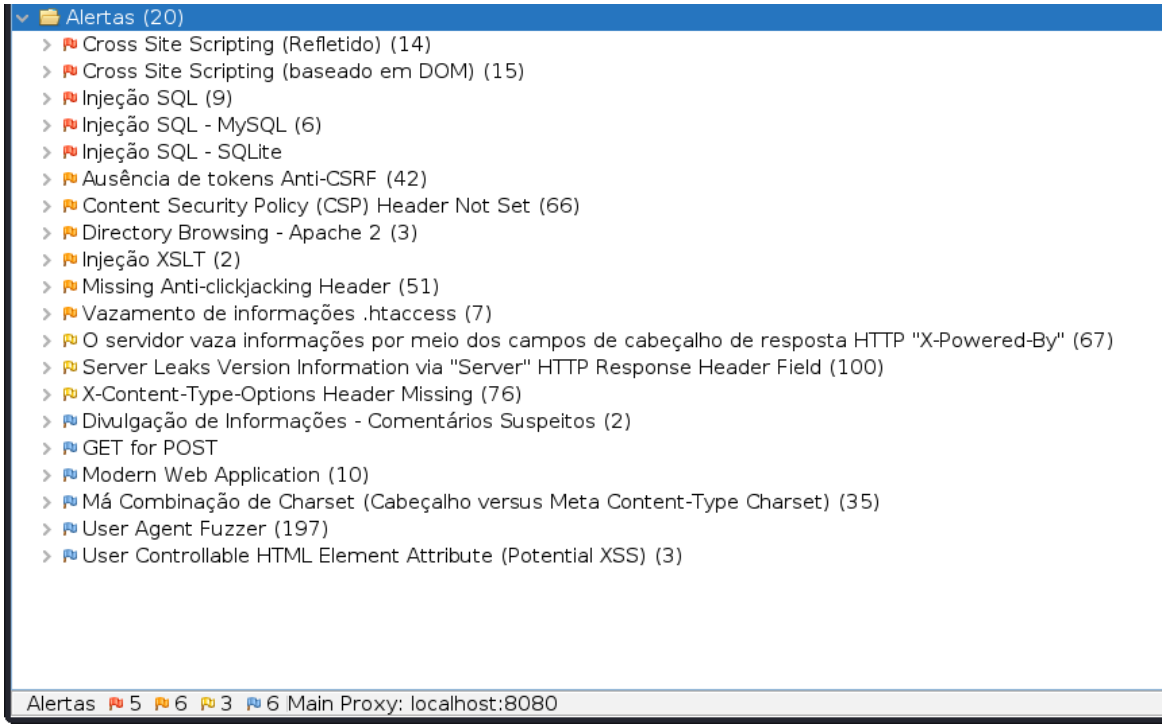

Em seguida irá aparecer uma nova página na aba principal, ela se parece com a imagem á seguir:




Para conduzir o ataque, basta inserir a url do alvo no campo “URL to attack” e depois pressionar o botão “Ataque” e esperar que o programa faça seu trabalho. Quando terminar a página ficará assim:




Também haverá uma lista de alertas de acordo com cada uma das vulnerabilidades encontradas, é possível encontrá-la na aba “Alerta”.




Existem inúmeros vetores de ataques possíveis de acordo com o owasp zap, cada bandeira tem um nível de risco de acordo com sua cor, a bandeira azul representa algo informacional, bandeira amarela risco baixo, laranja médio e por fim vermelho alto. Um exemplo seria:


A url “http://testphp.vulnweb.com/artists.php?artist=4-2” pode ser suscetível à injeção sql, pois o parâmetro “artist” aceitou e executou o valor “4-2”.

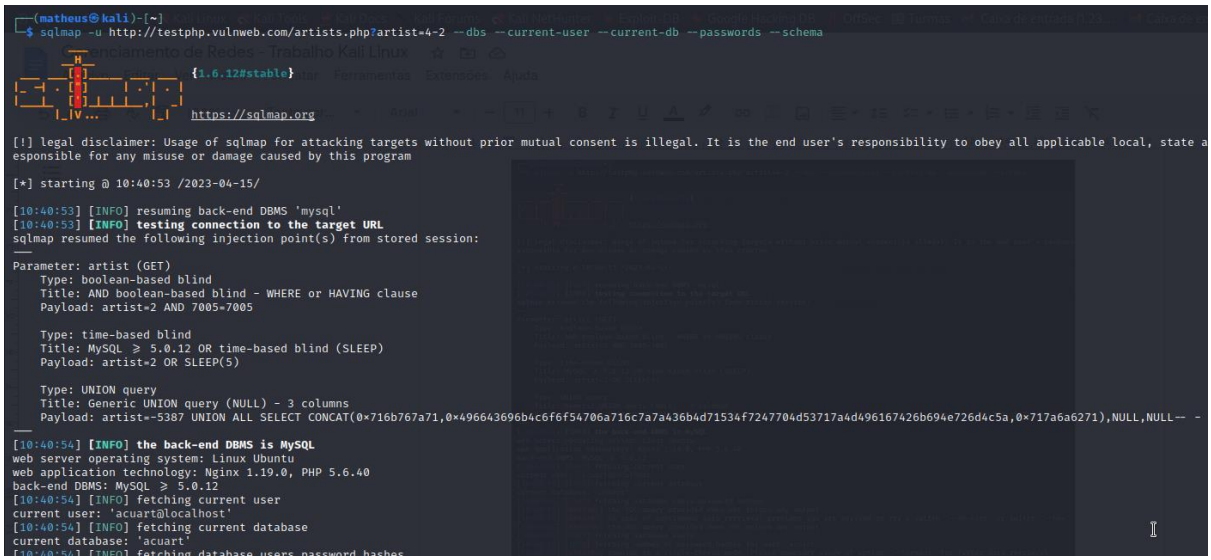
A partir dessas informações já é possível conduzir alguns ataques aos sites. Vamos fazer isso utilizando a ferramenta SQLmap, pois sabemos que o alvo usa SQL, o OWASP ZAP também revelou pontos de injeção SQL.

