

Evaluating “Ring -3” Rootkits

Patrick Stewin*

*Security in Telecommunications

Technische Universität Berlin, D-10587 Berlin, Germany

patrickx{at}sec.t-labs.tu-berlin.de

In 2009, security researchers discovered a new, very powerful rootkit environment on x86 platforms [1]. That environment is based on *Intel’s Active Management Technology (iAMT)* [2], which is completely isolated from the host. One part of iAMT is implemented as an embedded μ -controller in the platform’s memory controller hub. That μ -controller is called *Manageability Engine (ME)* and includes a processor (ARCTangent-A4), read-only memory (ROM), static random access memory (SRAM) and direct memory access (DMA) engines. Furthermore, iAMT provides an isolated network channel (out-of-band (OOB) communication). To illustrate the power of the stealth environment, [1] called the iAMT environment in conjunction with rootkits “ring -3”, following the x86 ring protection model. For our evaluation we implemented a prototype in form of a USB keyboard keystroke logger [3].¹ Since we were unable to get an Intel developer board providing the “ring -3” environment, we had to use the exploit discovered by [1] to infiltrate our target platform. We monitor the keyboard buffer of the Linux based target platform via DMA. To find the physical address of the keyboard buffer we apply a search algorithm, that finds the USB product string and follows some pointers to the structure containing the buffer address. To exfiltrate captured keystrokes our prototype uses iAMT’s OOB communication capabilities.

[1] discussed countermeasures against “ring -3” rootkits, but they also provide approaches to defeat such countermeasures. Furthermore, it is doubtful if all the proposed countermeasures can be applied in practice.² The goal of our evaluation is to find a reliable detection mechanism for “ring -3” rootkits. We assume that we can provoke delays when accessing the same resources as our prototype. For example, our prototype has to scan the host memory to find certain data structures and it also has use the network interface card to send keystroke codes. Another possibility is to initiate various DMA transfers using multiple devices. Only one device can be the bus master at a certain point in time. The next step is to design an experimental set-up that allows the measurement of delays and finally derive a reliable detection mechanism for “ring -3” rootkits.

References

- [1] A. Tereshkin and R. Wojtczuk, “Introducing Ring -3 Rootkits,” Black Hat USA, Jul. 2009. [Online]. Available: <http://www.blackhat.com/presentations/bh-usa-09/TERESHKIN/BHUSA09-Tereshkin-Ring3Rootkit-SLIDES.pdf>
- [2] Kumar, A., Goel, P., Saint-Hilaire, Y.: *Active Platform Management Demystified - Unleashing the power of Intel vPro Technology*. Intel Press (2009)
- [3] Stewin, P., Seifert, J.-P.: “In God We Trust All Others We Monitor” [Extended Abstract]. In: *CCS ’10: Proceedings of the 17th ACM Conference on Computer and Communications Security*. ACM, p.639â641. (2010). Online verfügbar: http://portal.acm.org/ft_gateway.cfm?id=1866381&type=pdf&CFID=6743120&CFTOKEN=21999560

¹The rootkit provided by [1] is not suited for an evaluation: it reveals itself, does not work permanently, is unable to read from host memory and does not provide any network capabilities.

²Note: Intel fixed the bug responsible for the exploit discovered by [1], i.e., the known exploit cannot be used to infiltrate a patched x86 based target platform anymore.