

SEMANA 05

Info Gathering - Business

" Give me six hours to chop down a tree and I will spend the first four sharpening the axe."

-- Abraham Lincoln

Coleta via Vagas de Emprego

Busca de informações públicas → coleta passiva → OSINT

Requisitos da vaga: (exemplo)

+ Administração do Firewall FORTINET

+ Conhecimento em Weblogic 11, PL/SQL e banco de dados Oracle 11g e superior...

→ as info ajudam a ter uma visibilidade maior do ambiente de ataque

Mapeando servidores

+ Pesquisa no linkedin da empresa

+ Mapear pessoas envolvidas com a tecnologia (Desenvolvedores, TI, Segurança)

+ Pesquisar o perfil desses funcionários -> github, boris, projetos, etc..

Coleta via Endereços de E-mail

+ Site para coleta e verificação de emails:

<https://hunter.io/>

→ Encontra emails de usuários da plataforma da empresa

→ Testa a validação do email

Vazamento de dados (Leaks)



+ É o vazamento de dados

+ Consulta o email em uma gama de bases: <https://haveibeenpwned.com/>

Consultando Leaks na Dark Web

Sites para consulta:

+ dehashed.com

+ <https://pwndb2am4tzkvold.onion/>

Ma só funciona com o tor

Utilizando a Rede Tor no Kali

+ Pode-se navegar pelo firefox passando pela rede Tor

+ devemos instalar o TOR e o proxychains

```
apt install tor proxychains  
service tor start  
netstat -nlpt
```

+ O serviço estará rodando na porta 9050

+ Devemos configurar o proxychains

```
nano /etc/proxychains.conf
```

→ Descer no final do arquivo e criar o sock 5: socks5 127.0.0.1 9050

→ Usar o # para comentar o strict

→ Retirar o # do dynamic

→ Salvar as configurações

+ Devemos configurar o navegador:

Preferences ⇒ Network Proxy ⇒ Settings (config do proxy) ⇒ Manual Proxy Configuration

Socks Host 127.0.0.1 Port 9050

Socksv5

Proxy DNS when using SOCKv5

HTTPS Proxy Port

SOCKS Host Port

☐ SOCKS v4 ☒ SOCKS v5

☐ Automatic proxy configuration URL

No proxy for

Example: .mozilla.org, .net.nz, 192.168.1.0/24

Connections to localhost, 127.0.0.1/8, and ::1 are never proxied.

☐ Do not prompt for authentication if password is saved

☒ Proxy DNS when using SOCKS v5

Script para Consulta de Leaks

~~~~~  
KARMA

~~~~~  
+ Basta copiarmos do github e colarmos no kali

```
git clone https://github.com/limkokhole/karma ; cd karma
```

+ Quando dentro do diretório, aplicamos:

```
pip3 install -r requirements.txt
python3 setup.py build
python3 setup.py install
service tor start
```

+ Para pesquisar:

```
karma search 'longatto' --password  
karma search 'tesla.com' --domain
```

Coleta de Dados no Pastebin

- + Site para postar conteúdos de forma rápida
- + Amplamente utilizado para publicação de Leaks
- + Maneira de fazer a pesquisa pelo Google:

```
site:pastebin.com "businesscorp.com.br"
```

- + Podemos criar uma conta gratuita para receber notificações via email para cada atualização relacionada a algum tema

Coleta de Dados no Trello

- + Ferramenta de administração
- + Permite o compartilhamento de info com um grupo ou equipe
- + **Vulnerabilidade:** Quando a função compartilhar atividades está ativada no modo público.
- + Modo de pesquisa:

```
site:trello.com ".com.br" projeto  
site:trello.com "businesscorp.com.br"  
site:trello.com "senhas"
```

Buscando Domínios Similares

- + O atacante pode registrar um domínio similar para realizar ataques direcionados à empresa ou aos clientes
- + Exemplos de links similares

www.businesscorp.co.br
www.businesscorp.com.br
www.businesscorp.com.br

- + Ferramenta no kali que automatiza a mudança de link: urlcrazy

```
urlcrazy businesscorp.com.br
```

- Faz variações de omissão, adição e troca de caracteres.
- Verifica quais já estão em uso

Pesquisando Cache de Sites

- + Site para consultas: www.web.archive.org

- + Permite acessar versões antigas dos sites
- + Podemos acessar info pertinentes que estavam expostas anteriormente

+ **Filme para assistir:** A Rede Social

Introdução ao Google Hacking

+ Buscando informações no servidor do Google:

site: nome do site especificado

inurl: busca na url

intitle: busca no título

intext: busca no texto

" ": pesquisa exata

filetype: busca por tipo de arquivo

ext: busca por extensão do arquivo

cache: busca no cache

- : omite na busca

+ Exemplos:

site: businesscorp.com.br **-www** **ext:** php

site: businesscorp.com.br **intitle:** "Admin"

site: businesscorp.com.br **"index of"** backup

Google Hacking

+ Tipos de arquivos e extensões:

php, asp, do, js, phps, txt, doc, docx, pdf

xls, xlsx, ppt, opvn, sql, bak, old

+ Títulos:

"index of", "login", "acesso restrito", "admin", "adm"

+ Palavras Chave:

"config", "senha", "senhas", "usuarios", "acesso",

"ftp", "bkp", "backup", "dados"

+ Combinações:

.com.br filetype:txt senha

→ sites .com.br que tenha, a palavra "senha" em algum lugar e que tenham arquivos txt

filetype:txt inurl:senha

→ digamos que acessei o link e ele foi removido do ar ⇒ podemos ver os caches (versões antigas dele)

cache: link.com

filetype:sql :com.br

→ base de dados (senhas, etc)

filetype:sql :gov.br backup

→ pode ser uma boa ideia pesquisar por extensões .bak ou .old

:com.br filetype:ovpn

→ conseguir o acesso à rede do cliente

Google Hacking aplicado ao Pentest

site: businesscorp.com.br

→ Retorna outros subdomínios/endereços que podem conter vulnerabilidades e vir a ser vetores de ataque.

+ No caso de realizar um pentest em empresas maiores como o Itaú, devemos ir afunilando a pesquisa para que os resultados se tornem mais analisáveis

+ Exemplo:

itau.com.br → + 17.10⁶ resultados

site:itau.com.br ext:php → 7 resultados

Google Dork/ GHDB/ Script

+ Dork: combinação de operadores de pesquisa no Google

+ Exemplo:

site:gov.br filetype:sql mysqldump

site: com.br inurl:ftp://ftp.

+ GHDB: Google Hacking Database → site com dorks já prontas

+ Montagem do script...

→ Como fazer uma pesquisa no navegador usando o terminal:

```
firefox 'https://google.com/search?q=site:businesscorp.com.br'
```

→ esse comando abre o site e já realiza a pesquisa

```
firefox 'https://google.com/search?q=site:businesscorp.com.br+inurl:rh'
```

+ Script

```
nano search.sh
```

```
#!/bin/bash
SEARCH = "firefox"
ALVO = "$1"

echo "Pesquisa no Pastebin"
$SEARCH "https://google.com/search?q=site:pastebin.com+$ALVO" 2>/dev/null

echo "Pesquisa no Trello"
$SEARCH "https://google.com/search?q=site:trello.com+$ALVO" 2>/dev/null

echo "Pesquisa por arquivos"
```

```
$SEARCH "https://google.com/search?q=site:$ALV0+ext:php+0R+ext:asp+0R+ext:txt"
2>/dev/null
```

Bing Hacking

+ O Bing é um buscador como o Googl, tanto que os parâmetros de filtro são os mesmos (site, filetype ...) → ele indexa info na Internet

+ **Recurso interessante:** Pesquisa por ip
ip "37.59.174.225"

+ Alguns sites podem hospedar-se em provedores (domínios) pagos ⇒ se algum deles tiver alguma vulnerabilidade, todos os outros tendem a ter

NDN - Non Delivering Notification

+ Coleta de informações via email

→ enviar um email inexistente para o alvo

+ Exemplo: dontreply1000@businesscorp.com.br

→ Pode expor:

- Subdomínios
- Info sobre endereçamentos de IP da rede local
- Versões de Software

The Harvester

+ Atualiza na última versão

```
apt install theharvester
```

+ Para chamar o programa:

```
theHarvester
```

+ Para entender as funcionalidades dele:

```
theHarvester -h
```

+ Para a coleta de info:

```
theHarvester -d businesscorp.com.br -l 100 -b google -f resultado.html
```



```
theHarvester -d fariasbrito.com.br -l 500 -b all
```

```
cd /etc/theHarvester  
ls
```

api-keys.yaml ...

```
nano api-keys.yaml
```

→ o arquivo vem zerado ⇒ devemos criar uma conta no github, hunter, shodan, ... pegar a api e posicionar nesse arquivo, para que o theHarvester possa ser executado corretamente.

