別紙.TIPS-3 実習資料

仙台 CTF 2019 セキュリティ技術勉強会 実習資料

TIPS-3 マルウェア解析

2019年9月28日 仙台 CTF 推進プロジェクト

目次

| 本実習の概要 | | 1 |
|----------------|-------------|----|
| 実習1(練習問題) ダウンロ | ードファイルの特定 | 2 |
| 実習1の解説 | | 3 |
| 実習2(練習問題)動作条件 | -の特定 | 9 |
| 実習2の解説 | | 10 |
| 実習3(練習問題) 起動され | る外部プログラムの特定 | 13 |
| 実習3の解説 | | 14 |
| 実習4 情報流出したファイル | ∠の特定 | 15 |
| 実習4の解説 | | 16 |

本実習の概要

あなたは、架空の企業「株式会社仙台シーテーエフ」に入社したばかりの新米情報セキュリティ 担当者です。

ある日、DMZ の開発用サーバが不正アクセスを受け、第三者への RDP ブルートフォース攻撃の踏み台として悪用されてしまいました。

開発用サーバを調査したところ、RDP ブルートフォース攻撃ツール「ncrack.exe」および不審プロ グラム「malware.exe」が実行されていたことが判明しました。

開発用サーバには、本番データ(顧客情報)が格納されており、情報流出が懸念されます。 不審プログラム「malware.exe」を解析し、情報流出の有無を確認してください。

| 不審プログラム (解析対象) | C:¥Users¥Administrator¥AppData¥Local¥Temp¥tools¥malware.exe |
|-------------------|---|
| 本番データ (顧客情報) | C:¥work¥sendaictf.csv |
| その他 | 不審プログラムと同じフォルダには「ncrack.exe」および関連ファイルも 設置されている。 |

◆開発用サーバのファイル

◆インシデントの状況図



[補足情報]

・ インシデント発生日は 2019 年 8 月 25 日(日)です。

実習1(練習問題) ダウンロードファイルの特定

実習内容

練習用不審プログラム「re01_urldownload.exe」を解析し、以下2点を特定してください。

① 不審プログラムがダウンロードしたファイル名(保存先のフルパス)

② 上記①のダウンロード元 URL

[実習用データ] 実習用仮想マシンに格納してあります。

フォルダ : /var/samba/public/tips3/ ファイル : re01_urldownload.exe

(補足) 実害の無いプログラムですが、ウイルス対策ソフトで検知される可能性があります。

回答記入欄

不審プログラムがダウンロードしたファイル名(保存先のフルパス)
 上記①のダウンロード元 URL

実習1の解説

解析ツール「Ghidra」で不審プログラムを解析します。

- 1. 実習用仮想マシン「Caine」を起動します。
- 2. 「Main Menu」-「Reverse Engineering」から「Ghidra」をクリックし起動します。 「Tip of the Day」ダイアログは「Close」をクリックして閉じてください。

| | 0 | Chidra: NO ACTIVE PROJECT 法 Elle Edit Breject Iools Help 输动输动输动输入。 | _=> |
|--|-----------|---|-----|
| | 4 | Tool Chest |) |
| Deverse Engineering | CM Chidra | Active Project: NO ACTIVE PROJECT | |
| Windows Forensics Tools Forensics Tools アクセサリ グラフィックス インターネット オフィス オフィス その他 ブログラミング サウンドとビデオ システムツール コニンパーサル・アクセス | | Ip of the Day Image: Construction of the Day Image: Did you know Did you know You can add and remove table columns as desired by night-clicking on a table header. Image: Did you know Imag | 2 |
| ■ Places ■ System ■ Lock Screen M Loo Out caine | omputer | Running Tools: INACTIVE | • |