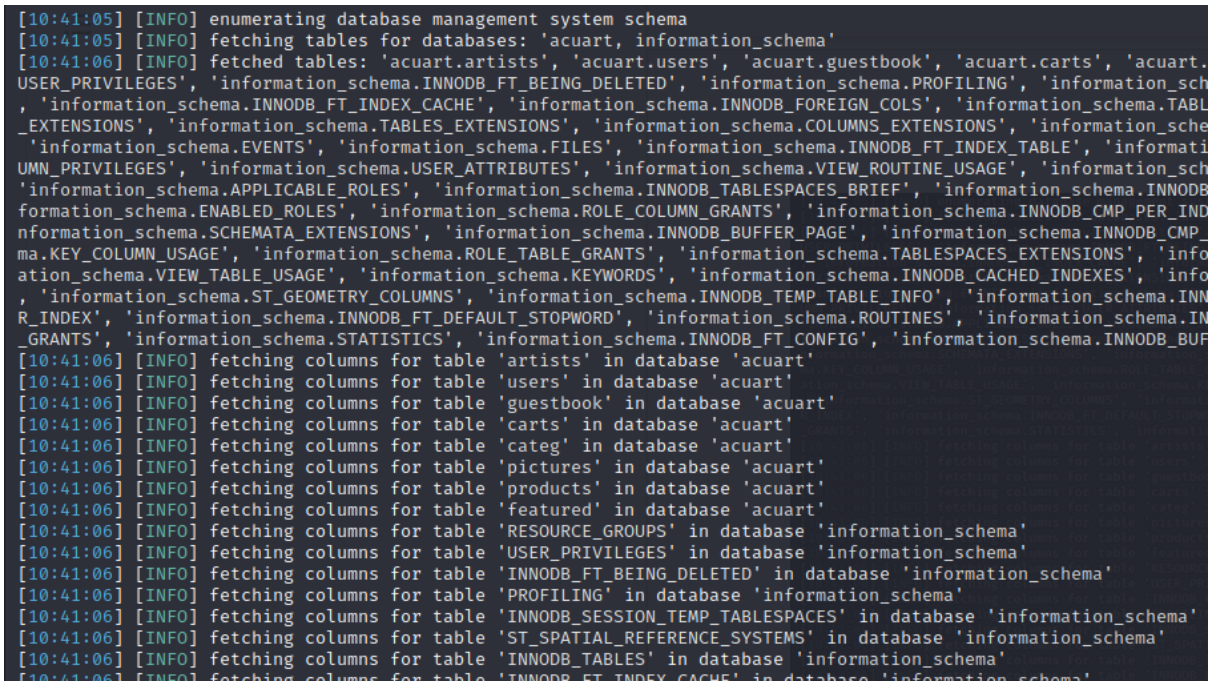
O SQLmap já vem instalado no Kali Linux, para utilizá-lo abra o terminal e digite o comando sqlmap depois coloque o parâmetro -u ele serve para especificar o alvo, é válido ressaltar que precisa ser uma URL com ponto de injeção SQL, isto é como a url de exemplo mencionada acima, após isso iremos utilizar alguns parâmetros adicionais estes são:

- **--dbs:** informa ao sqlmap para identificar quais bancos de dados estão disponíveis no sistema de gerenciamento de banco de dados. Essa opção permite que o sqlmap execute o comando SHOW DATABASES em busca de bancos de dados disponíveis.
- **--current-user:** solicita ao sqlmap para identificar o usuário atualmente conectado ao banco de dados.
- **--current-db:** informa ao sqlmap para identificar o banco de dados atualmente selecionado.
- **--passwords:** solicita ao sqlmap para tentar recuperar as senhas armazenadas no banco de dados. Isso é feito por meio de técnicas de injeção de SQL que exploram as vulnerabilidades encontradas pelo sqlmap.
- **--schema:** informa ao sqlmap para tentar recuperar informações do esquema do banco de dados, como tabelas, colunas, índices e assim por diante. Isso é feito por meio de técnicas de injeção de SQL que exploram as vulnerabilidades encontradas pelo sqlmap.

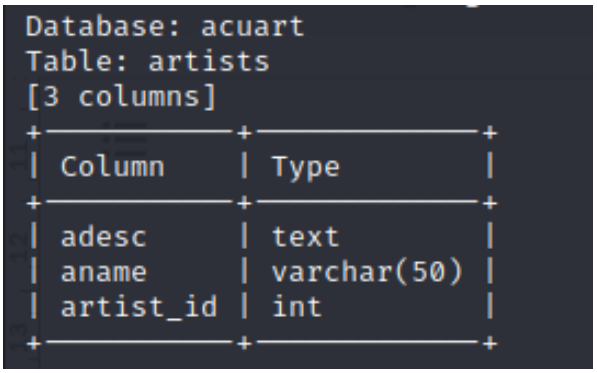
o comando fica assim no final:

- sqlmap -u http://testphp.vulnweb.com/artists.php?artist=4-2 --dbs --current-user --current-db --passwords --schema

