Disk Knight Worm Analysis

Luca D’Amico

https://www.lucadamico.dev

25-Jun-2023

Abstract....3
Ambiente, metodologie e strumenti usati....4
Informazioni sul binario....5
Analisi del malware....6
Installazione....6
Esecuzione....7
Propagazione....9
IOCs....11
Regola di rilevamento YARA....12
Rimozione del malware....13
Rimozione di Disk Knight dal pc infetto....13
Rimozione di Disk Knight dai drive USB infetti....14
Extra: soluzione al bug della tray icon....15
Conclusione....18

Abstract

Questo documento è un'analisi tecnica del malware Disk Knight.

Disk Knigh sembra essere stato concepito come un antivirus per bloccare la diffusione dei malware trasmessi tramite le chiavette usb. Questi tipi di worm sfruttano l'esecuzione automatica all'accesso della periferica grazie al file autorun.inf.

A causa di alcune scelte implementative pessime e alla presenza di problemi nel suo codice, questo software si è trasformato in un worm fuori controllo, diffondendosi automaticamente all'inserimento di dispositivi di archiviazione USB nei pc infetti ed infettando a sua volta i computer dove queste chiavette venivano inserite.

Ha avuto il suo momento di diffusione massimo verso la fine del 2007 e l'inizio del 2008.

La seguente relazione analizzerà i tre stadi principali del malware ovvero installazione, esecuzione e propagazione. Verranno quindi catalogati gli IOC e verrà scritta una regola di detection usando Yara. Infine, verrà applicata una modifica al codice del worm per far apparire correttamente la sua tray icon e verranno elencati i comandi per eliminare questo malware dal sistema e dalle pendrive infette.

Ambiente, metodologie e strumenti usati

Per effettuare l'analisi è stata utilizzata una macchina virtuale con Windows XP SP3. Benché sia possibile usare una versione più recente di Windows, si è scelto di utilizzare la versione più diffusa al momento del rilascio del worm. Inoltre, le versioni del sistema operativo successive richiedono l'autorizzazione tramite UAC.

Non è stato installato alcun tipo di antivirus nella macchina virtuale.

Sono stati usati i seguenti strumenti in fase di analisi:

- CFF Explorer, PE-Bear: ottenimento di informazioni dal file eseguibile
- P32Dasm 2.80: ottenimento di importanti informazioni sulle procedure VB6 del worm
- Process Monitor 3.20: ottenimento informazioni sul processo del worm durante la fase di detonazione
- X32dbg: debugging durante l'analisi dinamica del malware

Informazioni sul binario

Di seguito vengono riportare alcune caratteristiche del worm ottenute da una prima analisi sul file eseguibile:

Binary name	Knight.exe
File size	412 KB
SHA-256	d25c1d1423ed31b5436678318ca815092102e88d06a130481bc0728d14d74bb4
Language detected	Visual Basic 6
TimeDateStamp	46cc3475 (22.08.2007 13:04:53 UTC)
FileVersion	4.02
VirusTotal URL	https://www.virustotal.com/gui/file/d25c1d1423ed31b5436678318ca815092102e88d06a130481bc0728d14d74bb4
Virus Total popular threat name	worm.diskknight/knight

Nella section .rsrc, ovvero quella delle risorse, è presente una entry chiamata “CUSTOM”, con all’interno due risorse “AUTORUN.INF” e “RECOVER.REG”. La prima verrà usata dal worm durante la fase di propagazione, come descritto in seguito. È anche presente una sezione con del codice HTML che il worm visualizzerà durante la funzione “Help” (descritta in seguito).

Analisi del malware

In questa sezione verranno descritte le tre fasi principali del worm, ovvero installazione, esecuzione e propagazione.

Installazione

La tecnica con cui Disk Knight si installa nel sistema è molto semplice: la funzione situata al VA 0x408900 usa la funzione CopyFileA per copiare l'eseguibile nella cartella di Windows (comunemente C:\Windows\) con nome Knight.exe.

Il percorso della directory dove risiede il sistema operativo viene recuperata grazie alla funzione SHGetFolderPathA passando il parametro CSIDL_WINDOWS (0x24), in questo modo vi è la certezza di ottenere la cartella corretta anche nei casi in cui il sistema operativo sia installato in un percorso non convenzionale.