Ghidra のメニュー「File」-「New Project」をクリックします。
 「Non-Shared Project」が選択されていることを確認し「Next」をクリックします。
 「Project Name」に任意のプロジェクト名を入力し「Finish」をクリックします。

| Chidra: NO ACTIVE PROJECT | | |
|---|---------------------------------------|---|
| Chi+O | | |
| Delete Project | New Project | × |
| Configure Install Extensions Import File I Satch Import. Open File System | Gelect Project Type | |
| Exit Ghidra Ctrl+Q | | New Project |
| Filter: | Kon-Shared Project Shared Project | |
| Tree View Table View | | Project Directory: //horne/caine |
| | | |
| | < <pre>Sack [Next>>]</pre> | 任意のプロジェクト名を入力 (例:sendaictf2019) |
| | | • |
| | | < <back next="">> Einish Cancel</back> |

4. Ghidra のメニュー「File」-「Import File」をクリックします。

「Select File to Import」ダイアログで「My Computer」をクリックし、ディレクトリ「/var/samba/ public/tips3」-「re01_urldownload.exe」を選択し「Select File To Import」をクリックします。

| File Edit Project Tools | Help | |
|--|-----------------------|---|
| New Project Open Project Reopen | Ctrl+N Ctrl+O | |
| Close Project Save Project | Ctrl+W Ctrl+S | |
| Delete Project | | |
| Archive Current Project Restore Project | | Select File to Import |
| Configure | | 🗇 📫 🔐 /var/samba/public/tips3 😵 🚇 📃 |
| Install Extensions | | B all ulderslord ava |
| Import File Batch myore Open File System Exit Chidra Filter: Tree View Table V Running Tools | I Ctrl+I Ctrl+Q | My Computer Bestop Home Recent |
| Creating project: /home/ | caine/sendaictf2019 | File name: Type: All Files (") 3 Select File To Import] Cancel |

5. 解析対象ファイルの確認ダイアログが2回表示されますが、そのまま「OK」をクリックします。

| 41 v | | | | |
|---|--------------------------------------|------|--|---|
| tive Project: sendaictf2019 | | Impo | t Results Summary | |
| sendaict#2016 Import /var/sam Format Language: Destination Folder: Program Name: Tree View Table inning Tools | ba/public/tips3/reo1_urldownload.exe | 1 | Research Johnson Project File Norman Readorly Program Name Compiles ID: Processor Address Size Minimum Address and Size Address Size Size Size Address Size Size Address Size Address | ref01 statistic ref01 statistic ratio ref01 ratio ref02 ratio ref02 |

6. プロジェクトウィンドウに解析対象ファイルが追加されました。 解析対象ファイルをダブルクリックし「Code Browser」を開きます。

| | Ghidra: sendaict[2019 | × |
|----------------|------------------------------|---|
| 2 |] the fact Tools Telb | |
| 7 | ool Chest | |
| | 🧖 🔮 | |
| 1 | ctive Project: sendaictf2019 | _ |
| | V log sendaictf2019 | |
| | 📄 re01_urldownload.exe | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | Filter. | |
| a final second | Tree View Table View | |
| F | unning Tools | |
| ſ | Workspace | |
| L | (Transpace | 1 |
| | | |
| - | | |

7. 「Analyze」ダイアログが表示されたら「Yes」をクリックします。

「Analysis Options」ダイアログは、そのまま「Yes」をクリックします。 自動解析処理が終了し「Auto Analysis Summary」ダイアログが表示されたら「OK」をクリック します。(PDB に関する警告が表示されますが問題ありません。)



B. Ghidra の画面が表示されます。
 (他の実習でも、同様の手順で解析対象ファイルをインポートします。)



9. 下図手順で main 関数を表示します。

(他の実習でも、同様の手順で main 関数を表示します。)



10. main 関数(FUN_00401000)の「Listing」と「Decompile」から、 Windows API「URLDownloadToFileA」の引数を確認します。

[関数の説明]