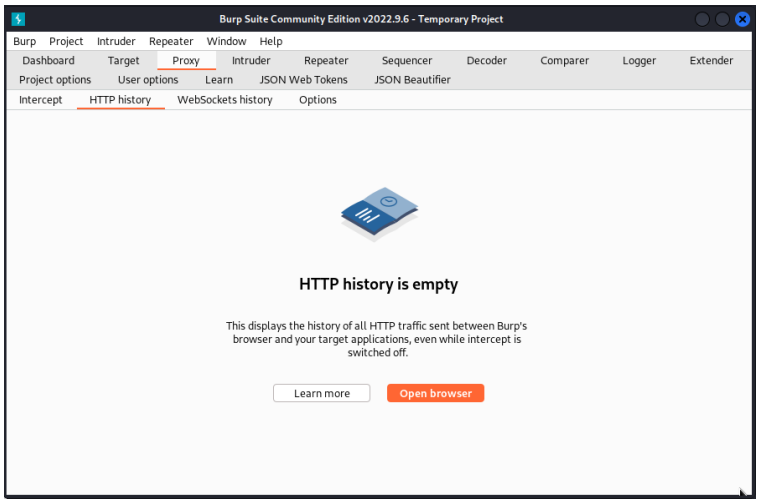


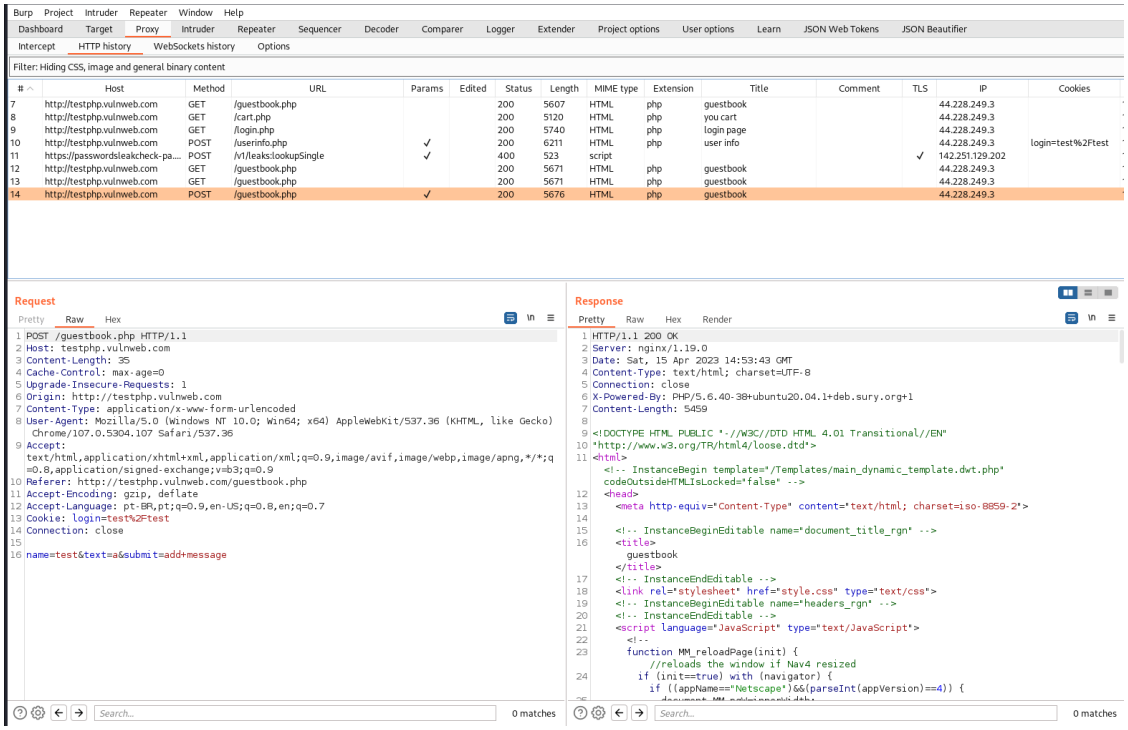

A injeção foi realizada com sucesso, contudo ele retornou muita informação, então nem tudo foi colocado, ele retornou inclusive o schema dos bancos.



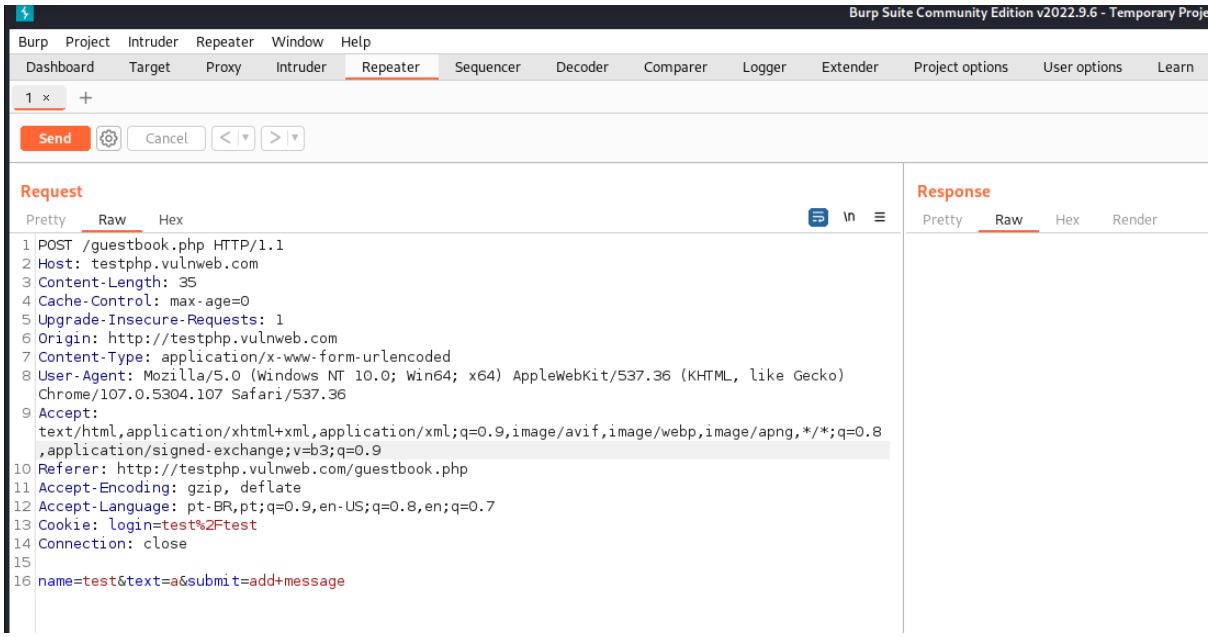
De acordo com Owasp Zap foi detectado um Cross-site-scripting (XSS) na página http://testphp.vulnweb.com/guestbook.php, para se aproveitar dele vamos utilizar o Burp Suite para explorar essa fraqueza, após iniciar a ferramenta vá para a aba proxy, acesse a sub area HTTP history e execute o navegador pelo qual o Burp está configurado.



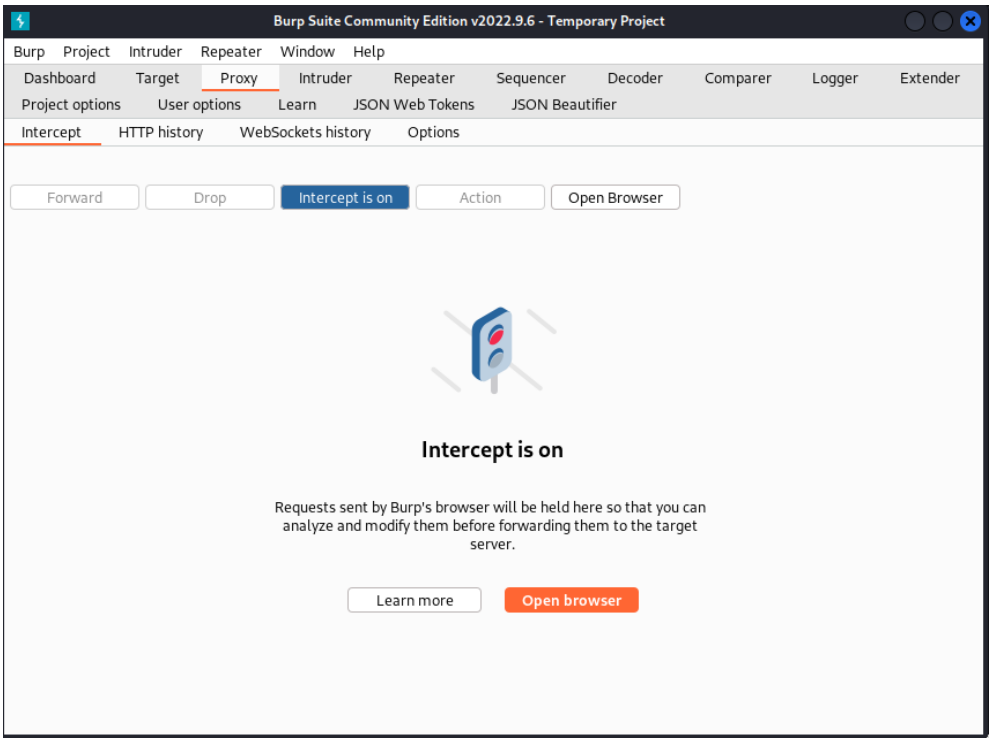

Uma das intenções maliciosas mais comuns com XSS é roubo de sessão vamos testar se é possível fazer isso, então vamos fazer o login no site http://testphp.vulnweb.com/, a função de cadastro foi desabilitada nele, então o site dá um login padrão este é a palavra “test” para usuário e senha, faça o login no site. Agora vamos para página http://testphp.vulnweb.com/guestbook.php, nela há como inserir comentários, digite um comentário aleatório e depois verifique a página do proxy na aba HTTP history