Coleta de Info através de Metadados

- + Encontrar documentos da empresa na internet (google hacking/ dorks)
- + Baixar os documentos (doc, docx, xls, xlsx, ppt, pptx, pdf)
- + Analisar metadados
- + Buscar por nome de usuários, versões de software e sistema operacional
- + Ferramenta do kali que permite fazer a leitura dos metadados: [exiftool](#)

```
root@pentest:~/Desktop# exiftool Pentest-Profissional.pdf  
ExifTool Version Number      : 11.76  
File Name                    : Pentest-Profissional.pdf  
Directory                   : .  
File Size                    : 9.6 MB  
File Modification Date/Time  : 2019:03:09 19:58:20-03:00  
File Access Date/Time       : 2019:11:23 18:37:37-03:00  
File Inode Change Date/Time  : 2019:11:23 18:37:37-03:00  
File Permissions             : rw-r--r--  
File Type                    : PDF  
File Type Extension          : pdf  
MIME Type                    : application/pdf  
Linearized                   : No  
Page Count                   : 11  
PDF Version                  : 1.4  
Title                        : Ementa-PP  
Producer                     : macOS Version 10.14.3 (Build 18D109) Quartz PDFContext  
Creator                      : PowerPoint  
Create Date                  : 2019:03:09 19:36:38Z  
Modify Date                   : 2019:03:09 19:36:38Z
```

Script para Análise de Metadados

- + Instalação do lynx

```
sudo apt-get update  
sudo apt-get -y install lynx
```

- + Forma de uso:


```
lynx --dump terra.com.br
```

+ Para fazer pesquisa no Google:

```
lynx google.com
```

ou

```
lynx --dump "https://google.com/search?q=site:dsecsecurity.com+ext:pdf" |  
grep ".pdf"  
| cut -d "=" -f2 | egrep -v "site|google" | sed 's/...$//' > dsec
```

OBS: 's/...\$//' substitui os 3 últimos caracteres por nada

```
for url in $(cat dsec); do wget -q $url; done
```

+ Para fazer a análise dos metadados:

```
exiftool *.pdf
```

+ Lembre que o google pode nos bloquear caso façamos muitas pesquisas recorrentes ⇒ devo usar com moderação

LAB - SEM 05 - Info Gathering - Business

LAB01: camila@businesscorp.com.br, rogerio@businesscorp.com.br, ti@businesscorp.com.br

→ Basta executar o wget no domínio businesscorp.com.br e em seguida filtrar a saída com um grep nos @:

```
cat index.html | grep "@"
```

LAB02: [Gh4ck1ng9988299311](#)

→ basta fazer o seguinte filtro no google:
Site:businesscorp.com.br "key"

LAB03: [c4ch3_1666277399911a](#)

Procuramos pelo arquivo robots.txt e percebemos que a permissão de indexação era dada apenas ao /configuracoes/comunicacao/projeto.txt
Para visualizar mensagens antigas desse arquivo, recorreremos ao site do webarchive.org, que nos dá a key e um site do trello

LAB04: [123qweAm,webmin](#)

→ basta entrar no link do trello que foi disponibilizado no LAB passado

LAB05: camila@businesscorp.com.br, [ca123456](#)

→ basta utilizar a seguinte dork:
site:pastebin.com "businesscorp"

LAB06: [RI.doc](#)

→ site:businesscorp.com.br intitle:index ⇒ index of /ri

LAB07: [rogerioseverovisk](#)

→ basta abrir o doc do lab passado

LAB08: 8812737123129912s

→ basta visitar a pagina do rogerio no linkedin

LAB09: Netgate-pfSense-Firewall

→ na mesma página do linkedin podemos ver essa informação nos requisitos procurados

Info Gathering - INFRA

Internet Assigned Numbers Authority - IANA

+ Responsável por coordenar alguns dos elementos chaves para manter a internet operacional

→ Gerenciamento dos root servers (Domain Names)

→ Coordenação dos números IP e ASN (Autonomous System Numbers)

→ Registro de protocolos

iana.org

iana.org/domains/root/servers

iana.org/numbers

RIRs (Regional Internet Registries)

iana.org/numbers



+ Netblock X ASN

Netblock (Bloco de Rede): Um range ou conjunto de endereços IP

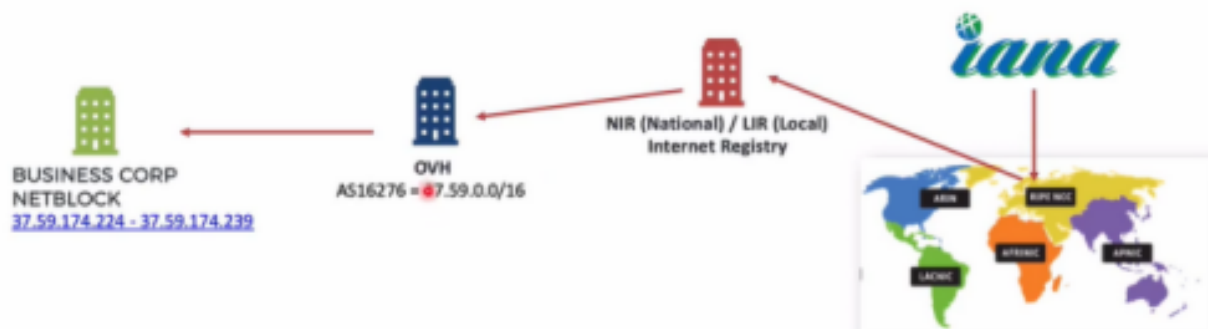
→ Empresa que precisa de um range de 12 IPs pode comprar um netblock

Autonomous System (Sistema Autonomo): Um ou mais blocos de rede sob o mesmo administrador

→ Empresa que precisa de centenas de endereços IPs pode comprar um ASN

+ Exemplo

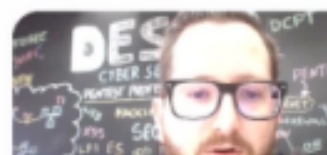
Exemplo:



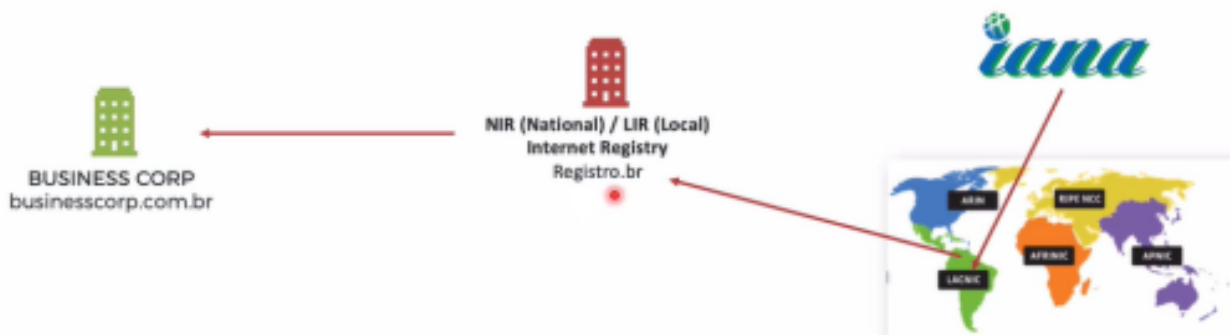
Exemplo:

IP = 37.59.174.225 disponibilizado pelo OVH e sob responsabilidade do NIR que é gerido pelo (RIR) RIPE NCC que é coordenado pela IANA.

