RSAConference2017

San Francisco | February 13 – 17 | Moscone Center

POWER OF OPPORTUNITY

SESSION ID: HTA-T09

How to Go from Responding to Hunting with Sysinternals Sysmon

Mark Russinovich
CTO, Microsoft Azure
Microsoft Corporation
@markrussinovich

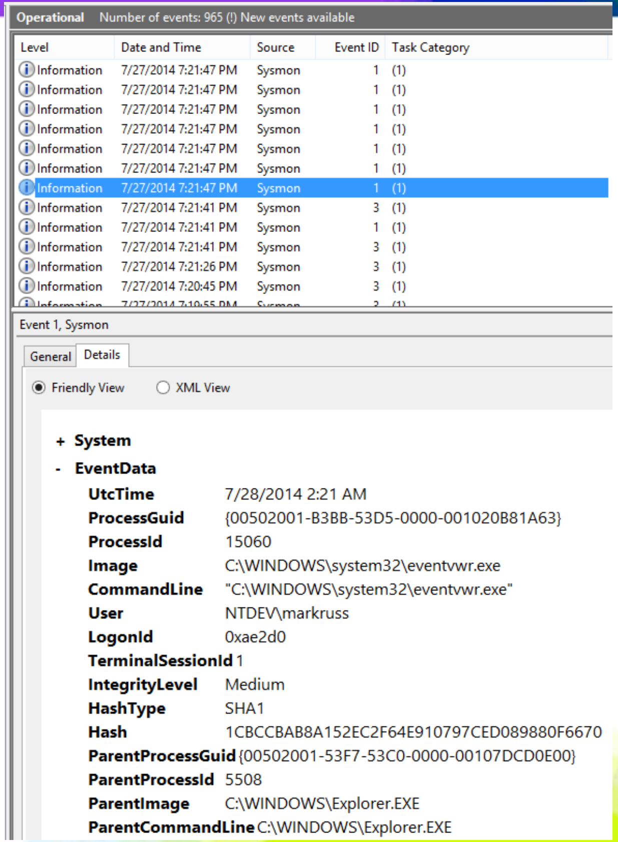
Microsoft

Hunting Attackers

- When attackers or malware get on your network, you need to hunt them down
  - What was their entry point?
  - Did they spread between systems?
  - What happened on a particular system?
- Built-in Windows tooling make it hard to answer these questions:
  - Limited information captured for process creates and DLL loading
  - Network connection information simultaneously too limited and verbose
  - No way to capture common attacker behavior (e.g. thread injection)

Sysinternals Sysmon (System Monitor)

- Free download from Sysinternals.com
- Background system monitoring utility
  - Record system events to the Windows event log
  - Can be used for system anomaly detection
  - Forensics can trace intruder activity across the network
- I wrote it for use within Microsoft corporate network
  - To understand attacker behavior and tools
  - Contributions from Thomas Garnier, David Magnotti, Mark Cook, Rob Mead and Giulia Biagini



Announcing Sysmon v6

- Schema output
- “Friendly” registry key names
- Named pipes
- Sysmon configuration change logging
- Status for revoked and expired signatures

Agenda

- Sysmon Basics
- Architecture and Advanced Filtering
- A Look at New Events
- Crafting a Configuration File
- Determining an Exploit Vector
- Hunting