URLDownLoadToFileA(0, ダウンロードする URL, ファイル保存先のフルパス, 0, 0)

| Program Trees X | Listing: re01_urldownload.exe | 🗅 🐘 🔽 🐺 🛃 👪 📑 • 🗙 | 😋 Decompile: FUN_0040 🍪 🐚 🛃 😸 🔫 🗙 |
|--|---|---|--|
| 🗔 🗁 🔁 | *re01_uridownload.exe 🗙 | | |
| Force of Luidownice Headers Headers text text text force market Symbol Tree force market Symbol Tree force market Classes Filter | 00401000 93 00401001 85 cc 00401003 86 00 c0 40 00401003 86 200 00 00401004 83 c4 04 00401010 65 00 00401010 65 00 00401010 65 00 00401010 65 00 00401010 65 00 00401020 c8 05 00 00 00401022 cc 00401022 cc 00401020 cc 00401022 cc 00401020 cc 004000000000000000000000000000000000 | Corp. Corp. Y EBP.ESP H s_01.What_URL_do_l_connect_to?_02_0040 L _printf O ESP.0x4 H 0x0 H 0x0 H s_0.01 H 0x0 H s_0.040 H 0x0 H s_0.040 H s_0.040 | 3 incluince of coords foot(sd) 4 [5 |
| | I DINKNOWN Stack | nvala param . | |
| Data Types | Console - Scripting | | a // > |
| re01_urldownload.e | Successfully compiled: WindowsResourceRefer | ence.java | |
| windows_vs12_32 | 50 Gat | | |
| | | | |

- 11. 上記により、不審プログラムは「http://abc.example.com/list」からダウンロードしたファイルを 「C:¥work¥list.txt」に保存することが分かります。
- なお、「Code Browser」ウィンドウを閉じると、プロジェクトウィンドウに戻ります。
 実習2以降は、プロジェクトウィンドウで手順4~手順8の操作を行い、解析対象ファイルをインポートしてください。

回答例

| 不審プログラムがダウンロードしたファイル名(保存先のフルパス) C:¥work¥list.txt | |
|--|--|
| 2 上記①のダウンロード元 URL http://abc.example.com/list | |

(空白ページ)

実習2(練習問題) 動作条件の特定

実習内容

練習用不審プログラム「re02_getlocaltime.exe」を解析し、不審プログラムが動作する条件を 特定してください。

[実習用データ] 実習用仮想マシンに格納してあります。

フォルダ : /var/samba/public/tips3/ ファイル : re02_getlocaltime.exe

(補足) 実害の無いプログラムですが、ウイルス対策ソフトで検知される可能性があります。

回答記入欄

不審プログラムが動作する条件

実習2の解説

解析ツール「Ghidra」で不審プログラムを解析します。

- 1. 実習用仮想マシン「Caine」を起動します。
- Ghidra」を起動しプロジェクトウィンドウで「/var/samba/public/tips3/re02_getlocaltime.exe」 をインポートし「Code Browser」を開きます。(詳細は、実習1の手順4~手順8を参照。)
- 3. 下図手順で main 関数を表示します。



main 関数の中で「GetLocalTime」関数を使用し現在時刻を取得しています。
 その後、if 文で現在時刻を確認し、「年(西暦)」が「0x7e3」でない場合、または現在の「月」が「1」でない場合はプログラムを終了する処理となっています。