IANA -> RIPE NCC -> NIR -> OVH -> BUSINESSCORP



Exemplo:



Exemplo:

IANA -> LACNIC -> REGISTRO.BR -> BUSINESSCORP

Coletando Info com o Whois

+ Primeiramente, devemos buscar o Whois da IANA:

iana.org/whois

businesscorp.com.br

Submit

```
% IANA WHOIS server
% for more information on IANA, visit http://www.iana.org
% This query returned 1 object
```

```
refer:      whois.registro.br
```

```
domain:     BR
```

```
organisation: Comit  Gestor da Internet no Brasil
address:     Av. das Na  es Unidas, 11541, 7. andar
address:     S o Paulo SP 04578-000
address:     Brazil
```

```
contact:    administrative
name:        Demi Getschko
organisation: Comit  Gestor da Internet no Brasil
address:     Av. das Na  es Unidas, 11541, 7. andar
address:     S o Paulo SP 04578-000
address:     Brazil
phone:       +55 11 5509 3505
fax-no:      +55 11 5509 3501
e-mail:      demi@registro.br
```

→ Veja que o campo “refer” aponta para o whois.registro.br

+ Faremos a busca mais detalhada no whois.registro.br

Domínio **businesscorp.com.br**

TITULAR

Desec Security Segurança da Informação LTDA

DOCUMENTO

23.019.510/0001-06

RESPONSÁVEL

Desec Security

PAÍS

BR

CONTATO DO TITULAR

JORLO47

CONTATO TÉCNICO

JORLO47

SERVIDOR DNS

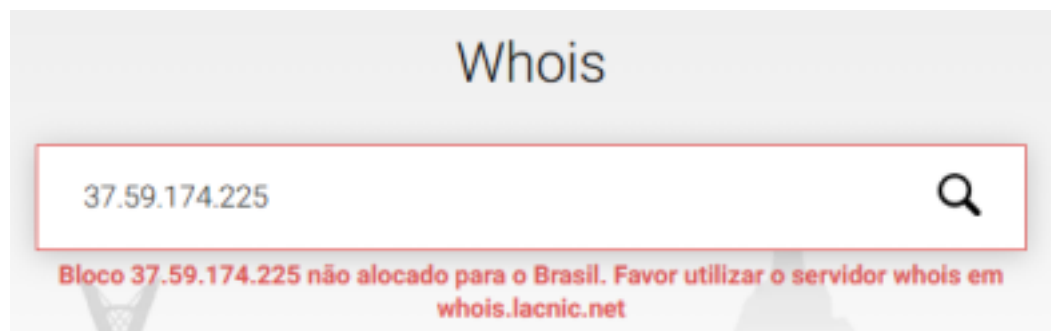
ns1.businesscorp.com.br 37.59.174.225 ~

SERVIDOR DNS

ns2.businesscorp.com.br 37.59.174.226 ~

→ a busca no registro.br nos apontou um endereço de ip:
37.59.174.225

→ Isso permite uma nova consulta no registro.br



→ A pesquisa nos aponta para o whois.lacnic.net, o qual faremos a busca

+ inetnum: 37.59.174.224 - 37.59.174.239 [range de IP's]
+ origin: AS16276

~~~~~  
Whois no terminal  
~~~~~

+ No terminal do kali, o whois também começa a pesquisa pelo site da IANA

```
whois businesscorp.com.br
```

```

root@pentest:~/Desktop# whois businesscorp.com.br

% Copyright (c) Nic.br
% The use of the data below is only permitted as described in
% full by the terms of use at https://registro.br/termo/en.html ,
% being prohibited its distribution, commercialization or
% reproduction, in particular, to use it for advertising or
% any similar purpose.
% 2019-11-12T16:04:13-03:00

domain:      businesscorp.com.br
owner:       Desec Security Segurança da Informação LTDA
ownerid:     23.019.510/0001-06
responsible: Desec Security
country:     BR
owner-c:     JORL047
admin-c:     JORL047
tech-c:      JORL047
billing-c:   JORL047
nserver:     ns1.businesscorp.com.br 37.59.174.225

```

+ Para aparecer o refer, basta que forcemos o processo com o seguinte comando:

```
whois -h whois.iana.org businesscorp.com.br
```

```

root@pentest:~/Desktop# whois -h whois.iana.org businesscorp.com.br
% IANA WHOIS server
% for more information on IANA, visit http://www.iana.org
% This query returned 1 object

refer:       whois.registro.br
domain:      BR
organisation: Comitê Gestor da Internet no Brasil
address:     Av. das Nações Unidas, 11541, 7º andar
address:     São Paulo SP 04578-000
address:     Brazil

contact:     administrative
name:        Demi Getschko
organisation: Comitê Gestor da Internet no Brasil
address:     Av. das Nações Unidas, 11541, 7º andar
address:     São Paulo SP 04578-000
address:     Brazil
phone:       +55 11 5509 3505
fax-no:      +55 11 5509 3501
e-mail:      demi@registro.br

```

Estudando Like a Pro - WHOIS

+ O objetivo é aprender a aprender sobre o funcionamento de uma ferramenta usando o whois como exemplo.

+ Usaremos o Wireshark, que é um analisador de protocolos, para ver o funcionamento do WHOIS em etapas

+ Aplica-se o filtro: tcp or udp port 43

OBS: porta 43 é a do DNS

+ Vemos que a primeira etapa é a resolução do DNS

+ Depois ele realiza a comunicação após completar o 3WHS na porta 53

+ Realiza uma query (consulta)

→ Query: businesscorp.com.br\r\n

+ Temos então a resposta (Answer)

+ Por fim, encerra-se a comunicação (FIN, ACK)

~~~~~

+ Com os resultados obtidos podemos repetir o processo fazendo conexão via netcat

```
apt install ncat
nc -v -6 2001:12ff:0:2::3 43
```

→ esse endereço grande é o ipv6, instalamos o ncat pois ele suporta esse protocolo

+ Depois de conectar, fazemos a query: businesscorp.com.br

**FATO IMPORTANTE:** O protocolo whois não é o mesmo para todas as regiões. Por isso está sendo substituído pelo RDAP

## ***Criando um WHOIS em Python***

+ Vamos fazer a comunicação com o servidor e depois realizar uma consulta

```
nano whois.py
```

```
#!/usr/bin/python3

import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("whois.iana.org", 43))
request = "businesscorp.com.br"+"\r\n"
s.send(request.encode())
resposta = s.recv(1024)

print (resposta)
```

+ Para executar esse script, devemos utilizar o python2

```
python2 whois.py
```

+ Agora faremos a simulação do verdadeiro comando whois que primeiro faz a busca na iana e depois busca no refer apontado por ela

```
#!/usr/bin/python3

import socket, sys
```



```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("whois.iana.org", 43))
request = sys.argv[1]+"\r\n"
s.send(request.encode())
resposta = s.recv(1024).split()
whois = resposta[19]
s.close

s1 = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s1.connect((whois, 43))
request1 = sys.argv[1]+"\r\n"
s1.send(request1.encode())
resp = s1.recv(1024)
print (resp)
```

## Registration Data Access Protocol - RDAP

+ Foi pensado para solucionar as deficiências existentes no protocolo WHOIS tradicional. Problemas com a internacionalização, o standart das respostas, inclusão de caracteres especiais.

+ Em [registro.br/rdap/](https://registro.br/rdap/) podemos acessar algumas URL's de consulta

+ Para extrair as info de um domínio:

[rdap.registro.br/domain/businesscorp.com.br](https://rdap.registro.br/domain/businesscorp.com.br)

→ texto horizontal

[client.rdap.org](https://client.rdap.org)

→ texto mais organizado na página

## Mapeando a INFRA - Pesquisa por IP

+ Pesquisa pelo IP do site:

```
host businesscorp.com.br
```

```
(root@DESKTOP-NJHHNK6) - [/home/kali]
# host businesscorp.com.br
businesscorp.com.br has address 37.59.174.225
businesscorp.com.br mail is handled by 10 mail.businesscorp.com.br.
```

→ estamos buscando descobrir se o cliente tem um netblock ou um ASN  
→ em alguns casos, o cliente usa um proxy ou outra ferramenta para ocultar seu real endereço de IP

+ Podemos então fazer a pesquisa no site da ARIN (American Registry for Internet Numbers)

<https://search.arin.net/rdap>

+ De maneira análoga, podemos pesquisar por meio do terminal

```
whois 37.59.174.225 | egrep "inetnum|aut-num"
```

```
(root@DESKTOP-NJHHNK6) - [/home/kali]
# whois 37.59.174.225 | egrep "inetnum|aut-num"
inetnum:          37.59.174.224 - 37.59.174.239
```

→ não retorna um aut-num pois o cliente não tem um ASN

+ Para o caso de empresas maiores, como é o caso do Itaú, o endereço apresentado é o de outra empresa. No nosso caso, a Akamai:

```
(root@DESKTOP-NJHHNK6) - [/home/kali]
# host itau.com.br
itau.com.br has address 23.73.216.51
```

→ Ao consultar o whois desse endereço, obtemos:

