Engineering Development Group

(U) 00AG9603
User's Manual

Rev. 1.0

Change Log

Doc Rev	Document Date	Revision By	Change Description	Reference	Approval Date

Table of contents

1. (U) Introduction

(TS) 00AG9603 is an implant comprised of 5 components: Solartime, Wolfcreek, Keystone, BadMFS, and the Windows Transitory File system.

Solartime modifies the partition boot sector to load some kernel code. That kernel code then modifies the Windows boot process so that when Windows loads boot time device drivers, an implant device driver can be loaded. The implant driver and Solartime boot code (aside from the partition boot sector modifications) are kept in a small user-specified file on disk. This file is encrypted.

Wolfcreek is the kernel code that Solartime executes. Wolfcreek is a self-loading driver, that once executed, can load other drivers and user-mode applications.

Keystone is responsible for starting user applications. Any application started by MW is done without the implant ever being dropped to the file system. In other words, a process is created and the implant is loaded directly into memory. Currently all processes will be created as svchost. When viewed in task manager (or another process viewing tool) all properties of the process will be consistent with a real instance of svchost.exe including image path and parent process. Furthermore, since the implant code never touches the file system (aside from the possibility of paging) there is very little forensic evidence that the process was ever ran.

BadMFS is a covert file system that is created at the end of the active partition. It is used to store all drivers and implants that Wolfcreek will start. All files are both encrypted and obfuscated to avoid string or PE header scanning.

The Windows Transitory File system is the new method of installing 00AG9603. Rather than lay independent components on disk, the system allows an operator to create transitory files for specific actions including installation, adding files to 00AG9603, removing files from 00AG9603, etc. Transitory files are added to the UserInstallApp (both the .exe or .dll versions).

2. (S) Implant Forensics

(S) 00AG9603 has a small forensic footprint.

Table : (S) 00AG9603 Installer MD5 Signature

00AG9603 Installer	MD5 Sum
UserInstallApp.exe (default name[1])	
UserInstallApp.dll (default name)	
tdbsip.sys (default name)	
xqlmi.dat (default name, pack	

1 (S) The user may rename the 00AG9603 Installer as necessary without impact to 00AG9603's operation.

file)	
wtpack.exe	

Table : (S) 00AG9603 Footprint Revision

Forensic Entry	Purpose	Changeable
File: encrypted container file	Holds boot code	Yes
Boot Sector: partition boot sector modification	Holds boot code	No
Registry key: HKLM\System\CurrentControlSet\Control\Windows\ SystemLookup	Holds BadMFS parameters	No
Covert Store: BadMFS will create an encrypted covert file system in the file specified in the zf file. Alternatively, the covert file system can be placed at the end of the active partition.	Holds driver and user-mode implants	No

3. (S) Implant Operation

3.1 (U) 00AG9603 Installer

(S) 00AG9603 now comes with two installer versions, both an executable and a fire-andcollect .dll installer. In order to install 00AG9603 on a target system, an operator must first create an "inst" transitory file via the wtpack executable. This transitory file must be finalized to the installation application of the operator's choice.

(S) Once an "inst" transitory has been finalized, the installation method then depends on which installer the operator has chosen to use. For the .exe installer, the installer should merely be run on the target machine with administrative privileges. The fire-and-collect installer should be loaded into an appropriate target process (i.e. one with administrative privileges).

(S) 00AG9603 requires administrative privileges to use the either install mechanism.

3.1.1 (U) Wtpack usage

(S) Both 00AG9603 install mechanisms lack command line options. Instead, all options are built through the creation of transitory files via the wtpack executable. Below are a list of wtpack.exe commands and options associated with those commands:

(U) Wtpack commands

Table : (S) wtpack commands

Commands	Options	Flags	Purpose
new	(inst | list | del | add | get | uninst) transitory_file_name	none	Creates a new transitory file. This transitory file can be for installation, listing files in the covert store, deleting files in the covert store, adding files to the covert store, getting the log file from the covert store, or uninstallation.
update	transitory_file_name {flags}		Updates a transitory file with additional information required to finalize it.
		-bp (Path on Target)	Specify the location of the BadMFS covert store partition on target. If this option is "PhysicalDrive", BadMFS will be used in the slack space at the end of the drive. N.B. certain drives do not have such slack space at the end.
		-wd (file name)	Specify the location of the wolfcreek driver to add to the transitory file.
		-cp (Path on Target)	Specify the location of the solartime container path that will be created on target. N.B. this path cannot

			have a drive letter. It also must be under \Windows\ folder.
		-st (file name)	The solartime pack file (xqlmi.dat) to be added to the transitory file.
		-f (file name)	File name to be deleted from the covert store.
		-bin (file name) {sub-options}	Add the specified binary (.exe, .dll, or .sys) to the transitory file.
print	transitory_file_name	none	Prints a summary of the contents of the transitory file.
finalize	transitory_file_name {user install application}	none	Adds the specified transitory file as a resource to the targeted user install application.

