VM RAM Control: What You Better to Know For Effective Cybersecurity

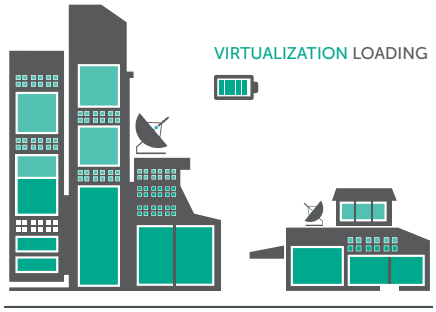
VDI (Virtual Desktop Infrastructure) is now standard in many organizations. Standalone physical endpoints are replaced with virtual machines which are indistinguishable to the user from physical PCs. The twin advantages of this approach from a business perspective are cost reduction and the simplification of the systems administration.

It's also easier to create a clean workstation from scratch with VDI, and to provide employees with machine access from any mobile device. Enterprises actively use all these facilities. Let's consider how to build in effective VDI security.

The reasons why business has embraced VDI are clear, but employees can be resistant to change. Accustomed to familiar working practices, they need to be able to work with VDI just the same way as with locally installed operating systems. VDI must be user-transparent, regardless of the internet browser used, USB use for data copying or the application running. IT departments provide this transparency with virtual USB-ports, familiar browsers, etc. And, incidentally, opportunities for familiar malware threats as well.

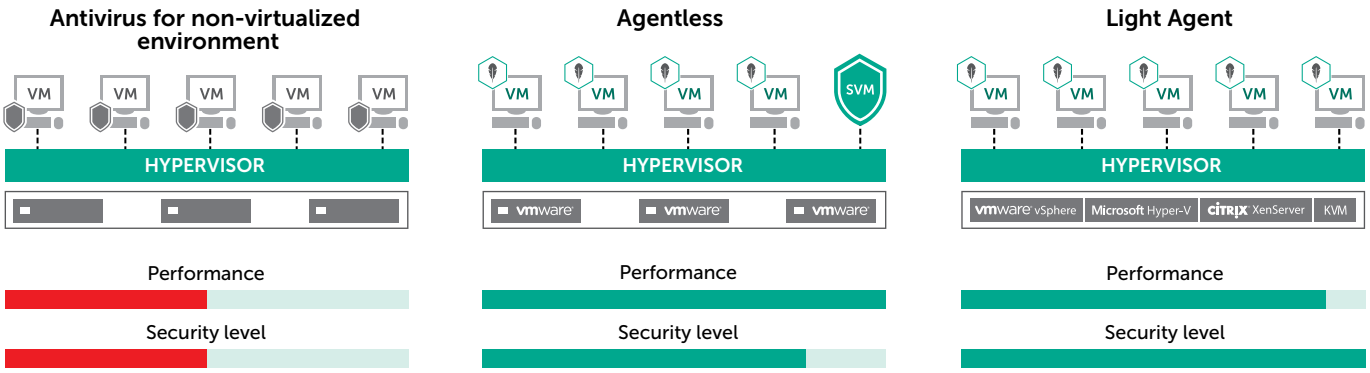
Why VDI infrastructure needs securing

VDI is widespread, but virtualization does not do away with the need for security. Many of the same vulnerabilities exist as for physical machines – there have been numerous cases of virtual infrastructure infection, as well as instances of malicious code using hypervisor vulnerabilities to move from the "sandbox" directly onto the host. So multi-layered defenses for your virtualized working space are a necessity. More information about the about the need for VDI security is available at: https://securelist.com/blog/securitypolicies/ 75279/vdi-non-virtual-problems-of-virtual-desktop-security-and-how-tosolve- them-for-real/.



Different approaches to protection and the importance of ram monitoring

The nature of virtual environments means that best-practice security approaches differ from those applicable to traditional physical infrastructures. It's worth briefly explaining why. Firstly, virtual hosts are typically lowerpowered than their physical counterparts, so resource-hungry products designed for physical endpoints can result in slower overall response times. There are also issues like "activity storms", where every virtual machine attempts the same action, e.g. updating its security database, simultaneously.



VDI solutions developers are always seeking new ways to minimize the load on virtual hosts. Security functions are centralized where possible, and duplication is avoided. In the ideal scenario from a systems performance perspective, nothing at all is installed on the secured host itself. A separate standalone host is responsible for securing the entire virtual infrastructure, and there is no software agent on the secured host. this approach does, however, have limitations, particularly in terms of memory (RAM) access on the secured host.

Even if RAM access is possible (as it can be under some conditions), information available to the security product is limited to memory content rather than processing activity. As described below, deep dynamic behavioral analysis, including that of processes in RAM, is essential to full VDI protection.

The "middle ground" between installing a heavy agent onto each host and deploying agentless protection with all its security shortcoming is using a light agent with sufficient functionality to secure the virtual host. Why is access to the host's memory so vital? The appearance of bodiless malware that exists only in memory requires it. One famous sample of this tactic is the using of legitimate Mimikatz software to launch attacks. The infection introduced works purely in the memory, seeking out user passwords hashes. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam

Why memory control is necessary, but not sufficient

The information security industry has known about such tactics for some years, and memory-scanning of physical and virtual machines is not new. Kaspersky Lab responded by developing a driver (also implemented in "light agent" applications for virtual host protection) that scans the operation system kernel, other drivers, user space processes, etc. in the memory, using a number of different rules.

