國家資通安全研究院
National Institute of Cyber Security

組織型駭客之反鑑識手法分享

通報應變中心中心
林易澍 經理
112年11月16日

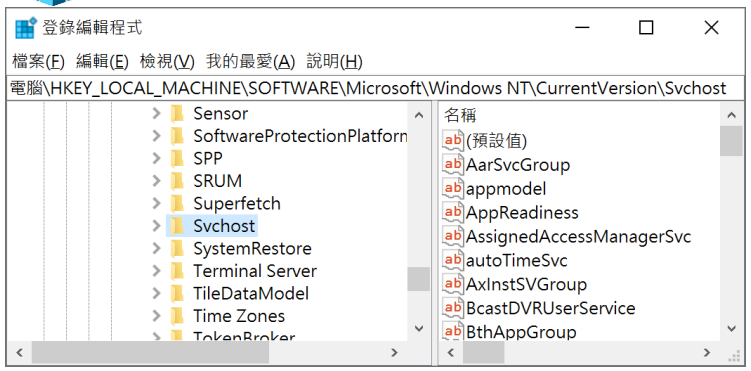
組織型駭客

- 組織型駭客通常指的是一個有組織的駭客團隊，主要是針對特定目標進行系統入侵、資料竊取、網絡破壞等非法活動
  - 多數攻擊屬於進階持續性滲透攻擊(APT)類型
  - 攻擊目標可能是政治、經濟、軍事、商業上的對手
    - 若是政治類型，則通常背後有國家的支持