```
NetRange:          23.72.0.0 - 23.79.255.255
CIDR:              23.72.0.0/13
NetName:           AKAMAI
NetHandle:         NET-23-72-0-0-1
Parent:            NET23 (NET-23-0-0-0-0)
NetType:           Direct Allocation
OriginAS:
Organization:      Akamai Technologies, Inc. (AKAMAI)
```

+ A solução para isso é então resolver os outros nomes que decorrem do host itau.com.br

## ***Border Gateway Protocol - BGP***

+ O BGP é onde os AS's se ligam e começam a interligar a internet

+ Há dois locais para consulta de IP's:

→ BGPview

→ bgp.he.net

Eles trazem info sobre quem é o ASN ⇒ de qual ASN o IP faz parte

+ Traz informações gráficas sobre a interligação dos AS's

+ A info na caixa DNS é de fato a resolução DNS de cada endereço apresentado

+ A busca também pode ser feita por nomes (nome da empresa)

+ A ideia é consultar IP's e blocos de rede para talvez extrair mais info no BGP

## ***Pesquisa no SHODAN***

## + Operadores - Shodan

| OPERADORES                           |                                |
|--------------------------------------|--------------------------------|
| hostname: busca no site especificado | geo: Busca por geolocalização  |
| os: busca por sistema operacional    | org: Busca por uma organização |
| port: busca por porta                | " " Busca por algum termo      |
| ip: busca por ip                     |                                |
| net: busca por rede                  |                                |
| country: busca por país              |                                |
| city: busca por cidade               |                                |

## + É o google dos hackers

### + O que podemos encontrar:

- Dispositivos conectados à internet como câmeras de segurança, roteadores, impressoras, servidores, dispositivos IoT, entre outros
- Info sobre redes e sistemas como endereços IP's, portas abertas, banners, serviços ativos e vulnerabilidades conhecidas
- Info sobre domínios e nomes de host como endereços IP associados, servidores de email, info WHOIS e DNS.
- Info sobre serviços e protocolos, como versões de software, vulnerabilidades conhecidas, configurações e senhas fracas.
- Info sobre sites e páginas da WEB como códigos-fonte, cabeçalhos HTTP, metadados e info sobre SSL

### + Exemplo de busca:

os:"windows xp" city:"London" port: "80"  
webcam country:br

## **Utilizando a API do Shodan**

### + Muito útil para realizar downloads dos resultados de pesquisas e fazer filtros via terminal

### + Os recursos mais avançados são para membros

### + Instalação do Shodan:

```
pip install shodan
```

### + Devemos copiar a API key do site logado na conta shodan e depois inserí-la:

```
shodan init -----key-----
```

### + Para ver quantas buscas conseguimos com determinados parâmetros, usamos o comando count

```
shodan count coutry:br port:445 contabilidade
```

### + Para mostrar os resultados, executamos:

```
shodan search --fields ip_str,org,port,hostnames country:br  
port:445 contabilidade
```

+ Podemos também fazer uma pesquisa por domínios ou IP's

```
shodan domain globo.com
```

```
shodan host 37.59.174.225
```

+ Para baixar os resultados da busca, devemos aplicar a api:

```
shodan download tanque port 10001 tanque country:br
```

→ tudo estará salvo no arquivo tanque.json.gz

+ Podemos usar o shodan para analisar o arquivo fazendo um parse

```
shodan parse --fields ip_str, port, org, hostnames --separator,  
tanque.json.gz
```

## ***Pesquisa no Censys***

+ Podemos criar uma conta e usar como o shodan

+ O que muda são os filtros

+ Exemplos de busca:

location.country\_code: BR

location.city: Salvador

location.country\_code: BR AND metadata.os: Windows

+ Traz detalhes como mapa, serviço, AS's, info da web, código html da index do site, whois também.

+ Também faz pesquisa por IP sem gerar log's na rede do cliente

+ Podemos fazer busca por um range:

ip:[37.59.174.224 TO 37.59.174.239]

→ também resolve os nomes

+ Combinando vários valores:

location.country\_code: BR AND metadata.os: Ubuntu AND 80.http.get.

title:"indexof"

ou AND ports:3306

## ***Pesquisa Domain Name System - DNS***

| REGISTROS |                                               |
|-----------|-----------------------------------------------|
| SOA       | START OF AUTHORITY (RESPONSÁVEL PELO DOMÍNIO) |
| A         | ENDEREÇO IPV4                                 |
| AAAA      | ENDEREÇO IPV6                                 |
| NS        | NAME SERVER (SERVIDORES DE NOMES)             |
| CNAME     | CANONICAL NAME (APELLIDO / ALIAS)             |
| MX        | MAIL EXCHANGE (SERVIDOR DE E-MAIL)            |
| PTR       | POINTER (MAPEIA IP PARA NOME)                 |
| HINFO     | HOST INFORMATION (INFORMAÇÕES DO HOST)        |
| TXT       | TEXT STRING (EXEMPLOS: SPF)                   |

+ Para realizar a pesquisa no terminal, executamos:

```
host -t A businesscorp.com.br
```

```
host -t mx businesscorp.com.br
```

```
host -t ns businesscorp.com.br
```

```
host -t txt businesscorp.com.br
```

## Entendendo a Transferência de Zona

+ Servidores primário e secundário:

```
host -t ns businesscorp.com.br
```

```
(root@DESKTOP-NJHHNK6) - [/home/kali]
# host -t ns businesscorp.com.br
businesscorp.com.br name server ns1.businesscorp.com.br.
businesscorp.com.br name server ns2.businesscorp.com.br.
```

ns1 é o primário e o ns2, o secundário

→ 2 é o mínimo de servidores de nome que precisamos ter para um domínio

→ O primário mantém registro com todas as entradas DNS (IP, subdomínios,...)

→ O secundário funciona como um backup ⇒ se o primário parar de funcionar, ele assume.

Para que isso funcione, ambos devem estar sincronizados (operando corretamente)

+ A transferência de zona funciona no protocolo DNS por meio da porta 53 TCP, enquanto a consulta DNS funciona pela UDP

## Script para Transferência de Zona

+ Vamos forçar a transferência de zona por meio do comando host -l

+ Veja que o ns1 não está autorizado a fazer a ZT,

mas o ns2 está

```
host -l businesscorp.com.br ns1.businesscorp.com.br
```

```
# host -l businesscorp.com.br ns1.businesscorp.com.br
Using domain server:
Name: ns1.businesscorp.com.br
Address: 37.59.174.225#53
Aliases:

Host businesscorp.com.br not found: 5(REFUSED)
; Transfer failed.
```

```
host -l businesscorp.com.br ns2.businesscorp.com.br
```

```
(root@DESKTOP-NJHHNK6) - [/home/kali]
# host -l businesscorp.com.br ns2.businesscorp.com.br
Using domain server:
Name: ns2.businesscorp.com.br
Address: 37.59.174.226#53
Aliases:

businesscorp.com.br name server ns1.businesscorp.com.br.
businesscorp.com.br name server ns2.businesscorp.com.br.
businesscorp.com.br has address 37.59.174.225
desafio.businesscorp.com.br has address 37.59.174.226
ftp.businesscorp.com.br has address 37.59.174.225
infrasecreta.businesscorp.com.br has address 37.59.174.225
intranet.businesscorp.com.br has address 37.59.174.228
mail.businesscorp.com.br has address 37.59.174.227
ns1.businesscorp.com.br has address 37.59.174.225
ns2.businesscorp.com.br has address 37.59.174.226
parsingok.businesscorp.com.br has address 37.59.174.225
piloto.businesscorp.com.br has address 37.59.174.230
rh.businesscorp.com.br has address 37.59.174.229
srvkey.businesscorp.com.br has address 37.59.174.235
www.businesscorp.com.br has address 37.59.174.225
```

+ Quando usamos a opção -a, a resposta virá mais completa

```
host -l -a businesscorp.com.br ns2.businesscorp.com.br
```

+ Montando o script: [dnszone.sh]

```
#!/bin/bash
for server in $(host -t ns $1 | cut -d " " -f 4);
```

```
do
host -l -a $1 $server
done
```

## ***Script para Pesquisa Direta (DNS)***

+ BRUTE FORCE DNS

+ Ao aplicar o comando host --domínio--, caso o endereço não exista, teremos um NXDOMAIN como retorno.

+ Sendo assim, usaremos uma wordlist [cat.txt ou br-wordlist.txt] para realizar o brute force

+ Criando o script: [brute\_dns.sh]

