

FRAMEWORK DE CONTEINERIZAÇÃO DOCKER
CTF-ARRAIÁCKER



INFORMAÇÕES DO DOCUMENTO

Condicionante	Responsáveis
Elaboração ou alteração	Giovana Kassime de Souza Chaerki
Orientação	Luiz Roberto Henz
Aprovação	Luiz Roberto Henz

SUMÁRIO

1. OBJETIVOS	4
2. APLICAÇÃO ESCOLHIDA	5
3. CONTÊINER BARRAQUINHA: BANCO DE DADOS :	6
4. CONTÊINER ARRAIÁCKER: APLICAÇÃO WEB (PHP/APACHE) ..	7
5. CONTÊINER: FASE FINAL (SHELL CONTAINER)	14
6. CONCLUSÃO	16

1. OBJETIVOS

Consolidar e demonstrar proficiência prática nos fundamentos da tecnologia Docker e na orquestração de contêineres através da concepção, desenvolvimento e documentação de um desafio de segurança computacional no formato Capture The Flag (CTF) que seja funcional, coeso e reproduzível.

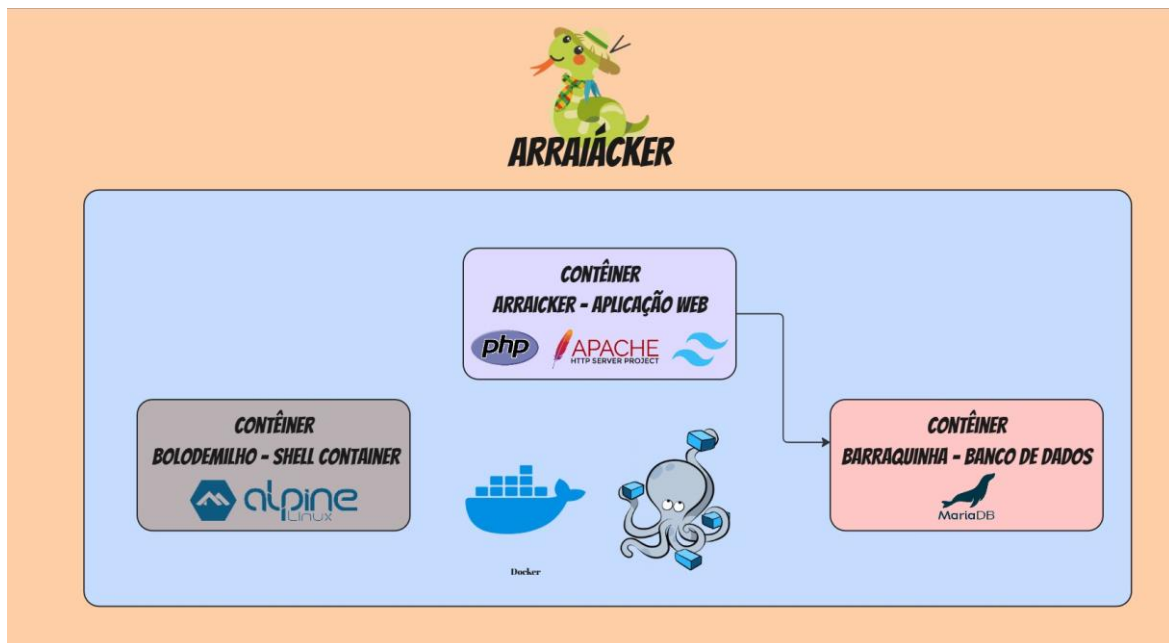
- **Orquestração Multi-Contêiner:** Implementar um ambiente composto por, no mínimo, 3 (três) contêineres distintos, gerenciados de forma coesa.
- **Integração de Banco de Dados(opcional):** Incluir um sistema de gerenciamento de banco de dados como um dos contêineres.
- **Aplicação da Criatividade:** Desenvolver uma narrativa ou um tema original e criativo

2. APLICAÇÃO ESCOLHIDA

Para atender aos objetivos de consolidar os estudos de Docker de forma prática e criativa, a aplicação escolhida consiste no desenvolvimento de um desafio de *Capture The Flag* (CTF) temático, intitulado "**Arraiácker**".

