EgoCrypt - Ransomware

Citeforma | CET 8493 | UFCD 9195 Enquadramento operacional de cibersegurança | Formador: João Almeida

Hugo Miguel Félix Bugalho | João Rodrigo Mota da Costa

Introdução

Neste documento, iremos documentar o EgoCrypt, um ransomware desenvolvido seguindo as regras e indicações do formador João Almeida. Iremos mostrar a base fundamental de como o nosso ransomware funciona, as formas como este pode fazer danos a um sistema e as

Inspiração

Tirando inspiração de famosos ransomware como o WannaCry, decidimos adaptar uma personagem mais "palhaça", que goza com a vitima enquanto pede o "resgate", neste caso feito via DogeCoin (DOGE), uma cripto-moeda mais conhecida como uma "MemeCoin", em que não é suposto ter um grande valor no mercado, mas sim um valor mais cómico e sentimental.

Como funciona?

Para efeitos educativos, o nosso ransomware não é destrutivo, desde que a vitima tenha acesso à chave de encriptação ou não tenha escolhido uma pasta importante do sistema operativo.

O que a vitima vê

O nosso ransomware está disfarçado como um programa que a vitima poderia querer instalar no sistema operativo. Usando o `filedialog` do tkinter, podemos abrir um explorador de ficheiros em qualquer sistema operativo para o utilizador poder escolher onde quer instalar o "programa".

Após a vitima escolher a pasta de instalação, a janela fecha e os ficheiros são encriptados. Por norma, o explorador de ficheiros começa na pasta onde o script está guardado, mas numa futura versão do script, gostariamos de adaptar o tkinter para abrir numa pasta importante

dentro do sistema operativo, tal como a pasta `Program Files` no Windows, ou a pasta root `/` em sistemas GNU/Linux, visto que muitos utilizadores tendem a carregar nos botões com ações positivas, como "Next", "Seguinte, "Próximo", "Sim", entre outros, tornando o ransomware mais destrutivo.

Após a encriptação de todos os ficheiros, o utilizador é apresentado com uma caixa de texto que mostra a "nota de resgate", indicando à vitima que os ficheiros estão encriptados e que necessita de enviar dinheiro dentro de 24 horas para poder decriptar os ficheiros, como uma forma de instalar o pânico e uma sensação de urgência. Após carregar em "OK", o utilizador é apresentado com um conjunto de caracteres que parecem ser uma carteira de criptomoeda, mas é na realidade a chave de encriptação. Numa versão futura do script, pensamos em adicionar um ficheiro .txt a explicar o ransomware e outras maneiras que a vitima pode pagar, como uma espécie de "README" que muitos programas (incluindo ransomware) incluem para o utilizador poder ler a qualquer altura.

O que acontece na realidade

Após a vitima escolher a pasta de instalação, o script usa o `os.listdir` para listar todos os ficheiros e depois corre um loop usando o `glob` em que vai procurar a diretoria de forma recursiva para encontrar todos os ficheiros e fazer o processo de encriptação.

A pedido do formador João Almeida, a encriptação dos ficheiros em si foi feita com o Fernet, parte do pacote de criptografia do Python, disponivel via `pip`. O script cria uma chave dinâmica, que esta é guardada para uma variável "chave", que é usada para criar um ficheiro `chave.key` e para ser apresentada no final à vitima.

O programa lê o conteúdo de cada ficheiro listado, guarda o mesmo para uma variável `conteudo` e o Fernet usa a chave para fazer a enciptação. Após a encriptação o conteúdo é escrito para outro ficheiro com o nome original, mas com a extensão `.ego`. Neste caso decidimos manter o nome original para dar uma "esperança falsa" à vitima ao pensar que pode ter o ficheiro "de volta" ao renomear o ficheiro para o seu nome original, mas ao fazer isto vai deparar que o ficheiro ou não abre de todo ou tem o conteúdo encriptado.

No código temos também uma linha que apaga o ficheiro original no final deste ciclo de encriptação, mas para efeitos educativos este está comentado.

Script de decriptação?

Incluido no nosso trabalho é um script de decriptação, que usa a chave de encriptação do Fernet guardada no script para ajudar a fazer o processo da forma inversa. Este script foi mais um *proof of concept* (validação do conceito) do que algo necessário.

Fluxograma

De forma a evitar a repetição e monotonia de um fluxograma normal, decidimos fazer uma representação da arquitetura do nosso ransomware em ASCII.

```
************************************************************
*                          ENTRADA                         *
************************************************************
                    [Inicio do programa]

# Pergunta ao utilizador qual é a diretoria para encriptar

************************************************************
*                       PROCESSAMENTO                      *
************************************************************
# Lista todos os ficheiros dentro da diretoria
# Gera uma chave do Fernet
# Chave é gurdada em "chave.key"
# Loop para encontrar todos os ficheiros
    # Ler o conteúdo do ficheiro
    # Fernet encripta o conteúdo
    # Conteúdo encriptado guardado nun ficheiro ".ego"
    # Ficheiro original é apagado

************************************************************
*                          SAÍDA                           *
************************************************************
# Mostra uma caixa de texto com a nota de resgate
# Mostra outra caixa de texto com a chave de encriptação

                    [Fim do programa]
```