RSAConference2017

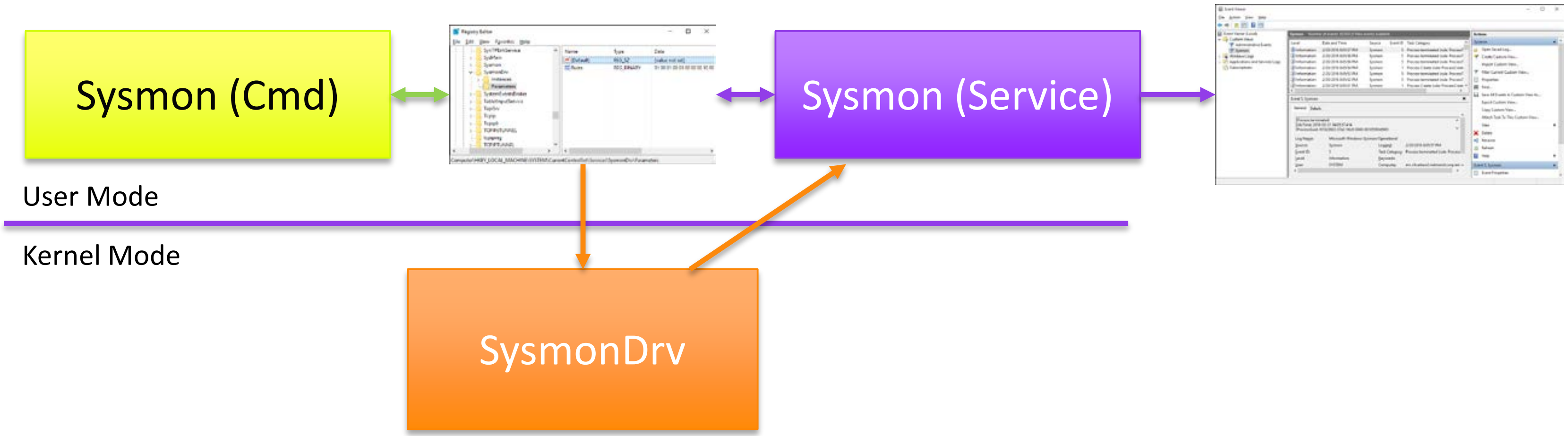
Sysmon Basics

Sysmon Command-Line Usage

- Windows service and device driver (~2 MB total)
  - Single binary includes 32-bit and 64-bit versions of both
  - Service doubles as command-line frontend
- Installation:
  sysmon -i -accepteula [options]
  - Extracts binaries into %systemroot%
  - Registers event log manifest
  - Enables default configuration

```
Usage:
Install:     sysmon -i [<configfile>]
               [-h <[sha1|md5|sha256|imphash|*],...>] [-n [<process,...>]]
               [-l [<process,...>]
Configure:   sysmon -c [<configfile>]
               [--|[-h <[sha1|md5|sha256|imphash|*],...>] [-n [<process,...>]]
               [-l [<process,...>]]]
Uninstall:   sysmon -u
```

Sysmon Architecture



Other Sysmon.exe Control Options

- Viewing and updating configuration:
  sysmon -c [options]
  - Updates take effect immediately
  - Options can be basic options or a configuration file
- Register event manifest for viewing logs only:
  sysmon -m
- Uninstall:
  sysmon -u

Sysmon Events

Category	Event ID
Sysmon Service Status Changed	0
Process Create	1
File Creation Time Changed	2
Network Connection	3
Sysmon Service State Change	4
Process Terminated	5
Driver Loaded	6
Image Loaded	7
CreateRemoteThread	8
RawAccessRead	9

Category	Event ID
Process Access	10
File Create	11
Registry Object CreateDelete	12
Registry Value Create	13
Registry Object Rename	14
File Create Stream Hash	15
Sysmon Configuration Changed	16
Pipe Created	17
Pipe Connected	18
Error	255

v6

Basic Configuration Options

- Installing with no options logs all the following with SHA1 hashes where applicable:

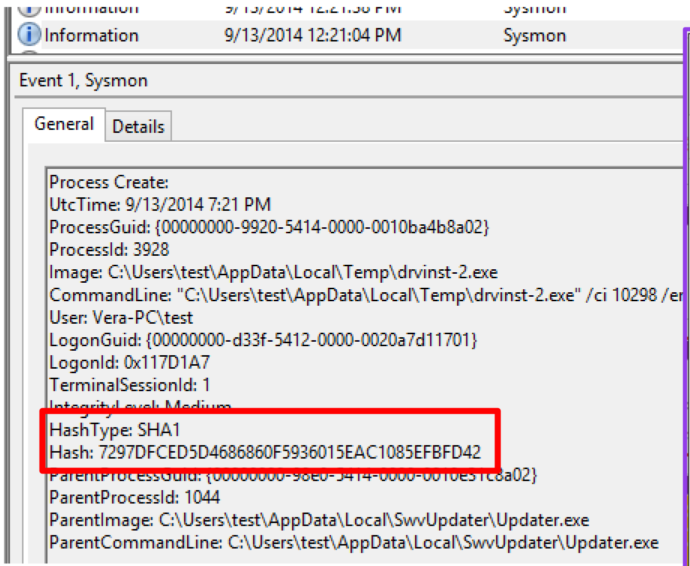
  Process create, Process terminate, Driver loaded, File creation time changed, Sysmon service state changed, Sysmon configuration changed

