****

**Visão**

Com a crescente demanda sobre Tecnologias, percebemos que muitas pessoas apesar de buscarem informações, não possuem fontes que queiram realmente passar o conhecimento da maneira como ela deve ser, livre e com embasamento técnico que permita ser aplicado e utilizado quando necessário, além de serem testados em sua criação, tornando esta informação útil e confiável.

**Missão**

O Laboratório foi criado com a intenção de buscar e disseminar o conhecimento de uma maneira clara e objetiva, de forma gratuita, auxiliando na evolução dos membros e da sociedade na qual estas informações são compartilhadas, buscando o crescimento de todos os envolvidos nesta criação de valores.



Caso você pense que com a leitura dos materiais da How2Security, você irá se tornar um Cracker capaz de invadir sistemas, se você espera encontrar aqui scripts infalíveis para invasão e, a partir deles, sair por aí invadindo computadores, essa não é a leitura indicada. Indicamos, sim a leitura do Código Penal (Lei 2.848/1940), principalmente a Lei Carolina Dickmann (Lei 12.737/2012), nos Artigos 154-A e 154-B.

*154-A Invadir dispositivo informático alheio, conectado ou não à rede de computadores, mediante violação indevida de mecanismo de segurança e com o fim de obter, adulterar ou destruir dados ou informações sem autorização expressa ou tácita do titular do dispositivo ou instalar vulnerabilidades para obter vantagem ilícita:*

*Pena – Detenção, de 3 meses a 1 ano, e multa*

Este material é um conjunto de informações compiladas de documentos e ferramentas do Mundo Underground testadas em ambiente de laboratório na nossa intranet. Desta forma, todo conhecimento aqui condensado é tangível, assim como as orientações das contramedidas.

Dessa forma, esperamos ter sido bem claros que, em momento algum, estamos com a pretensão de ensinar a você como se tornar um invasor. Estaremos sim, mostrando muitas das técnicas utilizadas pelos crackers e, em alguns casos, pelos scripts kiddies, para que você, como administrador de redes, seja capaz de identificá-las em tempo hábil para se defender, antes que alguém com desejos menos nobres ô faça por você.

Assim sendo, todo o conteúdo dessa literatura tem apenas o objetivo didático de informar e preparar os administradores de redes dos novos tempos. Em momento algum nos responsabilizamos pelo mau uso desse conhecimento ou por danos causados em seu equipamento ou de terceiros, assim como também não somos responsáveis pelos códigos e ferramentas aqui citados.

Sandro Melo

Adaptado por Wellington Silva aka Well

**0 – Links Relacionados ao Index do Web Site do Alvo**

Iremos criar um shellscript para retirar todos os links da página principal do web site do alvo, para levantarmos informações sobre outros possíveis serviços executando em outros servidores. Vamos utilizar dois portais para testarmos nosso shellscript. Eu escolhi o UOL Diveo do UOL e o MSN da Microsoft por serem um portais de conteúdo, e ser complexo de filtrar os dados.

[](http://192.168.5.128/lib/exe/detail.php?id=tec_kb00139&media=tec_kb00139-01.png)

