Disk Knight Worm Analysis

Luca D’Amico

https://www.lucadamico.dev

29-Jun-2023

Abstract

This is a technical analysis of Disk Knight malware.

Disk Knight seems to have been conceived as an antivirus to block the rise of malware spread via USB sticks. This type of worm exploits the automatic execution when the USB device is accessed, thanks to the presence of an autorun.inf file.

Due to some bad implementation choices and the presence of bugs in its code, this software has turned into an out-of-control worm, spreading itself automatically when USB storage devices are inserted into infected PCs and in turn infecting the computers where these were entered.

It reached its peak in late 2007 and early 2008.

This document will analyse the three main stages of the malware: installation, execution and spread. We will then catalogue the IOCs and write a detection rule using Yara. Finally, a patch to the worm's code will be applied to make its tray icon appear correctly and the commands to eliminate this malware from the system and from infected USB sticks will be listed.

Environment, methodologies and tools used

To carry out this analysis, a Windows XP SP3 virtual machine was used. Although it is possible to use a newer version of Windows, I have chosen to use the most popular version at the time of the release of the worm. Also, later OS versions require authorization via UAC.

No antivirus of any kind has been installed in the virtual machine.

The following tools were used during the analysis:

- CFF Explorer, PE-Bear: to get information from the executable file
- P32Dasm 2.80: to obtain valuable information about the VB6 procedures of the worm
- Process Monitor 3.20: obtaining information about the worm process during the detonation phase
- X32dbg: to debug the malware during dynamic analysis

Binary information

The following are some characteristics of the worm obtained from a first analysis on the executable file:

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

In the .rsrc section i.e., the resources section, there is an entry called "CUSTOM", with two resources "AUTORUN.INF" and "RECOVER.REG" inside. The former will be used by the worm during the propagation phase, as described below. There is also a section with HTML code that the worm will display during the "Help" function (described below).

Malware analysis

In this section the three main stages of the worm will be described: installation, execution and spread.

Installation

The technique used by Disk Knight to install itself in the system is very simple: the function located at VA 0x408900 uses the CopyFileA function to copy the executable to the Windows directory (commonly C:\Windows\) with the name Knight.exe.

The path of the directory where the operating system resides is retrieved thanks to the SHGetFolderPathA function, passing the CSIDL_WINDOWS (0x24) parameter, in this way there is the certainty of obtaining the correct folder even in cases where the operating system is installed in an unconventional path.

The function responsible for starting Disk Knight is located at VA 0x00408958 and uses a call to ShellExecuteA with the following arguments:

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

Before terminating the current process execution, a file named recover.reg is dropped into the installation directory.

Execution

At this point Disk Knight is running from the binary installed in the Windows directory.

When the process is active, if you try to start the worm from a different folder than the one where the operating system is installed, the malware will try to install Knight.exe by overwriting it, but this attempt will fail resulting in a crash as the binary is currently in use. Regardless, Knight.exe file will still restart.

At this point, the worm checks if its process is already running with a very simple method, that is by calling a FindWindowA:

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

If it manages to get a handle, then the worm is already running and therefore the current process will terminate.

Once executed from the directory where Windows is installed, a function located at VA 0x00423E90 takes care of creating a registry key to allow the worm to persist on reboots of the operating system. The key is:

“HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Disk Knight” and contains the path to Knight.exe in the Windows directory.

ATTENTION: at this point, the Disk Knight control icon should appear in your system tray, but due to bugs in the code, the function in charge of creating it will never be reached by the call to 0x41BB90, making the application NOT controllable by the user! For more information and for fixing this bug, there is a dedicated section in this document.

To detect newly inserted USB devices, the worm uses a hook to intercept window procedure messages:

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

At address 0x41B9F0 there is a function that checks the type of message and if it is equal to WM_DEVICECHANGE, the function located at 0x421EB0 is called.

This function checks the inserted device collecting the necessary data and finally the function at 0x41B8D0 will be called.

Spread

When a new USB stick is inserted, the function located at VA 0x0041ED70 takes care of the infection, which is in turn called by the function located at VA 0x41B8D0.

This happens according to the following steps:

1a) A call is made to SetFileAttributesA on both Knight.exe and autorun.inf:

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

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

Where X is the letter of the drive being infected and 0x80 is the FILE_ATTRIBUTE_NORMAL attribute.

1b) If the operation is successful, and therefore the files are present on the usb stick, a call to DeleteFileA is made on both Knight.exe and autorun.inf, removing them:

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

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

Where X is the letter of the drive being infected.

2) The infection is carried out, which consists of installing the two files:

- Knight.exe is copied from the Windows directory to the USB device using the CopyFileA API:

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

Where X is the letter of the drive being infected.

- autorun.inf is extracted from the resource segment of Knight.exe, from the section called "CUSTOM" and placed on the root of the USB stick.

3) Both files are hidden using the SetFileAttributesA API:

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

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

Where X represents the letter of the drive being infected and 0x7 the following attributes: FILE_ATTRIBUTE_HIDDEN, FILE_ATTRIBUTE_READONLY, FILE_ATTRIBUTE_SYSTEM

The autorun.inf file causes that by accessing the USB device, regardless of the type of access selected, Disk Knigh will be started and consequently the computer will be infected.