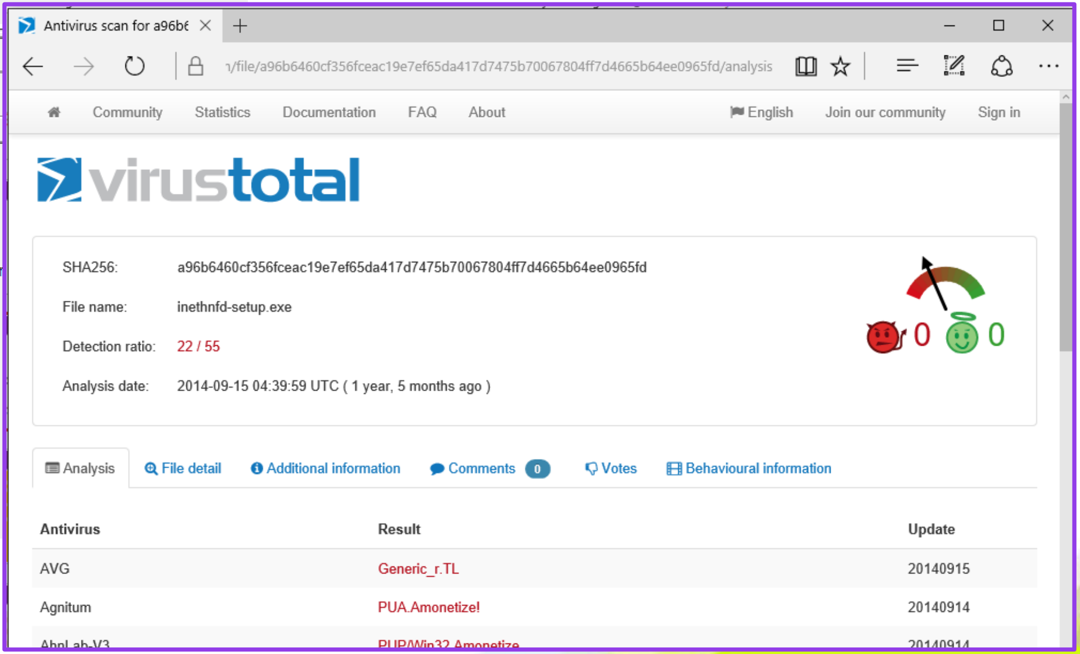
- Additional basic options:

Option	Description
-h [SHA1] [MD5] [SHA256] [IMPHASH] [*]	Hash algorithm(s)
-n [process,…]	Logs network events
-l [process,…]	Logs image load events
--	Restores default configuration (-c only)

Hashes and VirusTotal

- You can extract a hash and paste it into VT search for a report:





Configuration

- Basic options are limited:
  - Cannot disable events via basic options (e.g. CreateRemoteThread, RawAccessRead)
  - Advanced filtering not possible (e.g. process name filters)
- Sysmon configuration file supports all configuration options:

  install: sysmon -i -accepteula c:\SysmonConfig.xml

  update: sysmon -c c:\SysmonConfig.xml

Configuration Top-Level Tags

- Schema version: current is 3.3
- HashAlgorithms:
  - Applies to all events
  - ‘*’ for all hash types
- CheckRevocation:
  - Controls cert revocation checks (default: no)
- EventFiltering:
  - Flexible filtering rules
  - If event type not specified, default capture rule applies

```xml
<Sysmon schemaversion="2.0">
  <!-- Capture all hashes -->
  <HashAlgorithms>*</HashAlgorithms>
  <EventFiltering>
    <ProcessCreate onmatch="include">
      <Image condition="contains">notepad</Image>
    </ProcessCreate>
    <FileCreateTime onmatch="include"/>
    <ImageLoad onmatch="include"/>
    <CreateRemoteThread onmatch="include"/>
    <ProcessTerminate onmatch="include">
      <Image condition="contains">notepad</Image>
    </ProcessTerminate>
    <DriverLoad onmatch="exclude"/>
    <NetworkConnect onmatch="include"/>
  </EventFiltering>
</Sysmon>
```

