SOPHOS

simple + secure

Unveiling the kernel: Rootkit discovery using selective automated kernel memory differencing

Ahmed Zaki and Benjamin Humphrey

Agenda

- Objective and method
- System design and implementation
  - Running drivers
  - Data extraction
  - Processing the data
  - Reporting and signatures
- And the result is …
  - Experiment A: High profile rootkits
  - Experiment B: Driver files
  - Experiment C: Random set of PE files
- Conclusion
- Future work

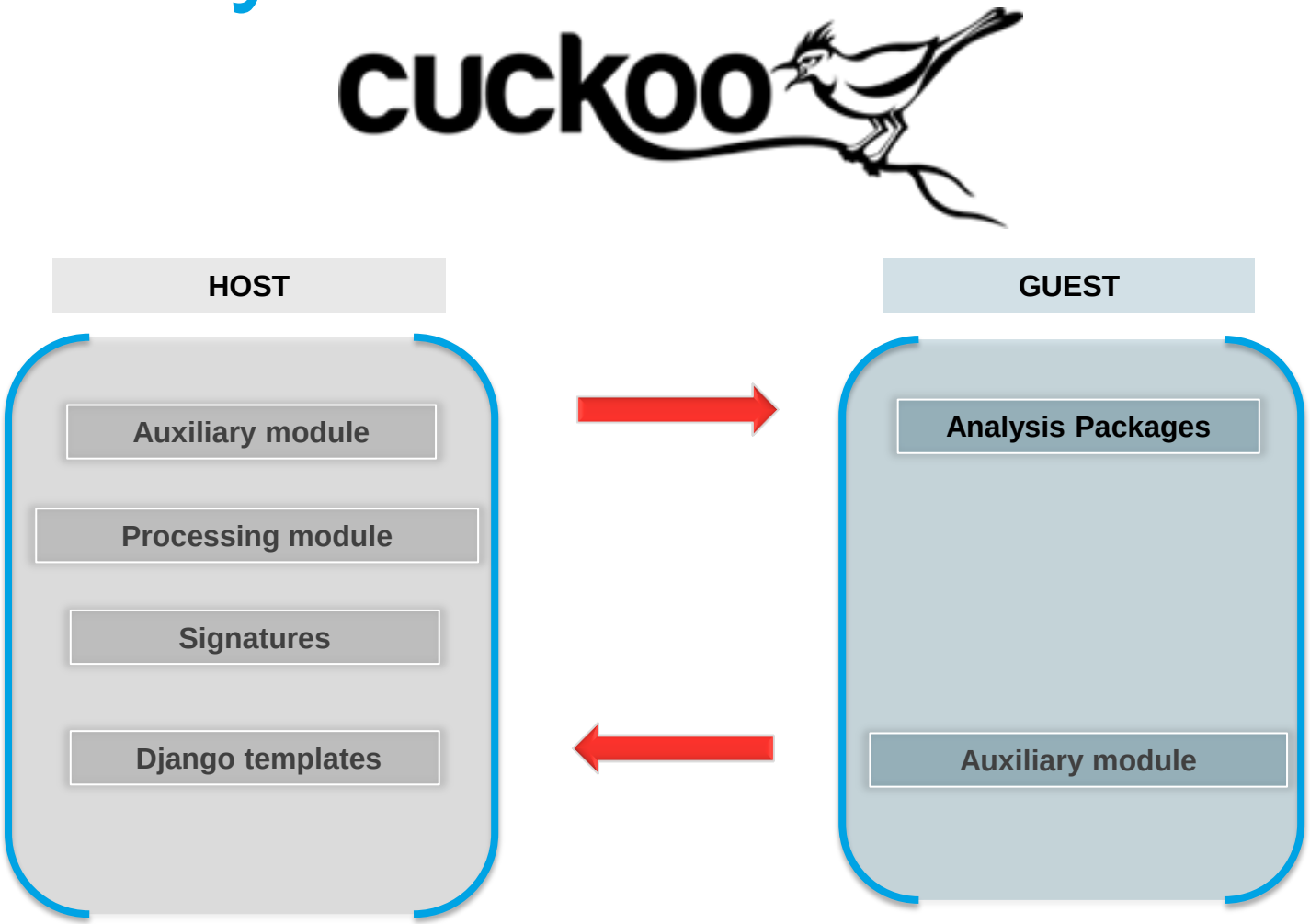
What is the objective?

- Automate the process of finding samples that exhibit kernelmode behaviour
- List the modifications made to the kernel
- Identify the maliciousness of specific modifications
- Import that data into other systems

How do we automate the process?

- Use existing tools
- Build our own
  - Using anti-rootkit tools
  - Using a custom diff based solution

Bird's eye view



SOPHOS

Running Driver files

(Servicename | StartType | ServiceType | LoadMode)

“Register service using scm
If not loadmode then:
  start service using NTLoadDriver
Else:
  start service using loadmode option
If not service is running:
   report FAIL ! “

Usage of the Sophos AV Engine

With the SAV engine we get:

- Existing software that has a presence in the kernel
- The ability to examine/dump areas of kernel memory
- The ability to write to a log file

