## Cryogenically Frozen Malware

vx-underground collection // by smelly\_vx and Ethereal



## Frozen Malcode:

For quite sometime now we've been sitting on a lesser known malware technique I've (smelly) titled *Cryogenically Frozen Malcode*. This term derives from the fact that the malicious binary is in a long-term frozen state and has a low-likelihood of execution (or *unthawing* and/or *resurrection*). This technique is not ideal - however under very specific environments or scenarios it may be viable. This technique is made possible by abusing registry components tied to specific properties within the Windows Installer API.

The <u>Windows Installer API's properties</u>, which derive from the Windows Installer binary, allow a product to configure it's uninstall properties which will be visible from the Control Panel. When a user selects uninstall (or, depending on the binary configurations, *modify settings*) it will refer to the property present within the registry titled *Uninstall*. (Un)fortunately, the documentation present on MSDN is slightly misleading - the documentation states that the binary uninstall path is located in <u>HKEY\_LOCAL\_MACHINE</u> i.e.

## HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall

However, this is not entirely true. In some scenarios applications will store uninstaller properties in the user-specific registry key <u>HKEY\_CURRENT\_USER</u> which does not require administrative and/or elevated privileges to write to i.e.

## HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Uninstall



The image above is a snippet of the file listing present within the Uninstall Key. Each entry contains attributes which the Control Panel reads when a user navigates through them.



In this particular paper our code demonstrates enumerating the Uninstall Key, locates a specified application, hijacks its UninstallString key and replaces it with a powershell command line which requests our malicious binary be run as admin. Ideally, a non-educated user would attempt to uninstall the application and allow UAC to execute the binary being presented. Why would a user not trust the Control Panel?

dows/CurrentVersion/Uninstall/wire			
1	Name	Туре	Data
	(Default)	REG_SZ	(value not set)
	Displaylcon	REG_SZ	C:\Users\\AppData\Local\wire\app.ico
	ab DisplayName	REG_SZ	Wire
	and DisplayVersion	REG_SZ	3.19.3731
	100 EstimatedSize	REG_DWORD	0x00010a56 (68182)
	anstallDate 🛃	REG_SZ	20200922
	and InstallLocation	REG_SZ	C:\Userstand:AppData\Local\wire
	100 Language	REG_DWORD	0x00000409 (1033)
	8 NoModify	REG_DWORD	0x00000001 (1)
	100 NoRepair	REG_DWORD	0x00000001 (1)
	and Publisher	REG_SZ	Wire
	and QuietUninstallString	REG_SZ	"C:\Users\}AppData\Local\wire\Update.exe"uninstall -s
	and UninstallString	REG_SZ	powershell.exe start-process C:\Users\lsource\repos\Persistence0x2\x64\Debug\Persistence0x2.exe -verb runas
	and URLUpdateInfo	REG_SZ	_
1			

The image above shows the modified UninstallString path.

powershell.exe start-process {binary-path.exe} -verb runas