| Program trees | re02_petiocaltime.exe | | = ■ ♥ ♀ ≤ ₩ ■ | | |
|--|--|---|---|--------------------------|--|
| re02_getlocaltim Headers text data | | / ram: 00401000- | -004098d3 | 234567 | undefined4 FUN_00401000(void) (_SYSTEMTIME local_14; |
| Program Tree × | undefined4 undefined2 undefined2 | ndefined4stdca EAX:4 Stack[-0x12]:2 Stack[-0x14]:2 | ill FUN_00401000(void) <return> local_12 local_14</return> | 8 9 10 11 12 | GetLocal imm(LP3Y3 LEMTIME)&local_14); if (local_14.wYear != 0x7e3) (local_14.wMonth ! 1)) [_exit(0);] _pinf(c_im_working!_0040c01c); |
| Finite Provide the second secon | FU D0401000 55 00401001 8b ec 00401003 83 ec 10 00401006 68 00 c0 40 00 70 02 000 | PUSH MOV SUB PUSH | XREFI EBP EBP ESP ESP 0x10 s_Q1.When_did_Lget_active?_0040c000 | 14 15 | retum o, |
| Filter: | 00401006 88 80 300 00 00401010 83 24 04 00401013 84 45 f0 00401017 ff 15 00 a0 | ADD LEA PUSH CALL | _pmm ESP,0x4 EAX=>local_14,[EBP + -0x10] EAX dword ptr [->KERNEL32,DLL::GetLocalTime] | | |
| (= · ⇒ · * · \ \ | 0040101d 0f b7 4d f0 00401021 81 F9 = 3 07 | MOVZX CMP | ECX,word ptr [EBP + local_14] | 1 | |
| ▼ A Data Types ► BuiltInTypes ► Øre02_getlocat | Console - Scripting | | | | in (1) |

5. 「0x7e3」にマウスカーソルを合わせるとポップアップウィンドウが表示され、10 進数の「2019」 であることが分かります。

| Program Trees X | Listing: re02_getlocaltime.exe | L) 📦 🔖 🛱 | 🖌 💩 📑 - 🗙 | C Decompile: FUN_004 | on 🥸 🐚 📝 🖓 👻 🛪 |
|--|---|---|-----------|---|---|
| Comparison of the second | Tro02_get(scattime ave X 00401000 55 00401000 58 ac 00401006 68 00 c0 40 00401018 50 00401018 50 a0 00401014 60 a0 00401012 61 ac 30 00401021 61 ac 30 00401027 75 09 00401029 61 75 57 | PUSH MOV EBP ESP,0x10 SUB ESP,0x10 PUSH s_01.When.did_lget_active? CALL printf ADD ESP,0x4 LEA EAX=shceal_14,IEBP + -0x10 PUSH EAX GALL dword pr[->KERNEL32.DL::C MOVZX Hex MOVZX word 7E3h Vexed 7E3h | _0040c000 | undefined4 FUN_00 J SYSTEMTIME loca | 401000(void) L14; did_let_active"_0040c000); ToxTall[08local_14); L0x7e3] (local_14.wMonth1= gl_0040c01c); |
| Filter. | 00401022 (83 fa 01 00401030 74 07 00401032 6a 00 00401032 6a 00 00401034 68 1c c0 40 | CMP JZ LAB_00401039 200401032 PUSH 0x0 CALL _exit 3_00401039 PTISHTm_workingl_0040c01c | XREF[| | |
| ► BuiltInTypes | 🖳 Console - Scripting | | | | 🛅 🖉 🗙 |
| windows_vs12 | | | | | |

回答例

不審プログラムが動作する条件 現在時刻が 2019 年 1 月の場合にのみ動作する。 (空白ページ)

実習3(練習問題) 起動される外部プログラムの特定

実習内容

練習用不審プログラム「re03_winexec.exe」を解析し、不審プログラムから起動される外部 プログラムを特定してください。

[実習用データ] 実習用仮想マシンに格納してあります。

フォルダ : /var/samba/public/tips3/ ファイル : re03_winexec.exe

(補足) 実害の無いプログラムですが、ウイルス対策ソフトで検知される可能性があります。

回答記入欄

不審プログラムから起動される外部プログラム名(フルパス)

実習3の解説

解析ツール「Ghidra」で不審プログラムを解析します。

- 1. 実習用仮想マシン「Caine」を起動します。
- Ghidra Jを起動しプロジェクトウィンドウで「/var/samba/public/tips3/re03_winexec.exe」を インポートし「Code Browser」を開きます。(詳細は、実習1の手順4~手順8を参照。)
- 3. 下図手順で main 関数を表示します。