Event Tags

- Each event is specified using its tag
- To see all tags, dump the full configuration schema:

  sysmon -s

- Onmatch can be “include” or “exclude”
  - Include and exclude refer to filter effect
  - Filters described later…

```
<tag onmatch=“include”>
    <include filter/>
    …
</tag>
```

```
<tag onmatch=“exclude”>
    <exclude filter/>
    …
</tag>
```

Event Tags With No Filters

- Useful for enabling specific event types
- If no filter, onmatch has opposite effect:
  - Include: don’t log any events
  - Exclude: log all events of the tag type
- This configuration enables the following:
  - ProcessCreate: because of onmatch exclude
  - ProcessTerminate: because it is omitted and by default enabled

```
<Sysmon  schemaversion="2.01">
  <EventFiltering>
    <ProcessCreate onmatch="exclude"/>
    <DriverLoad onmatch="include"/>
    <ImageLoad onmatch="include"/>
    <FileCreateTime onmatch="include"/>
    <NetworkConnect onmatch="include"/>
    <CreateRemoteThread onmatch="include"/>
    <RawAccessRead onmatch="include"/>
  </EventFiltering>
</Sysmon>
```

Filter Conditions

- Filters are specified as event field conditions:
  - Field is any field in event schema
  - Condition types can be used with any field

<*eventtag* onmatch=“include”>

<*field* condition=“*conditiontype*”>*value*</*field*>

…

</*eventtag*>

- Combine include and exclude filters

ConditionType
is
Is not
contains
excludes
begin with
end with
less than
more than
image

Filter Examples

- Include only Google Chrome network activity:

```
<NetworkConnect onmatch="include">
    <Image condition="contains">chrome.exe</Image>
</NetworkConnect >
```

- Include thread injections into winlogon and lsass:

```
<CreateRemoteThread onmatch="include">
    <TargetImage condition="image">lsass.exe</TargetImage>
    <TargetImage condition="image">winlogon.exe</TargetImage>
</CreateRemoteThread >
```

- Exclude all Microsoft-signed image loads:

```
<ImageLoad onmatch="exclude">
    <Signature condition="contains">microsoft</Signature>
</ImageLoad>
```

RSAConference2017

A Look at the New Events

Sysmon Configuration Change

- Logs configuration file updates
  - Name of configuration file
  - Hash of configuration file’s contents
- Cannot be filtered (neither can Sysmon service status)

ProcessAccess
UtcTime
Configuration
ConfigurationFileHash

Process Access

- Useful for detecting credential dumping
  - Will detect process monitors (Taskmgr, Procexp)
  - Should focus on sensitive processes (Lsass, Winlogon)
- Generated from ObRegisterCallbacks
- ProcessGuid, LogonGuid uniquely identify process (PID and LogonId can be reused)