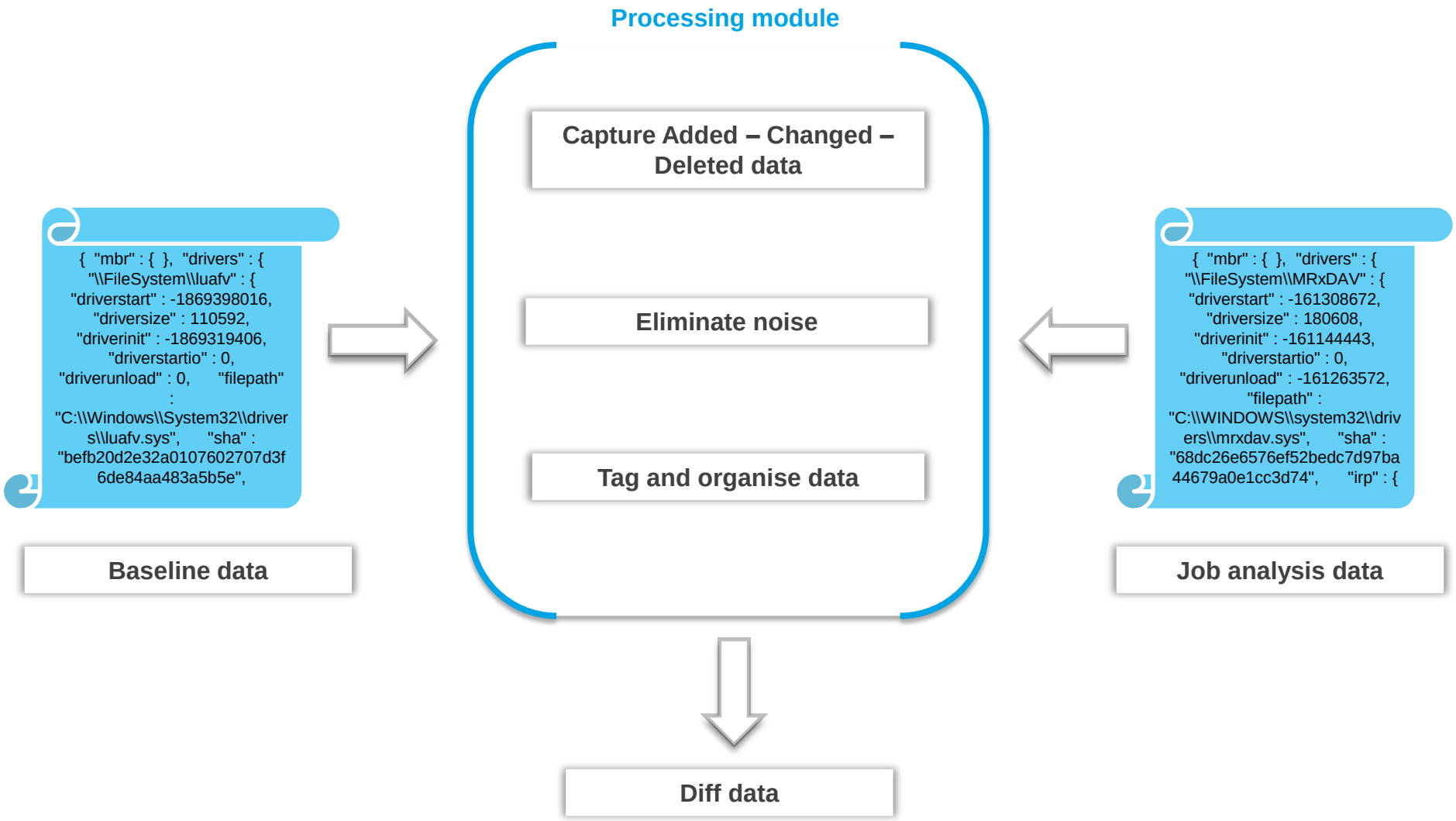
NOTE: Due to modular design we don't *need* to use the Sophos AV engine.

Examining the kernel

What areas are we looking at?

Drivers	Modules	SSDT
IDT	Callbacks	Disk Information

Processing the data



SOPHOS

Processing the data

SOPHOS

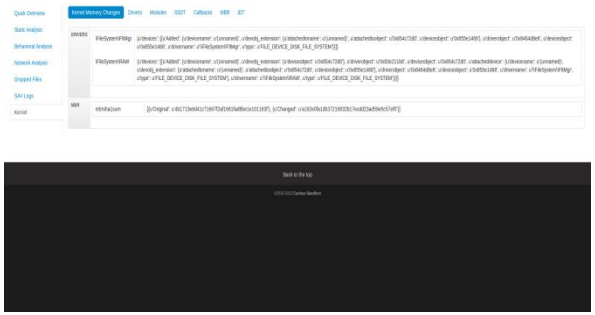
Data flow

Data → Signatures → Reporting


```
{"drivers": {"\\Driver\\4C3553F1":
{"Added": {"driverinit":
"0xf7b97983", "driverunload":
"0xf7b9794a", "irp":
{"IRP_MJ_CREATE_MAILSLOT
": "0xf7b97965",
"IRP_MJ_SET_QUOTA":
"0xf7b97965",
"IRP_MJ_SET_SECURITY":
"0xf7b97965",
"IRP_MJ_SET_VOLUME_INFO
RMATION": "0xf7b97965",
"IRP_MJ_WRITE":
"0xf7b97965", …"
```