```
#!/bin/bash
for palavra in $(cat cat.txt); do
host $palavra$1 | grep -v "NXDOMAIN"
done
```

## ***Script para Pesquisa Reversa (DNS)***

+ Descobrimos o range de IP público da rede corporativa, poderemos fazer a varredura nos endereços

+ Construindo o script [dns\_rev.sh]

```
#!/bin/bash
for ip in $(seq 224 239); do
host -t ptr 37.59.174.$ip | grep -v "37-59-174" | cut -d " " -f 5
done
```

## ***Analizando SPF - Sender Policy Framework***

+ Visa identificar quais servidores estão autorizados a enviar emails em nome do seu domínio

Exemplos:

Sem registro SPF = Vulnerável a falsificação de e-mail (Mail Spoofing)

v=spf1 include:servidorpermitido.com ?all = suscetível (neutro)

v=spf1 include:servidorpermitido.com -all = suscetível ("é tratado como suspeito")

v=spf1 include:servidorpermitido.com -all = Configuração Recomendada

+ A má configuração do SPF torna o ambiente suscetível ao ataque de mail spoofing

+ Investigação via terminal:



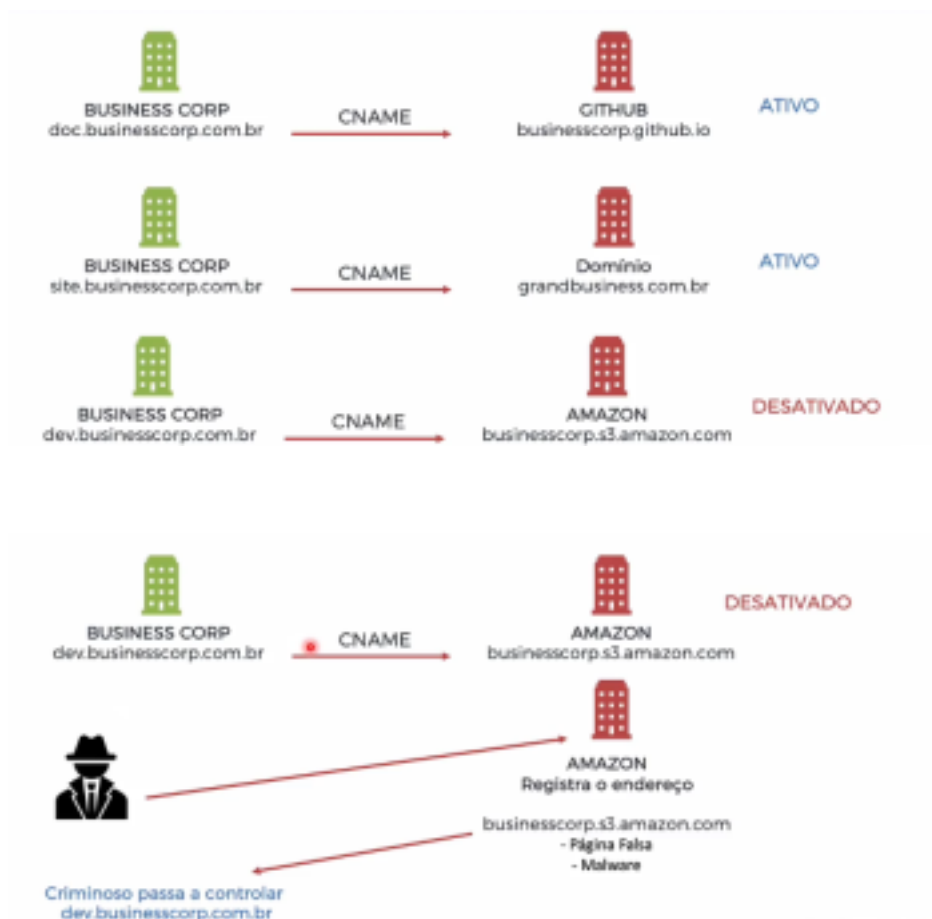
```
host -t txt businesscorp.com.br
```

+ Site que permite criar e enviar os e-mails falsos: [emkei.cz](https://emkei.cz)

## Entendendo o Subdomain Takeover

+ Subdomain takeover = Aquisição de Subdomínios

+ O objetivo é tomar o controle de algum subdomínio real da empresa



+ Como detectar a vulnerabilidade:  
Quando o endereço estiver disponível para registro, teremos uma condição de vulnerabilidade

Como detectar?

```
host -t cname doc.businesscorp.com.br
```

`doc.businesscorp.com.br` APONTA PARA `businesscorp.github.io` — Brute Force DNS

Endereço resolvido está ativo?



## Script para Subdomain Takeover

- + Fazemos a validação consultando os endereços na internet
- + Endereços não encontrados ou que deram erro são bons indícios de um vetor de ataque com subdomain takeover
- + O comando `host ---subdominio---` não retorna nada quando o subdomínio está inativo, mas o comando `host -t cname` mostra para onde ele apontava antes
- + Montando o script: [subtakeover.sh]

```
#!/bin/bash
for palavra in $(cat cat.txt); do
host -t cname $palavra$1 | grep "alias for"
done
```

- + Caso o link seja endereçado ao site da Amazon, podemos criar uma conta gratuita no site dela e na hora de registrar um domínio gratuito, fazemo-lo com exatamente o nome do domínio que achamos

## Tomando Controle de Subdomínios

- + Cenário: Com nosso bruteforce (DNS), descobrimos um subdomínio

Brute force DNS [CNAME] = dev.businesscorp.com.br

```
host -t cname dev.businesscorp.com.br
dev.businesscorp.com.br APONTA PARA devbusinesscorp.s3-sa-east-1.amazonaws.com
```

Endereço resolvido está ativo?



- + Validando o endereço com o terminal:

```
host -t cname dev.businesscorp.com.br
```

- + Quando acessamos o endereço na internet, obtemos que o bucket não existe.
- + Criamos uma conta gratuita na Amazon
- All services ⇒ Storage ⇒ s3 → seleciono ele pois é o serviço que eu quero criar o registro
- + O cliente usava um bucket e depois desativou. Vamos criar também um bucket com o mesmo nome que ele utilizava
- + Criando o bucket, podemos fazer o upload (por exemplo) de um arquivo html
- + Agora basta adicionar as configurações

properties → static website hosting → “use this bucket to host a website” -->  
index document arquivo.html (arquivo do upload)  
permission → edit → block all public access off  
properties → metadata → \*content type text/html  
permission → public access → everyone  
overview → make public

+ Agora o subdomínio da empresa está público, mas a empresa não controla mais ele.

## ***Outras Ferramentas para DNS Recon***

+ Temos o dig que funciona de maneira semelhante à do host

```
dig -t ns businesscorp.com.br +short
```

```
dig -t mx businesscorp.com.br +short
```

```
dig www.businesscorp.com.br +short
```

```
dig rh.businesscorp.com.br +short
```

```
dig -t axfr businesscorp.com.br @ns2.businesscorp.com.br
```

→ para realizar a transferência de zona, devemos acrescentar esse @

+ Outra ferramenta bem conhecida é o **dnsenum**

```
dnsenum --enum businesscorp.com.br
```

+ Há também o **dnsrecon**

```
dnsrecon -d businesscorp.com.br
```

+ **fierce**

```
fierce -dns businesscorp.com.br
```

+ Cada ferramenta dessas vem com uma wordlist

```
cd /usr/share/dnsenum
```

## ***Coleta Passiva através de Serviços Online***

+ Passiva pelo motivo de não interagirmos de maneira direta com o domínio (alvo)

[virustotal.com](https://www.virustotal.com) → podemos fazer o upload de arquivos (PDF's, por ex), que ele verifica a existência ou n de vírus. Além disso, faz varredura nos sites de maneira passiva (é ele, e não eu quem interage com o site).

[dnsdumpster.com](https://dnsdumpster.com) → faz a varredura do domínio trazendo resultados organizados e até mesmo gráficos.

[securitytrails.com](https://securitytrails.com) → faz a varredura um pouco mais precisa.

## Coleta através de Certificados Digitais



+ No site da desec, esse cadeado verde representa um certificado de segurança

+ [transparencyreport.google.com](https://transparencyreport.google.com)  
+ [crt.sh](https://crt.sh)

→ Fazem busca de arquivos públicos acerca de certificados digitais emitidos para cada subdomínio, sendo então uma forma de localizar outros subdomínios.

## LAB - SEM 05 - Info Gathering INFRA

LAB01: 37.59.174.224-37.59.174.239

host business... → usamos o endereço (37.59.174.225) para realizar uma busca no site da ARIN → whois/rdap

LAB02: AS16276

whois 37.59.174.225

LAB03: infrasecreta.businesscorp.com.br

forçamos o zone transfer com o script dnszone.sh

