Universidade Federal do Rio de Janeiro
Instituto de Matemática
Departamento de Ciência da Computação
Grupo de Resposta a Incidentes de Segurança

Rio de Janeiro, RJ - Brasil

Ferramentas para proteção de malware
Analisando o contexto atual

GRIS-2013-A-001

Manoel Domingues Junior

A versão mais recente deste documento pode ser obtida na página oficial do GRIS: http://www.gris.dcc.ufrj.br.

GRIS - Grupo de Resposta a Incidentes de Segurança
Av. Brigadeiro Trompowski, s/n°
CCMN – Bloco F1 - Decania
Cidade Universitária - Rio de Janeiro/RJ
CEP: 21949-900
Telefone: +55 (21) 2598-9491

Última atualização em: 16 de setembro de 2013

Sumário

1 Proteções de código

Existem disponíveis no mercado diversas soluções criadas para proteger o código de programas, entre as mais comuns existem os encriptadores e os ofuscadores de código.

A encriptação de código consiste na inclusão de funções que alteram o fluxo da aplicação fazendo a mesma passar por um processo de descriptografia em tempo de execução. Isso significa que partes do software ficarão em memória sem criptografia de forma a ser possível recuperar essas partes com um software que leia a memória.

Além da encriptação, e também usada as vezes como sinônimo para o mesmo procedimento é a técnica de ofuscação de código. Usar a ofuscação de código em um programa significa transformar o código existente em um código com a mesma funcionalidade só que de difícil compreensão. Normalmente é possível aplicar a ofuscação de código tanto no código fonte como no código objeto ou até mesmo no código executável.

Podemos então pensar que a Eliminação de Informação Simbólica (ESI/EIS) é uma forma de ofuscação, mas isso não é verdade. Enquanto a EIS consiste em criptografar e modificar strings identificadoras, ou seja, trabalha no escopo dos dados, a ofuscação atua na alteração da estrutura do código.

As transformações utilizadas na ofuscação de código variam desde a simples insersão de código "inútil" para dificultar a análise do programa – até transformações que podem redefinir os fluxos de dados ou esconder chamadas de funções. Algumas soluções também inserem funcões críticas de checagem do código em execução.

A finalidade de encriptar ou ofuscar o código varia bastante dentro das aplicações sendo separada em dois grandes grupos: proteger o código de engenharia reversa ou evitar a detecção de códigos maliciosos por sistemas de defesa através da análise estática de código.

O primeiro grupo contêm diversas soluções comerciais que trabalham com ofuscação e encriptação de código. Algumas que podemos citar são: Proguard, Retroguard para códigos Java e a Morphit para código em C/C++, mas existem diversas outras disponíveis. Para o segundo grupo, parece existir um número um pouco maior de aplicações, muito embora, as aplicações presentes no primeiro grupo possam ser utilizadas no segundo.

No artigo, propomos analisar as ferramentas existentes no segundo grande grupo e comparar os resultados de cada uma. Para isso, criamos um ambiente virtual que é explicado em detalhes na seção de apêndice.

1.1 Ferramentas contempladas

É notória a quantidade de ferramentas existentes para determinado objetivo na internet. Com a disponibilização de ferramentas automatizadas e de novas interfaces de programação a quantidade de ferramentas produzidas e disponibilizadas na internet sofreu um aumento exponencial de forma a tornar praticamente impossível determinar todas as ferramentas que realizam determinada atividade.

Para poder determinar quais ferramentas fariam parte do escopo do artigo, buscamos pelas ferramentas mais relevantes, usando como critério o número de vezes em que cada uma é mencionada na internet.[1]

Portanto, abaixo se encontra as ferramentas que serão contempladas na nossa análise:

- EXECryptor
- StarForce Crypto
- Private EXE Protector
- Themida
- Abronsius Code Obfuscator
- Indetectable Simple Crypt

[1] Usamos o números de resultados médio dos buscadores da web (Google, Bing e Yahoo)

- Win Troyanizer Porjoiner
- CigiCigi BCS Kriptomatik
- ShadeHacK Crypter
- Jodedor 5x1
- Masa crypter
- TYV Crypter
- Cactus Metamorph
- Billar Crypter
- Eye Crypter
- K! Cryptor
- Small Crypter
- Open Crypter
- Acid Burns Crypter

1.2 Testes contemplados

Com a finalidade de analisar os resultados obtidos na utilização de cada ferramenta, descrevemos abaixo os testes realizados no resultado obtido em cada uma.