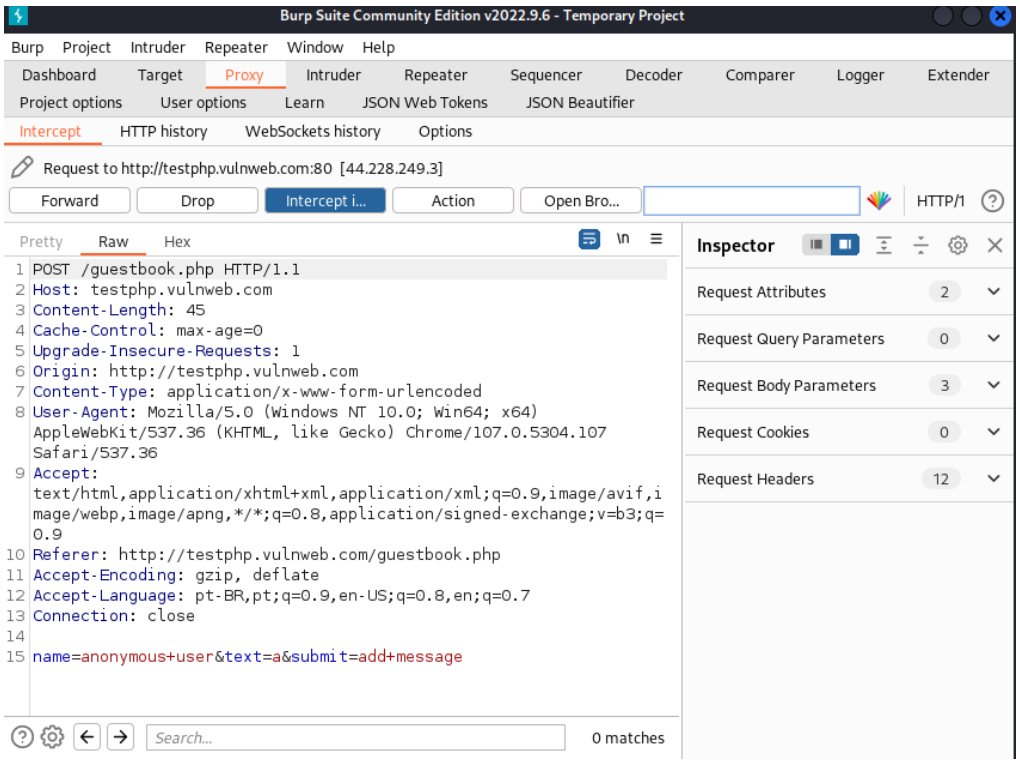
Clique com o botão direito na requisição do comentário, e selecione a opção “send to repeater”, isso é feito para enviar a requisição para aba repeater para podermos modificá-la e executá-la novamente. Acesse a aba Repeater, ela deverá aparecer assim




Já é possível observar que os cookies revelam informações sensíveis, no caso “Cookie: login=test%2Ftest”, quando decodificamos a codificação de URL atribuída ao valor “test%2Ftest” ele fica igual á “test/test”, isto é o nome de usuário ea senha respectivamente.Mesmo assim a vulnerabilidade não se limita a isso, é possível modificar a requisição realizada em tempo real ao ligar o interceptador de requisição, para isso acesse a aba “intercepter” e ligue-o.



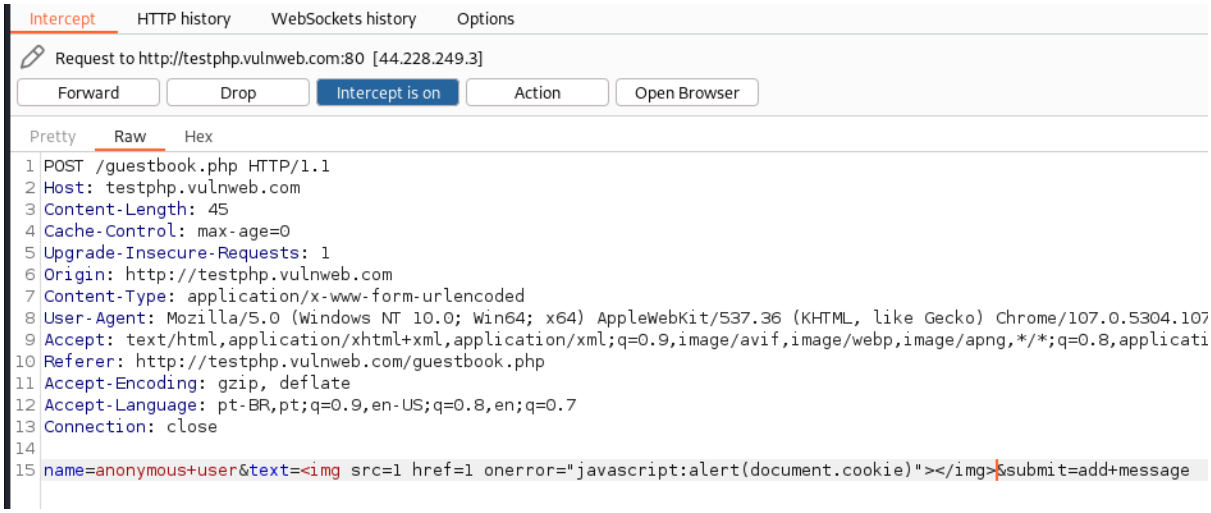
Depois insira outro comentário na página ele irá interpretá-lo, ou seja, o comentário não foi realizado pois o Burp o parou.

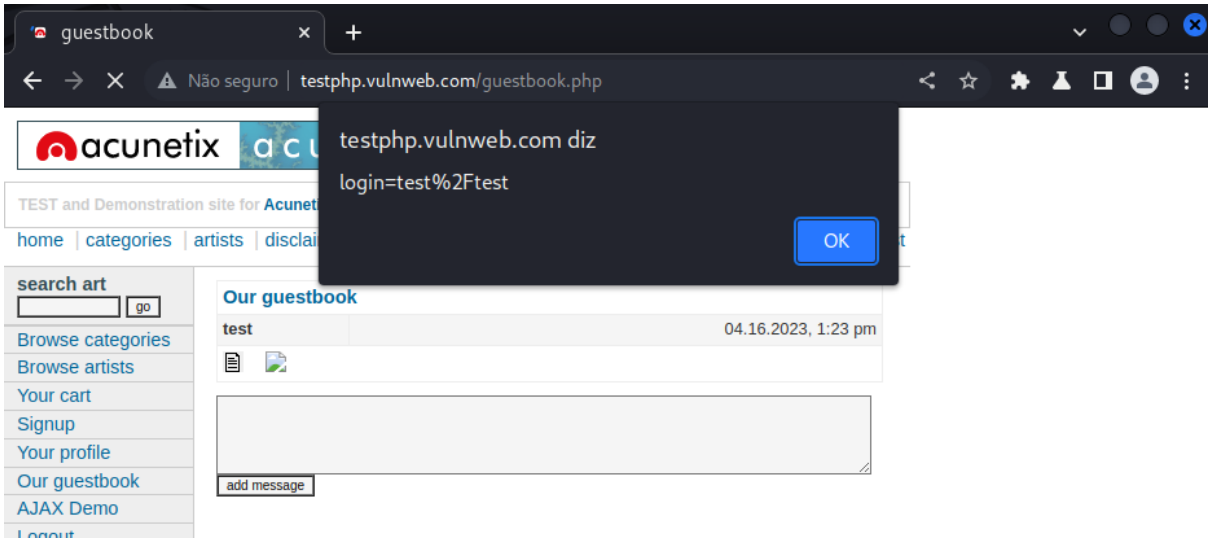



Vamos utilizar o payload a seguir, tentar explorar um XSS, esse payload faz o site printar os cookies, já conhecemos eles mas isso é realizado apenas para testar o XSS:

<img src=1 href=1 onerror="javascript:alert(document.cookie)"></img>

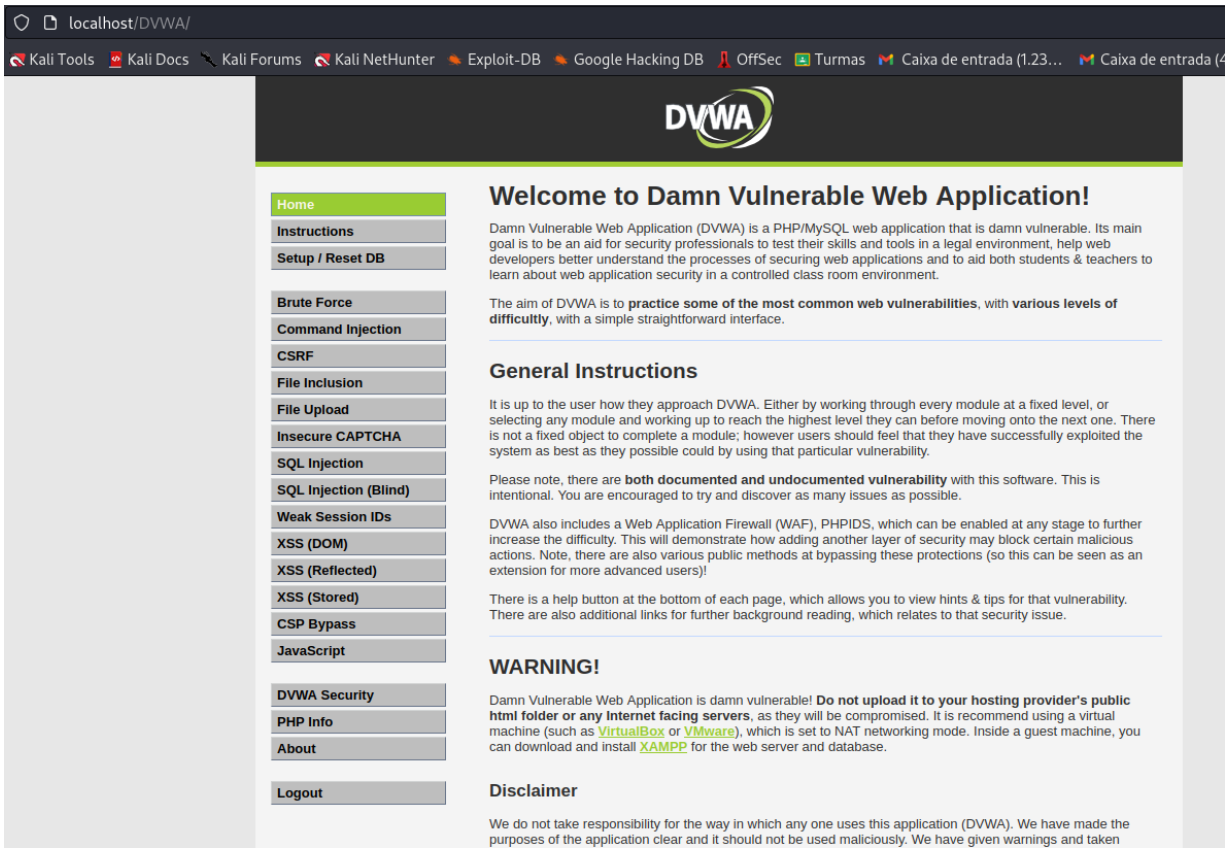
Esse código será inserido no atributo “texto” da mensagem. ela ficará assim






É possível notar que foi realizada uma execução de script, logo há XSS.

Para o segundo alvo Damn Vulnerable Web Application (DVWA) é necessário instalá-lo e executá-lo na máquina localmente. Após isso basta realizar novamente os passos descritos acima, mas dessa vez com o localhost pois é nele onde o DVWA é executado. Essa aplicação é dividida em seções onde cada uma contém uma vulnerabilidade específica, vamos observar como as ferramentas de varredura se saem.