3.2 (S) Installing BadMFS

(S) 00AG9603 uses the BadMFS covert file system to store many of the implants and data required to run. BadMFS has two options for installation, one using a specified file and the other using slack space at the end of a hard drive. Which option is used depends on what is specified under the –bp flag. To use the specified file option, an operator must give a **complete path** to a file that will hold the covert store on disk. To use the slack space option, the operator must specify "PhysicalDrive". Note that many drives do not have such slack space, therefore installation under this method is not guaranteed.

The maximum filesystem size for BadMFS is 200 MB.

Once BadMFS is installed using the 00AG9603 installer, the location of BadMFS must be provided to any transitory file created. The location is specified with the –bp flag when building a transitory file.

3.3 (S) Installing 00AG9603 (inst transitory file)

(S) To install 00AG9603, you must create and finalize an "inst" transitory file. This transitory file must include the BadMFS path on target (-bp), the wolfcreek driver (-wd), the solartime container path that will be created on target (-cp), and the solartime pack file (-st). Note that the container path (-cp) must not contain a drive letter, and it must be placed under \Windows folder (**i.e. the path must be \Windows\...**).

Example creation of inst transitory file:

wtpack.exe new inst "inst_transitory_file"
wtpack.exe update "inst_transitory_file" –bp "BadMFS location"
wtpack.exe update "inst_transitory_file" –wd "wolfcreek driver"
wtpack.exe update "inst_transitory_file" –cp "solartime container file (created on target)"
wtpack.exe update "inst_transitory_file" –st "solartime pack file"

3.4 (S) Adding a File To The Covert File System (add transitory file)

(S) To add a file to the BadMFS covert file system, you must create an "add" transitory file. The file must be finalized to the installation binary, which will then be run on target. Whenever files are added to the covert file system, a 3 digit number is appended to the beginning of the file name to encode information about the file for internal 00AG9603 use. In the cases of .exe's, .dll's, and .sys files, an additional file is also created (with a similar name) that contains the command line parameters to be passed to the .exe. To delete an .exe or .sys file, both of the files matching the implant name should be deleted. N.B. Multiple files can be added to an "add" transitory file.

Example creation of add transitory file:

wtpack.exe new add "add_transitory_file"

wtpack.exe update "add_transitory_file" –bp "BadMFS location"

wtpack.exe update "add_transitory_file" –bin "file to add" {sub-options}

3.4.1 *(S) Sub-options for -bin*

(S) There are several sub-options for the –bin option to add a binary file to BadMFS. The following list contains all options available. N.B. While most of these are optional, some are required depending on the type of binary being added to the covert store.

Sub-Option	Potential Values	Notes
-execp	persistent execution interval in minutes	
-execd	delay for initial execution in seconds	
-execa	absolute execution time	Must be in UTC. Format of YYYY:MM:DD:HH:MM:SS
-inject	target process for .dll injection	For .DLL only. Must be specified in that case.
-dtype	type for drivers (sys, auto, boot)	For .SYS only. Default value is 'auto'
-cmdline	command line options	Must be the last option. All contents after this flag will be added to the command line. Command lines for .dlls must be of the following format: "env_var=variable value". This is to support the NOD persistence spec.

3.4.2 *(S) Limitations for binary files*

- No executable with Graphical User Interfaces (GUI) can be run. This is because the parent process is always services.exe. Services executes in a different window station than the logged on user, so there is no way for it spawn the GUI. • Any binary must match the architecture it is being run on (i.e. a 64 bit version of Bulldozer on a 64 bit version of Windows). This also means that you cannot run a 32 bit executable on 64 bit Windows. Note: If a mismatched binary (i.e. 32 bit executable on a 64 bit OS) is run, it will fail gracefully.
- The application cannot interact with the console (such as cmd.exe).
- The application cannot be compiled to user side-by-side assemblies. This is a feature in Windows that tries to eliminate "dll hell" by storing what specific versions of Windows dll's are required in a manifest which is compiled into the binary. When Windows starts the executable, it pulls those specific versions of the dll's from a dll database on the machine.
- The application cannot require that a specific user dll be loaded with it. If this is a requirement, the application should pack the dll in a resource and extract it at runtime.
- No .dll can be added to the covert store on Windows XP. This is due to .dll injection not being currently supported on Windows XP.

3.5 (S) Deleting a file from the covert file system (del transitory file)

(S) To delete an executable from the covert file system, one must create and finalize a "del" transitory file. The file name specified must match the file name in the covert file system exactly. Note, that to delete an executable, you might also have to delete its command line file (see the "add" transitory file section).
N.B. You may delete multiple files in one "del" transitory file.