Figura 01 – Portal de Conteúdo MSN da Microsoft

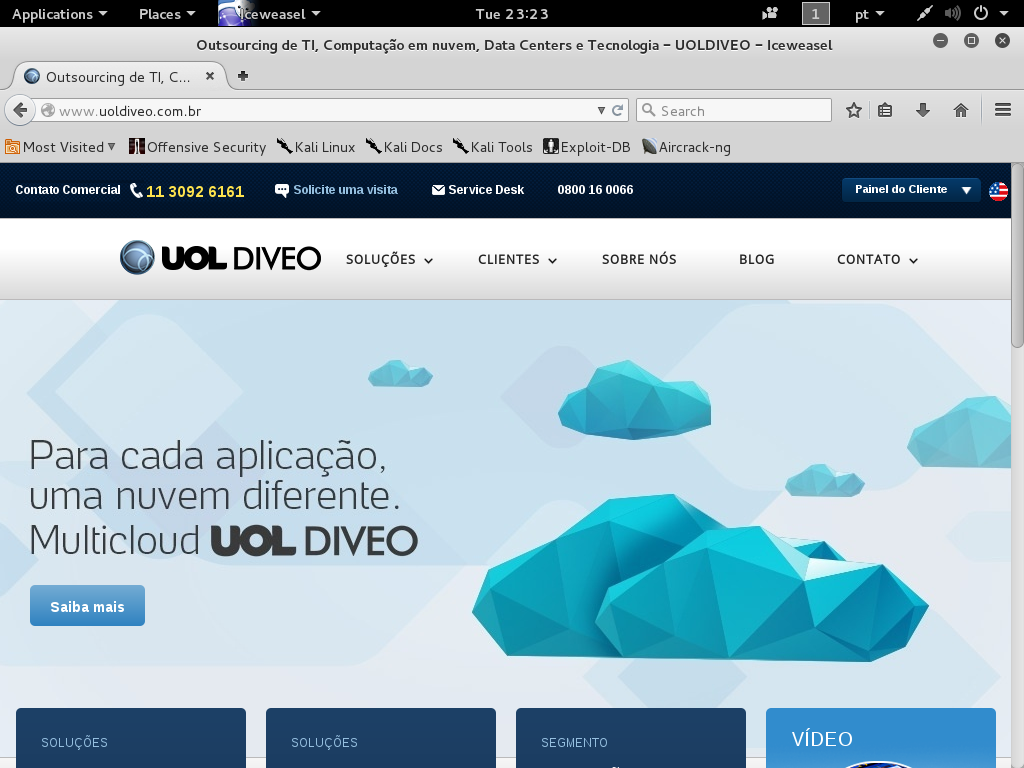


Figura 02 – Portal UOL Diveo

Vamos realizar isso de forma braçal, para sairmos da forma lammer para um nível jedi. Desta forma iremos entender o que estamos fazendo para adquirir estas informações.

**1 – Baixando o Index.html e Levantando os Requisitos**

Primeiramente vamos baixar o **index.html** com o utilitário ***wget*** para baixar.

root@kali-well:~scripts# wget http://msn.com

--2016-10-25 21:43:57-- http://msn.com/

Resolving msn.com (msn.com)... 23.101.196.141

Connecting to msn.com (msn.com)|23.101.196.141|:80... connected.

HTTP request sent, awaiting response... 301 Moved Permanently

Location: http://www.msn.com/ [following]

--2016-10-25 21:43:58-- http://www.msn.com/

Resolving www.msn.com (www.msn.com)... 204.79.197.203

Connecting to www.msn.com (www.msn.com)|204.79.197.203|:80... connected.

HTTP request sent, awaiting response... 302 Found

Location: http://www.msn.com/pt-br/ [following]

--2016-10-25 21:43:58-- http://www.msn.com/pt-br/

Reusing existing connection to www.msn.com:80.

HTTP request sent, awaiting response... 200 OK

Length: 59963 (59K) [text/html]

Saving to: ‘index.html’

index.html 100%[======================================>] 58.56K 151KB/s in 0.4s

2016-10-25 21:43:59 (151 KB/s) - ‘index.html’ saved [59963/59963]

root@kali-well:~scripts#

Agora usando um pouco de expressão regular vamos filtrar as informações que nos interessa com o utilitário ***grep***. Para procuramos as linhas que contenha links externos. Normalmente eles estão entre as tags ***<a href=”...”>link</a>***, e informam o protocolo usado, com isso, vamos pesquisar com base no **http** da seguinte forma:

root@kali-well:~scripts# grep "http" index.html

<!DOCTYPE HTML PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.2//EN" "http://www.openmobilealliance.org/tech/DTD/xhtml-mobile12.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt-BR" dir="ltr">

<link rel="canonical" href="http://www.msn.com/pt-br/" />

<link rel="alternate" hreflang="ar-ae" href="http://www.msn.com/ar-ae" /><link rel="alternate" hreflang="ar-eg" href="http://www.msn.com/ar-eg" /><link rel="alternate" hreflang="ar-sa" href="http://www.msn.com/ar-sa" />

--==[ Resumido ]==--

<img src="//c.bing.com/c.gif?Red3=MSNLI\_pd&amp;rid=f4e6d4ba-35f6-48cb-99ac-3b8f405c0300&amp;lng=pt-br&amp;dgk=downlevel.pc&amp;imd=0&amp;pn=startpage&amp;rf=&amp;tp=http%3A%2F%2Fwww.msn.com%2Fpt-br%2F&amp;ts=noscript" width="1" height="1" alt="" />

root@kali-well:~scripts#

Agora vamos utilizar o ***cut*** para eliminar aquilo que não é necessário, pois nossa meta é filtrar os links. O **cut** com as opções ***delimiter (-d)*** e a opção ***fields (-f)*** podemos colocar como delimitador o **/** que existem entre o protocolo e a URL, e solicitando o 3º campo da seguinte forma:

root@kali-well:~scripts# grep "http" index.html | cut -f3 -d"/"

