

Caos Wireless: Capturando e Analisando Pacotes

Segurança da Informação

Charles Tim Batista Garrocho

Instituto Federal de São Paulo – IFSP
Campus Campos do Jordão

`garrocho.ifspcj.o.edu.br/SEGA6`

`charles.garrocho@ifsp.edu.br`

Curso Superior de TADS



INSTITUTO FEDERAL

O ICEMAN

Em 5 de setembro de 2007, o Serviço Secreto dos EUA prendeu um hacker sem fio chamado Max Ray Butler. Também conhecido como o Iceman, o Sr. Butler vendeu dezenas de milhares de contas de cartão de crédito através de um site.

Mas como ele coletou essa informação privada? Analisando conexões de internet sem fio não criptografadas provou ser um dos métodos que ele usou para obter acesso a informações de cartão de crédito.

O Iceman alugou quartos de hotel usando identidades falsas. Ele então usou antenas de alta potência para interceptar as comunicações para os pontos de acesso sem fio dos hotéis nas proximidades para capturar a informação pessoal dos hóspedes.

Os especialistas classificam esse tipo de ataque "sofisticado e complexo". Essa afirmação prova ser perigosa, pois podemos executar vários desses ataques em scripts Python curtos.



INSTITUTO FEDERAL

Exemplo de Exif em uma Foto de um Smartphone

```
tim@charles:~$ exiftool foto.jpg
ExifTool Version Number      : 10.10
File Name                    : foto.jpg
Directory                    : .
File Size                    : 1794 kB
File Modification Date/Time   : 2015:10:27 11:05:34-02:00
File Access Date/Time        : 2017:11:05 12:57:12-02:00
File Inode Change Date/Time   : 2017:11:05 12:57:06-02:00
File Permissions              : rwxr-----
File Type                    : JPEG
File Type Extension          : jpg
MIME Type                    : image/jpeg
Exif Byte Order               : Big-endian (Motorola, MM)
Make                         : Motorola
Camera Model Name             : XT1069
X Resolution                  : 72
Y Resolution                  : 72
Resolution Unit               : inches
Modify Date                   : 2015:10:11 12:54:49
Y Cb Cr Positioning           : Centered
Exposure Time                 : 1/60
```

Exemplo de Exif em uma Foto de um Smartphone

```
GPS Date Stamp           : 2015:10:11
Compression               : JPEG (old-style)
Thumbnail Offset         : 4086
Thumbnail Length         : 29254
Image Width              : 1836
Image Height             : 3264
Encoding Process         : Baseline DCT, Huffman coding
Bits Per Sample          : 8
Color Components         : 3
Y Cb Cr Sub Sampling     : YCbCr4:2:0 (2 2)
Aperture                 : 2.0
GPS Altitude             : 0 m Above Sea Level
GPS Date/Time            : 2015:10:11 15:54:48Z
GPS Latitude             : 21 deg 12' 32.07" S
GPS Longitude            : 43 deg 45' 3.75" W
GPS Position             : 21 deg 12' 32.07" S, 43 deg 45' 3.75" W
Image Size               : 1836x3264
Megapixels               : 6.0
Shutter Speed            : 1/60
Thumbnail Image          : (Binary data 29254 bytes, use -b option
)
Focal Length             : 3.5 mm
Light Value              : 7.9
tim@charles:~$
```

DERAL

Buscando as Informações de GPS da Foto

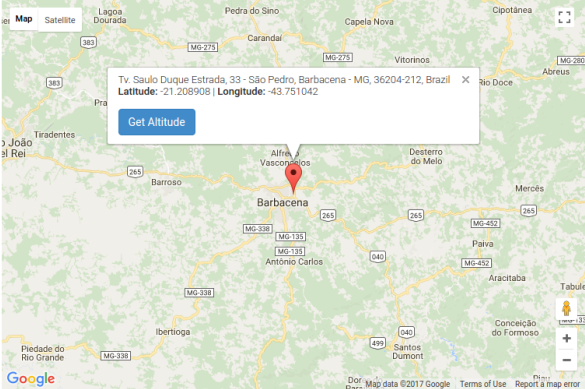
[Home](#) [Google Maps Directions](#) [Converter](#) [Street View](#) [API](#) [Geolocation](#) [Where am I](#) [Maps](#) [Custom](#)

Address
[Get GPS Coordinates](#)

DD (decimal degrees)*
Latitude
Longitude
[Get Address](#)

Lat,Long

DMS (degrees, minutes, seconds)*
Latitude ☐ N ☒ S ° ' "
Longitude ☐ E ☒ W ° ' "
[Get Address](#)





INSTITUTO FEDERAL

Fazendo o Download de Imagens com BeautifulSoup

Disponível em <http://www.crummy.com/software/BeautifulSoup/>, o BeautifulSoup nos permite analisar rapidamente documentos HTML e XML.