IOCs

The following table shows the IoCs obtained from the analysis of the executable file and the behaviour of the worm during execution.

File	%windir%\Knight.exe	File Attr: Hidden, Read Only, System
File	%windir%\recover.reg	
File	%windir%\0.log	
File (on USB device)	X:\Knight.exe	File Attr: Hidden, Read Only, System
File (on USB device)	X:\autorun.inf	File Attr: Hidden, Read Only, System
Process	Disk Knight (Knight.exe/Administrator)	
Registry key	Path: HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run Key: Disk Knight Value: %windir%\Knight.exe	
Registry key	Path: HKLM\SOFTWARE\Knight\Settings Keys: drive, protectmode	
SHA256	d25c1d1423ed31b5436678318ca815092102e88d06a130481bc0728d14d74bb4	Knight.exe

YARA detection rule

This is a specific detection rule for the analysed Disk Knight sample:

```
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

}
```

Malware removal

The removal of Disk Knight must take place in two stages and necessarily in this order:

1) Removing the malware from infected PC
2) Malware removal from all infected USB drives

Before carrying out this procedure, it is mandatory to disable the automatic start via autorun.inf file, to avoid re-infection when removing the malware from the USB devices. It is therefore necessary to modify a key in the system registry, by executing the following command on an instance of cmd.exe:

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

Removing Disk Knight from infected PC

Run the following commands on an instance of 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

At this point Disk Knight will no longer be present on the PC and it is possible to continue the procedure to remove the malware from the infected USB sticks.

Removing Disk Knight from infected USB drives

Make sure you have disabled the automatic start via autorun file and run the following commands on an instance of cmd.exe, taking care to change the letter "X" with the one of the drives you intend to disinfect:

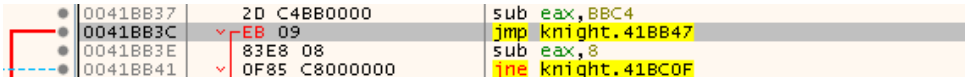
1) attrib -h -s -r "X:\autorun.inf"
2) del /f "X:\autorun.inf"
3) attrib -h -s -r "X:\Knight.exe"
4) del /f "X:\Knight.exe"

The USB device will now be clean.

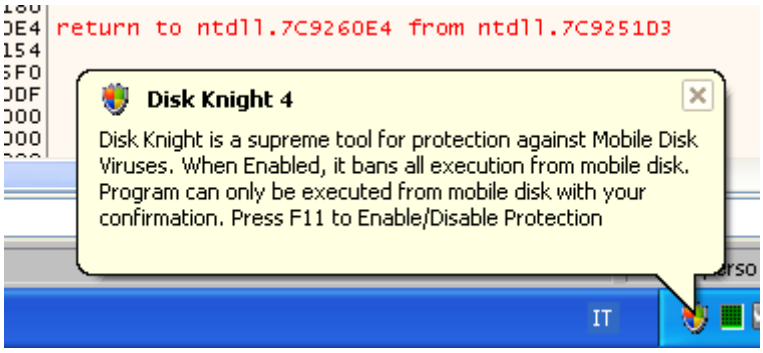
Extra: fixing the tray icon bug

During the analysis of the worm, I noticed that by forcing a conditional jump with the debugger it is possible to make the Disk Knight tray icon appear correctly and access its options.

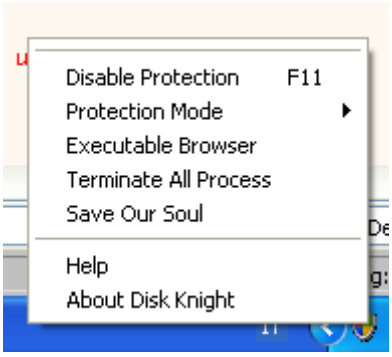
The patch consists in changing the instruction at VA 0x41BB3C from JE to JMP:



Right after resuming the execution, the tray icon will appear showing the following message:

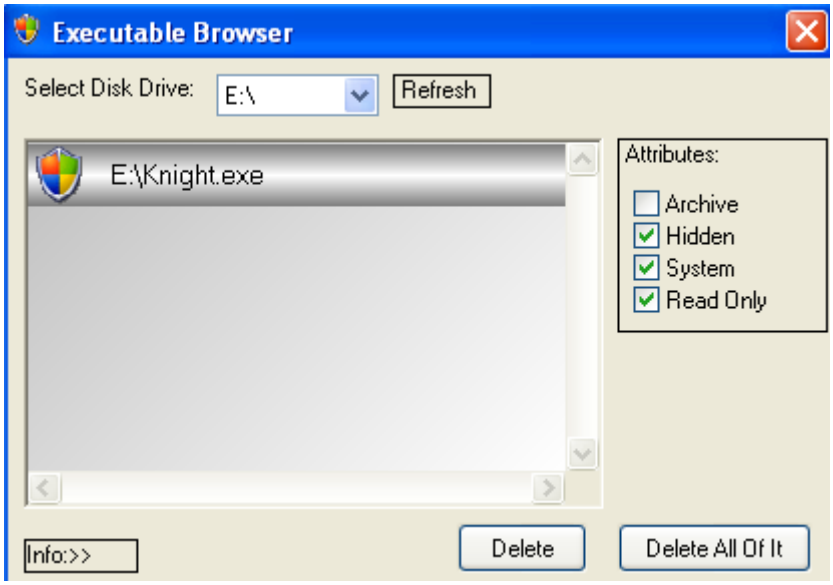


You can interact with the icon using the right-click of the mouse, to reveal the following menu:



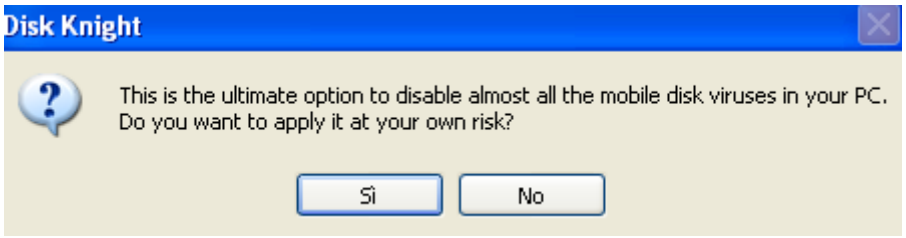
These options allow you to choose the "level of protection" offered by Disk Knight (for example, blocking executables from USB devices).

The Executable Browser feature allows you to obtain a list of executables present in the USB drive and to check the attributes associated with them:



The “Terminate All Process” function will terminate all active processes of the operating system.

By clicking on “Save Our Soul” the following message will be shown:

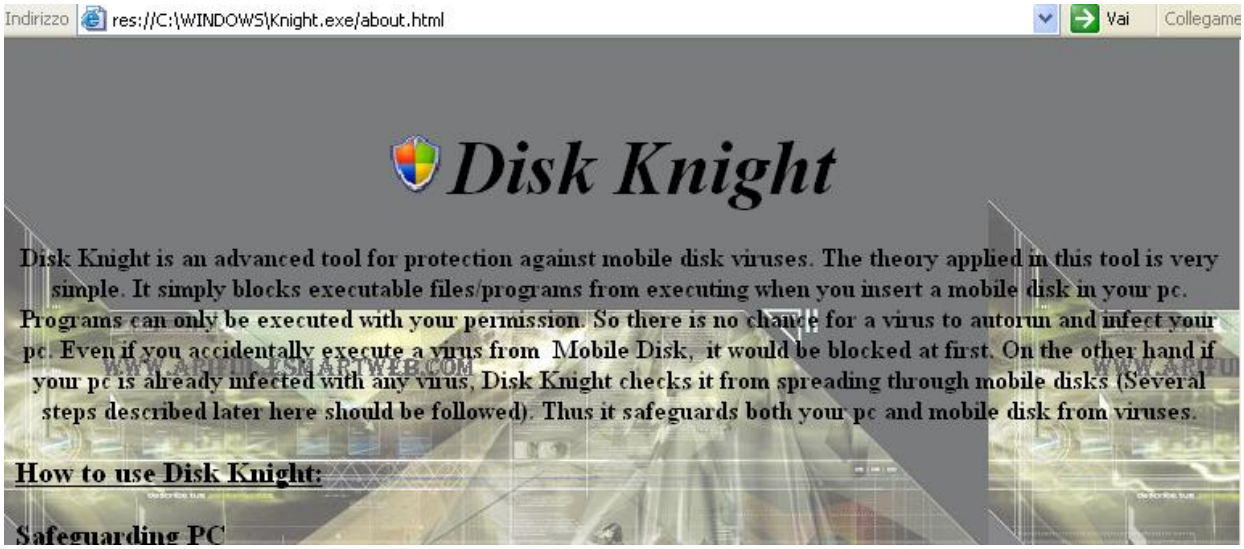


If you decide to continue, all active processes will be closed, and various registry changes will be made such as disabling programs to run automatically at startup. Finally, the computer will restart.

Interestingly, the “Disable Protection” option doesn't seem to work and although it can be clicked, Disk Knight will still propagate to all inserted USB devices.

At the bottom of the menu there are two options, Help and About Disk Knight.

Clicking on Help will open the system's default browser showing an html page extracted from the Disk Knight resource segment. This page explains the characteristics of the malware:



Clicking on About instead the following window will be presented:

Clicking OK will cause your browser to open the alleged author's web page. This page is offline and no working snapshots can be found using the Wayback Machine.

Conclusion

This analysis has highlighted how important it is to consider carefully the implementation choices when writing software and how important it is to make sure that there are no such serious bugs in your code before making a public release.

Fortunately, in this case there was no major damage or even loss of data, limiting the problem to the out-of-control diffusion of Disk Knight, especially on computers without and antivirus.

I hope you enjoyed this analysis and I sincerely thank the Italian (and the global one too :) malware analysis and reverse engineering scene.

For more technical papers, please visit my website:
https://www.lucadamico.dev