La funzione incaricata di avviare Disk Knight è situata al VA 0x00408958 ed utilizza una chiamata a ShellExecuteA con i seguenti argomenti:

```
ShellExecuteA(NULL, "open", "C:\WINDOWS\Knight.exe", "protect", "C:\WINDOWS", SW_SHOWNORMAL);
```

Prima di terminare l'esecuzione del processo attuale, viene creato il file recover.reg nella cartella di installazione.

Esecuzione

A questo punto Disk Knight è in esecuzione dal binario installato nella cartella di Windows.

Quando il processo è attivo, se si tenta di avviare il worm da una cartella diversa rispetto a quella dove è installato il sistema operativo, il malware cercherà di installare Knight.exe sovrascrivendolo, ma questo tentativo fallirà con conseguente crash poiché il binario è attualmente in uso. Il file Knight.exe verrà comunque riavviato.

Il worm a questo punto verifica se esiste già un suo processo in esecuzione con un metodo molto semplice, ovvero chiamando una FindWindowA:

```
FindWindowA("Disk Knight", "ThunderRT6FormDC");
```

Se riesce ad ottenere un handle, allora significa che il worm è già in esecuzione e quindi il processo attuale terminerà.

Una volta eseguito dalla directory dove è installato Windows, una la funzione situata al VA 0x00423E90 si occupa di inserire una chiave ne registro per permettere al worm di persistere ai riavvii del sistema operativo. La chiave è

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Disk Knight e contiene la path di Knight.exe nella directory di Windows.

ATTENZIONE: a questo punto sarebbe dovuta comparire la tray icon per il controllo di Disk Knight, ma a causa di bug del codice, la funzione incaricata di crearla non verrà mai raggiunta dalla call a 0x41BB90, rendendo l'applicazione NON controllabile dall'utente! Per maggiori informazioni e per il patching di questo bug, è presente una sezione dedicata in questo documento.

Per rilevare i nuovi dispositivi USB inseriti, il worm usa un hook per intercettare i messaggi della window procedure:

```
SetWindowLong(ThunderRT6FormDC whandle, GWL_WNDPROC,  0x41B9F0)
```

All'indirizzo 0x41B9F0 si trova una funzione che verifica il tipo di messaggio e se è uguale a WM_DEVICECHANGE, viene lanciata la funzione situata a 0x421EB0.

Questa funzione controlla il dispositivo inserito raccogliendo i dati necessari e infine la funzione a 0x41B8D0 verrà chiamata.

Propagazione

Quando una nuova chiavetta USB viene inserita, la funzione situata al VA 0x0041ED70 si occupa dell'infezione, chiamata a sua volta dalla funzione posizionata al VA 0x41B8D0.

Questo avviene secondo i seguenti passaggi:

1a) Viene effettuata una chiamata a SetFileAttributesA sia su Knight.exe che su autorun.inf:

```
SetFileAttributesA("X:\\Knight.exe", 0x80);
SetFileAttributesA("X:\\autorun.inf", 0x80);
```

Dove X rappresenta la lettera del drive che sta venendo infettato e 0x80 l'attributo FILE_ATTRIBUTE_NORMAL

1b) Se l'operazione ha successo, e quindi i file sono presenti sulla chiavetta usb, viene effettuata una chiamata a DeleteFileA sia su Knight.exe che su autorun.inf, rimuovendoli:

```
DeleteFileA("X:\\Knight.exe");
DeleteFileA("X:\\autorun.inf");
```

Dove X rappresenta la lettera del drive che sta venendo infettato

2) Viene effettuata l'infezione, che consiste nell'installazione dei due file:

- Knight.exe viene copiato dalla cartella di Windows alla periferica usando la funzione CopyFileA:

```
CopyFileA("C:\\WINDOWS\\Knight.exe", "X:\\Knight.exe", 0);
```

Dove X rappresenta la lettera del drive che sta venendo infettato.

- autorun.inf viene estratto dal segmento delle risorse di Knight.exe, dalla sezione chiamata "CUSTOM" e posizionato sulla radice della chiavetta USB.

3) Entrambi i file vengono nascosi sfruttano l'API SetFileAttributesA:

```
SetFileAttributesA("X:\\Knight.exe", 0x7);
SetFileAttributesA("X:\\autorun.inf", 0x7);
```

Dove X rappresenta la lettera del drive che sta venendo infettato e 0x7 i seguenti attributi: FILE_ATTRIBUTE_HIDDEN, FILE_ATTRIBUTE_READONLY, FILE_ATTRIBUTE_SYSTEM

Il file autorun.inf fa sì che accedendo alla periferica USB, indipendentemente dal tipo di accesso selezionato, disk knight venga avviato e conseguentemente il computer infettato.