4. main 関数の中で「WinExec」関数を使用し「C:¥Windows¥system32¥notepad.exe」(メモ帳)を 起動しています。

なお、第2引数に「0」が指定されているため、この不審プログラムから起動されたメモ帳は、 ウィンドウが非表示となり、画面上では何も起きていないように見えます。

[関数の説明]

WinExec(起動したいプログラムのフルパス, ウィンドウの表示状態^{**1}) **1 NULL(0)を指定するとウィンドウが非表示となる。

回答例

不審プログラムから起動される外部プログラム名(フルパス) C:¥Windows¥system32¥notepad.exe

実習4 情報流出したファイルの特定

実習内容

開発用サーバで実行された不審プログラム「re04_malware_easy.exe」を解析し、以下2点を 特定してください。

- ① 情報流出したファイル名(フルパス)
- ② 上記①の送信先 FQDN

[実習用データ]

実習用仮想マシンに格納してあります。

フォルダ : /var/samba/public/tips3/ ファイル : re04_malware_easy.exe

(補足) 実害の無いプログラムですが、ウイルス対策ソフトで検知される可能性があります。 本プログラムは、開発用サーバで実行された不審プログラム「malware.exe」を、実習 で解析しやすいよう修正したものです。

回答記入欄

① 情報流出したファイル名(フルパス)

② 上記①の送信先 FQDN

実習4の解説

解析ツール「Ghidra」で不審プログラムを解析します。

- 1. 実習用仮想マシン「Caine」を起動します。
- Ghidra」を起動しプロジェクトウィンドウで「/var/samba/public/tips3/re04_malware_easy.exe」 をインポートし「Code Browser」を開きます。(詳細は、実習1の手順4~手順8を参照。)
- 3. main 関数(FUN_004011a0)を表示すると、以下の処理を行っていることが分かります。
 - (1) printf 関数でコンソールに「Q.Do you think I leaked your confidential data?」を印字。
 - (2) GetLocalTime 関数で現在時刻を取得し「2019 年 8 月」以外の場合は処理を終了。
 - (3) GetCurrentDirectory 関数でカレントディレクトリを取得。
 - (4) URLDownloadToFileA 関数で「http://c2.attacker.invalid/targets」からダウンロードした ファイルを、カレントディレクトリにファイル名「targets」で保存。
 - (5) WinExec 関数でカレントディレクトリの「ncrack.exe」を実行。その際(4)でダウンロードした ファイル「targets」を引数で指定。
 - (6) 関数「FUN_00401000」を呼び出し、以下の処理を実行。
 - (7) fopen 関数、fread 関数で本番データ「C:¥work¥sendaictf.csv」を読み込み。
 - (8) FQDN と思われる文字列「c2.attacker.invalid」を引数として、不明な関数「Ordinal_N」を 呼び出し。(N は数字 1~3 桁) なお、「Symbol Tree」ウィンドウから「Ordinal_N」は WinSock ライブラリ(WSOCK32.DLL) の関数であることが分かる。
 - (9)「Ordinal_N」は、本番データに対して何らかの処理を実施。
 - (10)「Ordinal_N」による処理は、WinSock によるネットワーク接続、データ送信の処理と類似しているが断定はできない。
 (WinSock のテストコードをコンパイルし、Ghidra で表示すると特徴が一致する。)

4. 関数「Ordinal_N」の詳細が不明なため断定はできませんが、状況から本番データ「C:¥work¥sen daictf.csv」が「c2.attacker.invalid」に送信された可能性が疑われます。



5. なお、IDA Pro で解析した場合は、WinSock の send 関数で本番データをネットワーク送信する 状況を読み取ることができます。



回答例 ※Ghidra による調査では「推測」

- 情報流出したファイル名(フルパス) C:¥work¥sendaictf.csv
- ② 上記①の送信先 FQDN c2.attacker.invalid

以上で演習は終了です。お疲れ様でした。