**WAPFORUM**

www.w3.org

**www.msn.com**

**www.msn.com**

www.outlook.com

www.bing.com

**<input type="hidden" name="httpsmsn" value="1">**

support2.microsoft.com

www.msn.com

clk.tradedoubler.com

clk.tradedoubler.com

clk.tradedoubler.com

clk.tradedoubler.com

www.msn.com

www.microsoft.com

go.microsoft.com

go.microsoft.com

go.microsoft.com

jfe.qualtrics.com

go.microsoft.com

www.bing.com

advertising.microsoft.com

c.msn.com

c.bing.com

c.bing.com

root@kali-well:~scripts#

Temos duas questões com o conteúdo apresentado: o primeiro é que temos muitos links duplicados, e podemos resolver isso com o utilitário ***sort*** com a opção ***unique*** (**-u**); o segundo problema, é que veio lixos (tais como WAPFORUM e <input...), para procuramos as linhas que contenha links, normalmente eles estão entre as tags <a href=”...”>link</a>, com isso, vamos pesquisar com base no ***href*** da seguinte forma:

root@kali-well:~scripts# grep "href" index.html | grep "http" | cut -f3 -d"/" | sort -u

advertising.microsoft.com

clk.tradedoubler.com

go.microsoft.com

jfe.qualtrics.com

support2.microsoft.com

www.bing.com

www.microsoft.com

www.msn.com

www.outlook.com

root@kali-well:~scripts#

Excelente, conseguimos extrair todos os links externos referente a página principal do portal MSN.

Agora vamos fazer a mesma coisa com o portal UOL Diveo, comece baixando a página principal.

root@kali-well:~scripts# wget http://www.uoldiveo.com.br

--2016-10-26 00:02:10-- http://www.uoldiveo.com.br/

Resolving www.uoldiveo.com.br (www.uoldiveo.com.br)... 200.221.64.49, 200.147.68.20, 2804:49c:3103:405:ffff:ffff:ffff:3

Connecting to www.uoldiveo.com.br (www.uoldiveo.com.br)|200.221.64.49|:80... connected.

HTTP request sent, awaiting response... 200 OK

Length: 20015 (20K) [text/html]

Saving to: ‘index.html’

index.html 100%[======================================>] 19.55K --.-KB/s in 0.1s

2016-10-26 00:02:10 (177 KB/s) - ‘index.html’ saved [20015/20015]

root@kali-well:~scripts#

Agora vamos extrair as informações dos links como fizemos anteriormente.

root@kali-well:~scripts# grep "href" index.html | grep "http" | cut -f3 -d"/" | sort -u

> <meta property="og:title" content="Solu��es em Outsourcing de TI, Data Centers e Tecnologia - UOLDIVEO"

root@kali-well:~scripts#

Hummmmm! Acredito que deu alguma coisa errada. Pois nossa pesquisa deveria retornar as URLs como no exemplo anterior. Vamos pensar! Quantas linhas tem o arquivo ***index.html***? Vamos ver com o utilitário **wc** com a opção **lines** (**-l**).

root@kali-well:~scripts# wc -l index.html

0 index.html

root@kali-well:~scripts#

Não tem nenhuma linha???? Já sei, são os malditos códigos compactados, em que o Source Code fica em uma única linha. Vamos quebrar as linhas fazendo um laço **for** e direcionar para um arquivo temporário da seguinte forma:

root@kali-well:~scripts# for URLs in $(grep “http” index.html) ;do echo $URLs >> index.html > index.tmp; done

root@kali-well:~scripts# wc -l index.tmp

1484 index.tmp

root@kali-well:~scripts#

Maravilha, fizemos a quebra de linha, agora vamos executar o comando novamente para vermos quais são os links externos que contém na página principal do portal UOL Diveo.