IOCs

La seguente tabella riporta gli IoC ottenuti dall'analisi del file eseguibile e dal comportamento del worm durante l'esecuzione.

File	%windir%\Knight.exe	File Attr: Hidden, Read Only, System
File	%windir%\recover.reg	
File	%windir%\0.log	
File su device USB	X:\Knight.exe	File Attr: Hidden, Read Only, System
File su device USB	X:\autorun.inf	File Attr: Hidden, Read Only, System
Processo	Disk Knight (Knight.exe/Administrator)	
Chiave di registro	Path: HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run Nome: Disk Knight Valore: %windir%\Knight.exe	
Chiave di registro	Path: HKLM\SOFTWARE\Knight\Settings Nomi: drive, protectmode	
SHA256	d25c1d1423ed31b5436678318ca815092102e88d06a130481bc0728d14d74bb4	Knight.exe

Regola di rilevamento YARA

Questa è una regola di rilevamento specifica per il sample di Disk Knight analizzato:

```
import "pe"

rule diskknight {
   meta:
      description = "Disk Knight detection (worm.diskknight/knight) - VERY SPECIFIC"
      author = "Luca D'Amico"
      date = "2023/06/24"
      hash0 = "d25c1d1423ed31b5436678318ca815092102e88d06a130481bc0728d14d74bb4"

   strings:
      $a1 = "http://www.ariful.esmartweb.com"
      $a2 = "action=Disk Knight(Protection Against Mobile Disk Viruses)"
      $a3 = "[Disk Knight]"

   condition:
      uint16(0) == 0x5A4D and
      pe.machine == pe.MACHINE_I386 and
      for any i in (0..(pe.number_of_resources)-1):
      (
         pe.resources[i].type_string == "C\x00U\x00S\x00T\x00O\x00M\x00" and
         (pe.resources[i].name_string ==
"A\x00U\x00T\x00O\x00R\x00U\x00N\x00.\x00I\x00N\x00F\x00" or
         pe.resources[i].name_string ==
"R\x00E\x00C\x00O\x00V\x00E\x00R\x00.\x00R\x00E\x00G\x00")
      ) and
      pe.imports("MSVBVM60.DLL") and
      all of them

}
```

Rimozione del malware

La rimozione di Disk Knight deve avvenire in due fasi e necessariamente in questo ordine:

1) Rimozione del malware dal pc infetto
2) Rimozione del malware da tutti i drive USB infetti

Prima di effettuare questa procedura occorre disattivare l'avvio automatico tramite file autorun.inf in modo da non essere nuovamente infettati durante la rimozione del malware dai dispositivi USB. Occorre quindi modificare una chiave nel registro di sistema, eseguendo il seguente comando su una istanza di cmd.exe:

reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\IniFileMapping\Autorun.inf" /v "" /t REG_SZ /d "@SYS:DoesNotExist" /f

Rimozione di Disk Knight dal pc infetto

Eseguire i seguenti comandi su una istanza di cmd.exe:

1) taskkill /F /IM "Knight.exe"
2) attrib -h -s -r "%windir%\Knight.exe"
3) del /f "%windir%\Knight.exe"
4) del /f "%windir%\recover.reg"
5) del /f "%windir%\0.log"
6) reg delete "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /v "Disk Knight" /f
7) reg delete "HKEY_LOCAL_MACHINE\SOFTWARE\Knight" /f

A questo punto Disk Knight non sarà più presente sul PC ed è possibile continuare la procedura di rimozione del malware dalle chiavette infette.

Rimozione di Disk Knight dai drive USB infetti

Assicurarsi di aver disattivato l'avvio automatico tramite file autorun ed eseguire i seguenti comandi su una istanza di cmd.exe, avendo cura di modificare la lettera "X" con quella del drive che si intende disinfettare:

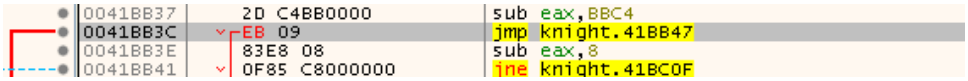
1) attrib -h -s -r "X:\autorun.inf"
2) del /f "X:\autorun.inf"
3) attrib -h -s -r "X:\Knight.exe"
4) del /f "X:\Knight.exe"

Adesso il dispositivo USB sarà pulito.