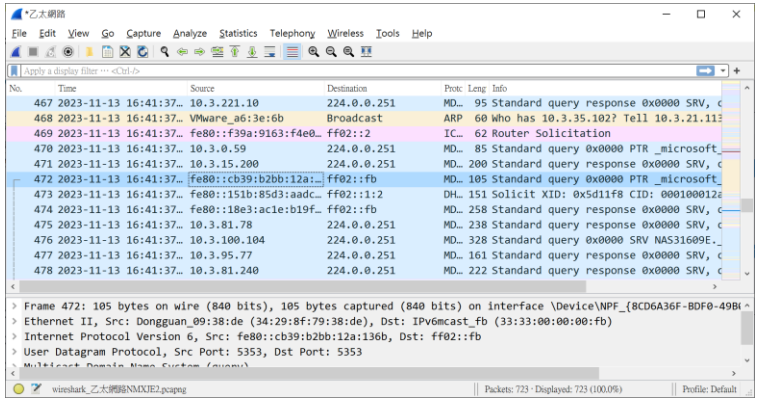
數位鑑識

- 受害的單位在被組織型駭客攻擊之後，我們可以利用數位鑑識的技術，針對受害的設備進行分析

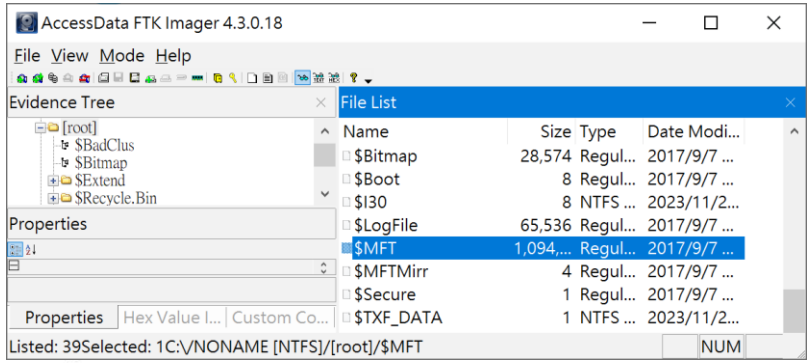
從登錄檔擷取紀錄



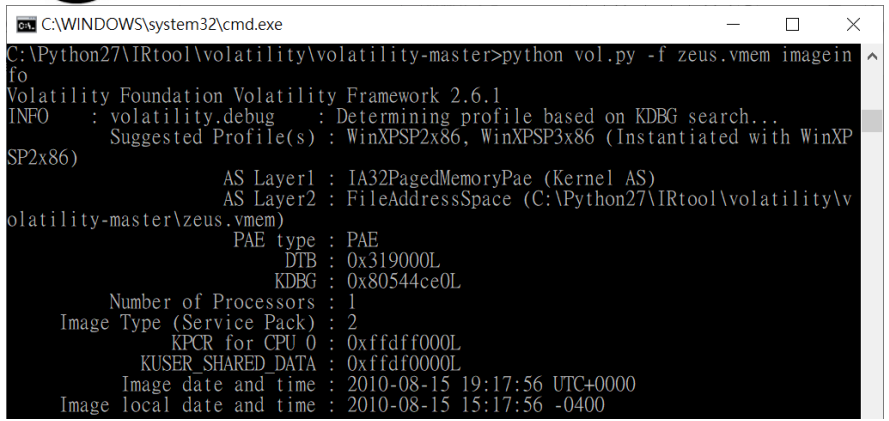
從網路封包擷取紀錄



從硬碟擷取日誌

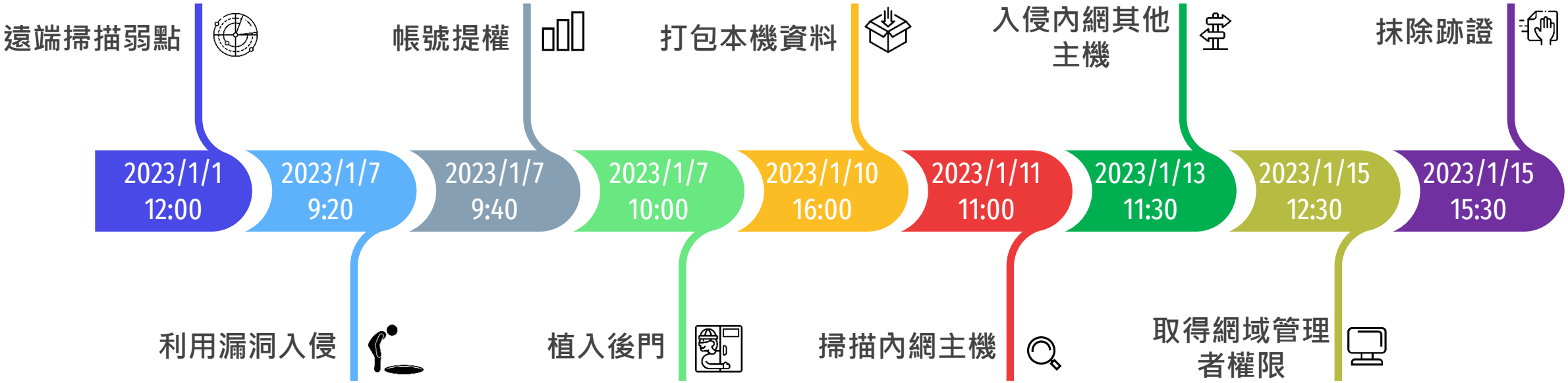


從記憶體擷取紀錄



數位鑑識

- 透過一系列的數位鑑識流程，可還原駭客活動軌跡
  - 了解駭客到底如何入侵，入侵後做了什麼事情，打包了哪些資料，如何擴散等等
  - 受害單位也可了解到如何改善其架構，以防駭客再次透過相似手法入侵

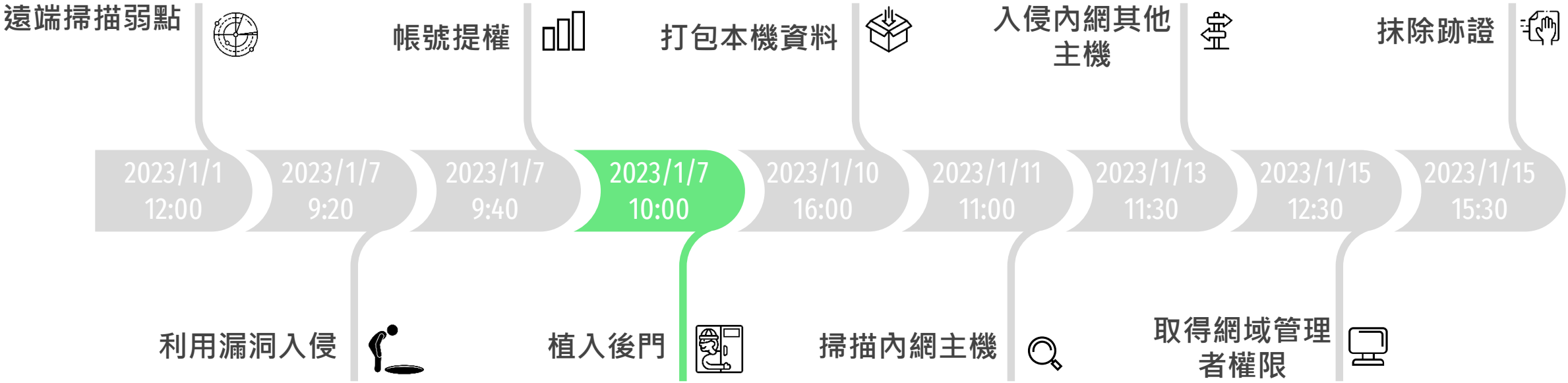


數位鑑識

- 這個工作聽起來怎麼好像很簡單？把所有的日誌/記錄全部找出來，看完之後拼湊出事件的狀況就好了？
  - 必須具備有找出日誌/記錄的知識，也就是需要知道做什麼事會留下什麼日誌/記錄，這些日誌/記錄會存放在哪裡？如何進行擷取？
  - 相關的日誌記錄數量非常龐大，且不會標明哪些行為駭客做的，哪些行為是正常使用者做的
    - 必須花大量時間仔細檢查日誌記錄
  - APT的攻擊通常會藏匿一段時間才被發現，可能是數個月，也可能是數年
    - 若我們不設法搞清楚駭客什麼時候進來的，就有看不完的日誌