Example of creating a "del" transitory file:

wtpack.exe new del "del_transitory_file"
wtpack.exe update "del_transitory_file" –bp "BadMFS location"
wtpack.exe update "del_transitory_file" –f "file for deletion"

3.6 (S) Listing the contents of the covert file system (list transitory file)

(S) To list the names of all files in the covert file system, one must create and finalize a "list" transitory file.

Example of creating a "list" transitory file:

wtpack.exe new list "list_transitory_file"
wtpack.exe update "list_transitory_file" –bp "BadMFS location"

3.7 (S) Getting the log file from covert store (get transitory file)

(S) 00AG9603 now includes a log file. This log file records basic information about the successful execution of an implant located in the covert store. To retrieve the contents of the log file, one must create and finalize a "get" transitory file. The contents of the log file will be printed to stdout (if using the .exe installer) or written back to the text pipe (if using the fire-and-collect .dll installer).

Example of creating a "get" transitory file:

wtpack.exe new get "get_transitory_file"

wtpack.exe update "get_transitory_file" –bp "BadMFS location"

3.8 (S) Uninstalling 00AG9603 (uninst transitory file)

(S) To uninstall 00AG9603, you must create and finalize an "uninst" transitory file. The uninstallation process will remove wolfcreek and solartime, and will also delete the covert store. N.B. After performing an uninstall, you will need to wait for a reboot before reinstalling.

Example of creating an "uninst" transitory file:

wtpack.exe new uninst "uninst_transitory_file"

wtpack.exe update "uninst_transitory_file" –bp "BadMFS location"

3.9 (S) Finalizing a transitory file

(S) To finalize a transitory file, you must run the "finalize" command on the transitory file you wish to use. You must also specify the location of the 00AG9603 installer executable you wish to use. The "finalize" command will place the transitory file you have selected as a resource inside the installer (either the .exe or .dll version).

N.B. Only one transitory file can be placed inside an installer at a time. This means only one "action" (inst, add, del, list, get, uninst) may be performed at any given time.

4. (U) Operational Notes

4.1 (S) Using 00AG9603 To Start Drivers

(S) 00AG9603 is capable of starting kernel mode drivers. The driver must first be added to the covert file system by using the "add" transitory file. See the section on adding file to the covert store for more information. On reboot, the any files with a .sys extension will be executed. There are some limitations to driver execution:

- Drivers will not have Structured Exception Handling (SEH) available even if the driver was built with SEH enabled. This will be added in a future version of 00AG9603.
- Once started, drivers cannot be unloaded by 00AG9603. However, drivers can terminate execution themselves (exit).

- If a driver start type of boot start (boot) is specified, the driver will be started at the same time as the system start drivers (sys). This is a limitation of the covert file system and will be fixed in a future version.

4.2 (S) Using 00AG9603 To Start Executables

(S) 00AG9603 is capable of starting executables. The executable must first be added to the covert file system by using "add" transitory file. See the section on adding files to the covert file system for more information. There are some limitations to starting executables:

- When viewing an 00AG9603-started process in Task Manager or another process viewer, the image name will be svchost.exe. It has been determined that svchost is the best (most reliable) process to use for process execution.
- When viewing an 00AG9603-started process in Task Manager or another process viewer, the command line string will display whatever the user passed as the command line when the file was added to the covert file system. If no command line string is specified, then 00AG9603 will use a default string ("c:\windows\system32\svchost.exe -k WerSvcGroup"). It is recommended, if possible, to not specify a command line due to its visibility in process viewing applications.

5. (S) OS Compatibility List

(S) 00AG9603 is compatible with the following 32-bit systems (latest service pack): XP, Windows 7.
(S) 00AG9603 is compatible with the following 64-bit systems (latest service pack): Server 2008 R2, Win7.

6. (U) Known Issues

(U) While 00AG9603 attempts to provide a robust environment for the user, there are some limitations that a user should be aware of prior to use. Table lists those issues that are currently known to the 00AG9603 development team.

Table : (S) Known Issues

Issue	Cause	Remediation
Windows XP does not currently support .dll persistence.	Windows XP and below use a different mechanism for creating threads, which does not allow the use of Keystone's .dll injection technique.	.dll persistence on XP will be supported with a later version of 00AG9603
Solartime does a heuristic check of the operating system at boot time to determine if it is possible to patch it. It is possible that this heuristic check will succeed, yet the OS has changed in a manner that would cause a crash if patched.	The heuristic algorithm is imperfect and can still have false positives.	Solartime has a more restrictive setting that will only allow the patch to proceed if the OS has not changed. The downside is, that if a new service pack or hotfix is applied, Solartime will not launch on bootup.