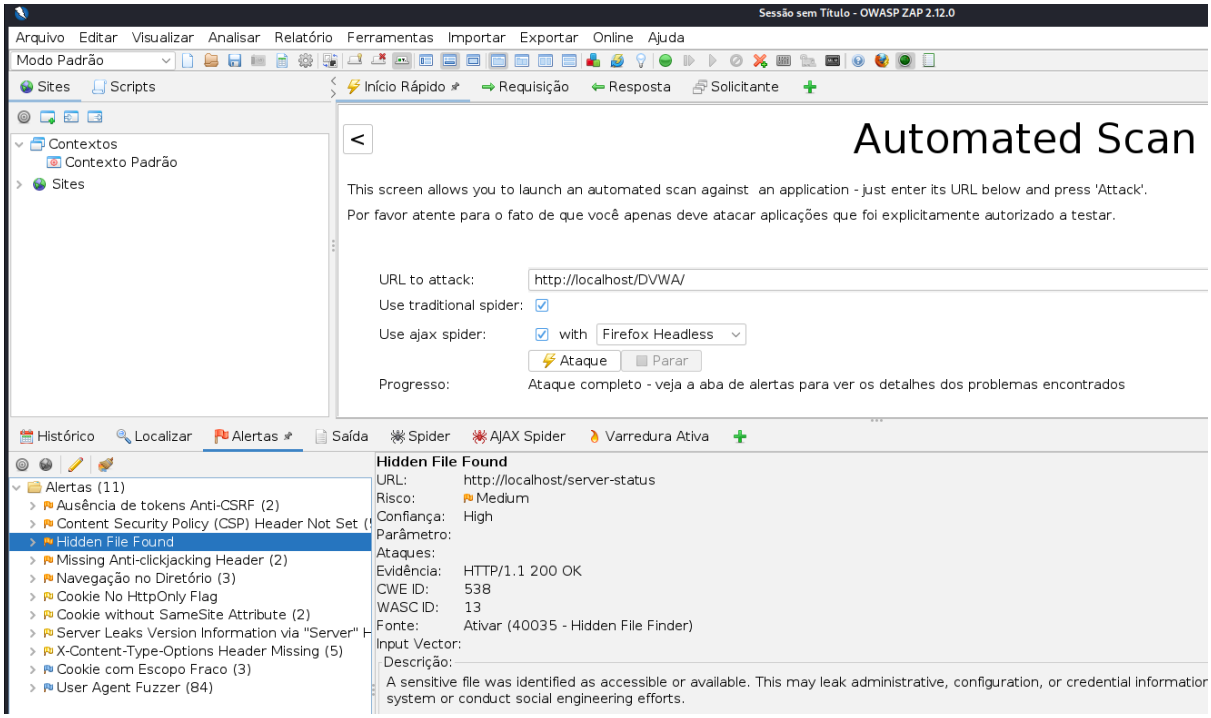
Primeiramente vamos começar com o nmap:

```
┌──(matheus㉿kali)-[/var/www/html]
└─$ sudo nmap -sV -O -sS localhost
[sudo] senha para matheus:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-15 11:26 -03
Stats: 0:00:50 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 66.67% done; ETC: 11:27 (0:00:22 remaining)
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000031s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE        VERSION
80/tcp   open  http           Apache httpd 2.4.54 ((Debian))
3306/tcp open  mysql          MySQL 5.5.5-10.6.11-MariaDB-1
8080/tcp open  ssl/http-proxy (bad gateway)
1 service unrecognized despite returning data. If you know the service/version, please submit the foll
SF-Port8080-TCP:V=7.93%T=SSL%I=7%D=4/15%Time=643AB435%P=x86_64-pc-linux-gn
SF:u%r(GetRequest,141,"HTTP/1\.1\x20502\x20Bad\x20Gateway\r\nContent-Type:
SF:\x20text/plain;\x20charset=UTF-8\r\nContent-Length:\x20231\r\n\r\nUma\x
SF:20exce\xc3\xa7\xc3\xa3o\x20ocorreu\x20ao\x20tentar\x20conectar\x20a:\x2
SF:0https://null/\nA\x20exce\xc3\xa7\xc3\xa3o\x20foi:\nnull:\x20Nome\x20ou
SF:\x20servi\xc3\xa7o\x20desconhecido\nO\x20documento\x20a\x20seguir\x20po
SF:de\x20ajudar\x20a\x20resolver\x20esta\x20falha:\nhttps://www\.zaproxy\.
SF:org/faq/why-cant-zap-connect-to-my-website/\x20")%r(HTTPOptions,122,"HT
SF:TP/1\.1\x20502\x20Bad\x20Gateway\r\nContent-Type:\x20text/plain;\x20cha
SF:rset=UTF-8\r\nContent-Length:\x20200\r\n\r\nUma\x20exce\xc3\xa7\xc3\xa3
SF:o\x20ocorreu\x20ao\x20tentar\x20conectar\x20a:\x20https://null/\nA\x20e
SF:xce\xc3\xa7\xc3\xa3o\x20foi:\nnull\nO\x20documento\x20a\x20seguir\x20po
SF:de\x20ajudar\x20a\x20resolver\x20esta\x20falha:\nhttps://www\.zaproxy\.
SF:org/faq/why-cant-zap-connect-to-my-website/\x20")%r(FourOhFourRequest,1
SF:64,"HTTP/1\.1\x20502\x20Bad\x20Gateway\r\nContent-Type:\x20text/plain;\
SF:x20charset=UTF-8\r\nContent-Length:\x20266\r\n\r\nUma\x20exce\xc3\xa7\x
SF:c3\xa3o\x20ocorreu\x20ao\x20tentar\x20conectar\x20a:\x20https://null/ni
SF:ce%20ports%2C/Tri%6Eity\.txt%2ebak\nA\x20exce\xc3\xa7\xc3\xa3o\x20foi:\
SF:nnull:\x20Nome\x20ou\x20servi\xc3\xa7o\x20desconhecido\nO\x20documento\
SF:x20a\x20seguir\x20pode\x20ajudar\x20a\x20resolver\x20esta\x20falha:\nht
SF:tps://www\.zaproxy\.org/faq/why-cant-zap-connect-to-my-website/\x20");
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 77.66 seconds
```

Aqui descobrimos os serviços utilizados. Depois vamos para o FFUF

```
┌──(matheus㉿kali)-[~]
└─$ ffuf -w /usr/share/wordlists/dirb/common.txt -u http://localhost/DVWA/FUZZ -c

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.5.0 Kali Exclusive <3
________________________________________________

 :: Method           : GET
 :: URL              : http://localhost/DVWA/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirb/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
________________________________________________

.htaccess               [Status: 403, Size: 274, Words: 20, Lines: 10, Duration: 1ms]
                        [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 1ms]
config                  [Status: 301, Size: 312, Words: 20, Lines: 10, Duration: 0ms]
database                [Status: 301, Size: 314, Words: 20, Lines: 10, Duration: 0ms]
docs                    [Status: 301, Size: 310, Words: 20, Lines: 10, Duration: 0ms]
external                [Status: 301, Size: 314, Words: 20, Lines: 10, Duration: 0ms]
.hta                    [Status: 403, Size: 274, Words: 20, Lines: 10, Duration: 67ms]
favicon.ico             [Status: 200, Size: 1406, Words: 5, Lines: 2, Duration: 0ms]
index.php               [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 1ms]
.git/HEAD               [Status: 200, Size: 23, Words: 2, Lines: 2, Duration: 145ms]
.htpasswd               [Status: 403, Size: 274, Words: 20, Lines: 10, Duration: 151ms]
php.ini                 [Status: 200, Size: 154, Words: 19, Lines: 6, Duration: 34ms]
phpinfo.php             [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 34ms]
robots.txt              [Status: 200, Size: 26, Words: 3, Lines: 2, Duration: 11ms]
tests                   [Status: 301, Size: 311, Words: 20, Lines: 10, Duration: 0ms]
:: Progress: [4614/4614] :: Job [1/1] :: 12 req/sec :: Duration: [0:00:05] :: Errors: 0 ::
```

Aqui é possível perceber várias pastas que podem ser acessadas a partir da raiz, incluindo “php.ini”. Por fim, a última ferramenta de varredura OWASP ZAP.




Nesta análise o owasp zap não conseguiu extrair muitas informações úteis em comparação com o alvo anterior.