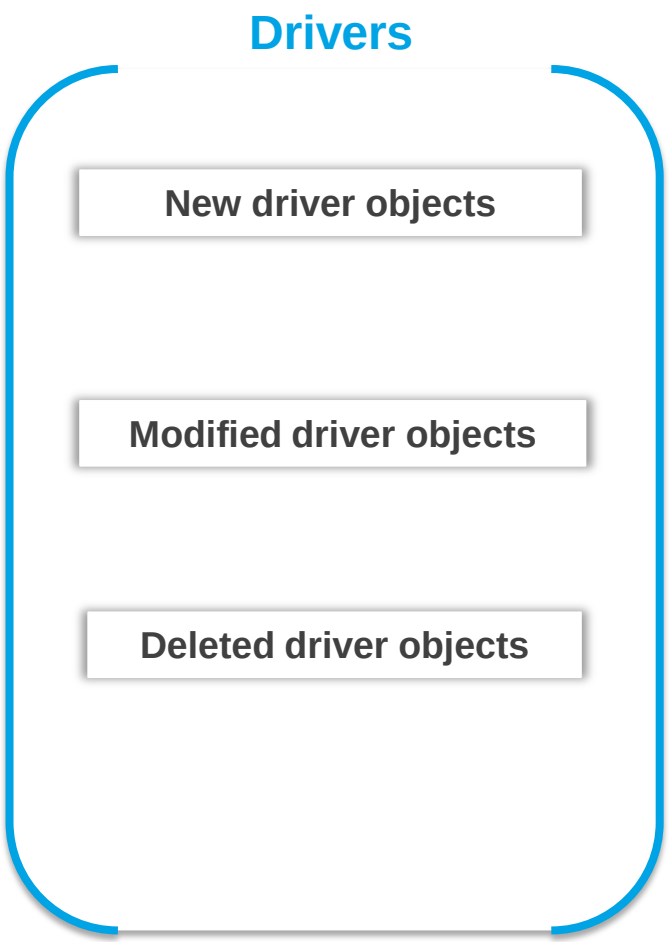

generic_new_driver
generic_modified_driver
generic_deleted_driver
generic_new_module
generic_deleted_module
generic_ssdt_hook
generic_idt_hook
generic_new_callback
generic_modifed_callback
generic_attached_device

. . .