Atendendo ao objetivo de **Aplicação da Criatividade**, o projeto **Arraiácker** adota o tema de uma tradicional Festa Junina brasileira, onde o participante do CTF terá que descobrir as flags como ponto de partida uma página com um formulário.

Todas as flags contidas neste relatório estão ocultas para que o usuário tenha que percorrer o caminho para encontrá-las de verdade.



3. CONTÊINER BARRAQUINHA: BANCO DE DADOS :

Tecnologias Utilizadas:

- **Imagem Docker:** *mariadb:latest*. Trata-se de um sistema de gerenciamento de banco de dados relacional, robusto e de código aberto, derivado do MySQL.
- **Inicialização de Dados:** Um script *init.sql* é executado na primeira vez que o contêiner é criado. Este script cria o banco de dados barraquinha e tem diversas tabelas.

Função no Desafio e Fluxo de Interação:

O propósito principal deste contêiner não é servir a uma aplicação convencional, mas sim ser o alvo direto de um ataque de **SQL Injection**. O fluxo ocorre da seguinte maneira:

- **Estrutura do Banco de Dados:** O script *init.sql* cria a tabela principal do desafio, **caipira**, que armazena os usuários e senhas, incluindo um usuário principal e uma nota secreta que serve para descobrir a página de gerar relatórios. Além destas, são criadas várias tabelas de distração (barracas, produtos etc.) para simular um ambiente real e confundir o atacante.
- **Conexão com a Aplicação Web:** A aplicação no contêiner arraiacker se conecta a este banco de dados usando as credenciais definidas no *docker-compose.yml* e no arquivo *config/database.php*.
- **Execução da Query Vulnerável:** O usuário não interage diretamente com o banco. A interação é feita através do formulário de login da aplicação web. Quando o usuário tenta fazer login, a aplicação arraiacker monta e executa uma consulta SQL vulnerável. É neste ponto que o banco de dados processa a entrada maliciosa, permitindo que o atacante manipule a consulta e extraia informações sensíveis.

4. CONTÊINER ARRAIÁCKER: APLICAÇÃO WEB (PHP/APACHE)

O contêiner arraiacker serve como a interface do desafio, sendo o ponto de entrada para o ataque e a plataforma onde a vulnerabilidade principal reside.

Tecnologias Utilizadas:

- **Servidor Web e Linguagem:** A aplicação é construída em **PHP** e servida por um ambiente que, pela estrutura de arquivos e build do Docker, é um servidor **Apache**.
- **Frontend:** Utiliza **Tailwind CSS** para a estilização.
- **Roteamento:** Um roteador simples em PHP, localizado em `public/index.php`, gerencia todas as rotas da aplicação.

Função no Desafio e Fluxo de Interação:

Este contêiner hospeda a vulnerabilidade de SQL Injection que permite ao participante extrair dados do banco barraquinha e outras flags de vulnerabilidades web.

- **Página Home:** O desafio começa na página home, acessada pela rota `/`. Aqui, o usuário se depara com um botão para ir para o primeiro desafio que é o `/login`.



-

- ARRAIÁCKER CTF**

Acesso Não Autorizado Detectado

Hmmm, um calpira curioso... Parece que você encontrou algo que não devia.

Recompensa: Nomes das Tabelas

Você conseguiu injetar um comando SQL. Como prêmio pela sua astúcia, aqui estão os nomes das tabelas do nosso sistema. Use com sabedoria!

