Writeup

Kevin Turkington

I. Rootkits Overview

In Aditya Kapoor's lectures he describes rootKits in a general overview and the difficulties associated with removing post infection. Rootkits are typically installed at the driver level, the delivery of the payload is usually done in may different ways specifically

- Spoofing driver signing checks
- Modifying windows boot path (MBR)
- General kernel exploits within the windows operating system
- Stealing valid digital signatures

Each allows the payload to look authentic or be undetected by the user,as well as allow access into kernel level modules within the system. From there the rootkit can do a number of things to protect itself from being patched/removed and listen in on user actions. For example in most cases rootkits work similar to man in the middle attacks done on the internet to listen in on users traffic on a network. To begin the rootkit will target a specific process loaded into memory and running and replace its "Hooks" (function pointers) to other windows APIs within the Systems Service Descriptor Table (redirecting system calls through a middle man module and returning modified or recorded data back to the intended module.)

II. Removing Rootkits

Once installed removing a rootkit can be extremely difficult for anti-virus engineers. Because rootkits have been known to spin up child threads to do one or multiple things such as:

- File foraging
- Memory foraging
- Attack AntiViruses
- Un-trusting or trusting different programs
- Remove existing dependencies on files
- Disassociated memory from a file
- Protect malware source code

Rootkits when installed and has access to kernel level process and files they have the ability to show, hide and modify content within files or files themselves from the user. Which can aid in obfuscating rootkit source files from the user. Rootkits have also been known to prevent specific lists of programs from being opened by the user, the most obvious purpose for this is to prevent anti-virus engineers from opening tools then

may need to investigate malware. Furthermore Rootkits can spin up watch dog threads, which can serve as a protection from patching Hooks to Windows kernel APIs within the system calls. Watch dog threads help replace patched drivers with their infected counterparts giving the illusion of fixing and creating an inert rootkit.

III. Bootkits Overview

While rootkits take up a more stealth approach to infection of a target, bootkits are vastly different. Bootkits take a goal is to edit the Master Boot Record (MBR) or Volume Boot Record (VBR) of a targets hard disk. Inorder to load rootkit malicious like functionality before the operating system is booted on the target machine. This allows the bootkit to run alongside windows without running the risk of being detected by antivirus softwares, because the malicous source code is run outside of the Operating Systems file structure. A great example of bootkit like functionality is TrueCrypt a ”pre-boot” hard drive encryption tool, it serves as an in between when booting an operating system keeping the system secure and only recoverable or bootable after proper authentication.