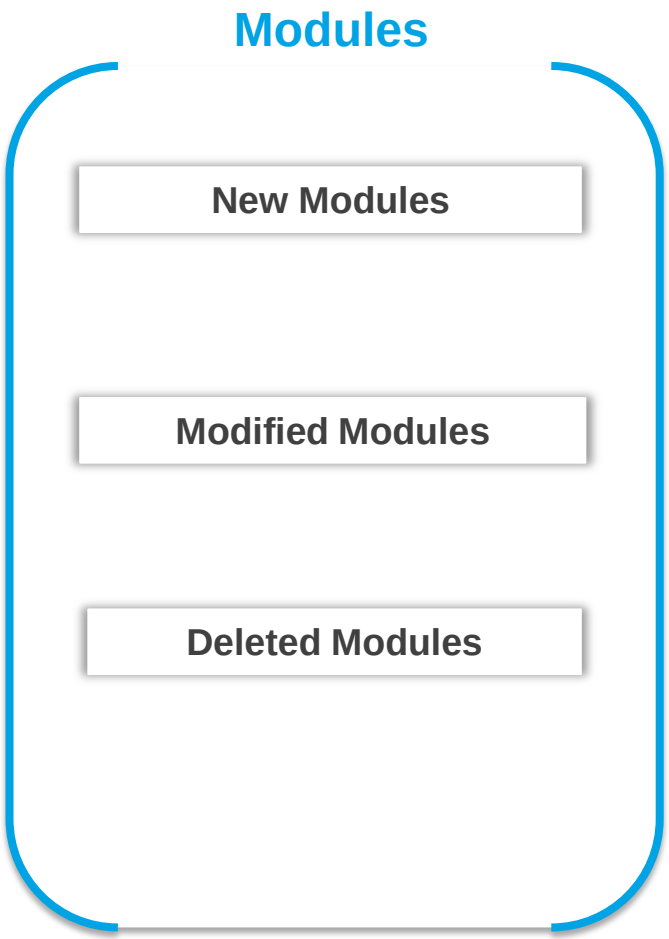
SOPHOS

Signatures



SOPHOS

Signatures

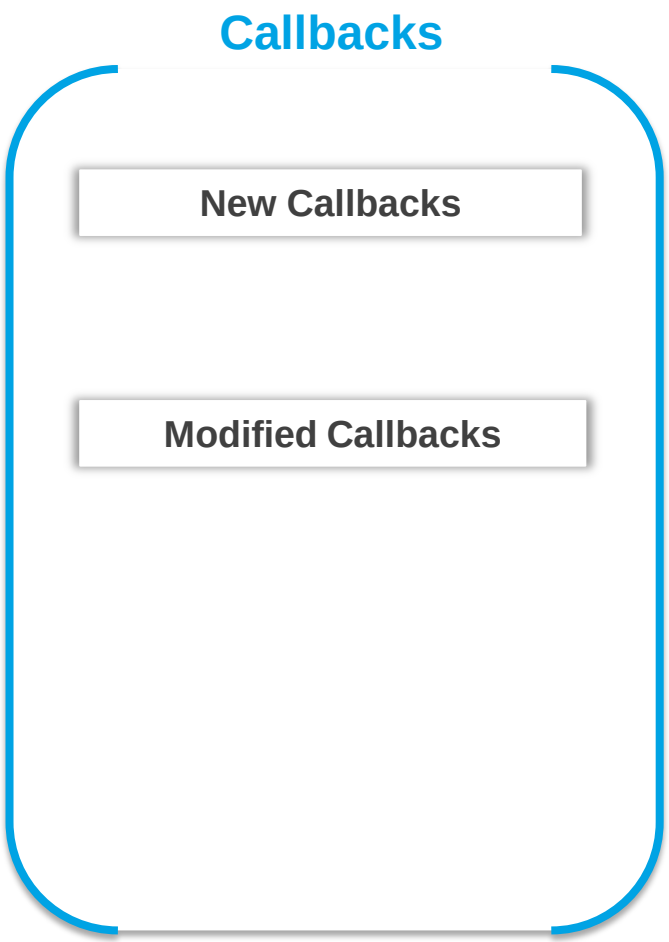


SOPHOS

Signatures

SOPHOS

Signatures

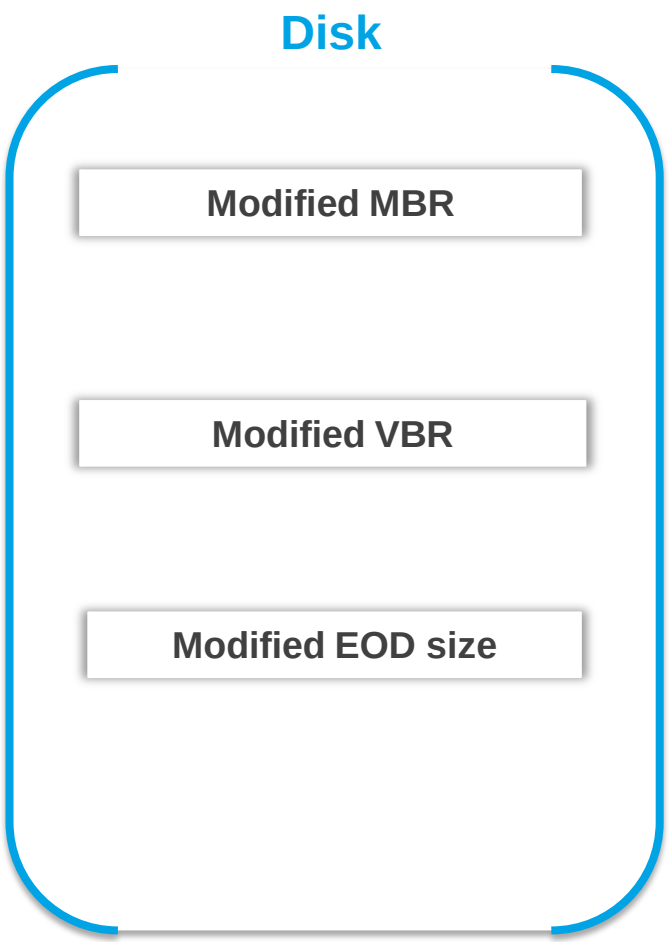


SOPHOS

Signatures

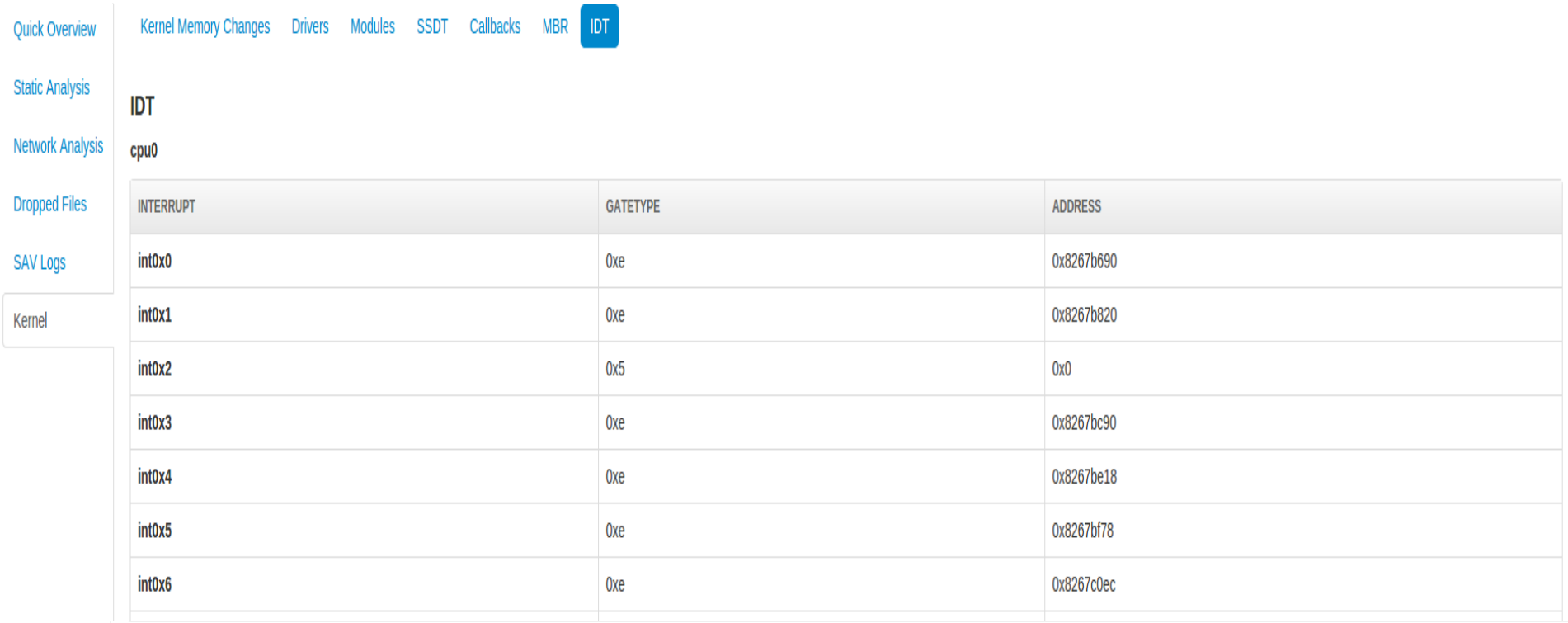
SOPHOS

Signatures



SOPHOS

Reports



SOPHOS


YOU SHALL NOT PASS

Tests

- High profile rootkits
  - TDL Derivatives
  - GAPZ
  - Turla
  - Necurs
- Experiment B: Malicious and clean driver set
- Experiment C: Random set of known malicious PE files

TDL Derivatives

SOPHOS

TDL Derivatives

Device Object(s) added

DeviceName: (unnamed)

