Esteganografia utilizando o OutGuess

Juan Felipe Serafim dos Santos

[1]Segurança da Informação - 2025.1

jfss@cin.ufpe.br

Resumo. *Esteganografia é a técnica de ocultar informações de modo que a existência da mensagem passe despercebida. Seu nome vem do grego steganos (coberto) e graphia (escrita), que significa "escrita escondida". Em textos, a esteganografia pode usar espaços invisíveis, sinônimos cuidadosamente escolhidos ou padrões específicos na redação para ocultar mensagens. em imagens, a esteganografia utiliza dados que são inseridos nos bits menos significativos de pixels, de forma que o olho humano não percebe qualquer alteração. Técnicas semelhantes também existem para áudios e vídeos, onde informações são embutidas de maneira imperceptível, mas não se limita a somente embutir imagens.*

1. Objetivo

O propósito desta atividade é demonstrar o processo de esteganografia utilizando a ferramenta OutGuess para ocultar informações, ou aplicativos, mensagens, arquivos de áudio, em imagens do formato .jgp, .ppm ou .pnm. Isso será feito ao adicionar outra imagem de extensão .jpg na imagem (de também extensão .jpg) que será compartilhada livremente.

2. Como a ferramenta funciona

De acordo com a documentação do OutGuess para o processamento de imagens JPEG: "[...] *o OutGuess preserva estatísticas com base em contagens de frequência. Como resultado,* ***nenhum teste estatístico conhecido é capaz de detectar a presença de conteúdo esteganográfico****. Antes de incorporar dados em uma imagem, o sistema OutGuess pode determinar o tamanho máximo da mensagem que pode ser ocultada,* ***mantendo as estatísticas com base em contagens de frequência****.*

[...] *Uma semente pode ser usada para modificar o comportamento do iterador. Ela é incorporada aos dados junto com o restante da mensagem. Ao alterar a semente, o OutGuess tenta encontrar uma sequência de bits que minimize o número de alterações nos dados que precisam ser feitas.*"

De início, será utilizada a seguinte imagem para servir de “camuflagem” para a mensagem a ser escondida:

Figura 1. Imagem original, 594.516 bytes

A imagem a ser embutida na imagem supracitada:

Figura 2. Imagem a ser embutida, 5.447 bytes

A execução do OutGuess fornece os parâmetros a serem utilizados:

```
juanfelipe@juanfelipe-PC:~/Área de Trabalho$ outguess
OutGuess 0.4 Universal Stego 1999-2021 Niels Provos and others

outguess [options] [<input file> [<output file>]]
        -h              print this usage help text and exit
        -[sS] <n>       iteration start, capital letter for 2nd dataset
        -[iI] <n>       iteration limit
        -[kK] <key>     key
        -[dD] <name>    filename of dataset
        -[eE]           use error correcting encoding
        -p <param>      parameter passed to destination data handler
        -r              retrieve message from data
        -x <n>          number of key derivations to be tried
        -m              mark pixels that have been modified
        -t              collect statistic information
        -F[+-]          turns statistical steganalysis foiling on/off.
                        The default is on.
```

Figura 3. Tela principal o OutGuess com parâmetros de ajustes

Ao utilizar os parâmetros “-p 100 -d logo-ufpe2.jpg ufpoe.jpg saida_outguess.jpg” , temos o seguinte resultado:

```
juanfelipe@juanfelipe-PC:~/Área de Trabalho$ outguess -p 100 -d logo-ufpe2.jpg ufpoe.jpg saida_outguess.jpg
Reading ufpoe.jpg....
JPEG compression quality set to 100
Extracting usable bits:   1034448 bits
Correctable message size: 18446744073709551160 bits, 1783245188366336.00%
Encoded 'logo-ufpe2.jpg': 43576 bits, 5447 bytes
Finding best embedding...
    0: 21751(49.9%)[49.9%], bias 12062(0.55), saved:     4, total:  2.10%
    4: 21725(49.8%)[49.9%], bias 12045(0.55), saved:     7, total:  2.10%
    6: 21591(49.5%)[49.5%], bias 12118(0.56), saved:    24, total:  2.09%
    7: 21705(49.8%)[49.8%], bias 11991(0.55), saved:    10, total:  2.10%
    8: 21719(49.8%)[49.8%], bias 11812(0.54), saved:     8, total:  2.10%
   25: 21626(49.6%)[49.6%], bias 11898(0.55), saved:    20, total:  2.09%
   87: 21490(49.3%)[49.3%], bias 11904(0.55), saved:    37, total:  2.08%
87, 33394: Embedding data: 43576 in 1034448
Bits embedded: 43608, changed: 21490(49.3%)[49.3%], bias: 11904, tot: 1033903, skip: 990295
Foiling statistics: corrections: 8648, failed: 0, offset: 217.876167 +- 342.617163
Total bits changed: 33394 (change 21490 + bias 11904)
Storing bitmap into data...
Writing saida_outguess.jpg....
```

Figura 4. Saída da execução

Figura 5. Arquivo com a imagem embutida, 1.356.347 bytes

Então, ao executar novamente o OutGuess para extrair a imagem embutida, serão utilizado os parâmetros “-r saida_outguess.jpg reverso_outguess.jpg”:

```
juanfelipe@juanfelipe-PC:~/Área de Trabalho$ outguess -r saida_outguess.jpg reverso_outguess.jpg
Reading saida_outguess.jpg....
Extracting usable bits:   1034448 bits
Steg retrieve: seed: 87, len: 5447
```

Figura 6. Saída da execução

E o arquivo escondido, extraído da imagem adulterada:

Figura 7. Imagem extraída, 5.447 bytes

Para verificar se o arquivo escondido foi adulterado durante a extração, foi calculado o via algoritmo de hash, o SHA256, o valor correspondente ao seu conteúdo. Como ambos arquivos resultaram num mesmo hash, então a garantia da integridade da informação foi mantida.

```
juanfelipe@juanfelipe-PC:~/Área de Trabalho$ sha256sum reverso_outguess.jpg
a00ec7e11e44ca8a187549f3959ce3d4f928bcc3fcd780cb383467a3b8db5809  reverso_outguess.jpg
juanfelipe@juanfelipe-PC:~/Área de Trabalho$ sha256sum logo-ufpe2.jpg
a00ec7e11e44ca8a187549f3959ce3d4f928bcc3fcd780cb383467a3b8db5809  logo-ufpe2.jpg
```

Figura 8. Cálculo de hash de ambos os arquivos

3. Análise da vulnerabilidade

O que foi utilizado até então foi a manipulação de bits redundantes da imagem original para ser inserido quaisquer informações nesses ditos “espaços”. Essa é uma forma de mascarar, de maneira sutil, informações sigilosas.

4. Sugestão de defesa

Uma das formas de detecção de conteúdo embutido em arquivos seria a utilização do chamado *Content Disarm and Reconstruction* ou o Desarmamento e Reconstrução de Conteúdo (CDR). Em tempo de execução, esse software “desmonta” o arquivo a ser executado, removendo o conteúdo nocivo, e “monta-o” novamente.

Referências

Fernando Seabra Chirigati, Rafael Shinji Aoki Kikuchi, T. L. G. (2006). Esteganografia. https://www.gta.ufrj.br/grad/09_1/versao-final/stegano/index.html.

Lam, V. (2023). Criar e impedir a esteganografia em cinco minutos. https://portugese.opswat.com/blog/create-and-prevent-steganography-in-five-minutes.

Provos, N. (2021). Outguess. https://github.com/resurrecting-open-source-projects/outguess.

Provos, N. (2024). Kali linux - outguess. https://www.kali.org/tools/outguess/.