Issue	Cause	Remediation
SEH doesn't work in drivers started by 00AG9603.	The SEH environment is not configured correctly during driver load.	This will be fixed in a future version of 00AG9603.
When viewing an 00AG9603-started process in Task Manager or another process viewer, the command line string will display whatever the user passed as the command line when the file was added to the covert file system.	Process viewers display whatever command line was passed to the executable.	Executables that are started by 00AG9603 should not use a command line if possible. This will allow 00AG9603 to display a svchost.exe appropriate command line, allowing it to blend in with everything else.
If the user chose to install BadMFS the end of the logical volume and if there is insufficient space at the end of the logical volume, the covert file system won't install. This is frequently the case with VmWare guest OS'.This is usually the case when install returns error code 617. NOTE: this is only if “PhysicalDrive” is specified in the –bp option to indicate that the covert file system is to be installed in the drive slack space. This does not apply to a file-based covert file system.	The covert file system needs a minimum of 2mb at the end of the volume to install correctly.	Shrink the volume using 3rd at party disk tools. The covert file system needs a minimum of 2mb to install correctly.
If the container file is deleted, but 00AG9603 has not been uninstalled, it will continue to work on reboot until the disk clusters that the container file occupies are overwritten by the file system. If this happens, the integrity check of the container file will fail and 00AG9603 will allow the boot process to continue as normal.	The 00AG9603 boot process references the location of the container file based on its file ID, not the file name. Because of this approach, it won't recognize when the container has been deleted.	None.
If Windows is installed on a non-standard drive (i.e. D:), processes started by 00AG9603 with a default command line will have a svchost.exe path of "c:\windows\system32\svchost.exe" . This would be inconsistent with the actual svchost.exe path on the system. NOTE: this only applies to applications started with no parameters.	00AG9603 does not dynamically determine the path of svchost.exe.	A future version 00AG9603 will dynamically determine svhost.exe's path.
If a driver start type of boot start (boot) is specified, the driver will be started at the same time as the system start drivers (sys).	This is a limitation of the covert system.	This will be fixed in a future version.

Issue	Cause	Remediation
If an existing file (not badmfs) is specified in the –bp option, it will used and modified by badmfs.	BadMFS does not check to see if the file specified is a valid badmfs archive.	In the future, BadMFS can check to see if the file is actually a be valid BadMFS archive.Until then, care must be taken when specifying the file name.
If an application that is started by 00AG9603 crashes, it is possible that a dialog box will pop up on the target machine stating that svchost.exe has crashed.	All user implants look like svchost.exe.	Fix the bug in the crashing implant.
An application that uses networking is failing when started on reboot.	00AG9603 starts executables very early and sometimes the network stack might not be fully up and available.	Use the –execd switch to delay the application executing for x number of seconds. This should allow the network stack time to become available.
When executing GUI programs with 00AG9603, the process might start, the GUI will not be visible.	This is because the host process used by Keystone is svchost.exe which is not capable of displaying GUI applications.	This might be fixed in a future version by allowing the user but more control over what host process is used.

7. (U) Installer Error Conditions

(U) Table lists the error codes that the 00AG9603 Installer produces.

Table (S) 00AG9603 Installer Error Codes

Error Code	Error Description
0	No error, everything was successful.
1	General failure code
101	Container not found (an update error).
102	Container found (on install - previous installation).
103	Container rename failure.
104	Container path failure.
105	Container Object ID failure.
106	Container pack failure.
107	Container unpack failure.
108	Container write failure.
109	Container read failure.
110	Container clean failure.
111	Container get path failure.
200	Dollar boot change failure.
201	Multiboot dollar boot.
202	Dollar boot write failure.
203	Dollar boot read failure.
257	Invalid transitory file parameter
258	Memory Allocation Error
260	Required item not found in transitory file
261	Transitory file import error
264	Failure to decompress transitory file
265	De obfuscation error in transitory file
301	Bad install package.
401	Existing install test failure.
501	EFI found failure.
502	Unsupported OS.
503	Unsupported file system.
504	Resource Error with Transitory File
600	Invalid parameters.
601	Path parse failure.
603	Driver install failure.
604	CPUID get failure.
605	Version get failure.
606	ACPI data get failure.
607	Covert store install failure.
608	Covert store uninstall failure.
609	Covert store add failure.
610	Covert store delete failure.
611	Covert store list failure.
612	Covert store get failure.

Error Code	Error Description
614	Covert store read file failure.
615	Covert store write file failure.
616	Invalid file path.
617	File start failure.
618	File path construct fail.
619	File open fail.
620	File get file size fail.
621	File read fail.
622	File write fail.
623	Memory alloc fail.
624	Construct command line fail.
625	Convert date time fail.
626	String operation fail.
627	Exception caught fail.
628	File mapping fail.
629	File too large fail.
630	Covert store scramble fail.
632	Entropy initialization fail.
633	Key initialization fail.
634	IV initialization fail.
635	Log file does not exist.
Cannot inject .dll in Windows XP	