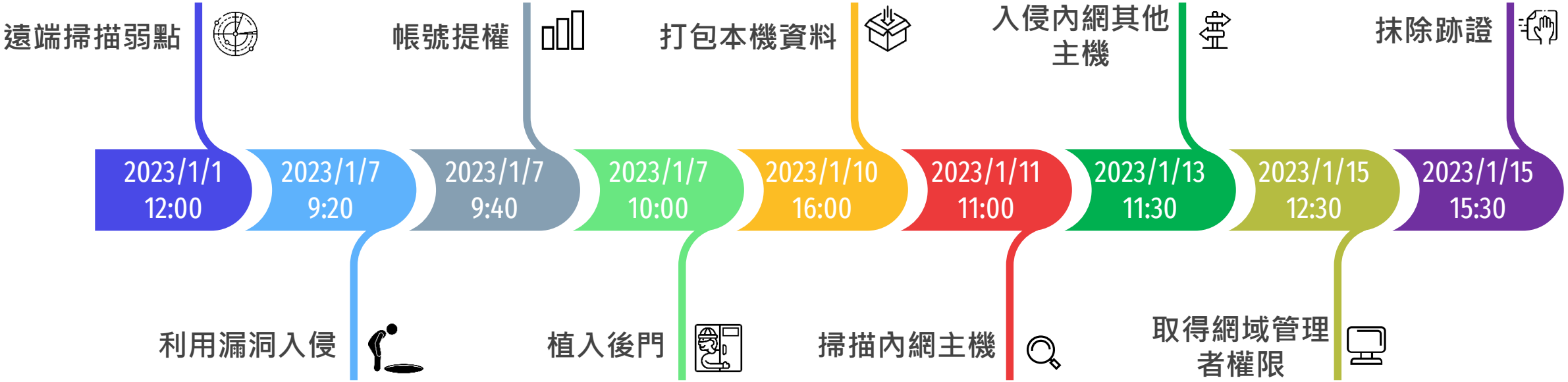
數位鑑識

- 該如何縮小分析的日誌記錄範圍？
  - 其實駭客的操作是具有連貫性的，因此我們只要能找到一個活動，再檢視往前跟往後一小段時間的日誌和記錄就可以還原出駭客活動軌跡
    - 一般而言最好下手的是**植入後門**階段



數位鑑識

- 駭客需要用後門持續掌控受害電腦，通常入侵後就會立刻安裝
  - 如果駭客只偷帳號密碼，若使用者改密碼就進不來了
  - 如果駭客是用漏洞入侵，若使用者修補漏洞就進不來了
- 但若駭客修改了後門植入的日期呢？



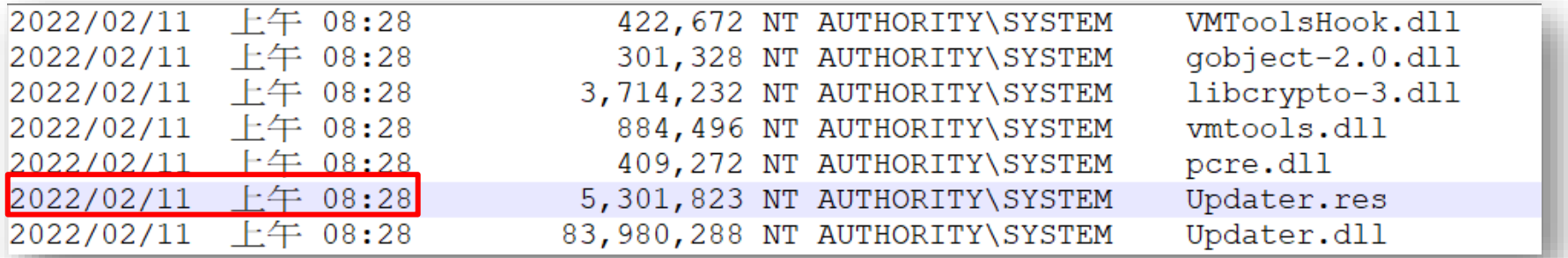
研究背景

研究背景(1/2)

- 調查某組織型駭客之中繼站時，由於該受害設備是虛擬機，因此請受害單位協助先行匯出受駭主機之映像檔
  - 機關於2022/12/14協助匯出映像檔
- 後續至現場鑑識時，除了檢視目前受害虛擬機之狀況外，也攜回映像檔分析
  - 於2022/12/26至受害單位檢測

研究背景(2/2)

- 進行受害映像檔分析時，發現現場找到的部分惡意程式並未出現在映像檔之中
  - 代表該惡意程式植入時間為2022/12/14~2022/12/26之間
  - 但該惡意程式於系統內顯示之建立時間卻為2022/2/11
    - 可知該時間明顯曾遭駭客修改
    - 另透過$MFT分析，嘗試釐清實際植入時間，發現駭客似乎利用特殊手法修改檔案時間，故需進行相關研究



2022/02/11	上午 08:28	422,672	NT AUTHORITY\SYSTEM	VMToolsHook.dll
2022/02/11	上午 08:28	301,328	NT AUTHORITY\SYSTEM	gobject-2.0.dll
2022/02/11	上午 08:28	3,714,232	NT AUTHORITY\SYSTEM	libcrypto-3.dll
2022/02/11	上午 08:28	884,496	NT AUTHORITY\SYSTEM	vmtools.dll
2022/02/11	上午 08:28	409,272	NT AUTHORITY\SYSTEM	pcre.dll
2022/02/11	上午 08:28	5,301,823	NT AUTHORITY\SYSTEM	Updater.res
2022/02/11	上午 08:28	83,980,288	NT AUTHORITY\SYSTEM	Updater.dll

NTFS系統架構

NTFS系統架構(1/5)

- 基本上所有Windows的硬碟格式都是NTFS
  - –雖然Windows有支援exFAT、ReFS格式但相對少見
  - –NTFS是非常成熟的硬碟格式，提供多樣功能與紀錄