```
#!/bin/bash
for server in $(host -t ns $1 | cut -d " " -f 4);
do
host -l -a $1 $server
done
```

LAB04: 37.59.174.225

Já vinha resolvido com o script anterior

LAB05: rh.businesscorp.com.br,piloto.businesscorp.com.br

o seguinte script realiza o dns reverso a partir do range de IP's encontrado no começo (dns\_rev.sh)

```
#!/bin/bash
for ip in $(seq 224 239); do
host -t ptr 37.59.174.$ip | grep -v "37-59-174" | cut -d " " -f 5
done
```

**LAB06:** 37.59.174.229,37.59.174.230

já vinham resolvidos com o script anterior

**LAB07:** 0989201883299

para exibir informações acerca do host, usamos o seguinte comando

```
host -t hinfo businesscorp.com.br
```

**LAB08:** 9283947588214

para analisar o spf, bastou que executássemos o seguinte:

```
host -t txt businesscorp.com.br
```

**LAB09:** 092935999311009

Para a realização do subdomain takeover, usamos uma wordlist chamada cat.txt e o seguinte script: (subtakeover.sh)

```
#!/bin/bash
for palavra in $(cat cat.txt); do
host -t cname $palavra$1 | grep "alias for"
done
```

## ***Info Gathering - WEB***

### ***Introdução - Web Recon***

- + Identificar o WebServer
- + Identificar a tecnologia
- + Diretórios
- + Arquivos
- + Possíveis controles e bloqueios
- + Métodos http permitidos
- + Listagem de diretórios
- + Estrutura da página
- + Código fonte
- + Analisar arquivos encontrados

~~~~~  
Para isso, usaremos as seguintes técnicas

- | | |
|----------------------------|-------------------------------|
| + Parsing HTML/JS | + Burp Suite (Proxy) |
| + Mirror Website | + Whatsweb |
| + Robots.txt/Sitemap.xml | + Wappalyzer |
| + Análise de resostas HTTP | + Análise de extensões e Icon |
| + Bypass User-Agent | + Análise de erros e banners |

~~~~~  
Mapear a página

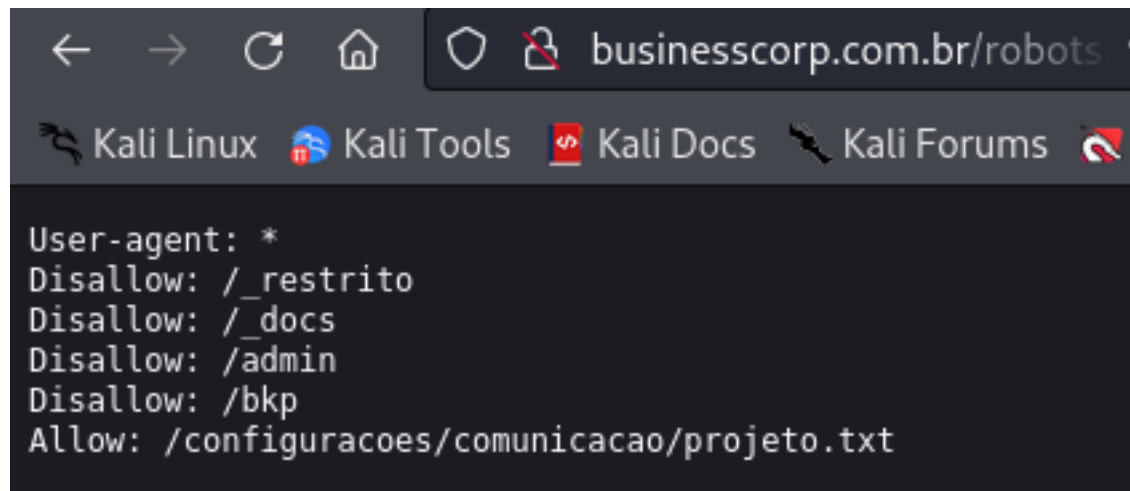
- |                                   |                   |
|-----------------------------------|-------------------|
| → Login?                          | → Upload?         |
| → Cadastro? É possível cadastrar? | → Download?       |
| → Busca?                          | → Erros?          |
| → Posts?                          | → Banco de dados? |

→ Redirecionamento?

→ Envio de dados?

## ***Robots e Sitemap***

- + O google tem vários robôs ou crawlers que varrem a internet e indexam as páginas que encontram e tem autorização para tal
- + O arquivo robots.txt tem o papel de controlar (negar) essa indexação

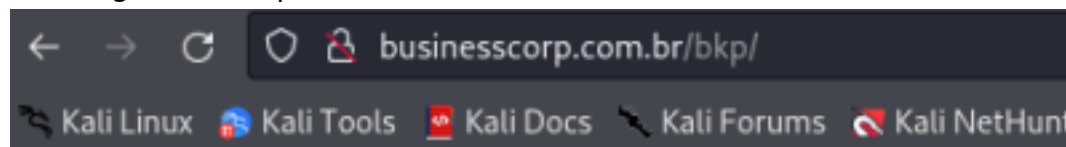


```
User-agent: *
Disallow: /_restrito
Disallow: /_docs
Disallow: /admin
Disallow: /bcp
Allow: /configuracoes/comunicacao/projeto.txt
```

- + O sitemap vai mapear todas as páginas do meu site, para que sejam indexadas de maneira mais rápida.

## ***Listagem de diretórios***

- + Listagem do /bcp:











## **Index of /bcp**

| <a href="#">Name</a>                                                                                               | <a href="#">Last modified</a> | <a href="#">Size</a> | <a href="#">Description</a> |
|--------------------------------------------------------------------------------------------------------------------|-------------------------------|----------------------|-----------------------------|
|  <a href="#">Parent Directory</a> |                               | -                    |                             |
|  <a href="#">script.sh</a>        | 27-Sep-2019 13:10             | 64                   |                             |

- + Listagem do /css:

## Index of /css

| <u>Name</u>                                                                                                        | <u>Last modified</u> | <u>Size</u> | <u>Description</u> |
|--------------------------------------------------------------------------------------------------------------------|----------------------|-------------|--------------------|
|  <a href="#">Parent Directory</a>  |                      | -           |                    |
|  <a href="#">default.css</a>       | 06-Feb-2015 01:47    | 19K         |                    |
|  <a href="#">font-awesome/</a>     | 06-Feb-2015 01:47    | -           |                    |
|  <a href="#">fontello/</a>         | 06-Feb-2015 01:47    | -           |                    |
|  <a href="#">fonts.css</a>         | 06-Feb-2015 01:47    | 6.1K        |                    |
|  <a href="#">fonts/</a>            | 06-Feb-2015 01:47    | -           |                    |
|  <a href="#">layout.css</a>        | 06-Feb-2015 01:47    | 10K         |                    |
|  <a href="#">media-queries.css</a> | 06-Feb-2015 01:47    | 7.7K        |                    |

+ A listagem de diretórios consiste em analisar diretórios que permitem navegação

## Mirrors Website

+ Consiste em baixar o website para a máquina local e então analisar o conteúdo.

+ Podemos também usar para clonar uma página existente e usar como vetor de ataque de engenharia social

+ Para realizar o download, aplicamos:

```
wget -m businesscorp.com.br
```

o -m é de mirror

+ Para que o arquivo robots seja excluído na captura, executamos:

```
wget -m -e robots=off rh.businesscorp.com.br
```

## Análise de erros, banners, extensões e código fonte

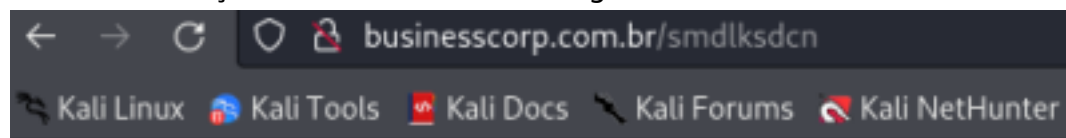
+ As vezes o desenvolvedor deixa algum comentario no código fonte

+ É um processo demorado, mas pode ajudar quando estivermos perdidos na investigação



+ Podemos achar algumas tags meta que dão informações acerca da versão da aplicação que está rodando

+ Quando fizermos uma requisição de um diretório inexistente, podemos obter informações coletadas da mensagem de erro



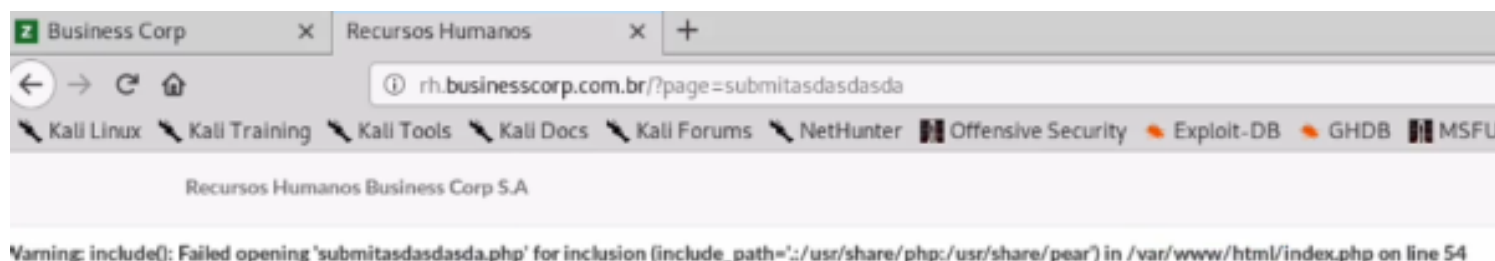
## Not Found

The requested URL /smdlksdcn was not found on this server.

*Apache/2.2.22 (Debian) Server at businesscorp.com.br Port 80*

→ isso informa que a aplicação está rodando em um servidor Linux, que é um Debian, e que está rodando um Apache na versão 2.2.22

+ Exemplo de patch exposed



→ é um problema, pois mostra o caminho no servidor onde está rodando a aplicação

## Pesquisa via Requisições HTTP

+ Faremos a captura do HEAD no business e no rh.business na porta 80

```
nc -v businesscorp.com.br 80
```

ou

```
nc -v rh.businesscorp.com.br 80
```

```
HEAD / HTTP/1.0
```

→ isso irá retornar informações acerca do servidor apache e do php

+ Para entender os métodos que o servidor suporta, podemos mandar uma requisição do tipo OPTIONS

```
OPTIONS /desec HTTP/1.0
```

```
root@pentest:~/Desktop# nc -v businesscorp.com.br 80
DNS fwd/rev mismatch: businesscorp.com.br != ip225.ip-37-59-174.eu
businesscorp.com.br [37.59.174.225] 80 (http) open
HEAD / HTTP/1.0
Host:businesscorp.com.br

HTTP/1.1 200 OK
Date: Thu, 26 Sep 2019 08:25:44 GMT
Server: Apache/2.2.22 (Debian)
Last-Modified: Wed, 25 Sep 2019 17:05:45 GMT
ETag: "20463-1bb6-59363a9ea0957"
Accept-Ranges: byte
Content-Length: 7094
Vary: Accept-Encoding
Connection: close
Content-Type: text/html
```

## ***Brute force - Arquivos e Diretórios***

+ Aqui faremos a busca de diretórios por meio do brute force usando o dirb

```
dirb http://businesscorp.com.br subdomains-10000.txt
```

```
(root@DESKTOP-NJHHNK6)-[/home/kali]  
# dirb http://businesscorp.com.br subdomains-10000.txt
```

DIRB v2.22  
By The Dark Raver

```
START_TIME: Fri Jan 19 17:03:55 2024  
URL_BASE: http://businesscorp.com.br/  
WORDLIST_FILES: subdomains-10000.txt
```

```
root@DESKTOP-NJHHNK6:~/Desktop# ./dirb http://rh.businesscorp.com.br php
```

```
_____ The 2.4.7  
Technologies: PHP 5.3.9-1ubuntu4.22
```

\_\_\_\_\_  
Arquivo por Diretórios e Arquivos

GENERATED WORDS: 9985

— Scanning URL: http://businesscorp.com.br/ —

quise encontrado: http://rh.businesscorp.com.br/robots.txt

quise php encontrado: http://rh.businesscorp.com.br/login.php

quise php encontrado: http://rh.businesscorp.com.br/home.php

⇒ DIRECTORY: http://businesscorp.com.br/admin/  
+ http://businesscorp.com.br/demo (CODE:200|SIZE:22657)

⇒ DIRECTORY: http://businesscorp.com.br/images/

⇒ DIRECTORY: http://businesscorp.com.br/intranet/  
+ http://businesscorp.com.br/info (CODE:200|SIZE:80)

⇒ DIRECTORY: http://businesscorp.com.br/db/

⇒ DIRECTORY: http://businesscorp.com.br/app/

⇒ DIRECTORY: http://businesscorp.com.br/css/

⇒ DIRECTORY: http://businesscorp.com.br/js/

## ***Estudando a Lógica do Programa***

+ O objetivo é fazer uma requisição na mão do tipo GET  
e analisar a response

+ O HTTP 200 indica que a página existe. Caso não existisse, o retorno  
seria 404

+ Os programa como o dirb mandam requisições http e analisam a resposta  
do servidor

```
root@pentest:~/Desktop# nc -v businesscorp.com.br 80
DNS fwd/rev mismatch: businesscorp.com.br != ip225.ip-37-59-174.eu
businesscorp.com.br [37.59.174.225] 80 (http) open
GET /app/ HTTP/1.0

HTTP/1.1 200 OK
Date: Thu, 26 Sep 2019 09:00:36 GMT
Server: Apache/2.2.22 (Debian)
X-Powered-By: PHP/5.4.45-0+deb7u2
Vary: Accept-Encoding
Content-Length: 185
Connection: close
Content-Type: text/html

<form method="POST">
  Username: <input name="username" type="text" /><br />
  Password: <input name="password" type="password" /><br />
  <input type="submit" value="Entrar" />
```

## Conhecendo o curl

+ O comando

```
curl businesscorp.com.br
```

retorna o conteúdo html da página

+ Para exibir mais detalhes, adicionamos o -v (verbose)

```
curl -v businesscorp.com.br
```

+ Para exibir apenas o head, podemos usar o --head

+ Existe também o modo silent do curl, que retorna uma resposta mais limpa com o uso do -s

+ Podemos mudar o user agent tbm, pra que não apareça mais o curl na requisição

```
curl -v -H "User-Agent: Mozilla"
```

+ Isso é legal pois podemos burlar sistemas de defesa que filtram ferramentas como dirb, nessus e curl, pois estaríamos sendo apresentado com outro nome que não seja o curl

+ Uma maneira de filtrar a saída do curl para que retorne apenas o código (200, 404, 408,...) é a seguinte:

```
curl -s -o /dev/null -w "%{http_code}" businesscorp.com.br
```

em que a "parte suja" da resposta será lançada em /dev/null

## Script pra Web Recon

+ Primeiramente, devemos ter uma wordlist  
-----vamos usar a subdomains-10000.txt-----