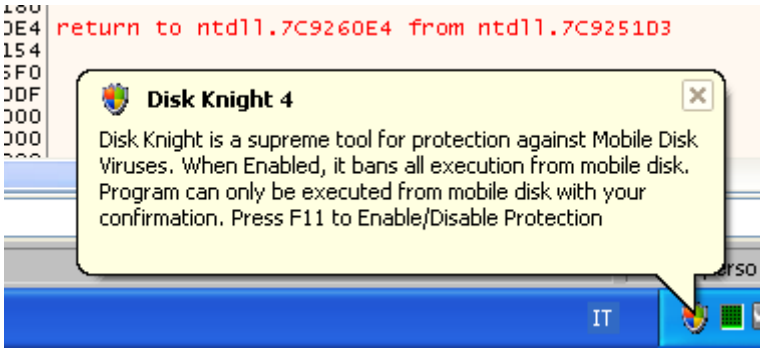
Extra: soluzione al bug della tray icon

Durante l'analisi del worm ho notato che forzando un salto condizionale con il debugger è possibile far apparire correttamente la tray icon di Disk Knight ed accedere alle sue opzioni.

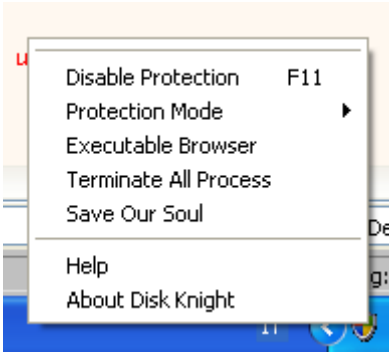
La patch consiste nel modificare l'istruzione al VA 0x41BB3C da JE a JMP:



Subito dopo aver ripreso l'esecuzione, la tray icon apparirà mostrando il seguente messaggio:

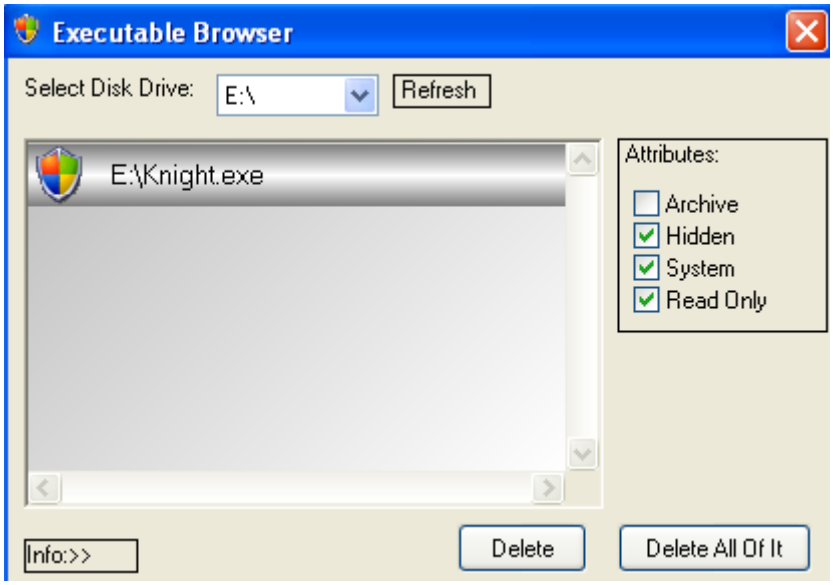


È possibile interagire con l'icona cliccando con il tasto destro del mouse per rivelare il seguente menu:



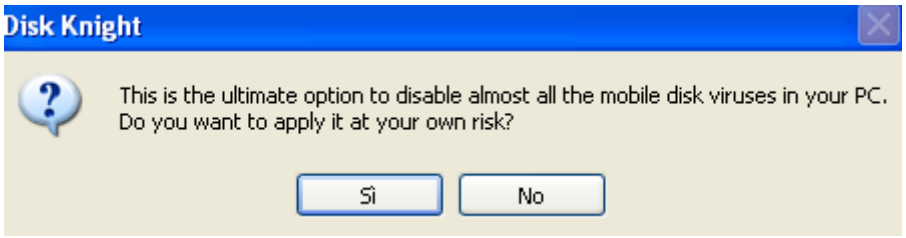
Queste opzioni permettono di scegliere il "livello di protezione" offerto da Disk Knight (ad esempio il blocco degli eseguibili da dispositivi USB).

