

Disk Level Malware Detection

Adrienne Felt, Nathanael Paul, David Evans, Sudhanva Gurumurthi

University of Virginia, Department of Computer Science

felt, nate, evans, gurumurthi@virginia.edu · http://www.cs.virginia.edu/malware/

What is malware detection at the disk level?

Modern disk drive processors are now capable of general purpose computation, and we can harness this new power to implement malware detection directly on the disk drive. All data flowing to and from the hard drive must pass through the disk drive processor. This key property makes the disk processor the final line of defense against malware, since it is privy to the low-level behavior of viruses that wish to alter data on the host. Disk-level malware detection uses the disk processor to identify threats based on patterns of I/O requests.

Current anti-malware methods are insufficient

String scanning is the traditional - and primary - method of virus detection.

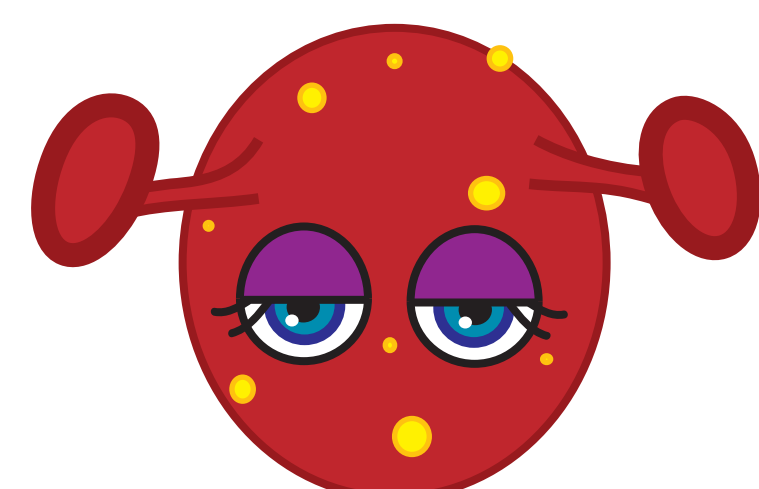
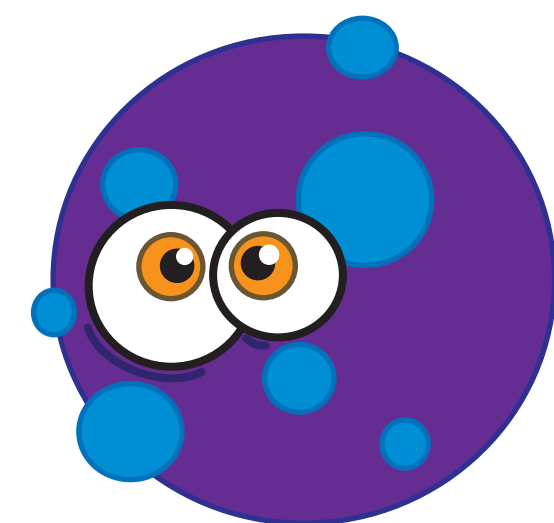
- Polymorphic and metamorphic viruses elude these detectors.
- It is easy to hand-craft variants that evade detection.

Emulation was designed in response to these complex viruses.

- Emulation is limited by its high computational cost and imprecision.
- Virus authors subsequently created anti-emulation techniques.

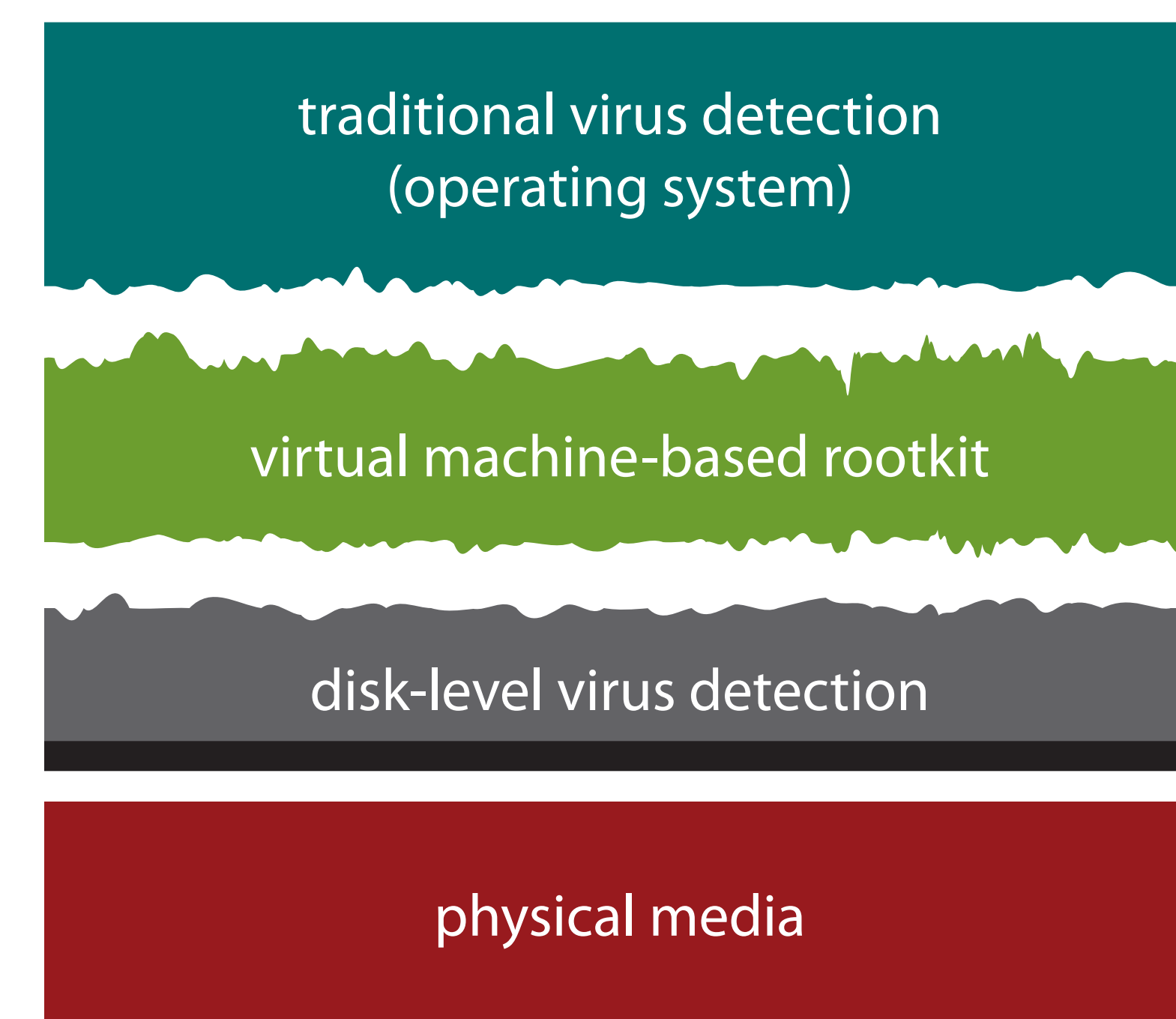
Rootkits are rapidly becoming a large threat.