DeviceObjectAddr: 0x84895710

DeviceType: FILE_DEVICE_DISK_FILE_SYSTEM

Driver: \FileSystem\FltMgr

Driver Object(s) modified

Name: \FileSystem\RAW

Name: \FileSystem\FltMgr

Kernel Memory Changes | Drivers | Modules | SSDT | Callbacks | MBR | IDT

DRIVERS

\FileSystem\RAW	{u'devices': [{u'Added': {u'devicename': u'(unnamed)', u'devobj_extension': {u'deviceobject': u'0x8482c030'}, u'driverobject': u'0x83e1f1b8', u'deviceobject': u'0x8482c030', u'attacheddevice': {u'devicename': u'(unnamed)', u'devobj_extension': {u'attachedtoname': u'(unnamed)', u'attachedtoobject': u'0x8482c030', u'deviceobject': u'0x84895710'}, u'driverobject': u'0x84a41c40', u'deviceobject': u'0x84895710', u'drivername': u'\\FileSystem\\FltMgr', u'type': u'FILE_DEVICE_DISK_FILE_SYSTEM'}, u'drivername': u'\\FileSystem\\RAW', u'type': u'FILE_DEVICE_DISK_FILE_SYSTEM'}}]}
\FileSystem\FltMgr	{u'devices': [{u'Added': {u'devicename': u'(unnamed)', u'devobj_extension': {u'attachedtoname': u'(unnamed)', u'attachedtoobject': u'0x8482c030', u'deviceobject': u'0x84895710'}, u'driverobject': u'0x84a41c40', u'deviceobject': u'0x84895710', u'drivername': u'\\FileSystem\\FltMgr', u'type': u'FILE_DEVICE_DISK_FILE_SYSTEM'}}]}

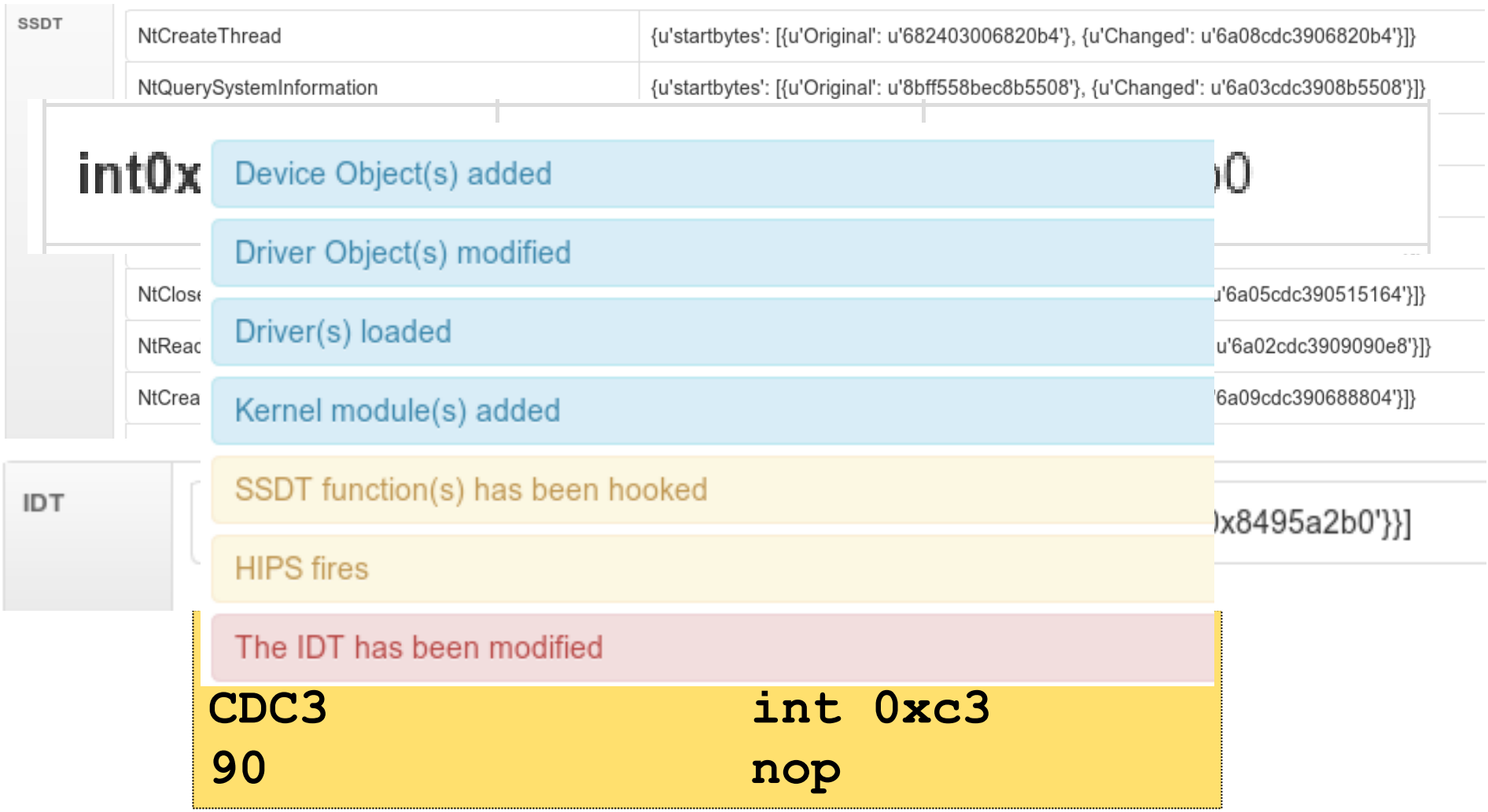
SOPHOS

GAPZ

The VBR has been modified

MBR	Partition0	{u'vbrsha1sum': [{u'Original': u'7a781423dbb768786a81633441f8d533594583f5'}, {u'Changed': u'64f08b44562578234af25a1cfef84d2bccf1a5'}]}
SSDT	NtDeletePrivateNamespace	{u'startbytes': {u'Added': u'8bff558bec83ec10', u'PagedIn': 1}}
	NtSaveKey	{u'startbytes': {u'Added': u'8bff558bec83e4f8', u'PagedIn': 1}}
	NtPulseEvent	{u'startbytes': {u'Added': u'6a1468a8e6982e8', u'PagedIn': 1}}