NTFS規格相關紀錄

NTFS files

File	Name	$MFT record #	Description
$Mft	Master File Table	0	Contains one base file record for each file and folder on an NTFS volume. If the allocation information for a file or folder is too large to fit within a single record, other file records are allocated as well.
$MftMirr	MFT mirror	1	Guarantees access to the MFT in case of a single-sector failure. It is a duplicate image of the first four records of the MFT.
$LogFile	Log file	2	Contains information used by NTFS for faster recoverability. The log file is used by Windows Server 2003 to restore metadata consistency to NTFS after a system failure. The size of the log file depends on the size of the volume, but you can increase the size of the log file by using the Chkdsk command.
$Volume	Volume	3	Contains information about the volume, such as the volume label and the volume version.
$AttrDef	Attribute definitions	4	Lists attribute names, numbers, and descriptions.
.	Root file name index	5	The root folder.
$Bitmap	Cluster bitmap	6	Represents the volume by showing free and unused clusters.
$Boot	Boot sector	7	Includes the BPB used to mount the volume and additional bootstrap loader code used if the volume is bootable.
$BadClus	Bad cluster file	8	Contains bad clusters for a volume.
$Secure	Security File	9	Contains unique security descriptors for all files within a volume.
$Upcase	Upcase table	10	Converts lowercase characters to matching Unicode uppercase characters.
$Extend	NTFS extension file	11	Used for various optional extensions such as quotas, reparse point data, and object identifiers.
		12-15	Reserved for future use.

source: http://technet.microsoft.com/en-us/library/cc781134(WS.10).aspx

NTFS系統架構(2/5)

- 在所有紀錄中，最重要的是$MFT紀錄
  - 該紀錄會標示每個檔案的詳細資訊，例如檔名、內容、時間等

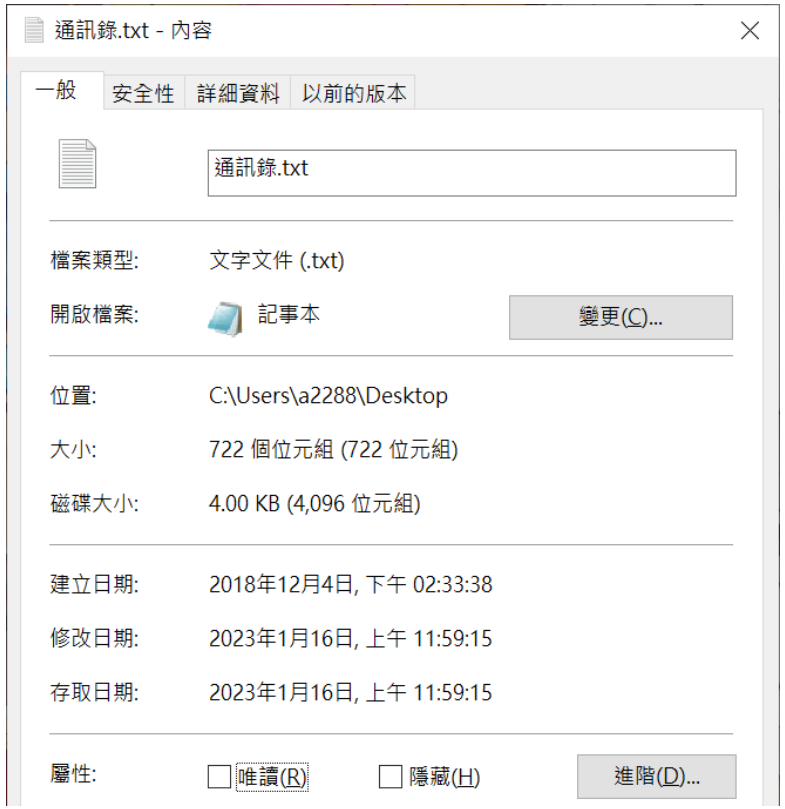
MFT紀錄

Some $MFT entry attributes

ID	Attribute Type	Description
0x10	Standard Information	Includes information such as time stamp and link count.
0x20	Attribute List	Lists the location of all the attribute records that do not fit in the MFT record.
0x30	File Name	A repeatable attribute for both long and short file names. The long name of the file can be up to 255 Unicode characters. The short name is the MS-DOS-readable, 8.3, case-insensitive name for the file. Additional names, or hard links, required by POSIX can be included as additional file name attributes.
0x40	Object ID	A volume-unique file identifier. Used by the link tracking service. Not all files have object identifiers.
0x50	Security Descriptor	Shows information about who owns the file and who can access the file.
0x60	Volume Name	Used only in the $Volume system file. Contains the volume label.
0x70	Volume Information	Used only in the $Volume system file. Contains the volume version.
0x80	Data	Contains file data. NTFS allows multiple data attributes per file. Each file typically has one unnamed data attribute. A file can also have one or more named data attributes, each using a particular syntax.
0x90	Index Root	Used to implement folders and other indexes.
0xA0	Index Allocation	Used to implement folders and other indexes.
0xB0	Bitmap	Used to implement folders and other indexes.
0xC0	Reparse Point	Used for directory junction points and volume mount points. They are also used by file system filter drivers to mark certain files as special to that driver.
0x100	Logged Tool Stream	Similar to a data stream, but operations on a logged tool stream are logged to the NTFS log file just like NTFS metadata changes. Used by EFS.