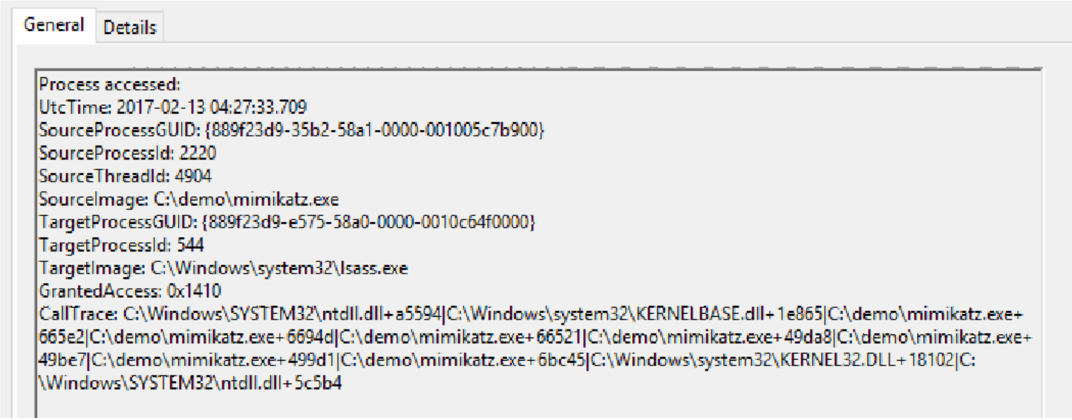
ProcessAccess
UtcTime
SourceProcessGuid
SourceProcessId
SourceThreadId
SourceImage
TargetProcessGuid
TargetProcessId
TargetImage
GrantedAccess
CallTrace

Tracking Mimikatz

- I recommend always including lsass.exe process access:

```
<ProcessAccess onmatch="include">
     <TargetImage condition="is">C:\windows\system32\lsass.exe</TargetImage>
</ProcessAccess>
```

- Mimikatz request 0x1410:
  - 0x1000: PROCESS_QUERY_ LIMITED_INFORMATION
  - 0x0400: PROCESS_QUERY_ INFORMATION
  - 0x0010: PROCESS_VM_READ



- Exclude GrantedAccess of 0x1000, 0x1400, 0x400

File Create and File Create Stream Hash

- Generated from file system mini-filter
- File create is useful for detecting autostart drops
- File create stream hash detects browser drops (MOTW – Mark of the Web)
  - MOTW: Zone.Identifier
  - Hashes default data stream (main file)
  - Logged for any matching create of an alternate data stream

FileCreate
UtcTime
ProcessGuid
ProcessId
Image
TargetFileName
CreateUtcTime

FileCreateHash
UtcTime
ProcessGuid
ProcessId
Image
TargetFileName
CreateUtcTime
Hash

Registry

- Captured by CmRegisterCallback
- Three sub-events:
  - Registry object (key or value) create/delete
  - Registry value set
  - Registry object rename
- 'Friendly' names for schema > 3.2
- Useful for watching:
  - Critical system configuration
  - Autostart locations

Object Create/Delete

RegistryEvent
EventType
UtcTime
ProcessGuid
ProcessId
Image
TargetObject

Value Set

RegistryEvent
EventType
UtcTime
ProcessGuid
ProcessId
Image
TargetObject
Details

Object Rename

RegistryEvent
EventType
UtcTime
ProcessGuid
ProcessId
Image
TargetObject
NewName

Named Pipes

- Captured by file system minifilter
- Two sub -events:
  - Named pipe create
  - Named pipe connect
- Useful for watching inter-process malware communication

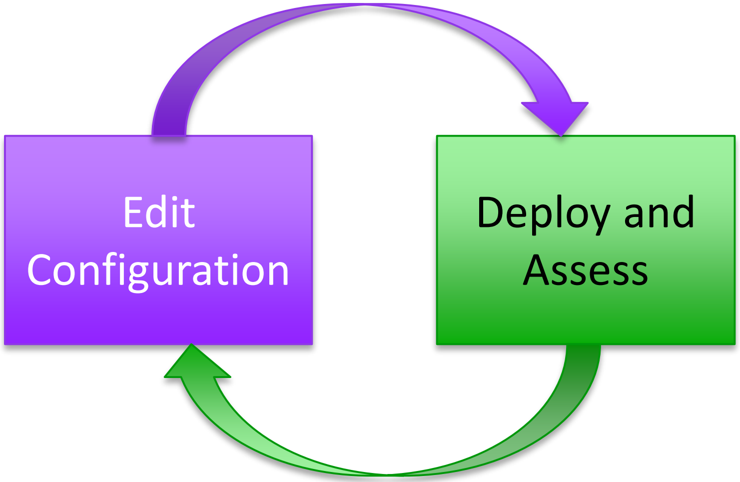
Pipe Create
PipeEvent
UtcTime
ProcessGuid
ProcessId
Pipe
Image

Pipe Connect
PipeEvent
UtcTime
ProcessGuid
ProcessId
Pipe
Image

Crafting Configuration

What's a Good Configuration?