To return to simple memory scanning, and deep behavioral analysis on the virtual host. Memory scanning is certainly valuable, but its usefulness shouldn't be overestimated. Without events recording at file systems level, system registry and OS API calls, memory scanning will catch near to nothing or will produce a raft of false positives.

Technical details

The main justification for memory scanning is that researching a packed malicious file is only possible in memory, the only place where it exists in unpacked form. But well-known packer programs are already "cracked" by modern anti-virus engines, that allowing packed malware to be analysed on the disk also.

Worst things are with protectors programs, which implement virtual machine, polymorphism and other technologies. However, simple memory

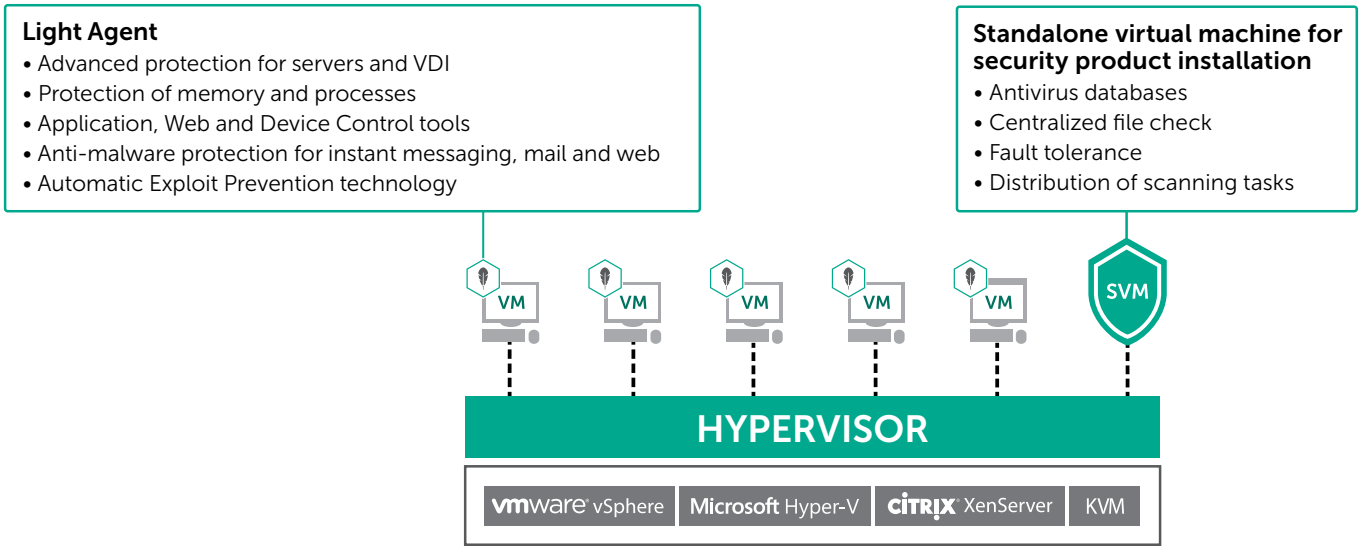
scanning undertaken doesn't generate much profit per se in the case of protected files. Without behavioral analysis, it's hard to judge the code assignment. And it's not always clear when to start scanning, i.e. to determining where to end decryption. That's assuming it has an end and is not a "decrypt on the fly" scheme.

The features of packers or protectors overlap with the memory handling mechanisms in modern operating systems. In Microsoft Windows, many operations take place asynchronously. When writing data to a file, it is initially buffered in memory, after which the OS writes the entire buffer to the disk. I.e. the precise moment of writing to disk is unclear, as is the processing context in which it will occur.

Another issue – all modern operating systems now use 'swap files' to enhance performance. From time to time, virtual memory content is unloaded to the swap file. In the event of a security incident, the virtual memory can be loaded into the swap file just before a physical memory scan. This is a problem for agentless solutions, as they only have access to physical memory. So using a swap file to virtual memory unload an interfere with memory scanning.

Finally, where an programming language interpreter is responsible for scripting language malware, only part of the malware may exist in the memory at the same time: malicious code not existed fully in the memory, but can still execute.

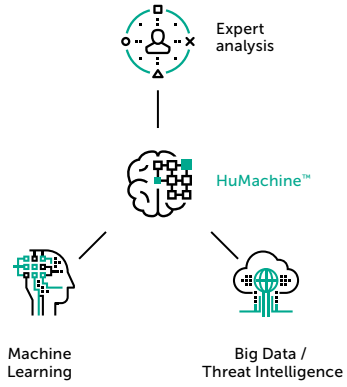
In terms of the specifics of virtual environment defenses, OS memory mechanisms also overlap with hypervisor features. Hypervisors can take control only during a very limited number of strictly specified times, which in turn limits possible reactions to malware detection in the memory.



What full multi-layered defense looks like

To sum up, RAM scanning is very useful in fighting malicious code, but should operate in parallel with detailed dynamic information about processes on the host. This data about is gathered in the memory as well as at file system level by specialized agents. A over-reliance on memory scanning can slow down analysis with no significant benefits in terms of protection.

Recognizing the differing needs of our customers, Kaspersky Lab implements both approaches: offering and agentless solution as well as light agent based product. This latter is capable of gathering all the data described, to better protect the virtual device. A detailed comparison of these two approaches is given here http://media.kaspersky.com/en/ business-security/Light-Agent-or-Agentless-KSV-Feature-Guide.pdf



www.kaspersky.com/enterprise
www.securelist.com

#truecybersecurity
#HuMachine