Agora vamos utilizar o sqlmap na área de injeções http://localhost/DVWA/vulnerabilities/sqli/ do site DWVA. Ao tentar consultar o id de valor 1 a URL é alterada da seguinte FORMA:

- http://localhost/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#

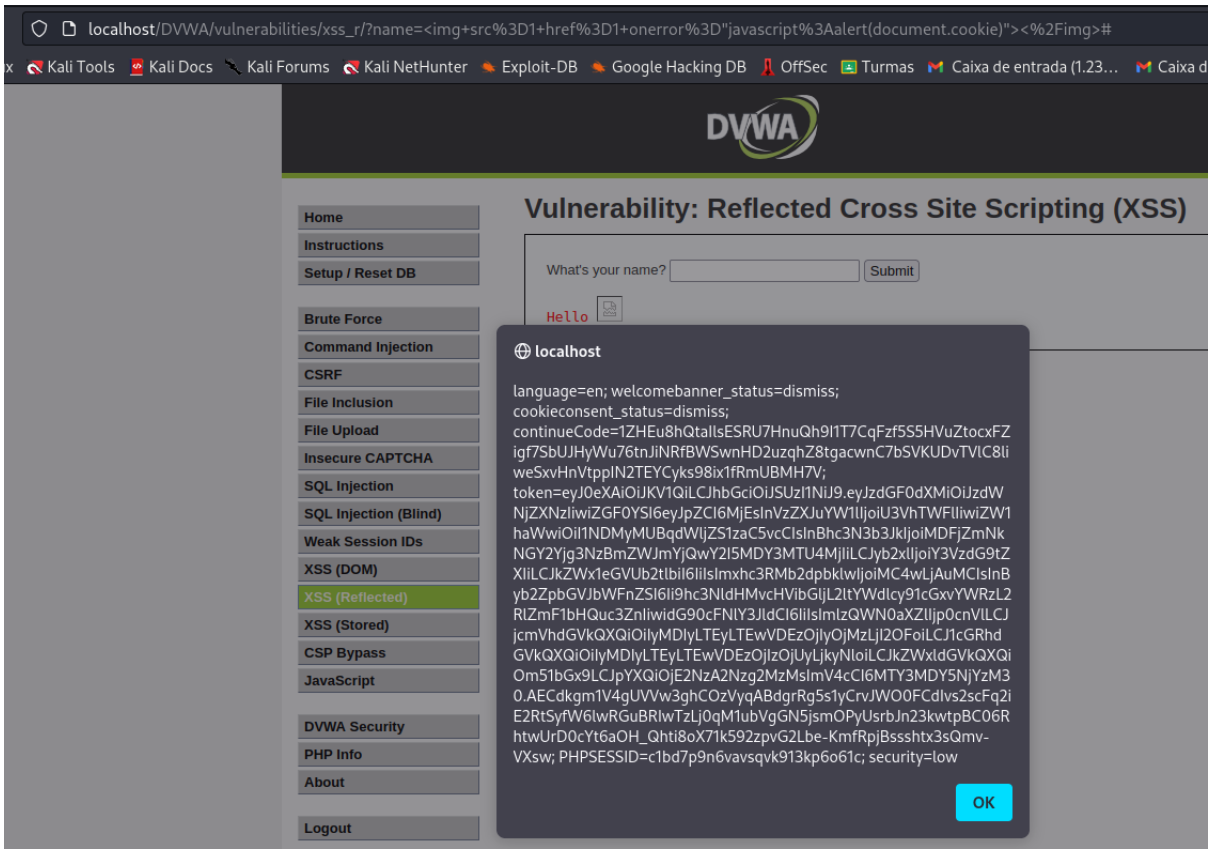
Neste caso há mais de um parâmetro na URL, quando esse é o caso é necessário especificar isso ao sqlmap, para fazer isso a url do sqlmap deve ser colocada entre aspas e após isso deverá ser inserido um novo parâmetro -p que especifica os parâmetros, os parâmetros devem ser inseridos em ordem conforme a URL, estar entre aspas e separados por uma vírgula. Dessa forma o comando fica assim:

- sqlmap -u 'http://localhost/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#' -p "id,Submit" --dbs --current-user --current-db --passwords --schema

Ao executar o comando no prompt, o programa pergunta se ele pode seguir um redirecionamento e logo em seguida fala se pode aceitar os cookies informados, isso nos lembra que o DVWA tem cookies, ao continuar o programa falha em invadir. Após explorar

um pouco mais isso ocorre pela falta da configuração correta de cookies.

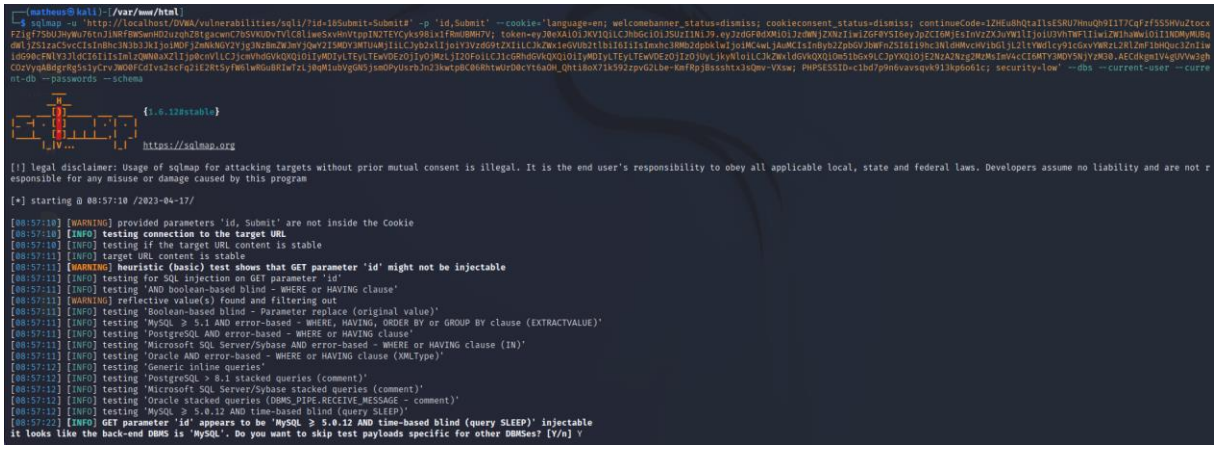
Dessa forma vamos tentar extrair os cookies com a inserção do payload de XSS utilizado anteriormente na página de XSS.

