**ASSUMPTION**: ***Trojan.Worktrik (Variant) – Trik Bot***

1. **What is the general functionality of the sample?**
2. **What are the indicators that this sample is malicious?**

* winmgr.exe is a new running process
* share.exe file properties changed from 882 Bytes 🡪 32 KB
* update.exe disappeared after running

(SEE RISK REPORT)

1. **How does this sample interact with the local system (e.g., system DLLs, files, etc.)?**

* WS2\_32.dll
  + **WS2\_32.DLL** implements the [Winsock](https://en.wikipedia.org/wiki/Winsock) API, which provides TCP/IP networking functions and provides partial, broken compatibility with other network APIs.
* WININET.dll
  + The WinInet API is a convenience API which simplifies the interaction to higher level protocols such as HTTP, FTP and even GOPHER! To begin using the WinInet API to talk to a remote host you first need to call InternetOpen() followed by InternetConnect() or InternetOpenURL(). Once the internet has been opened and a connection established, to perform and HTTP request you can call the functions HttpOpenRequest() to make a request handle and HttpSendRequest()to send the request.  InternetReadFile() may then be called to read any response from the server. It can also be used to read data from an FTP session. The InternetWriteFile() function can also be used generically in place of any protocol specific function where data needs to be sent across the wire. The HTTP functions provide the programmer the ability to configure most of the options one would expect in an HTTP request header such as the User-Agent string. Not all of these values have to be specified though, and the system default will be used if none is specified. The malware programmer doesn't have as much flexibility to introduce subtle anomalies in the request structures though like they can with sockets, and left with default settings the malware's HTTP requests will look virtually identical to legitimate Internet Explorer traffic on the network.
* SHLWAPI.dll – MS Shell Lightweight Utility
  + SHLWAPI.dll is a library which contains functions for UNC and URL paths, registry entries, and color settings.
* URLMON.dll
  + The URL Monikers API provided by the DLL urlmon.dll provides yet another API for performing internet communications. In the back end it uses COM but I choose to list this as a separate category from the later discussion of COM because using this API is an abstraction away from the ugly, obscure-t0-reverse methods of direct COM interaction. The most popular function in the URLMon arsenal, from a malware perspective, is the one-punch knockout function URLDownloadToFile(). Rarely in the Win32 API does one function do so much work. You provide this function a URL (for any protocol IE understands), a filename and it uses COM to force Internet Explorer to download the resource to the specified filename. This is very popular with dropper malware which simply needs to download an EXE from a website and launch it. You might also run into the URLDownloadToCacheFile() function which will download a specified URL to the browser cache and return the name of the file it downloaded to.  URLOpenStream() and URLOpenPullStream() can be used to download a URL to a buffer in memory, but these functions are rarely used in malware.
* KERNEL32.dll
  + KERNEL32.DLL exposes to applications most of the Win32 base APIs, such as [memory management](https://en.wikipedia.org/wiki/Memory_management), [input/output (I/O)](https://en.wikipedia.org/wiki/Input/output) operations, [process](https://en.wikipedia.org/wiki/Process_(computing)) and [thread](https://en.wikipedia.org/wiki/Thread_(computing)) creation, and synchronization functions. Many of these are implemented within KERNEL32.DLL by calling corresponding functions in the [native API](https://en.wikipedia.org/wiki/Native_API), exposed by NTDLL.DLL.
* USER32.dll
  + **USER32.DLL** implements the Windows USER component that creates and manipulates the standard elements of the Windows user interface, such as the desktop, windows, and menus. It thus enables programs to implement a [graphical user interface (GUI)](https://en.wikipedia.org/wiki/Graphical_user_interface) that matches the Windows look and feel. Programs call functions from Windows USER to perform operations such as creating and managing windows, receiving window messages (which are mostly user input such as mouse and keyboard events, but also notifications from the operating system), displaying text in a window, and displaying message boxes.
  + Many of the functions in USER32.DLL call upon GDI functions exported by GDI32.DLL to do the actual rendering of the various elements of the user interface. Some types of programs will also call GDI functions directly to perform lower-level drawing operations within a window previously created via USER32 functions.
* ADVAPI32.dll
  + **ADVAPI32.DLL** provides security calls and functions for manipulating the registry.
* SHELL32.dll – MS Windows Shell Library
  + shell32.dll is a library which contains Windows Shell API functions, which are used when opening web pages and files.
* OLE32.dll – Object Linking and Embedding (MS [Propietary])
  + OLE allows an editing application to export part of a document to another editing [application](https://en.wikipedia.org/wiki/Software_application) and then import it with additional content. For example, a [desktop publishing](https://en.wikipedia.org/wiki/Desktop_publishing)system might send some text to a [word processor](https://en.wikipedia.org/wiki/Word_processor) or a picture to a [bitmap editor](https://en.wikipedia.org/wiki/Bitmap_editor) using OLE. The main benefit of OLE is to add different kinds of data to a document from different applications, like a text editor and an image editor. This creates a [Compound File Binary Format](https://en.wikipedia.org/wiki/Compound_File_Binary_Format) document and a master file to which the document makes reference. Changes to data in the master file immediately affect the document that references it. This is called "linking" (instead of "embedding"). Its primary use is for managing [Compound File Binary Formats](https://en.wikipedia.org/wiki/Compound_File_Binary_Format), but it is also used for transferring data between different [applications](https://en.wikipedia.org/wiki/Application_software) using [drag and drop](https://en.wikipedia.org/wiki/Drag_and_drop) and [clipboard](https://en.wikipedia.org/wiki/Clipboard_(software)) operations.

1. **What files and registry keys does this sample create, modify and access?**

The Trojan creates the following registry entries so that it runs every time Windows starts:

* HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Run\"Microsoft Windows Manager" = "%Windir%\M-505045024322940506830284960384065\winmgr.exe"
* HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\"Microsoft Windows Manager" = "%Windir%\M-505045024322940506830284960384065\winmgr.exe"
* HKEY\_USERS\S-1-5-21-2445195769-2503366633-525057035-500\Software\Microsoft\Windows\CurrentVersion\Run\"Microsoft Windows Manager" = "%Windir%\M-505045024322940506830284960384065\winmgr.exe"

1. **What is the network behavior (including hosts, domains and IP addresses accessed)?**
2. **What are the time and local system dependent features?**
3. **What is method and means by which this sample communicates to the external environment?**
4. **What is the original infection vector and propagation methodology?**

* Trojan Worm

1. **What use does this sample make of encryption for storage, communication**?
2. **What self-modifying or encrypted code does this sample employ?**

* share.exe file properties changed from 882 Bytes 🡪 32 KB
* update.exe disappeared after running

1. What ancillary information is available concerning the development of this sample (compiler type, country of origin, author names/handles, etc.)

* This malware imports MSVCRT.dll
  + **MSVCRT.DLL** is the [C standard library](https://en.wikipedia.org/wiki/C_standard_library) for the [Visual C++ (MSVC)](https://en.wikipedia.org/wiki/Visual_C%2B%2B) compiler from version 4.2 to 6.0. It provides programs compiled by these versions of MSVC with most of the standard C library functions. These include string manipulation, memory allocation, C-style input/output calls, and others. **MSVCP\*.DLL** is the corresponding C++ library.