Aula 11

Engenharia Reversa e Análise de Malware

Ronaldo Pinheiro de Lima
crimescibernéticos.com@gmail.com

Aula 11

7. Trabalhando com DLLs

7.1. Estrutura Básica de uma DLL
7.2. Enumerando Exports
7.3. Executando DLLs
7.4. Restrições de Processo Host
7.5. Debugging DLL
7.6. Component Object Model (COM)
7.7. Lab 07-01

DLLs

- Dynamic Link Libraries
- compartilhamento de código entre aplicações
- arquivo PE
- não executa sozinha
- exporta funções para outras aplicações
- memória compartilhada entre processos
- não precisa distribuir DLLs do SO com o EXE
- execução dentro de um processo (processo host)
- atrativo p/ malware, ocultar ações, dificuldade de análise
- ter acesso a memória do processo

Estrutura Básica de uma DLL

- internamente são iguais aos EXEs
- IMAGE_FILE_HEADER – Characteristics – flag is DLL
- possuem mais Exports que os EXEs
- DllEntryPoint não é um Export
- DllEntryPoint é chamado automaticamente

Exports

- podem ter nomes significativos ou não
- enumerar com CFF Explorer e IDA Pro

Executando DLLs

- requer um processo host
- processo host genérico do Windows: rundll32.exe
- linha de comando:

C:\>rundll32.exe <dllpath>,<export> [argumentos opcionais]

O que o rundll32.exe faz:

1. Chama GetCommandLineW para verificar os parâmetros
2. Carrega a DLL com LoadLibraryW, executa o **DllEntryPoint**
3. Obtém endereços do Export com GetProcAddress
4. Chama a função Export, fornecendos argumentos opcionais

Restrições de Processo Host

- algumas DLLs apenas se executam em processos específicos

```
v7 = strlwr(szModName);
if ( strstr(v7, "explorer.exe") )
{
  CreateEventA(0, 0, 0, "prx673912690");
  v8 = decodestr(off_10025054, off_10025030, &unk_1003787C);
  lstrcpyA(byte_1003A950, v8);
  hLib = LoadLibraryA("kernel32");
  *CopyFileA = GetProcAddress(hLib, "CopyFileA");
  CreateThread(0, 0, ProxyThread, 0, 0, 0);
  if ( CheckOnFile() == 2 )
    SetTimer(0, 0, 0xC8u, StealPOSCookies);
  return SetTLS2();
}
v10 = strlwr(szModName);
if ( !strstr(v10, "iexplore.exe") )
{
  v11 = strlwr(szModName);
  if ( !strstr(v11, "regedit.exe") )
  {
    v12 = strlwr(szModName);
    if ( !strstr(v12, "regedt32.exe") )
    {
      v13 = strlwr(szModName);
      if ( !strstr(v13, "firefox.exe") )
        return Cleanup(v3, hinstDLL);
    }
  }
}
```

Evitando Restrições de Processo Host

- renomear o rundll32.exe para o nome desejado
- injetar a DLL no projeto desejado
- RemoteDLL (http://securityxploded.com/remotedll.php)
- use o Python! Livro Gray Hat Python e muito mais na net!
- após injeção analisar o comportamento com as ferramentas

Debugging DLL

- LOADDLL.EXE (proc host do OllyDbg e Imm Debugger)
- funcionamento parecido do rundll32.exe
- debugger coloca breakpoint automático no DllEntryPoint

Component Object Model (COM)

- comunicação entre aplicações
- utilizar códigos (interfaces) de outros programas
- não executa o processo do outro programa
- determinar se o malware utiliza COM
- difícil de analisar
- COM client – COM server
- Microsoft fornece muitos objetos COM para serem utilizados
- **Saber se o malware utiliza COM, presença do Imports**:

  OleInitialize ou CoInitialize

COM Server Malware

- COM server malicioso para outras aplicações utilizarem (DLL)
- BHO (Browser Helper Objects) – plug-ins para o IE
- Não possuem restrições
- Permite executar e manipular código dentro do IE
- Monitorar tráfego, rastrear uso, se comunicar com a Internet
- *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Browser Helper Objects*

- **Fáceis de detectar, exportam inúmeras funções:**

  DllCanUnloadNow

  DllGetClassObject

  DllInstall

  DllRegisterServer e DllUnregisterServer

Lab 07-01 Análise de DLL maliciosa

Material necessário:

- VM Windows XP 32-bit
- RemoteDLL
- PEiD, Exeinfo PE, RDG
- FakeNet, Wireshark
- Sysinternals Suite
- IDA Pro Free
- Arquivo: **Lab-07-01.rar**

Obrigado!

A explicação detalhada de todos os tópicos está na apostila.

Ronaldo Pinheiro de Lima
crimescibernetico.com@gmail.com
http://www.crimescibernetico.com
@crimescibernet