```
#!/bin/bash
for palavra in $(cat subdomains-10000.txt):
do
curl -s -H "User-Agent: MrCatTool" -o /dev/null -w "%{http_code}" $1/$palav>
if [ $(cat resposta.txt) != "404" ]
then
echo "Diretório encontrado: $palavra"
fi
rm resposta.txt
done
```

+ Esse é o webrecon.sh

+ Adicionamos a opção -H para que os filtros de user-agent não afetem o funcionamento do curl

## Whatweb

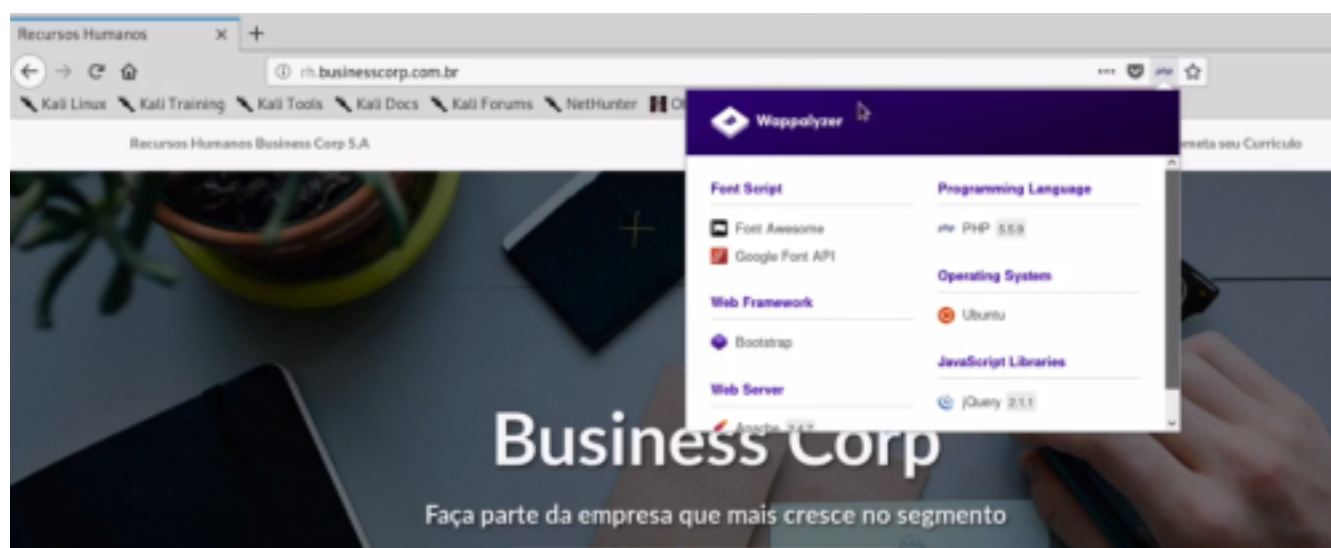
+ whatweb é uma ferramenta do kali que realiza um mapeamento rápido das tecnologias encontradas no site

```
root@pentest:~/Desktop# whatweb businesscorp.com.br
http://businesscorp.com.br [200 OK] Apache[2.2.22], Country[FRANCE][FR], Email[camila@businesscorp.com.br,rogerio@businesscorp.com.br,ti@businesscorp.com.br], Google-API[ajax/libs/jquery/1.10.2/jquery.min.js], HTML5, HTTPServer[Debian Linux][Apache/2.2.22 (Debian)], IP[37.59.174.225], JQuery[1.10.2], Modernizr, Script[text/javascript], Title[Business Corp]
```

## Wappalyzer

+ Podemos entrar no site wappalyzer.com e realizar seu download

+ Ele funciona como uma extensão do browser



+ Traz as informações do sistema ou frameworks utilizados

# Script para identificar páginas na internet

```
#!/bin/bash
lynx -dump "http://google.com/search?num=500&q=site:"$1"+ext:"$2""
| cut -d "=" -f2 | grep ".$2" | egrep -v "site|google"
```

## LAB - SEM 05 - Info Gathering WEB

**LAB01:** d81j237sh102k3a88njsnna12

```
dirb http://businesscorp.com.br /usr/share/dirb/wordlists/small.txt -a "Cavalo"
```

Seguimos a dica de mudar o nome do user-agent e aplicamos o bruteforce com a small.txt do dirb.

Esperamos bastante até que se chegasse na verificação do diretório db  
A key estava no arquivo encontrado em businesscorp.com.br/db/update

**LAB02:** 1nf0gh4t3r1ng89271882

Ao realizar a busca pelo /sitemap da business, encontramos um outro diretório indicado que é o /painelcliente

Nele, ao inspecionarmos o código fonte, obtemos a key

```
<br>
vlab | sitemap
<br>
<!--
Olhar o código fonte da página sempre é uma boa prática de recon! Key: 1nf0gh4t3r1ng89271882
<br>
-->
```

**LAB03:** g80889113568fkp9

Quando realizamos um bruteforce, encontramos um diretório chamado ~administrator

Lá teremos um arquivo.txt com a key desejada

**LAB04:** 65784920123nvw0f4

Aplicamos o mesmo método do LAB08, mas com extensão .txt  
o arquivo era info.txt, de onde obtivemos a key

**LAB05:** W3bR3nc0nisN3c3ss4ry10

Quando buscamos no google por site:businesscorp.com.br api,  
encontramos um diretório chamado apiCliente  
Basta analisar o arquivo.xml para encontrar a key

**LAB06:** bkmc5502874hdkiw91244hh

Ao realizar o bruteforce com o dirb, mudando o nome do user-agent,  
encontramos um diretório chamado /adminhelp  
Nele, um arquivo de texto que conterá a key

```
dirb http://rh.businesscorp.com.br /usr/share/dirb/wordlists/big.txt -a "Cavalo"
```

A opção -a serve para mudar o user-agent

**LAB07:** 00289jfhsyw72ll399s1

Executando o mesmo dirb do lab06, encontramos o arquivo /webdata, onde estará a key

**LAB08:** ed05a6d4d2fb2c6a35fe40c0e53386f2

Criamos um script para executar um brute force em arquivos php no site da rh.business

```
for palavra in $(cat small_php.txt)
do
    curl -s -o /dev/null -H "User-Agent: Cavalo" -w "%{http_code}"
    rh.businesscorp.com.br/$palavra
    echo "Encontrado: rh.businesscorp/$palavra"
done
```

Essa wordlist foi uma modificação de uma das wordlists padrão do curl, chamada small.txt. Adicionamos um .php ao final de cada palavra com o seguinte comando

```
sed 's/$/.php/' small.txt > small_php.txt
```

A wordlist pôde ser encontrada no diretório /usr/share/dirb/wordlists

O diretório que retornou 200 foi o /backup.php

**LAB09:** Apache/2.4.7

Obtivemos essa resposta analisando o banner apresentado ao executar uma requisição que dava erro

rh.businesscorp.com.br/jbjfksjdf

**LAB10:** PHP/5.5.9

Fizemos a busca no whatweb

<https://whatweb.net/>

```
http://rh.businesscorp.com.br [200 OK] Apache[2.4.7],
Bootstrap, Country[FRANCE][FR],
HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.7 (Ubuntu)],
IP[37.59.174.229],
jQuery, PHP[5.5.9-1ubuntu4.22],
Script, Title[Recursos Humanos],
X-Powered-By[PHP/5.5.9-1ubuntu4.22],
X-UA-Compatible[IE=edge]
```

## **LAB - SEM 05 - Info Gathering Web - VPN**

**LAB01:** Microsoft-IIS/7.5

Usamos a dica do Longatão das massas:



Quando você faz um http request para o servidor informando um arquivo de um determinado tipo (php, aspx, jsp) o servidor vai ver se compreende aquele tipo de solicitação.

Por exemplo, no caso de chegar um request para aspx ele vai verificar:

- Eu entendo aspx? não.. retorna erro default..

- Eu entendo aspx? sim.. responde com a tecnologia que ele tem.

E com isso acabamos descobrindo a tecnologia utilizada. (By Longatto)

Revisar o módulo "Information Gathering - WEB", aula "Conhecendo o Curl"

Então executamos um curl direcionado ao host 172.16.1.60

```
curl -v 172.16.1.60
```

A resposta trouxe o seguinte:

```
[root@DESKTOP-4311110 ~]# curl -v 172.16.1.60/arquivo.html
* Trying 172.16.1.60:80 ...
* Connected to 172.16.1.60 (172.16.1.60) port 80
> GET /arquivo.html HTTP/1.1
> Host: 172.16.1.60
> User-Agent: curl/8.4.0
> Accept: */*
>
< HTTP/1.1 404 Not Found
< Content-Type: text/html
< Server: Microsoft-IIS/7.5
< X-Powered-By: ASP.NET
< Date: Sun, 21 Jan 2024 00:50:46 GMT
< Content-Length: 1245
<
```

LAB02: [ASP.NET](#)

Respondido no lab passado

LAB03: [2.0.50727](#)

Para encontrar essa informação, continuamos mudando a extensão do arquivo aleatório solicitado.

Quando solicitamos aspx, a key chegou:

```
(root@DESKTOP-NJHNNK6)-[/home/kali]
# curl -v 172.16.1.60/arquivo.aspx
* Trying 172.16.1.60:80 ...
* Connected to 172.16.1.60 (172.16.1.60) port 80
> GET /arquivo.aspx HTTP/1.1
> Host: 172.16.1.60
> User-Agent: curl/8.4.0
> Accept: */*
>
< HTTP/1.1 404 Not Found
< Cache-Control: private
< Content-Type: text/html; charset=utf-8
< Server: Microsoft-IIS/7.5
< X-AspNet-Version: 2.0.50727
< X-Powered-By: ASP.NET
< Date: Sun, 21 Jan 2024 00:51:05 GMT
< Content-Length: 1507
<
```