Turla a.k.a Snake, Uroborus



SOPHOS

Necurs

\FileSystem\FltMgr

Device Object(s) added

Driver Object(s) modified

Driver(s) loaded

A callback has been added

tension': {u'attachedtoname': u'0x85765ed8'}, u'driverobject': {u'devicename': u'(unnamed)', ltoobject': u'0x85765ed8', deviceobject': u'0x8581eb68', u'drivername': u'\\Driver\\808a56c5daeb2cc4', u'type': u'FILE_DEVICE_DISK_FILE_SYSTEM'}, ISK_FILE_SYSTEM'}}]}

LoadImageNotify

[{u'Added': {u'filepath': None, u'driver': u'unknown/hidden', u'module': u'\\SystemRoot\\system32\\ntkrnlpa.exe', u'sha': None, u'address': u'0x8289add5', u'type': u'LoadImageNotify'}}, {u'Added': {u'filepath': None, u'driver': u'\\Driver\\808a56c5daeb2cc4', u'module': u'unknown/hidden', u'sha': None, u'address': u'0x84874510', u'type': u'LoadImageNotify'}}]

Name: \Driver\808a56c5daeb2cc4

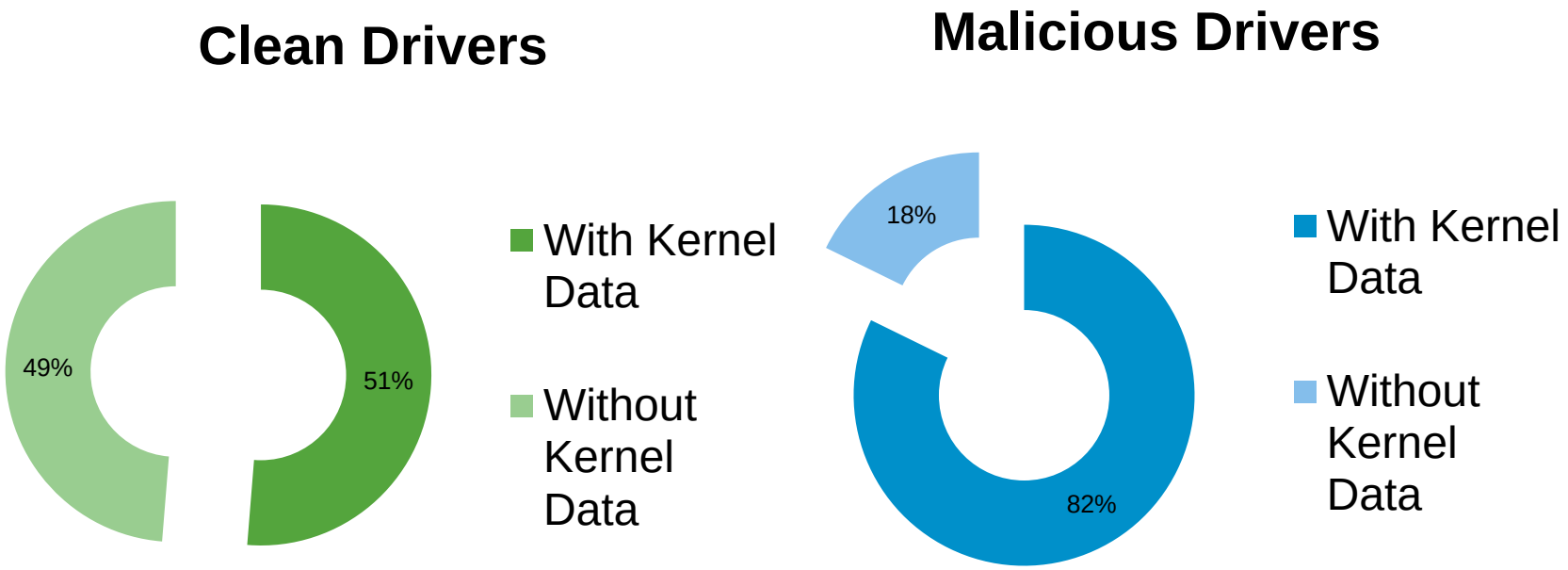
Sha: None

SOPHOS

High profile rootkits

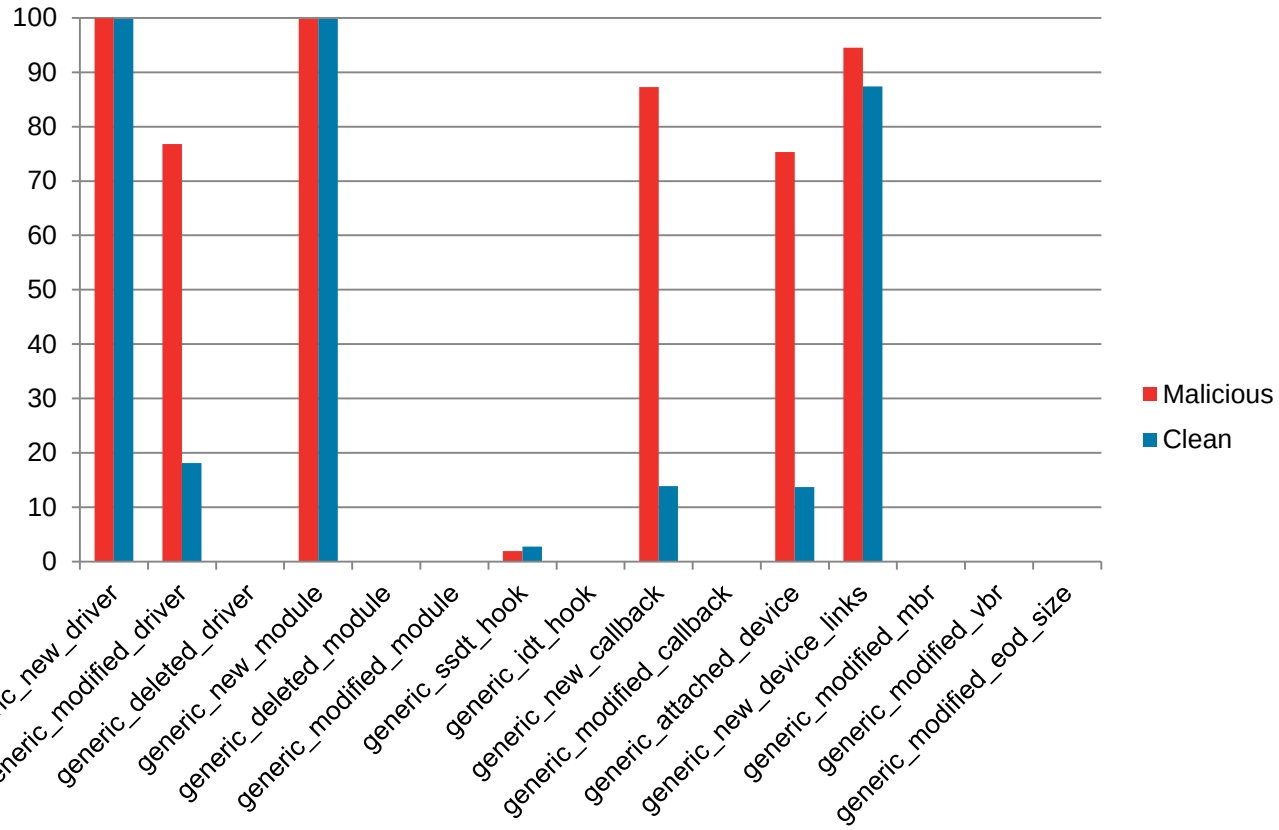
- We do not always get enough information to classify specific families
- We are getting enough information to warrant further investigation by an researcher

Experiment B



- Total number of malicious drivers 1854.
- Total number of clean drivers 1053.