root@kali-well:~scripts# grep "href=" index.tmp | grep "http" | cut -f3 -d"/" | sort -u

a> <

bpag.uol.com.br

fonts.googleapis.com

h.imguol.com

index.html">solicite uma visita<

jsuol.com

> <meta property="og:title" content="Solu��es em Outsourcing de TI, Data Centers e Tecnologia - UOLDIVEO"

painel.uoldiveo.com.br

servicos.html">Servi�os<

uoldiveo.com.br

uoldiveo.imguol.com

www.linkedin.com

www.uni5.com

www.uoldiveo.com.br

www.vagas.com.br

root@kali-well:~scripts# grep "href=" index.tmp | grep "http" | grep -v " " | cut -f3 -d"/" | sort -u

bpag.uol.com.br

fonts.googleapis.com

h.imguol.com

jsuol.com

painel.uoldiveo.com.br

uoldiveo.com.br

uoldiveo.imguol.com

www.linkedin.com

www.uni5.com

www.uoldiveo.com.br

www.vagas.com.br

root@kali-well:~scripts#

No primeiro resultado tivemos alguns lixos, que resolvemos na segunda pesquisa removendo os espaços em branco, coisas que não temos nas URLs. Para remover os espaços utilizamos a opção ***invert match (-v)*** do utilitário ***grep***.

Bem continuando nosso shellscript, vamos encontrar os endereços IP dos servidores que encontramos. Assim, além do FQDN teremos os endereços IP e para isso usaremos o utilitário ***host***.

Antes de continuarmos vamos enviar o resultado para outro arquivo temporário.

root@kali-well:~scripts# grep "href=" index.tmp | grep "http" | grep -v " " | cut -f3 -d"/" | sort -u > index.tmp2

root@kali-well:~scripts#

Agora vamos fazer outro laço for no arquivo e consultando um servidor DNS para resolver os nomes dos links em endereços IP.

root@kali-well:~scripts# for hostname in $(cat index.tmp2) ;do host $hostname; done

bpag.uol.com.br has address 200.221.137.226

fonts.googleapis.com is an alias for googleadapis.l.google.com.

googleadapis.l.google.com has address 216.58.202.106

googleadapis.l.google.com has IPv6 address 2800:3f0:4001:811::200a

h.imguol.com is an alias for cache.uol.com.br.

cache.uol.com.br has address 200.221.7.95

cache.uol.com.br has address 200.147.4.50

cache.uol.com.br has address 200.147.68.8

cache.uol.com.br has IPv6 address 2804:49c:3101:405:ffff:ffff:ffff:17

cache.uol.com.br has IPv6 address 2804:49c:3103:405:ffff:ffff:ffff:1

jsuol.com has address 200.147.15.241

jsuol.com has address 200.147.67.184

jsuol.com has address 200.221.2.70

jsuol.com has IPv6 address 2804:49c:3101:401:ffff:ffff:ffff:35

jsuol.com has IPv6 address 2804:49c:3103:401:ffff:ffff:ffff:2

painel.uoldiveo.com.br has address 200.147.112.193

uoldiveo.com.br has address 200.221.64.46

uoldiveo.com.br has address 200.147.68.9

uoldiveo.com.br has IPv6 address 2804:49c:3103:405:ffff:ffff:ffff:3

uoldiveo.com.br mail is handled by 10 mx3.uoldiveo.com.

uoldiveo.com.br mail is handled by 10 mx3.uolinc.com.

uoldiveo.imguol.com is an alias for cache.uol.com.br.

cache.uol.com.br has address 200.147.4.50

cache.uol.com.br has address 200.147.68.8

cache.uol.com.br has address 200.221.7.95

cache.uol.com.br has IPv6 address 2804:49c:3101:405:ffff:ffff:ffff:17

cache.uol.com.br has IPv6 address 2804:49c:3103:405:ffff:ffff:ffff:1

www.linkedin.com is an alias for pop-esp2-alpha.www.linkedin.com.

pop-esp2-alpha.www.linkedin.com has address 108.174.12.129

pop-esp2-alpha.www.linkedin.com has IPv6 address 2620:109:c00b:105::6cae:c81

www.uni5.com has address 200.147.63.68

www.uoldiveo.com.br is an alias for conteudo3.uol.com.br.

conteudo3.uol.com.br has address 200.147.68.20

conteudo3.uol.com.br has address 200.221.64.49

conteudo3.uol.com.br has IPv6 address 2804:49c:3103:405:ffff:ffff:ffff:3

www.vagas.com.br is an alias for www.vagas.com.br.cdn.cloudflare.net.