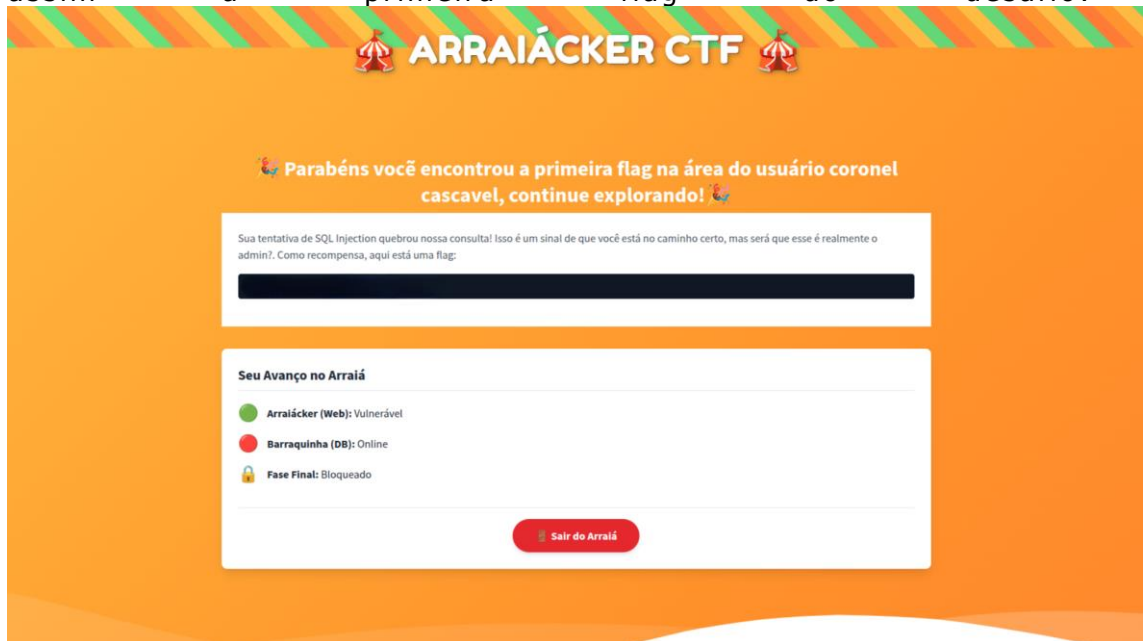
```
Array
(
    [0] => Array
        (
            [Tables_in_barracaquinha] => barracas
        )
    [1] => Array
        (
            [Tables_in_barracaquinha] => calpira
        )
    [2] => Array
        (
            [Tables_in_barracaquinha] => cronograma_eventos
        )
    [3] => Array
        (
            [Tables_in_barracaquinha] => fornecedores
        )
    [4] => Array
        (
            [Tables_in_barracaquinha] => sqlmap
        )
)
```

Resultado da sua Exploração

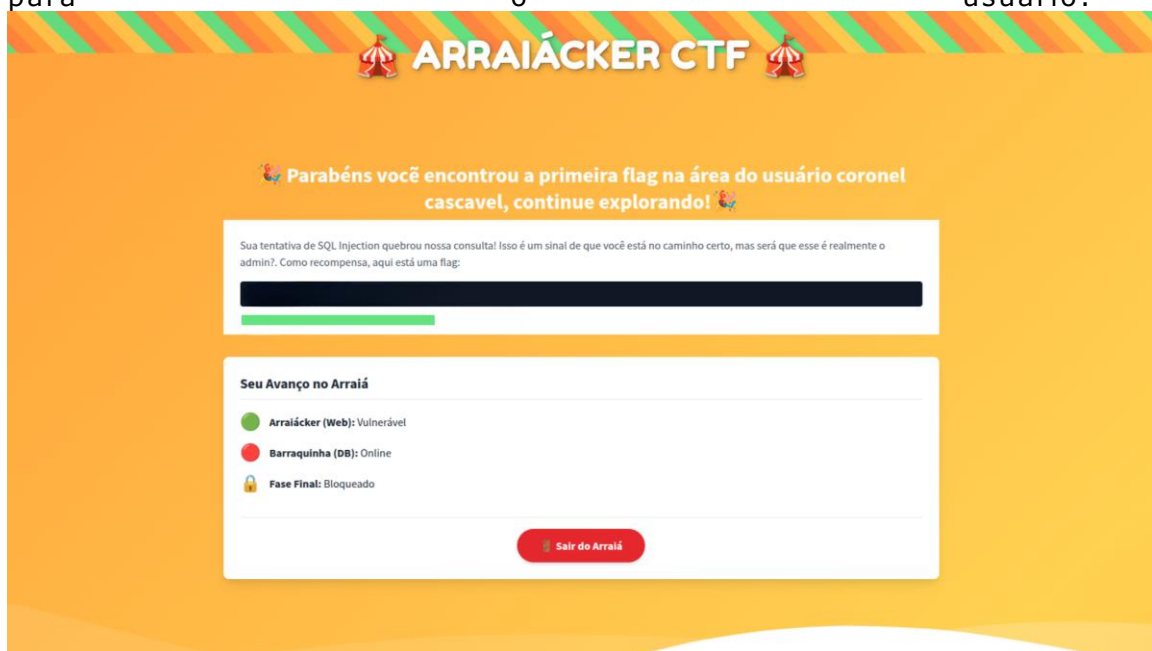
Executamos uma consulta na tabela "calpira" usando seu input. Veja o que encontramos:

```
Array
(
    [0] => Array
        (
            [id] => 1
            [username] => maria.pipoca@arraiacker.com
            [password] => pipoca123
            [role] => user
        )
    [1] => Array
        (
            [id] => 2
            [username] => chico.quanta@arraiacker.com
            [password] => adrofesta
            [role] => admin
        )
    [2] => Array
        (
            [id] => 3
            [username] => ana.canjica@arraiacker.com
            [password] => alhoverde
            [role] => guest
        )
)
```