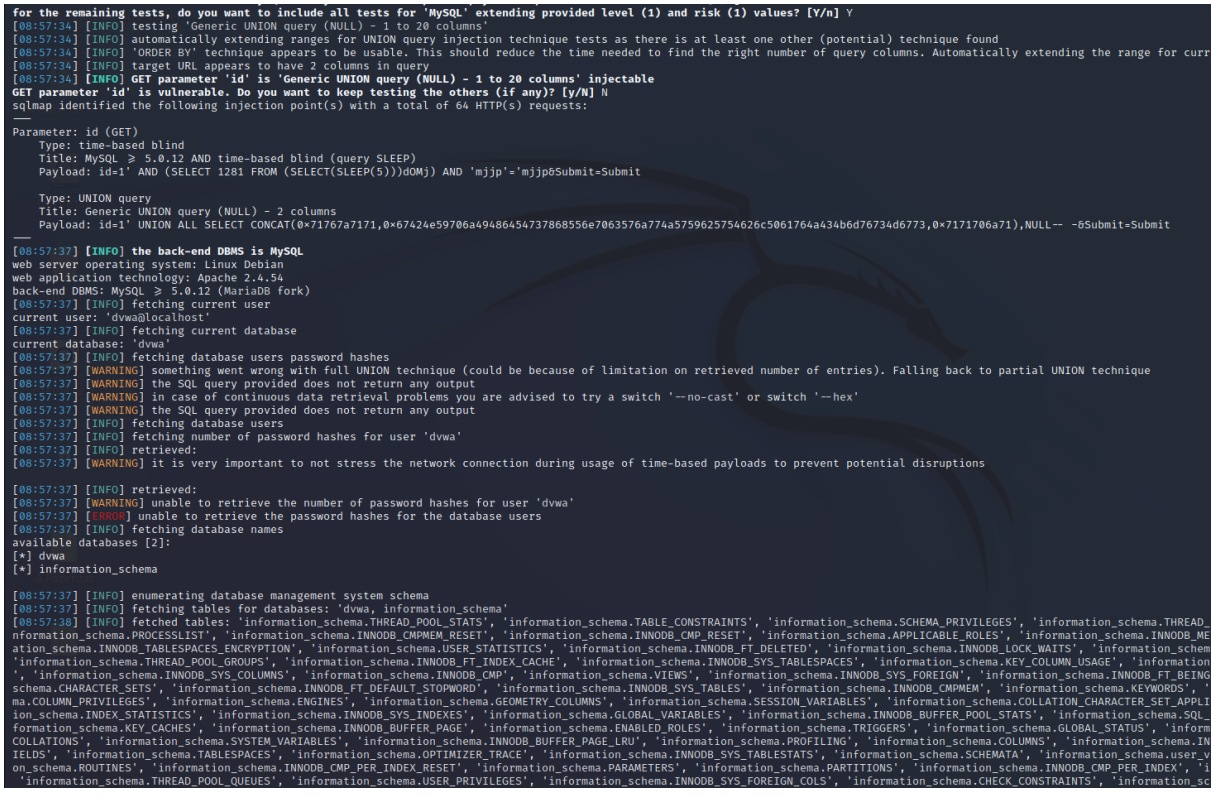



O teste obteve sucesso, agora poderemos utilizar os cookies no sqlmap. Para inserir os cookies no sqlmap use o parâmetro “--cookie=”, depois coloque todo o cookie entre aspas após o símbolo “=”. o Comando ficará extenso e cada um terá o seu de acordo com a sessão.

- sqlmap -u 'http://localhost/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#' -p 'id,Submit' --cookie='language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss; continueCode=1ZHEu8hQtaIlsESRU7HnuQh9I1T7CqFzf5S5HVuZtocxFZigf7SbUJHyWu76tnJiNRfBWSwnHD2uzqhZ8tgacwnC7bSVKUDvTVlC8liweSxvHnVtppIN2TEYCyks98ix1fRmUBMH7V;token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MjEsInVzZXJuYW1lIjoiU3VhTWFlIiwiZW1haWwiOiI1NDMyMUBqdWljZS1zaC5vcCIsInBhc3N3b3JkIjoiMDFjZmNkNGY2Yjg3NzBmZWJmYjQwY2I5MDY3MTU4MjIiLCJyb2xlIjoiY3VzdG9tZXIiLCJkZWx1eGVUb2tlbiI6IiIsImxhc3RMb2dpbklwIjoiMC4wLjAuMCIsInByb2ZpbGVJbWFnZSI6Ii9hc3NldHMvcHVibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHQuc3ZnIiwidG90cFNlY3JldCI6IiIsImlzQWN0aXZlIjp0cnVlLCJjcmVhdGVkQXQiOiIyMDIyLTEyLTEwVDEzOjIyOjMzLjI2OFoiLCJ1cGRhdGVkQXQiOiIyMDIyLTEyLTEwVDEzOjIzOjUyLjkyNloiLCJkZWxldGV

kQXQiOm51bGx9LCJpYXQiOjE2NzA2Nzg2MzMsImV4cCI6MTY3MDY5NjYzM30.AECdkgm1V4gUVVw3ghCOzVyqABdgrRg5s1yCrvJWO0FCdIvs2scFq2iE2RtSyfW6lwRGuBRIwTzLj0qM1ubVgGN5jsmOPyUsrbJn23kwtpBC06RhtwUrD0cYt6aOH_Qhti8oX71k592zpvG2Lbe-KmfRpjBssshtx3sQmv-VXsw;PHPSESSID=c1bd7p9n6vavsqvk913kp6o61c; security=low' --dbs --current-user --current-db --passwords --schema




A injeção SQL foi performada com sucesso e, dessa forma, foi possível obter informações possivelmente valiosas para pentesters.

## Conclusão Crítica

Através desse trabalho foi possível utilizar e aprender sobre as ferramentas de invasão e o Kali Linux para invadir e até explorar algumas vulnerabilidades nos ambientes escolhidos, contudo esses domínios utilizados foram criados para ser invadidos, e suas vulnerabilidades foram criadas para serem exploradas, em um ambiente real não será tão fácil descobrir e explorar vulnerabilidades em comparação com os ambientes de treinamento utilizados (salvo se o alvo não dispor de nenhuma forma de segurança).

Não foi necessário um estudo profundo em relação às ferramentas e suas funcionalidades, nem formas de prevenção de detecção da utilização das ferramentas pois a execução padrão delas já atendia a necessidade, isso ocorre porque as vulnerabilidades eram fáceis de detectar com as ferramentas utilizadas, ou seja, bastava apenas especificar alguns parâmetros e as ferramentas concluíam as análises automaticamente.

## Referências

LINUX, Kali. **Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution**. Disponível em: https://www.kali.org/. Acesso em: 15 abr. 2023.

NMAP.ORG. **Nmap: the Network Mapper - Free Security Scanner**. Disponível em: https://nmap.org/. Acesso em: 15 abr. 2023.

ZAP, Owasp. **OWASP ZAP**. Disponível em: https://www.zaproxy.org/. Acesso em: 15 abr. 2023.

SQLMAP. **Sqlmap: automatic SQL injection and database takeover tool**. Disponível em: https://sqlmap.org/. Acesso em: 15 abr. 2023.

LINUX, Kali. **Ffuf | Kali Linux Tools**. Disponível em: https://www.kali.org/tools/ffuf/. Acesso em: 15 abr. 2023.

INFORCHANNEL. **Brasil é principal alvo de ataques cibernéticos na América Latina**. Disponível em: https://inforchannel.com.br/2023/04/14/brasil-e-principal-alvo-de-ataques-ciberneticos-na-america-latina/. Acesso em: 15 abr. 2023.

BRASIL, Cnn. **Site do Superior Tribunal Militar sofreu ataque hacker no dia da posse de presidente**. Disponível em: https://www.cnnbrasil.com.br/nacional/site-do-superior-tribunal-militar-sofreu-ataque-hacker-no-dia-da-posse-de-presidente/. Acesso em: 15 abr. 2023.