www.vagas.com.br.cdn.cloudflare.net has address 104.16.25.223

www.vagas.com.br.cdn.cloudflare.net has address 104.16.21.223

www.vagas.com.br.cdn.cloudflare.net has address 104.16.22.223

www.vagas.com.br.cdn.cloudflare.net has address 104.16.24.223

www.vagas.com.br.cdn.cloudflare.net has address 104.16.23.223

root@kali-well:~scripts#

Observe a saída que informa que o endereço do host uoldiveo.imguol.com é um alias para o host cache.uol.com.br e isso não é interessante, porém esta saída sim: cache.uol.com.br has address 200.147.4.50.

Vamos acrescentar um filtro com o **grep** utilizando a expressão “has address”.

root@kali-well:~scripts# for hostname in $(cat index.tmp2) ;do host $hostname | grep "has address"; done

bpag.uol.com.br has address 200.221.137.226

googleadapis.l.google.com has address 216.58.202.42

cache.uol.com.br has address 200.221.7.95

cache.uol.com.br has address 200.147.4.50

cache.uol.com.br has address 200.147.68.8

jsuol.com has address 200.147.15.241

jsuol.com has address 200.147.67.184

jsuol.com has address 200.221.2.70

painel.uoldiveo.com.br has address 200.147.112.193

uoldiveo.com.br has address 200.147.68.9

uoldiveo.com.br has address 200.221.64.46

cache.uol.com.br has address 200.221.7.95

cache.uol.com.br has address 200.147.4.50

cache.uol.com.br has address 200.147.68.8

pop-esp2-alpha.www.linkedin.com has address 108.174.12.129

www.uni5.com has address 200.147.63.68

conteudo3.uol.com.br has address 200.147.68.20

conteudo3.uol.com.br has address 200.221.64.49

www.vagas.com.br.cdn.cloudflare.net has address 104.16.25.223

www.vagas.com.br.cdn.cloudflare.net has address 104.16.22.223

www.vagas.com.br.cdn.cloudflare.net has address 104.16.21.223

www.vagas.com.br.cdn.cloudflare.net has address 104.16.23.223

www.vagas.com.br.cdn.cloudflare.net has address 104.16.24.223

root@kali-well:~scripts#

Muito bom temos todos os endereços relacionados com a página principal, vamos fazer mais um refinamento, a ideia é termos apenas os endereços IP para podemos automatizar outras ferramentas. Para isso, vamos delimitar esse resultado utilizando os espaços em branco e pegando o 4º campo.

root@kali-well:~scripts# for hostname in $(cat index.tmp2) ;do host $hostname | grep "has address" | cut -f4 -d" "; done

200.221.137.226

216.58.222.10

200.147.68.8

200.221.7.95

200.147.4.50

200.147.15.241

200.147.67.184

200.221.2.70

200.147.112.193

200.221.64.46

200.147.68.9

200.147.4.50

200.147.68.8

200.221.7.95

108.174.12.129

200.147.63.68

200.221.64.49

200.147.68.20

104.16.23.223

104.16.25.223

104.16.24.223

104.16.21.223

104.16.22.223

root@kali-well:~scripts#

Pronto conseguimos levantar os endereços IP referente aos domínios pesquisados, e podemos utilizar isso em nosso footprint.

**2 – Juntando Tudo em um Shellscript**

O reconhecimento ativo envolve uma sondagem direta a sistemas do alvo, com o objetivo de identificar hosts específicos, endereços IP ou blocos de endereços IP, serviços de rede, aplicações publicadas, entre outras informações.

root@kali-well:~scripts# vi finderweb.sh

#!/bin/bash

### CORES

amarelo="\e[33;1m"

azul="\e[34;1m"

verde="\e[32;1m"

vermelho="\e[31;1m"

fim="\e[m"

# BROWSER AGENT

AGENT="Mozilla/5.0 (Windows NT 10.0; WOW64)"

banner()

{

echo -e "$azul##############################################################$fim"

echo -e "$azul# By Wellington Luiz da Silva aka w3ll #$fim"

echo -e "$azul# #$fim"

echo -e "$azul# FINDER WEB Version 1.0 - How2Sec Lab #$fim"

echo -e "$azul# #$fim"

echo -e "$azul# Created: 20/10/2016 Updated: 20/10/2016 #$fim"

echo -e "$azul##############################################################$fim"

echo -e "\n"

