Slide 1

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

Slide 2

My name is Tom Ueltschi and I've been working for Swiss Post for over 9 years.
My current focus is: Malware Analysis, Threat Intel, Threat Hunting and Red Teaming.
Some of you may know me from my Ponmocup talks or trust groups that I'm active in.

Slide 3

Just a quick disclaimer. Views and opinions are my own.
The work presented is from my dayjob, although I also spent lots of spare time to prepare this talk.
It's more ideas and examples, not a solution or product you can plugin.
I prepared lots of slides, so I'll go over some of them quickly and try to focus on the big points.
The slides will become available to review later and all public resources are listed at the end (references slides)

Slide 4

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

Slide 5

So this is the best I could come up with.
First an introduction about Sysmon and in general.
Then covering different sources for «knowing bad» for detection (searching for known bad) and hunting.

Slide 6

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

Slide 7

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

Slide 8

This talk is about host-based detection, not about prevention or network-based detection.
I would put this approach in the EDR space along with solutions like Carbon Black etc.

Slide 9

So which is better, network-based or host-based detection?
My opinion is you need both.
Sometimes I almost feel like host-based detection can be better or more efficient though.

Slide 10

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

Slide 11

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

Slide 12

The triangle reminded me of the Pyramid of Pain, which should be mentioned in every great talk ?
I want to be able to detect and hunt for tools and TTPs.

Slide 13

Also, there should be a mention of the Kill Chain and the word Cyber (at least once) in every good talk.
I want to be able to detect all post exploitation phases of an intrusion.

Slide 14

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

Slide 15

So why use Sysmon?
Sysmon gives you incredible visibility into system activity on Windows hosts… and it's FREE.
Having this great data available in your Windows event logs for investigations and forensics is really useful.
However, if you can ingest Sysmon data into your SIEM it's even much more useful.
But your analyst(s) need to know what to search for, what's normal or abnormal, and what's suspicious or malicious.
Every company network is different and what works at one company may not work at another at all.

Slide 16

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

Slide 17

These are the Sysmon event types from version 4.
This presentation focusses on mostly three of them: process create, network connections and create remote thread, used for DLL / process injection

Slide 18

Some interesting fields for process create are:
- Image path and full command line from process and its parent process
- Different hashes (MD5, SHA-1, SHA-256, IMPHASH configurable)
- User starting the process
- ProcessGuid to correlate with other events

Slide 19

Some interesting fields for network connection are:
- Image path, user, protocol and if this host initiated the connection
- IP, hostname and port for source and destination
- Process guid for correlation

Slide 20

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

Slide 21

There were also some examples listed from a Splunk blog.

Slide 22

When I tweeted to Mark Russinovich thanking for the presentation…

Slide 23

… he replied with «cool to see people using Sysmon at scale» ?

Slide 24

Here's a brief high-level overview of our deployment.
Sysmon and Splunk Forwarder are installed on all workstations.
I put a lot of effort and time into tuning the Sysmon- and Splunk Forwarder configs to filter some data before storing in event log and before sending it to Splunk.
We have this deployed on over 20K hosts and ingesting about 15GB of data per day into Splunk.

Slide 25

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

Slide 26

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

Slide 27

A couple years ago SANS put out this great DFIR poster
Know normal, know abnormal, find evil

Slide 28

Many of the mentioned anomalies «when searching for malicious processes» can be implemented using Sysmon.
Parent- / child-process relationships
Command line arguments
Wrong or known malicious image path

Slide 29

To repeat, let me quote this:
«In an intrusion case, spotting the difference between abnormal and normal is often the difference between success and failure»

Slide 30

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

Slide 31

Just knowing this you can search for and alert on abnormal «svchost.exe» processes which are
- missing a «-k» parameter
- not started by services.exe
- running from a wrong path
If you don't have too many different Windows versions in your network you can even whitelist known good hashes

Slide 32

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

Slide 33

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

Slide 34

A couple days later there were 8 AVs detecting it.

Slide 35

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

Slide 36

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

Slide 37

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

Slide 38

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

Slide 39

This alert is detecting any of these behaviors
- VBS (Retrive<random>.vbs) scripts executed by cscript.exe
- xcopy being used to copy the legitimate JRE to a path under APPDATA
- Java executable started from this abnormal path (never seen used legitimately)

Slide 40

This alert is detecting
- «reg add» being used to create a registry Run key for persistence

Slide 41

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

Slide 42

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

Slide 43

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

Slide 44

I was looking thru my own malware samples and found the same sample from the blog.
I also found other samples with the same behavior, up to 6 days earlier.
The common behavior is an office process (e.g. winword.exe) spawning a system process (e.g. explorer.exe, svchost.exe) to be abused for process hollowing.