La funzione Executable Browser permette di ottenere una lista degli eseguibili presenti nel drive USB e verificare gli attributi associati ad essi:



La funzione “Terminate All Process” terminerà tutti i processi attivi del sistema operativo.

Cliccando su “Save Our Soul” verrà mostrato il seguente messaggio:

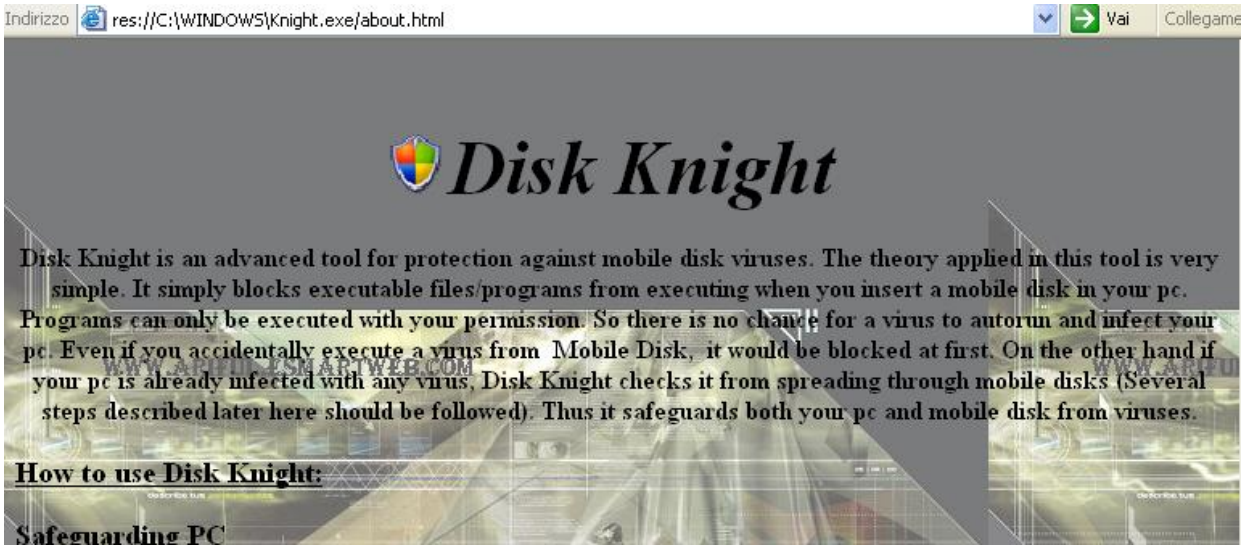


Se si decide di continuare, verranno chiusi tutti i processi attivi e verranno effettuate varie modifiche al registro di sistema come la disattivazione dell’esecuzione automatica dei programmi all’avvio. Infine, il computer verrà riavviato.

È interessante notare che l’opzione “Disable Protection” non sembra funzionare: nonostante possa essere cliccata, Disk Knight continuerà a propagarsi sui dispositivi USB inseriti.

Nella parte inferiore del menù sono presenti due opzioni, Help e About Disk Knight.

Help aprirà il browser predefinito del sistema mostrando una pagina html estratta dal segmento delle risorse di Disk Knight. Questa pagina illustra le caratteristiche del malware:



Cliccando su About invece verrà presentata la seguente finestra:

Il click su OK causerà l'apertura del browser sulla presunta pagina web dell'autore. Tale pagina risulta offline e non è possibile trovare istantanee funzionanti usando Wayback Machine.

Conclusione

Questa analisi ha messo in risalto quanto sia importante ponderare attentamente le scelte implementative durante la scrittura di software e quanto sia altrettanto importante assicurarsi che non siano presenti bug così gravi nel proprio codice prima di effettuare un rilascio pubblico.

Fortunatamente in questo caso non ci sono stati grossi danni e neanche perdita di dati, limitando il problema alla diffusione fuori controllo di Disk Knight, soprattutto sui computer sprovvisti di antivirus.

Spero che questa analisi sia stata di vostro gradimento e ringrazio vivamente la scena italiana (e anche quella globale :) di malware analysis e reverse engineering.

Per ulteriori analisi e documenti tecnici, visitate il mio sito:
https://www.lucadamico.dev