- One that doesn't overwhelm your systems
  - Excessive resource usage
  - Excessive log volume
- Crafting is iterative:
  - Exclude known sources
    - E.g. OneDrive for file time stamp changes
  - Include sensitive targets:
    - E.g. Lsass.exe for credential theft
- When investigating likely breach, bias for data



Basic Event Recommendations

Category	Recommendation
Process Create	Log all
File Creation Time Changed	Exclude known processes
Network Connection	Log all non-browser activity
Process Terminated	Enable when investigating a breach
Driver Loaded	Log all non-Windows and non-Microsoft
Image Loaded	Enable when investigating a breach
CreateRemoteThread	Exclude known processes
RawAccessRead	Exclude known processes

Basic Event Recommendations (Cont)

Category	Recommendation
Process Access	Include sensitive targets
File Create	Include autostart entry points
Registry Object CreateDelete	Include autostart entry points
Registry Value Create	Include autostart entry points
Registry Object Rename	Include autostart entry points
File Create Stream Hash	Include MOTW names
Pipe Created	Enable for targeted investigation
Pipe Connected	Enable for targeted investigation

Example: Identifying Known Drivers

- Set this filter and reboot (assuming system is clean):

```
<DriverLoad onmatch="exclude"/>
```

- Query log for loaded drivers:

```
Get-WinEvent -LogName "Microsoft-Windows-
Sysmon/Operational"| Where-Object { $_.Id -eq 6 } | ForEach-
Object { $_.Properties[4].Value } | Sort-Object -Unique
```

- Make exclude filters from the list:

```
<DriverLoad onmatch="exclude">
  <Signature condition="is">Alps Electric Co.</Signature>
  <Signature condition="is">Conexant Systems</Signature>
```

SwiftOnSecurity's Configuration

- @SwiftOnSecurity (Securitay) has published a Sysmon configuration
  - Has been using Sysmon for over a year
  - Deployed across thousands of systems
  - Commented configuration explains rationale

  https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml

- Additional analysis from Lennart Koopman

  https://medium.com/@lennartkoopmann/explaining-and-adapting-tays-sysmon-configuration-27d9719a89a8#.rwt51mfrg

RSAConference2017

Hunting

Splunk

- Splunk enables collection and rich queries of Sysmon data
- Configuring Splunk for Sysmon: https://github.com/splunk/TA-microsoft-sysmon
  - Install Splunk universal forwarder on Sysmon systems
  - Install Splunk Sysmon TA on search heads
  - Set Sysmon configuration to exclude Splunk binaries

<Image condition=“contains">splunk</Image>
<Image condition=“contains">streamfwd</Image>
…

Demo: Hunting a Phishing Email

Operations Management Suite

- System monitoring and configuration for Windows and Linux systems (VMs, physical, cloud, etc.)
- Includes support for agent that can forward arbitrary logs to Operational Insights service
- Logs can be used for:
  - Standing dashboard queries
  - Visualization
  - Ad-hoc exploration

Demo: Credential Exfiltration

RSA Conference2017

Conclusion

Best Practices and Tips

- Install it on all your systems
  - Proven at scale
  - Data will be there when you need it for DFIR
- Configure all event types for maximum visibility
  - Filter out noise, especially uninteresting image loads
  - Test overhead on mission-critical systems
  - Make sure event log is large enough to capture desired time window
- Forward events off box
  - To prevent deletion by attackers
  - For analyzing aggregate network behavior
  - For tracing activity between systems (e.g. pass-the-hash)

Summary

- Sysmon can give you deep insights into intrusions and infections
- Send cases, tips and feature requests to me:

  mark.russinovich@microsoft.com
  @markrussinovich

- Sysmon and other Sysinternals tools are documented in "Troubleshooting with the Sysinternals Tools"