1.3 Confiabilidade da ferramenta

Nesse teste consideramos o executável da ferramenta como um artefato para análise. Isso ocorre, pois diversas ferramentas foram construídas por um único desenvolvedor ou por um grupo deles, os quais revelam em sua maioria a utilização da ferramenta para fins maliciosos.

Dessa forma, analisaremos inicialmente a ferramenta quanto a sua detecção em ferramentas anti-malware. Após essa análise, submeteremos a ferramenta a uma análise comportamental para mapear possíveis conexões suspeitas, modificação de arquivos do sistema ou modificações no registro do sistema.

1.4 Detecção dos resultados

Com a finalidade de verificar a eficiência da ferramenta, submeteremos alguns softwares com rotinas maliciosas detectados por todas as ferramentas anti-malware existentes[2] ao processamento da ferramenta e analisaremos a resposta e sua taxa de detecção.

Com isso, analisaremos o quanto a ferramenta é capaz de dificultar a detecção de um software previamente conhecido sem modificações.

1.5 Padrões em resultados

Diversas ferramentas inserem marcas em seus resultados com fins de divulgar a origem de tal resultado ou simplesmente promover o uso da ferramenta.

Para obter os resultados desse teste, serão executadas verificações de padrões em alguns resultados da ferramenta, mostrando se é possível encontrar padrões nos resultados ou não.

[2]através do sistema VirusTotal.com

2 Testes

2.1 EXECryptor

EXECryptor é um sistema de segurança para proteção de programas, evitando a engenharia reversa, análise estática e modificações não autorizadas. Em sua documentação ele afirma usar uma tecnologia de segurança nova e original que fornece aos desenvolvedores de software um nível sem precedentes de proteção para aumentar significativamente suas receitas.

Suas principais características são: anti-depuração, anti-trace, mecanismo matematicamente comprovado, compactação de código, proteção de vários tipos de arquivo (EXE, DLL, ActiveX) e compatibilidade com diversas linguagens de programação.

http://www.strongbit.com/execryptor.asp

2.2 StarForce Crypto

StarForce Crypto protege as áreas de código executável de valor intelectual e comercial. A ferramenta oferece uma proteção confiável que torna muito difícil analisar o código do software. StarForce Crypto é uma ferramenta de proteção baseada em criptografia que transforma o código executável em instruções de máquina StarForce virtuais tornando a análise e modificação do código do software consideravelmente mais difícil.

http://www.star-force.com/solutions/product/starforce-crypto/

2.3 Private EXE Protector

O Private EXE Protector (PEP) é uma ferramenta que polimorfismo de 32 bits para ofuscar aplicações para Microsoft Windows. Ela suporta diversas linguagens de programação, entre elas ASM, C++, VC++, Delphi, C, Python, entre outras.

Possui recursos anti-falsificação e sistema de análise de software. O PEP trabalha com métodos tradicionais, como a compactação de arquivos, criptografia de fragmento de código, carregamento metamórfico, proteção de depuração e manipulação de arquivos, e apresenta novas técnicas incluindo a proteção de dados com a técnica de recursos roubado e execução de código parcial em uma máquina virtual.

http://www.setisoft.com/

2.4 Themida

Themida é uma ferramenta para proteção de software que utiliza técnicas de criptografia e ofuscação de código em uma tecnologia denominada pelo seu fornecedor de SecureEngine.

De forma resumida, o código fonte da aplicação é encapsulado em diferentes partes e criptografado individualmente, para somente no final ser reunido e novamente criptografado de forma global.

A ferramenta também conta com características anti-depuração, anti dump de memória e utilização de algoritmos de criptografia diferentes em diferentes partes do código.

http://www.oreans.com/themida.php

2.5 Abronsius Code Obfuscator

ACO é uma ferramenta criada para programadores de Visual Basic 6 com o objetivo de ofuscar o código fonte de softwares escritos nessa linguagem para tornar sua detecção mais difícil em softwares anti-vírus.

O software também inclui outras funcionalidades, como um gerador de stubs únicos, análise de detecções e um compilador.

Infelizmente não foi possível encontrar informações sobre a origem do software pois o fórum ao qual ele é referenciado foi fechado.[3]

[3]URL do fórum: http://www.hackhound.org/forum/index.php?topic=42713.0. O programador responsável pela criação da ferramenta é conhecido como Abronsius.