- Insufficient time for the log to be generated was a common reason for failure to get back kernel data. Miss the log by a second or two. It's a trade off.

And the results is



SOPHOS

Callbacks

TYPE	FILEPATH	DRIVER	MODULE
CreateThreadNotify			
	C:\Windows\System32\drivers\savonaccess.sys	\FileSystem\SAVOnAccess	\SystemRoot\system32\DRIVE
CreateProcessNotify			
	None	unknown/hidden	\SystemRoot\system32\ntkrnlp
	C:\Windows\System32\drivers\ksecdd.sys	\Driver\KSecDD	\SystemRoot\System32\Drivers
	C:\Windows\System32\drivers\cng.sys	\Driver\CNG	\SystemRoot\System32\Drivers
	C:\Windows\System32\drivers\tcpip.sys	\Driver\Tcpip	\SystemRoot\System32\drivers
	None	unknown/hidden	\SystemRoot\system32\CI.dll
	C:\Windows\System32\drivers\PEAuth.sys	\Driver\PEAUTH	\SystemRoot\system32\drivers

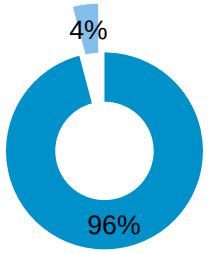
SSDT	NtEnumerateKey	{u'startbytes': [{u'Original': u'6a6068d88c6982e8'}, {u'Changed': u'e967eaa8186982e8'}]}
	NtFlushInstructionCache	{u'startbytes': [{u'Original': u'6a2c6838206982e8'}, {u'Changed': u'e9badfb6186982e8'}]}

	None	unknown/hidden

SOPHOS

Experiment C

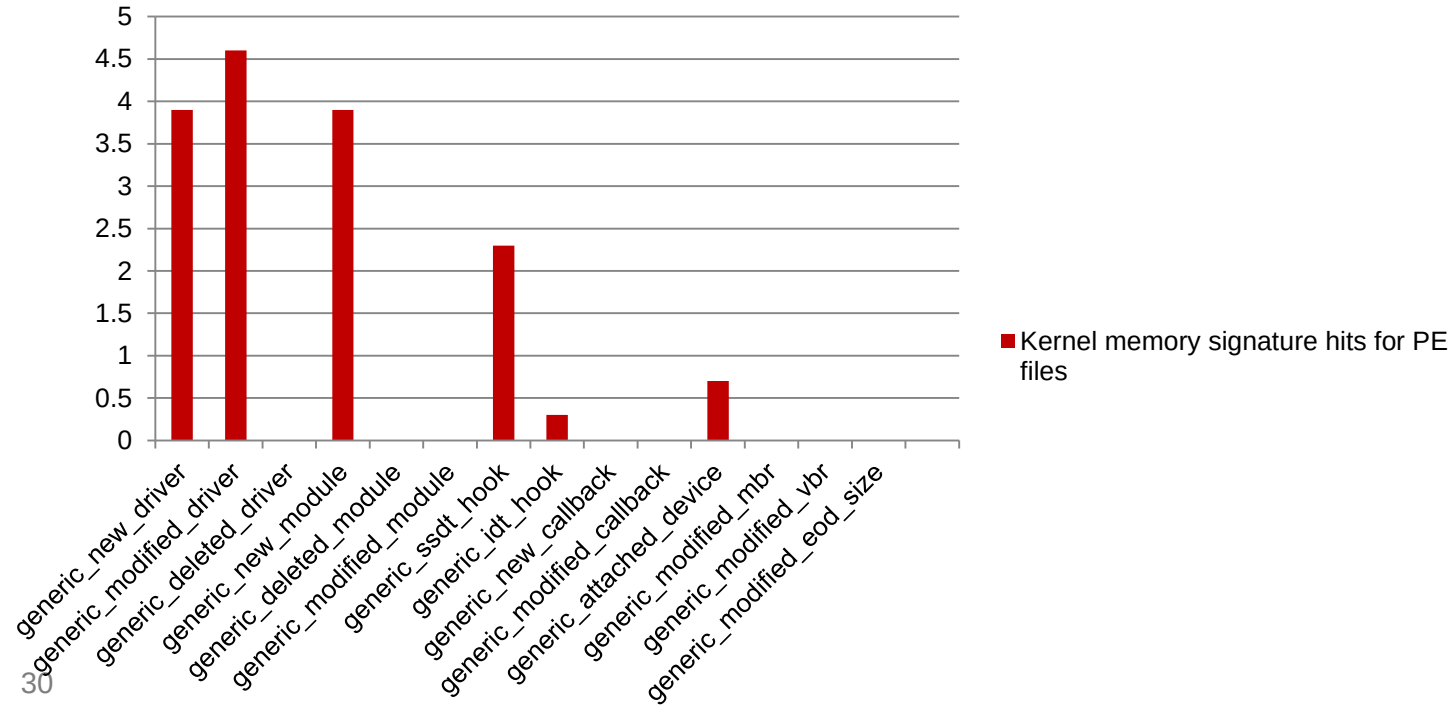
- 319 of known malicious PE files.