- **Página Flag 2:** O objetivo é descobrir o usuário e a senha corretos para fazer o login legítimo e ser redirecionado para /flag2, obtendo assim a primeira flag do desafio:



- É deixado um elemento oculto com um token codificado em base64 para o usuário:



- **Página Flag 3:** A segurança do serviço que protege a Flag 3 é falha porque se baseia em um mecanismo de autenticação fraco. Em vez de usar sessões seguras ou tokens dinâmicos e assinados, o sistema apenas valida a presença de um token estático em um cabeçalho HTTP customizado. Uma vez que o atacante descobre o segredo e o cabeçalho correto, ele obtém acesso irrestrito ao /dashboardmilhao.

Exemplo usando o "curl":

```
curl -i -H "X-Auth-Token: aquivaiotoken" http://localhost:8589/
```

Resultado:

```
<a target="_blank" href="http://localhost:8589/dashboardmilhao?aquifaltaalgo">Parabéns continue por aqui</a>
```

ARRAIÁCKER CTF

Dashboard de Controle do Arraiácker
Visão geral de todas as operações do evento. Parabéns, Chefe!

Caipiras (Usuários)

Role:

Role:

Role:

Role:

Barracas do Evento

Tipo: Brincadeira | Responsável: Pedro Pescador

Tipo: Bebida | Responsável: Chico Quantão

Tipo: Brincadeira | Responsável: Rosinha Correio

Tipo: Comida | Responsável: Maria da Silva

Produtos & Preços

Preço: R\$ 8,00 | Estoque: 150

Preço: R\$ 7,50 | Estoque: 120

Preço: R\$ 5,00 | Estoque: 500

Preço: R\$ 6,00 | Estoque: 400

Cronograma da Festa

Horário	Evento	Local
18:00:00	Abertura dos portões e das barracas	Entrada Principal
19:30:00	Quadrilha Infantil "Os Pipoqueinhas"	Palco Principal
21:00:00	Acendimento da Fogueira Principal	Pátio Central

- Aqui é deixado um bilhete com o caminho para a próxima fase do desafio que é /gerador_de_relatorio.

🔒 Notas Secretas dos Organizadores

Senhas WiFi do Evento

WiFi_Visitantes: FestaJunina2025! | WiFi_Staff: AcessoRestritoStaff** | WiFi_Caixa: NaoMexer!!

Combinação do Cadeado do Gerador

A senha é o ano do primeiro arraí: 1998. Mas parece que o Zé perdeu a chave de novo.

Lembrete Urgente

Falar com o Coronel sobre a segurança da barraca de prêmios. A senha do cofre dele é muito fraca, algo como o nome do cavalo dele. Seria "Trovão"? Preciso confirmar.

TODO Urgente - TI

O novo gerador de relatórios (baseado em XML) está em teste no endpoint /gerador_de_relatorio

[Encerrar Sessão e Sair do Painel](#)

- Página Gerador de Relatório:** A última etapa do desafio testa a habilidade do participante em explorar vulnerabilidades em processadores de XML. O ponto de entrada é uma funcionalidade de "Gerador de Relatório", que foi configurada de forma insegura, permitindo a execução de um ataque de XML External Entity (XXE).

ARRAIÁCKER CTF


Gerador de Relatórios Customizados

Cole o template XML abaixo para gerar o relatório do evento.