2.6 Indetectable Simple Crypt

O Indetectable Simple Crypt é uma ferramenta que implementa a encriptação usando algoritmos RC4 e possui como outras funcionalidades a possibilidade de inserir mensagens ao abrir o arquivo e um gerador de chaves criptográficas interno.[4]

2.7 Win Troyanizer Porjoiner

O Win Troyanizer Porjoiner[5] é uma ferramenta escrita com o objetivo de ser usado com trojans.

Suas principais características incluem sua compatibilidade com o compactador UPX, modificar o tipo de arquivo executável, modificar as informações sobre o arquivo, habilitar a desativação do firewall do Windows e até inserir uma assinatura no arquivo final.

2.8 CigiCigi BCS Kriptomatik

Essa ferramenta tem a principal função de realizar a criptografia e compactação do executável para fins maliciosos.

Suas principais características são o suporte a mulpiplos formatos de compactação, utilizar criptografia RC4, XOR entre outras e difinir mecanismos de propagação do software, como via USB, contatos de email, entre outros.

2.9 ShadeHacK Crypter

O ShadeHacK Crypter é uma ferramenta de encriptação com suporte a 3 algoritmos que possui características como: mover o cabeçalho de arquivos PE, mudar as chaves de encriptação dos arquivos, entre outras.

2.10 Jodedor 5x1

A ferramenta Jodedor 5x1[6] mais conhecida como 5x1 combina uma ferramenta com 5 utilidades: encriptador de código, juntador de arquivos, downloader, empacotador e escritor de final de arquivo.

Ela possui funcionalidades bem completas, como migrar o processo onde o código será executado ou estipular um tempo de espera para a execução do código.

2.11 Masa Crypter

Masa Crypter[7] é uma ferramenta de encriptação com opções de realinhar o arquivo PE e mudar o OEP.

2.12 TYV Crypter

TYV Crypter[8] é uma ferramenta de criptografia com suporte a dois métodos de criptografia: o stubs e o CryptApiy RC4 podendo realizar a mudança de chave de criptografia. A ferramenta também possui um anti-debbuging.

2.13 Cactus Metamorph

O Cactus Metamorph[9] é uma ferramenta repleta de funcionalidades que permite entre vários recursos: mapear as áreas do código que foram modificadas, gerar logs das mudanças realizadas nos arquivos e até inserir partes de outro executável no executável em questão para aumentar o nível de ofuscação.

Sua última versão disponível é a 0.3.

[4] O programador responsável pela criação da ferramenta é conhecido como sm0Kes.
[5] O programador responsável pela criação da ferramenta é conhecido como Hax991.
[6] O programador responsável pela criação da ferramenta é conhecido como m3m0_11.
[7] O programador responsável pela criação da ferramenta é conhecido como Masangel.
[8] O programador responsável pela criação da ferramenta é conhecido como m3m0_11.
[9] O programador responsável pela criação da ferramenta é conhecido como MadAntrax.

2.14 Billar Crypter

O Billar Crypter[10] é uma ferramenta para criptografia de código.

2.15 Eye Crypter

O Eye Cripter[11] é uma ferramenta de criptografia com funções específicas para realizar a evasão de alguns sistemas de anti-vírus, como o Anti-malware Kapersky e o Avira Antivir.

2.16 K!cryptor

o K!cryptor é uma ferramenta para criptografia de código com funções exclusivas para realizar a evasão de sistemas anti-malware como: Kapersky, NOD32, AVG, Avast, McAfee, Panda e Norton.

2.17 Small Crypter

O Small Crypter é uma ferramenta de criptografia de código que apresenta recursos de injeção de código.

2.18 Open Crypter

O Open Crypter é uma ferramenta de criptografia.

2.19 Acid Burns Crypter

O Acid Burns Crypter é uma ferramenta de criptografia.

3 Matriz de resultados

4 Conclusões

5 Referências

http://www.sawp.com.br/blog/?p=201
https://sites.google.com/site/1millondevirus/encriptadores
http://troyanosyvirus.com.ar
http://foro.elhacker.net/analisis_y_diseno_de_malware/recopilatorio_herramientas_manuales_y_codigos_de_los_usua
t255646.0.html

[10] O programador responsável pela criação da ferramenta é conhecido como 4n0nym0us.
[11] O programador responsável pela criação da ferramenta é conhecido como Fakundo.