NTFS系統架構(3/5)

- 在$MFT中，有兩個欄位會帶有檔案相關時間
  - Standard Information(SI)欄位會記錄一組(4個)時間，我們經常看到的時間即為Standard Information Time
  - 4個時間分別為
    - Creation Time
    - Last Write Time
    - Last Access Time
    - Metadata Time (隱藏欄位)



NTFS系統架構(4/5)

- 在$MFT中，有兩個欄位會帶有檔案相關時間
  - File Name(FN)欄位會記錄另一組(4個)時間，但僅能用鑑識工具進行檢視
  - 4個時間分別為
    - Creation Time (隱藏欄位)
    - Last Write Time (隱藏欄位)
    - Last Access Time (隱藏欄位)
    - Metadata Time (隱藏欄位)

NTFS系統架構(5/5)

- 若要針對檔案時間進行調整，僅能修改Standard Information Time，無法修改File Name Time
  - 使用者可用指令/API輕易修改Standard Information Time
  - 無法修改File Name Time之原因為沒有可用API，修改File Name Time之API不但為未公開API，使用時更會被微軟的Patch Guard機制擋下
    - 若繞過Patch Guard的保護，等於可以直接控制Windows Kernel，難度非常高
    - 因此過往的迷思，都是FN無法修改，僅有SI可以修改

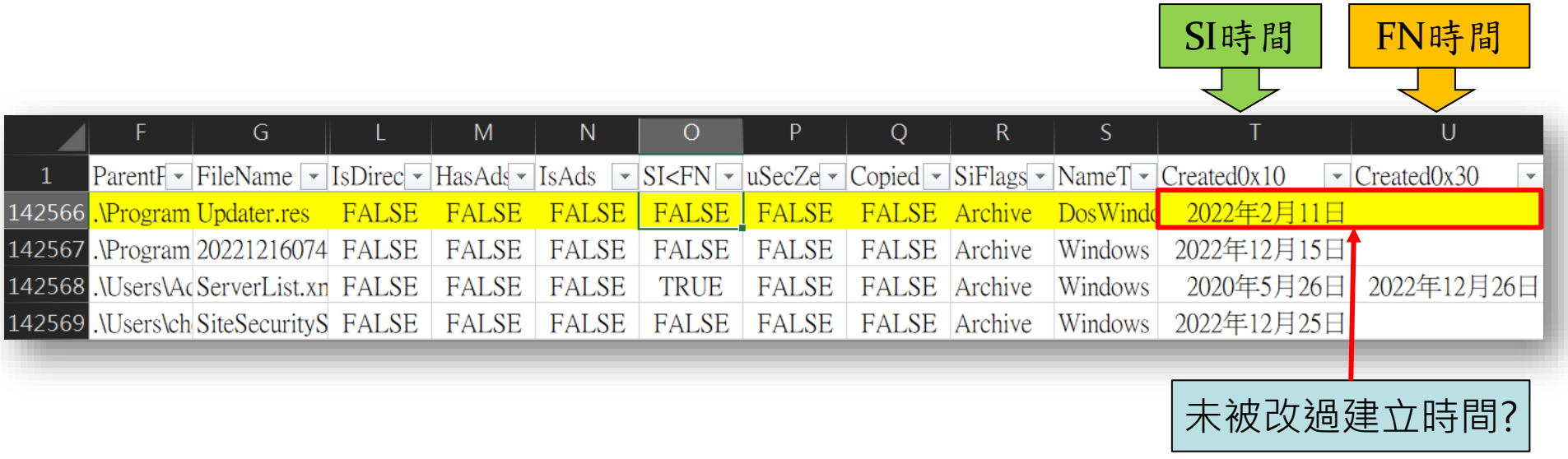
國家資通安全研究院
National Institute of Cyber Security

駭客手法分析與實證

駭客手法分析與實證(1/9)

國家資通安全研究院
National Institute of Cyber Security

- 在本案發現駭客似乎可以修改File Name Time
  - 我們使用鑑識工具對硬碟中$MFT進行分析，竟然發現該惡意程式的Standard Information Time和File Name Time是一樣的
    - 該工具若未顯示FN時間，則代表此檔案的FN時間和SI時間相同
    - 惡意程式為黃色，明顯可看出FN時間和SI時間相同，即未被改過建立時間



	F	G	L	M	N	O	P	Q	R	S	T	U
1	ParentF	FileName	IsDirec	HasAds	IsAds	SI<FN	uSecZe	Copied	SiFlags	NameT	Created0x10	Created0x30
142566	.\Program	Updater.res	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	Archive	DosWindo	2022年2月11日	
142567	.\Program	20221216074	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	Archive	Windows	2022年12月15日	
142568	.\Users\Ac	ServerList.xn	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	Archive	Windows	2020年5月26日	2022年12月26日
142569	.\Users\ch	SiteSecurityS	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	Archive	Windows	2022年12月25日	

駭客手法分析與實證(2/9)

國家資通安全研究院
National Institute of Cyber Security

- 我們在分析該惡意程式相關時間，意外發現該程式Standard Information中的Metadata Time很接近正確的植入時間
  - 該程式為12/14~12/26間植入，8個時間中僅有Metadata Time符合此時段

是否為實際植入時間?