Leonard Richardson lançou a última versão do BeautifulSoup em 29 de maio de 2012.

Para atualizar a versão mais recente no Linux, use o seguinte comando para buscar e instalar a biblioteca beautifulsoup4:

```
tim@charles:~$ sudo apt-get install python-beautifulsoup
```

Esta biblioteca irá nos permitir baixar todas as imagens de uma determinada página na WEB.



INSTITUTO FEDERAL

Busca de coordenadas GPS em Imagens WEB

O script aceita como entrada a url da página que será baixado as imagens. A biblioteca **optparse** é utilizada para analisar a entrada.

```
52 def inicio():
53     analisador = optparse.OptionParser('use %prog "+\
54         "-u <url alvo>')
55     analisador.add_option('-u', dest='url', type='string',
56         help='especifique o endereco url')
57
58     (opcoes, args) = analisador.parse_args()
59     url = opcoes.url
60     if url == None:
61         print analisador.usage
62         exit(0)
63     else:
64         imgTags = buscarImagens(url)
65         for imgTag in imgTags:
66             nomeArqIMG = baixarImagem(imgTag)
67             if nomeArqIMG is not None:
68                 testeParaExif(nomeArqIMG)
```

Busca de coordenadas GPS em Imagens WEB

Para baixar o conteúdo da página WEB da url é utilizado a biblioteca **urllib2**.

```
12 def buscarImagens(url):
13     print '[+] Buscando imagens em ' + url
14     conteudoURL = urllib2.urlopen(url).read()
15     soup = BeautifulSoup(conteudoURL)
16     imgTags = soup.findAll('img')
17     return imgTags
```

A biblioteca **BeautifulSoup** então é aplicada na página baixada de forma a encontrar imagens.

A função retorna um vetor de **tags** de imagens encontradas.



INSTITUTO FEDERAL

Busca de coordenadas GPS em Imagens WEB

Aqui nesta função é tratado o download de cada imagem. Para isso é encontrado o endereço exato da imagem.

```
20 def baixarImagem(imgTag):
21     try:
22         fonteIMG = imgTag['src']
23         print '[+] Baixando imagem ', urlsplit(fonteIMG)[2]
24         conteudoIMG = urllib2.urlopen(fonteIMG).read()
25         nomeArqIMG = basename(urlsplit(fonteIMG)[2])
26         arqIMG = open(nomeArqIMG, 'wb')
27         arqIMG.write(conteudoIMG)
28         arqIMG.close()
29         return nomeArqIMG
30     except:
31         return None
```

A biblioteca **urllib2** novamente é utilizada, agora para baixar a imagem.



INSTITUTO FEDERAL

Busca de coordenadas GPS em Imagens WEB

Nesta função é buscado os dados Exif e verificado se existem as tags de GPS. Se existirem, então é imprimido na tela uma confirmação.

```
34 def testeParaExif(nomeArqIMG):
35     try:
36         dadosExif = {}
37         arqIMG = Image.open(nomeArqIMG)
38         info = arqIMG._getexif()
39         if info:
40             for (tag, valor) in info.items():
41                 decoded = TAGS.get(tag, tag)
42                 dadosExif[decoded] = valor
43             exifGPS = dadosExif['GPSInfo']
44             print dadosExif
45             if exifGPS:
46                 print '[' + nomeArqIMG + \
47                     ' contem GPS MetaData'
48     except:
49         pass
```

Buscando Imagens

Abaixo é apresentado um exemplo de execução deste script.

```
tim@charles:~$ python buscar_exif.py -u http://facebook.com
[+] Buscando imagens em http://facebook.com
[+] Baixando imagem /rsrc.php/v3/yc/r/GwFs3_KxNjS.png
[+] Baixando imagem /rsrc.php/v3/yb/r/GsNjNwuI-UM.gif
[+] Baixando imagem /rsrc.php/v3/yb/r/GsNjNwuI-UM.gif
[+] Baixando imagem /rsrc.php/v3/yb/r/GsNjNwuI-UM.gif
tim@charles:~$
```

Faça um teste com a seguinte url:

<http://www.depoisdosquinze.com/2017/03/30/dica-de-viagem-um-roteiro-especial-por-campos-do-jordao/>



INSTITUTO FEDERAL

Você deverá procurar na Internet um blog sobre guerra ou assuntos afins que não identificam os locais da imagem. O objetivo é identificar os locais onde as fotos foram tiradas.

Após isso desenvolva um script que irá receber a url do blog e irá fazer as seguintes ações:

- Baixar as imagens em uma pasta;
- Verificar se existe as coordenadas GPS;
- Guardar em uma pasta separada apenas as imagens com GPS;
- Apagar todas as imagens que não contém GPS.

Extra: Script receber url e buscar todas as demais urls do site e também realizar a investigação forense de todas as imagens deste site.