}

banner

# Validando se o usuario passou a URL

if [ "$1" == "" ] ;then

echo -e "$vermelho""[-] ERROR: You need to specify a URL$fim"

echo -e "$verde""[+] Use: $0 http://www.exemplo.com.br\n$fim"

exit 1

fi

# Apagando pesquisas anteriores

rm index.ht\*

# Fazendo o download passando como parametro o User-Agent do Browser

# Isso para fazer bypass de WAF com filtros de Browsers

wget --user-agent="$AGENT" $1

# Laco FOR para fazer a quebra de linha em conteudos compactados

# Enviamos o resultado para um arquivo temporario com o nome PID.urls.1 em /tmp

for URLs in $(grep "http" index.html) ;do

echo $URLs >> /tmp/$$.urls.1

done

# Criando a pasta onde serao salvos os resultados

DIR=`echo "$1" | cut -f3 -d"/"`

mkdir -p $DIR

# Seprarando os Links Externos (FQDN)

grep -v ^$ /tmp/$$.urls.1 | grep "href=" | grep "http" | cut -d"/" -f3 | sort -u >> /tmp/$$.urls.2

echo "FQDN => Hosts Encontrados" >> $DIR/finderweb.aud

echo "=========================" >> $DIR/finderweb.aud

cat /tmp/$$.urls.2 >> $DIR/finderweb.aud

echo "N de Dominios encontrados: " `cat /tmp/$$.urls.2 | wc -l` >> $DIR/finderweb.aud

echo -e "\n" >> $DIR/finderweb.aud

# Correlacionando os Hosts com os Enderecos IP

for Hostname in $(cat /tmp/$$.urls.2) ;do

host $Hostname | grep "has address" >> /tmp/$$.urls.3

done

echo "HOST2IP => Correlacionando os Hosts com os Enderecos IP Encontrados" >> $DIR/finderweb.aud

echo "==============================================================" >> $DIR/finderweb.aud

cat /tmp/$$.urls.3 >> $DIR/finderweb.aud

echo -e "\n" >> $DIR/finderweb.aud

# Extraindo os Enderecos IP dos Hosts

cat /tmp/$$.urls.3 | cut -d" " -f4 | sort -u >> /tmp/$$.urls.4

echo "IP => Extraindo os Enderecos IP dos Hosts Encontrados" >> $DIR/finderweb.aud

echo "=====================================================" >> $DIR/finderweb.aud

cat /tmp/$$.urls.4 >> $DIR/finderweb.aud

# Mostrando os Resultados

cat $DIR/finderweb.aud

# Removendo os Arquivos Temporarios

rm /tmp/$$.urls\*

echo -e "\n"

echo -e "$verde""[+] Search results saved in: $DIR""$fim"

<ESQ>:wq

"finderweb.sh" 93L, 5651C written

root@kali-well:~scripts#

Com nosso shellscript vamos testar.

root@kali-well:~scripts# chmod 755 finderweb.sh

root@kali-well:~scripts# ./finderweb.sh

################################################################################

# By Wellington Luiz da Silva aka w3ll #

# #

# FINDER WEB Version 1.0 - How2Sec Lab #

# #

# Created: 20/10/2016 Updated: 20/10/2016 #

################################################################################

[-] ERROR: You need to specify a URL

[+] Use: ./finderweb.sh http://www.exemplo.com.br

root@kali-well:~scripts# ./finderweb.sh http://www.uoldiveo.com.br

################################################################################

# By Wellington Luiz da Silva aka w3ll #

# #

# FINDER WEB Version 1.0 - How2Sec Lab #

# #

# Created: 20/10/2016 Updated: 20/10/2016 #

################################################################################

--2016-10-26 02:08:28-- http://www.uoldiveo.com.br/

Resolving www.uoldiveo.com.br (www.uoldiveo.com.br)... 200.221.64.49, 200.147.68.20, 2804:49c:3103:405:ffff:ffff:ffff:3

Connecting to www.uoldiveo.com.br (www.uoldiveo.com.br)|200.221.64.49|:80... connected.