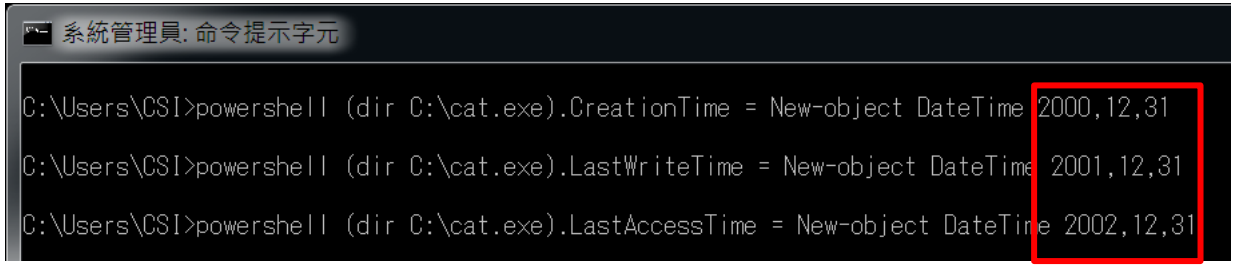
	J	S	T	U	V	X	Y	Z	AA
1	FN_FileN	SI_CTime	SI_ATime	SI_MTime	SI_RTime	FN_CTime	FN_ATime	FN_MTime	FN_RTime
114425	Updater.res	2022/2/11 0:28	2022/2/11 0:28	2022/12/14 23:41	2022/2/11 0:28	2022/2/11 0:28	2022/2/11 0:28	2022/2/11 0:28	2022/2/11 0:28
114426	201D14~1.P	2022/12/15 23:41	2022/12/15 23:41	2022/12/15 23:41	2022/12/15 23:41	2022/12/15 23:41	2022/12/15 23:41	2022/12/15 23:41	2022/12/15 23:41
114427	SERVER~1.	2020/5/26 18:15	2022/12/26 3:32	2022/12/26 3:32	2022/12/26 3:32	2022/12/26 3:32	2022/12/26 3:32	2022/12/26 3:32	2022/12/26 3:32
114428	SITESE~1.T	2022/12/25 22:16	2022/12/25 22:16	2022/12/25 22:16	2022/12/25 22:16	2022/12/25 22:16	2022/12/25 22:16	2022/12/25 22:16	2022/12/25 22:16

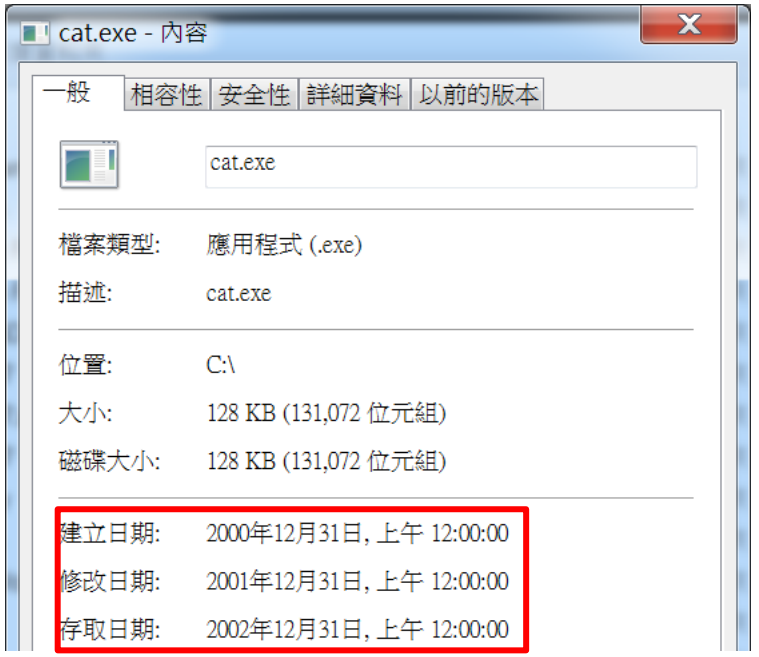
駭客手法分析與實證(3/9)

- 因此我們於實驗環境對此進行實測
  - 隨意準備一個檔案，其時間如下

駭客手法分析與實證(4/9)

- 我們透過powershell指令改變其Standard Information之3個時間

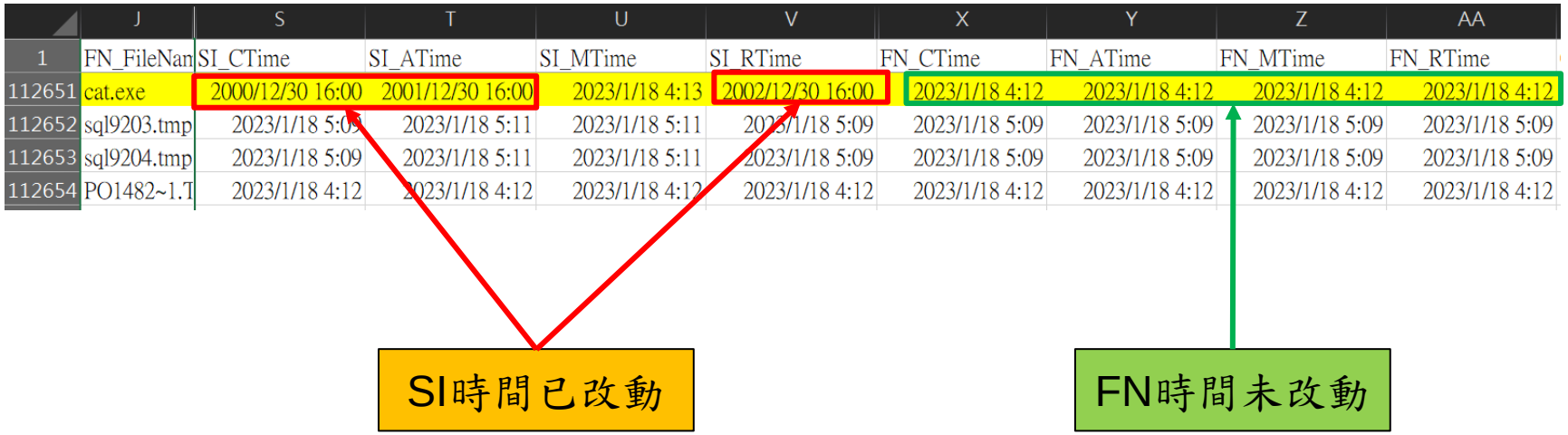




駭客手法分析與實證(5/9)