Template XML:

Ex: <relatorio><titulo>Vendas</titulo></relatorio>


[Gerar Relatório](#)

**ARRAIÁCKER CTF**

Desafio de Cybersecurity com tema junino, caso precise de ajuda com alguma fase, acesse a documentação abaixo

[LINKS](#)

- [GitHub](#)
- [Documentação](#)

**DESENVOLVEDOR**

- [Sobre o Projeto](#)
- [Contato](#)

Arraiácker CTF 2025 - Desafio de Docker e Cybersecurity | Criado por **Kassime**

- Se o usuário inserir um XML simples e bem formado, sem nenhuma "palavra-chave de ataque", o servidor responde com uma mensagem padrão, confirmando que o relatório foi gerado:

Gerador de Relatórios Customizados

Cole o template XML abaixo para gerar o relatório do evento.

Template XML:

Digitando qualquer coisa

Gerar Relatório

Relatório Processado

Relatório gerado, mas nada de interessante encontrado. Continue tentando.

- Para explorar a vulnerabilidade, o participante precisa adicionar as "palavras-chave de ataque", que são a declaração `<!DOCTYPE ... ENTITY>`. Essa declaração define uma entidade externa e a associa a um arquivo no servidor que contém o link para a flag final /flag4:

Gerador de Relatórios Customizados

Cole o template XML abaixo para gerar o relatório do evento.

Template XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [ <!ENTITY xxe SYSTEM "file:///etc/passwd"> ]>
<stockCheck><productId>&xxe;</productId></stockCheck>
```



Gerar Relatório


🎉 Parabéns, Hacker do Sertão! 🎉

Sua entidade externa foi processada! Você detectou a falha no nosso parser XML.

Pegar a Última Flag

- **Página Flag 4:** Aqui é deixado o prêmio final que é as credenciais de acesso ao terceiro e último contêiner.

 **ARRAIÁCKER CTF FINAL** 




🏆 PARABÉNS, MESTRE CAIIRA! 🏆

Sua jornada pelo Arraiácker o trouxe até o tesouro final. Use as credenciais abaixo para o acesso SSH na porta 2222 e complete o desafio.

Usuário:

Senha:

**ARRAIÁCKER CTF**
Desafio de Cybersecurity com tema junino

[LINKS](#)
[Docker Hub](#)
[GitHub](#)

[DESENVOLVEDOR](#)
[Sobre o Projeto](#)
[Contato](#)

🔥 Arraiácker CTF 2025 - Desafio de Docker e Cybersecurity | Criado por **Kassime**

5. CONTÊINER: FASE FINAL (SHELL CONTAINER)

Ele vai conter dois usuários, o usuário organizador e o usuário admin, onde vai estar a última flag que é o caminho de uma página no arraiacker no qual a sua será passado uma frequência descoberta neste container também, chegando assim na flag final.

- **Fase 1:** O objetivo desta fase é que o usuário entre com as credenciais encontradas na última flag do container "Arraiácker", entrando na home do caipira ele tem uma flag: _____

```
b5f46ab6e8a5:~$ whoami
caipira
b5f46ab6e8a5:~$ ls -la
.          ..          .parabens.txt
b5f46ab6e8a5:~$ cat .parabens.txt
Parabéns! Você conseguiu acesso como 'organizadores'. Bom trabalho!

A jornada está apenas começando.

Para provar seu valor e continuar, você precisa escalar seus privilégios.
Uma dica para encontrar a senha do administrador esta por aqui.b5f46ab6e8a5:
```

- **Fase 2:** O objetivo desta fase é o usuário descobrir um arquivo oculto em uma pasta, e nele vai conter um texto criptografado

```
43f0f2676cab:~$ find / -name "*txt*" 2>/dev/null
```

[REDACTED]

```
- /var/lib/docker/images/overlay2/43f0f2676cab-113.txt
```

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]43NA==

- /var/lib/docker/images/overlay2/43f0f2676cab-113.txt 1/1 100%

- **Fase 3:** O objetivo é que o usuário descriptografe o arquivo anterior e entre com o usuário admin e a senha descriptografada;

```
Kassime@WS-SC10A-T001864:~/Área de trabalho/Framework-de-conteineriza-o$ ssh admin@localhost -p 2222
admin@localhost's password:
Welcome to Alpine!