- Virtual machine-based rootkits are nearly impossible to detect using host-level techniques because they run below detectors.



Polymorphism and metamorphism automatically alter code structure between generations without changing the virus's behavior.

Advantages of working at the disk level



Location

Can monitor all I/O requests, and the disk is not vulnerable to layer-below attacks.

Isolation

Separated from host and can operate while host may be compromised.

Difficult to circumvent

Altering code does not necessarily change disk accesses.

Low overhead

There is a speed gap between the disk processor and mechanical data transfer system.

Signature creation process

To watch virus activity, we ran Windows in a virtual machine on a Linux platform and observed the virus behavior on both levels.

type	PID,TID	request time	counter	file accessed
CREATE	560,564	18:39:12:156	0x0000006D	\Device\HarddiskVolume1\WINDOWS\ExpIorer
WRITE	560,564	18:39:12:156	0x00000070	\Device\HarddiskVolume1\WINDOWS\ExpIorer
CLOSE	560,564	18:39:12:156	0x00000073	<NO NAME>
CREATE	560,564	18:39:12:156	0x00000074	\Device\HarddiskVolume1\WINDOWS\ExpIorer
WRITE	4,24	18:39:12:156	0x00000075	<NO NAME>
CREATE	560,564	18:39:12:156	0x0000007A	\Device\HarddiskVolume1\WINDOWS\apphelp.c

r/w	Linux P/TID	time	block	length info
r	3216,3215	06:43:19	254108770	4096,4096
w	528,528	06:43:19	254069298	4096,4096
w	528,528	06:43:19	254069306	4096,4096
w	528,528	06:43:19	254069314	4096,4096
w	528,528	06:43:19	254080810	4096,4096
w	528,528	06:43:19	254080818	4096,4096

Windows trace

EFish (process ID 560) drops an unencrypted virus sample, ExpIorer.exe, into the Windows directory. The virus data is written at an offset of 0 blocks. The system process (PID 4) echoes the write.

Linux trace

In Linux, we can see these same reads and writes. VMWare (process IDs 4260-4264) writes the virus data ("MZgdi...") to disk.

Example signature for W32/EFish

W32/EFish is an unencrypted, slow polymorphic virus. It is a highly aggressive parasitic infector. Its typical pre-infection behavior can be described as follows:

- EFish writes to the unencrypted virus sample. It is named "ExpIorer" with a capital "I" to trick users.
- "ExpIorer" then reads itself back in to execute.
- The virus registers itself as a service and causes a "WDM call returned error", which Windows logs.

WRITE:	"MZgdi32.dll [...] ExpIorer"
OFFSET:	0
WRITE:	"MZgdi32.dll [...] ExpIorer"
OFFSET:	0
READ:	"MZgdi32.dll [...] ExpIorer"
OFFSET:	0
READ:	"(\\c\\c\\c \\c\\c \\d\\d \\d\\d:\\d\\d \\d\\d\\d.[15]) : WDM call returned error: 4200__"
OFFSET:	0
WRITE:	"(\\c\\c\\c \\c\\c \\d\\d \\d\\d:\\d\\d \\d\\d\\d.[15]) : WDM call returned error: 4200__"
OFFSET:	not 0
WRITE:	"(\\c\\c\\c \\c\\c \\d\\d \\d\\d:\\d\\d \\d\\d\\d.[15]) : WDM call returned error: 4200__"
OFFSET:	0

Supported by the National Science Foundation Cyber Trust Program ("Disk-Level Malware Detection and Response," NSF 0627527).