國家資通安全研究院
National Institute of Cyber Security

- 透過鑑識工具分析MFT的內容，發現僅有Standard Information(SI)的時間改動，而File Name(FN)的時間未改動
  - MFT紀錄之時間均為UTC，故需+8小時始為當前時區



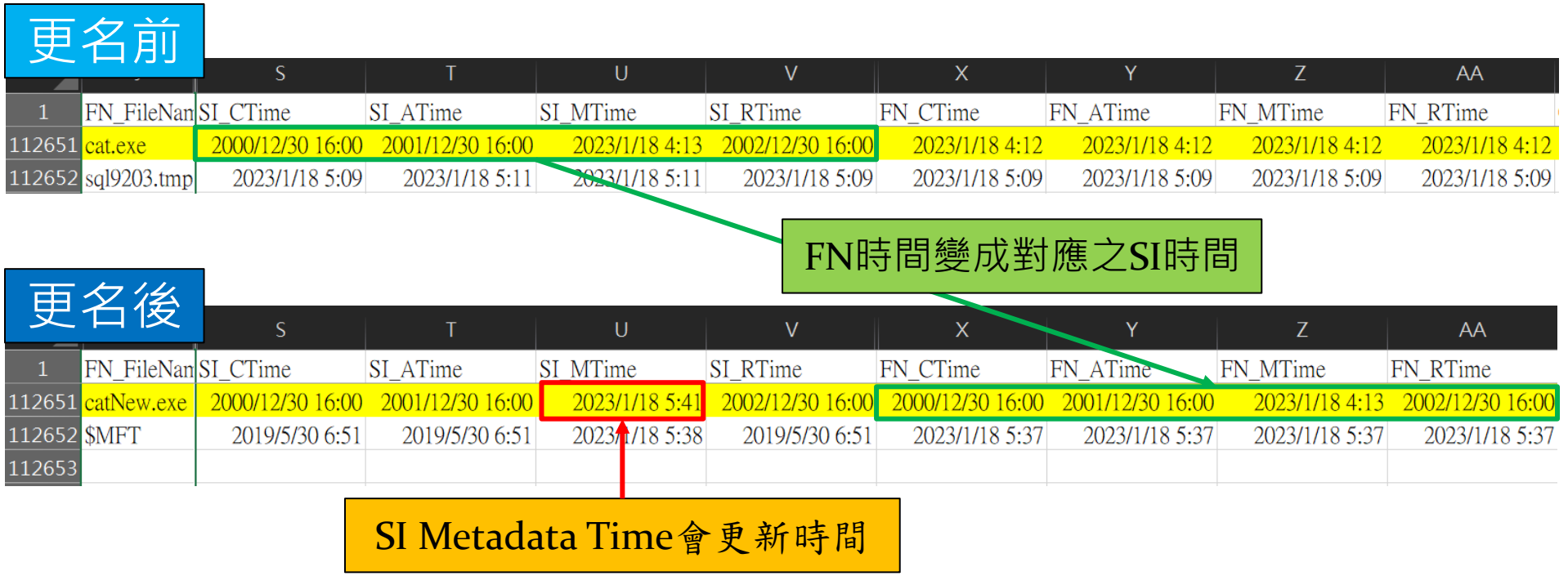
	J	S	T	U	V	X	Y	Z	AA
1	FN_FileNan	SI_CTime	SI_ATime	SI_MTime	SI_RTime	FN_CTime	FN_ATime	FN_MTime	FN_RTime
112651	cat.exe	2000/12/30 16:00	2001/12/30 16:00	2023/1/18 4:13	2002/12/30 16:00	2023/1/18 4:12	2023/1/18 4:12	2023/1/18 4:12	2023/1/18 4:12
112652	sql9203.tmp	2023/1/18 5:09	2023/1/18 5:11	2023/1/18 5:11	2023/1/18 5:09	2023/1/18 5:09	2023/1/18 5:09	2023/1/18 5:09	2023/1/18 5:09
112653	sql9204.tmp	2023/1/18 5:09	2023/1/18 5:11	2023/1/18 5:11	2023/1/18 5:09	2023/1/18 5:09	2023/1/18 5:09	2023/1/18 5:09	2023/1/18 5:09
112654	PO1482~1.T	2023/1/18 4:12	2023/1/18 4:12	2023/1/18 4:12	2023/1/18 4:12	2023/1/18 4:12	2023/1/18 4:12	2023/1/18 4:12	2023/1/18 4:12

駭客手法分析與實證(6/9)

- 經查SANS公布有關檔案時間之鑑識研究，可知不會更動到SI中3個時間之行為僅有檔案更名與同磁碟區檔案移動兩種

駭客手法分析與實證(7/9)

- 接著進行檔名變更，Standard Information的Metadata Time會變成修改檔名的時間
  - 同時因為檔名調整的影響，會發現所有的File Name時間均變成其對應的Standard Information時間



駭客手法分析與實證(8/9)