HTTP request sent, awaiting response... 200 OK

Length: 20018 (20K) [text/html]

Saving to: ‘index.html’

index.html 100%[==============================>] 19.55K --.-KB/s in 0.05s

2016-10-26 02:08:28 (418 KB/s) - ‘index.html’ saved [20018/20018]

FQDN => Hosts Encontrados

=========================

bpag.uol.com.br

fonts.googleapis.com

h.imguol.com

jsuol.com

painel.uoldiveo.com.br

uoldiveo.com.br

uoldiveo.imguol.com

www.linkedin.com

www.uni5.com

www.uoldiveo.com.br

www.vagas.com.br

N de Dominios encontrados: 11

HOST2IP => Correlacionando os Hosts com os Enderecos IP Encontrados

==============================================================

bpag.uol.com.br has address 200.221.137.226

googleadapis.l.google.com has address 216.58.202.42

cache.uol.com.br has address 200.147.68.8

cache.uol.com.br has address 200.221.7.95

cache.uol.com.br has address 200.147.4.50

jsuol.com has address 200.147.15.241

jsuol.com has address 200.147.67.184

jsuol.com has address 200.221.2.70

painel.uoldiveo.com.br has address 200.147.112.193

uoldiveo.com.br has address 200.221.64.46

uoldiveo.com.br has address 200.147.68.9

cache.uol.com.br has address 200.221.7.95

cache.uol.com.br has address 200.147.4.50

cache.uol.com.br has address 200.147.68.8

pop-esp2-alpha.www.linkedin.com has address 108.174.12.129

www.uni5.com has address 200.147.63.68

conteudo3.uol.com.br has address 200.147.68.20

conteudo3.uol.com.br has address 200.221.64.49

www.vagas.com.br.cdn.cloudflare.net has address 104.16.24.223

www.vagas.com.br.cdn.cloudflare.net has address 104.16.25.223

www.vagas.com.br.cdn.cloudflare.net has address 104.16.21.223

www.vagas.com.br.cdn.cloudflare.net has address 104.16.23.223

www.vagas.com.br.cdn.cloudflare.net has address 104.16.22.223

IP => Extraindo os Enderecos IP dos Hosts Encontrados

=====================================================

104.16.21.223

104.16.22.223

104.16.23.223

104.16.24.223

104.16.25.223

108.174.12.129

200.147.112.193

200.147.15.241

200.147.4.50

200.147.63.68

200.147.67.184

200.147.68.20

200.147.68.8

200.147.68.9

200.221.137.226

200.221.2.70

200.221.64.46

200.221.64.49

200.221.7.95

216.58.202.42

**[+] Search results saved in: www.uoldiveo.com.br**

root@kali-well:~scripts#

Primeiramente tornamos o script executável, e executamos sem passar como parâmetro uma URL, validando desta forma as informações passadas para o script. Logo em seguida executamos novamente passando a URL do portal e ele fez o processamento e entregou o resultado de forma correta.

**Nota:** Podemos baixar o script em https://github.com/how2security/finderweb.

**3 – Referências Bibliográficas**

**[1]** MUNIZ, JOSEPH and LAKHANI, AAMIR – Web Penetration Testing with Kali Linux, 2013, Birmingham – Mumbai, Packet Publishing

**[2]** GRAVES, KIMBERLY – CEH – Official Ethical Hacker Review Guide, Indianapolis - Indiana, 2007, Wiley Publishing Inc.

**[3]** STUTTARD, DAFYDD and PINTO, MARCUS – The Web Application Hacker’s Handbook, Indianapolis - Indiana, 2º Edition, Wiley Publishing Inc.

**[4]** SCAMBRAY, JOEL, McCLURE, STUART and KURTZ, GEORGE – Hackers Expostos Segredos e Soluções para a Segurança de Redes, São Paulo, 2001, MAKRON Books Ltda.

**[5]** LONG, JOHNNY - GOOGLE HACKING - Para Teste de Invasão [tradução Sérgio Pereira Couto] 2005, DIGERATI (BOOKS)

**[6]** TANENBAUM, Andrew S. – Redes de Computadores, 4º Ed, São Paulo, 2003, Editora Campus.

**[7]** MELO, Sandro – Estudo de Técnicas para Exploração de Vulnerabilidades em Redes TCP/IP, 2º Ed, Rio de Janeiro, 2006, Editora Alta Books Ltda.

**[8]** AHARONI, Mati; Offensive Security Lab Exercises - Curso de Pen-Test oferecido pela Offensive Security, 2007