Set of PE files

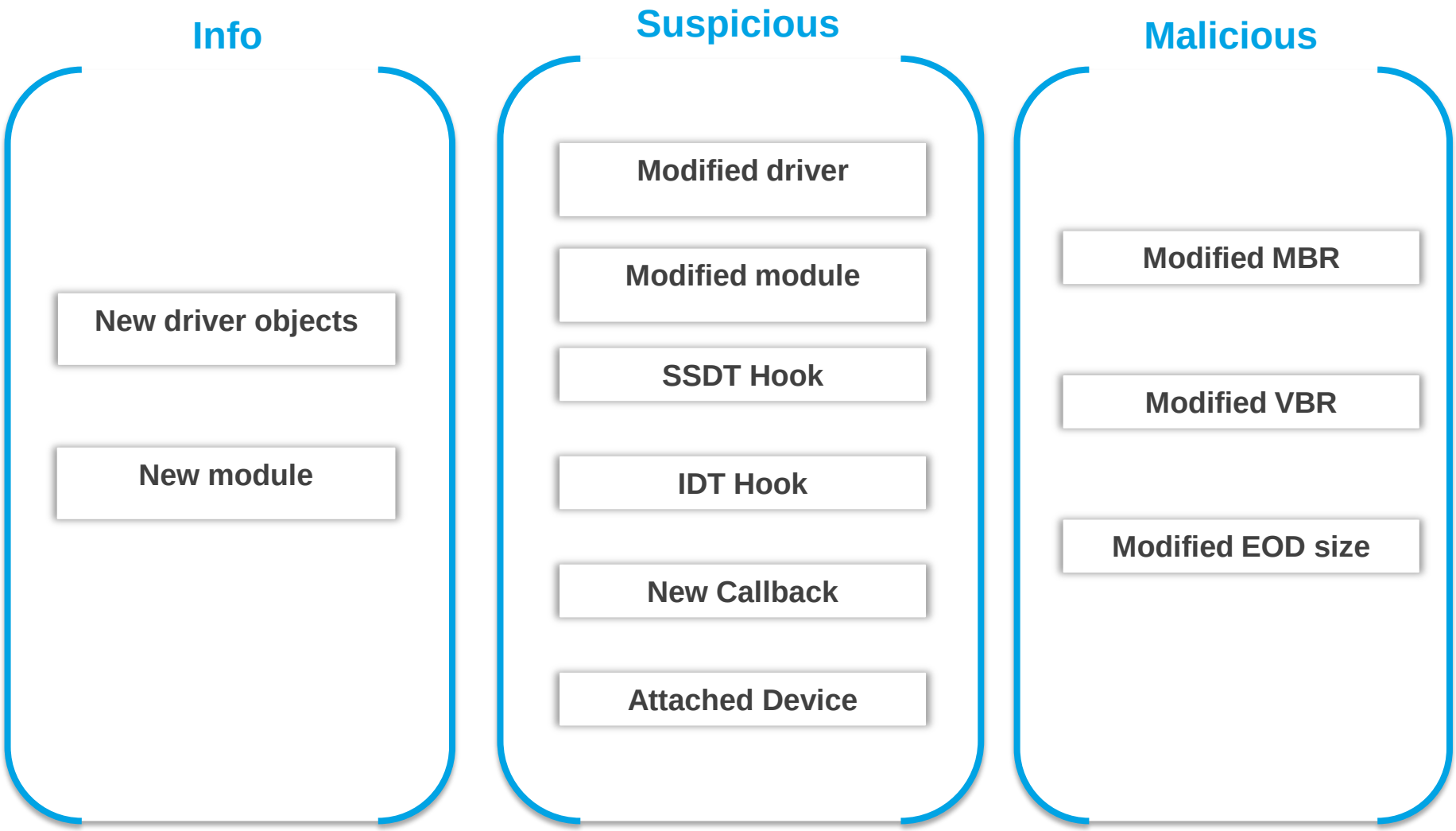
- With Kernel memory
- Without Kernel Data

Kernel memory signature hits for PE files



SOPHOS

Weighting the signatures



SOPHOS

Conclusion

- Malicious activity can be identified via modifications rather than creation
- Malicious drivers are unlikely to employ anti-sandboxing techniques
- Good enough to identify kernel activity
  - Not exhaustive analysis

Future work

- Exploring other areas of the kernel
  - Object table
  - DKOM
  - 64bit drivers
- Sample clustering
- Usermode rootkits

Questions?