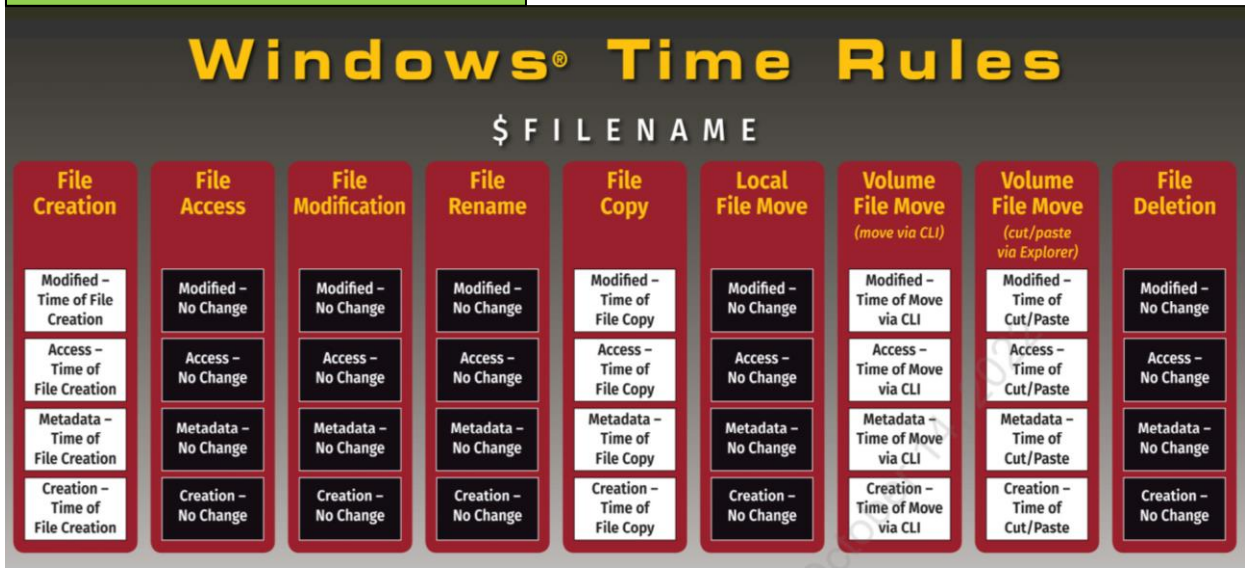
- 由以上實驗可知，駭客修改檔案時間之流程推測如下步驟
  1. 透過工具修改檔案SI的3個時間(不包含metadata time)或4個時間(包含metadata time)
  2. 將檔案更名或移動
  3. 再度修改檔案SI的metadata time，以達成所有SI和FN時間都是駭客想要的內容
- 本案的駭客可能是忘了進行動作3，所以才會有SI的metadata time與其他時間不同之情形

駭客手法分析與實證(9/9)

國家資通安全研究院
National Institute of Cyber Security

- SANS新版之檔案操作時間對應表已於今年調整

2022年的SANS資料



2023年的SANS資料

偵測方式

偵測方式(1/2)

- 由於NTFS檔案格式具有許多日誌紀錄，我們可以透過查找其他日誌了解駭客是否使用前述反鑑識手法
  - 分析NTFS的$Extend\$UsnJrnl日誌，可以知道檔案是否有改名/改路徑的紀錄
  - 但該日誌紀錄的東西較多，通常僅能保存1日，之後便會覆蓋舊有紀錄

	A	B	C	D	E	F
1	Offset	FileName	USN	Timestamp	Reason	MFTRefe
327519	0x01E746	cat.exe	3.89E+08	2023/1/18 13:41	RENAME_OLD_NAME	112649
327520	0x01E747	catNew.exe	3.89E+08	2023/1/18 13:41	RENAME_NEW_NAME	112649
327521	0x01E747	catNew.exe	3.89E+08	2023/1/18 13:41	CLOSE+RENAME_NEW_NAME	112649
327522	0x01E747	ftk_0823db7e-4e	3.89E+08	2023/1/18 13:41	CLOSE+FILE_DELETE	67350

偵測方式(2/2)

- 由於NTFS檔案格式具有許多日誌紀錄，我們可以透過查找其他日誌了解駭客是否使用前述反鑑識手法
  - 分析NTFS的$Extend\$LogFile日誌，可以知道檔案的FN time修改前後之紀錄
  - 但該日誌紀錄的東西過多，通常僅能保存數小時，之後便會覆蓋舊有紀錄

	G	J	K	AE	AF	AG	AH
1	lf_RedoOperation	lf_FileName	lf_CurrentAttribute	lf_FN_CTime	lf_FN_ATime	lf_FN_MTime	lf_FN_RTime
207616	DeleteAttribute	cat.exe	$FILE_NAME	2023/1/18 4:12	2023/1/18 4:12	2023/1/18 4:12	2023/1/18 4:12
207617	CreateAttribute	catNew.exe	$FILE_NAME	2000/12/30 16:00	2001/12/30 16:00	2023/1/18 4:13	2002/12/30 16:00
207618	AddIndexEntryAllocation	catNew.exe	$INDEX_ALLOCATION:$I30				
207619	SetNewAttributeSizes		??				
207620	UpdateNonResidentValue	cat.exe	$DATA:$J				

結論

- 不只我們懂得做數位鑑識，組織型駭客也很了解這個領域
  - 駭客知道要如何修改跡證以防被我們還原其入侵軌跡
- 過往的知識可能並不是永遠正確
  - 不是FN Time不能修改，只是沒有API可以修改，但駭客可以不透過API修改時間
- 近期有部分組織型駭客開始透過此法修改檔案時間進而干擾跡證分析
  - 駭客若操作沒有依照SOP完成所有步驟，就有可能留下痕跡可供分析
- 雖可透過$UsnJrnl和$LogFile偵測該手法利用之可能性，但相關日誌保存時間均十分短暫，一旦入侵時間較長即難以找出實際遭駭時間

國家資通安全研究院
National Institute of Cyber Security

報告完畢 敬請指教