The Alpine Wiki contains a large amount of how-to guides and general
information about administrating Alpine systems.
See <https://wiki.alpinelinux.org/>.

You can setup the system with the command: setup-alpine

You may change this message by editing /etc/motd.

dea6cc793fdf:~$ whoami
admin
```

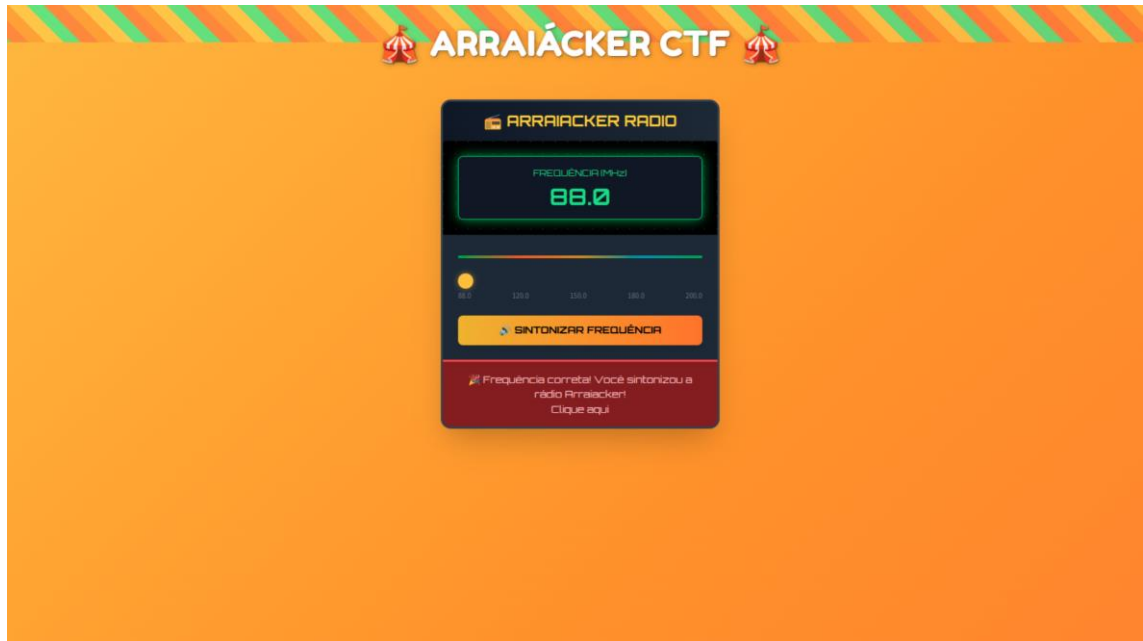
- **Fase 4:** O objetivo é que o usuário encontre o arquivo na home do admin chamado parabensadmin.txt

```
Acesso total concedido! Voc.. agora .. o administrador deste sistema.  
Voc.. provou sua habilidade. Agora, o verdadeiro pr..mio.  
A flag final n..o .. um texto.
```

- E o arquivo secreto em uma pasta:

```
 siga adiante /r4d90fl4g  
 - 1/1 100%
```

- **Fase 5:** O objetivo é que o usuário retorne a página web Arraiácker e entre no caminho `/r4d90fl4g` e coloque a frequência e assim obtenha a flag final:



- **Última flag:**



6. CONCLUSÃO

O projeto "Arraiácker" atingiu com sucesso seus objetivos ao desenvolver um desafio de Capture The Flag (CTF) funcional, coeso e criativo. A arquitetura demonstrou proficiência em orquestração ao empregar um ambiente com três contêineres interdependentes (aplicação web, banco de dados e shell), enquanto a original temática de Festa Junina enriqueceu a narrativa e os desafios técnicos.

O roteiro de exploração abrangeu múltiplas vulnerabilidades, desde SQL Injection e XXE na interface web até a escalada de privilégios no contêiner final. O uso do Docker garante a total reprodutibilidade do ambiente, consolidando o "Arraiácker" como uma ferramenta de aprendizado eficaz, que ilustra de forma prática e completa a aplicação de tecnologias de contêineres em cenários de cibersegurança.



itaipu
parquetec