Slide 45

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

Slide 46

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

Slide 47

This alert detects process injection from office processes into a system process (explorer or svchost).
After finishing this slide I found out that «create remote thread» is not used in the process hollowing technique.
So this alert won't detect this behavior.
(a new event type in Sysmon version 5 should be able to detect this though)

Slide 48

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

Slide 49

This is a high-level overview of our automated malware analysis process.
Inputs are files or emails, where attachments are extracted and decompressed if necessary.
The sample is uploaded to a sandbox and the analysis results are downloaded when it's finished.
The post processing is extracting
- files, reg keys, processes created
- dns, http requests and tcp connections
- Yara rule matching on files, memory strings, pcap
Behavior analysis looks for specific patterns and is then used to create Splunk searches and alerts

Slide 50

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

Slide 51

Let take a look back at the Java Adwind RAT family.
From 132 Java malware samples analyzed, 122 (>90%) would be detected by the «reg add» alert (first seen in Jan 2015),
and 118 (almost 90%) would be detected by the other two detections (xcopy JRE, Java process started from known bad path – first seen in Oct 2015)

Slide 52

Now let's take a look at some Keylogger (and password stealer) families.
Here are 5 keylogger families which all use the «/stext» parameter and some TXT filename after that.
The TXT filenames vary between keylogger families. In red are the number of samples analyzed per family, in total almost 350.
They also all have in common that they abuse the NirSoft tools for password «recovery»

Slide 53

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

Slide 54

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

Slide 55

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

Slide 56

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

Slide 57

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

Slide 58

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

Slide 59

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

Slide 60

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

Slide 61

NirSoft's Product Key Recovery uses a «/stext» parameter.
And probably some more.
Just recently I found other kaylogger families using «/scomma» or « -f » parameter, which is not covered here.

Slide 62

Now let's take a look at ransomware, especially Locky, which has been heavily spammed out this year.
I try to (almost) daily analyse at least one sample from each malspam run seen (and blocked) at work.
Then I look for changes in behavior and adjust alerts accordingly.
Let's compare two Locky samples from April and August of this year.

Slide 63

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

Slide 64

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

Slide 65

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

Slide 66

This alert detects a «rundll32.exe» process started with either
- TEMP path and «qwerty» parameter in command line
- parent process is used for know malspam filetypes (JS, VBS, WSF, HTA, DOC/XLS)

Slide 67

These are some of the behaviors detected from Locky samples.
- locky, zepto, odin files dropped
- One of these two HTML filenames dropped
- «vssadmin delete shadows» called
- rundll32 with «qwerty» parameter started

Slide 68

These are Yara rules detecting the different Locky variants from their POST request URI patterns.
These URI patterns change every few months, weeks, (days)
In (late August?) September Locky started using XOR to encrypt the executable payload download
(presumably to bypass executable download blocking)

Slide 69

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

Slide 70

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

Slide 71

In November they started chainging the DLL parameters almost daily or for every malspam run.
One variant used a *.44 instead of a DLL extension for the executable.

Slide 72

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

Slide 73

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

Slide 74

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

Slide 75

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

Slide 76

The Powershell Webclient.Downloadfile behavior has been seen in over 80 samples since Feb 2015.
The (simple) Powershell obfuscation was first seen end of September.

Slide 77

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

Slide 78

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

Slide 79

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

Slide 80

Just to add to this, here's a sample from Nov 18th where they started using «string replacement» which can't be easily deobfuscated without a complex script.
(Splunk could call a [Python] script to deobfuscate more techniques – to be tried out soon maybe)

Slide 81

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

Slide 82

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

Slide 83

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

Slide 84

David created a web site for the threat hunting project.

Slide 85

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

Slide 86

One hunt for example is lateral movement detection via process monitoring

Slide 87

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

Slide 88

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

Slide 89

At the end of this description there is a statement:
«Sysmon is a very good free tool that can do nearly anything you'd need»

Slide 90

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

Slide 91

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

Slide 92

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

Slide 93

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

Slide 94

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

Slide 95

Cobalt Strike makes heavy use of Powershell and its features.
But even when using cobalt strike, red teamers commonly use system commands and tools like «whoami»

Slide 96

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

Slide 97

A lot of features in CS make use of DLL / process injection.

Slide 98

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

Slide 99

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

Slide 100

So before you can start hunting for certain things you need to ask yourself some questions.
Can you distinct between workstations and servers / NAS / filers?
Is SMB traffic between workstations (WS) normal?
Is «whoami /groups» normal activity from users / admins?
How common is DLL / process injection? (can be legit)
Can you distinguish benign from malicious injection?
How common is Powershell usage?
EncodedCommand? Invoke-Expression (IEX)?
Parent processes / user accounts running legit Powershell?

Slide 101

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

Slide 102

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

Slide 103

This randomly named executable spawns a rundll32.exe process.

Slide 104

And then it uses DLL injection to inject the CS beacon payload into the rundll32 process.
You can hunt for this searching for the rarest source or target images from injections.

Slide 105

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

Slide 106

A few more ideas for hunting.
Search for processes connecting to the proxy (or Internet directly if not blocked)
and look for rarest processes by count of hashes, hostnames, image paths or names
Search for Powershell using encoded command and filter out legitimate usage.

Slide 107

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

Slide 108

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

Slide 109

This query detects usage of Powershell encoded command.
Often the abbreviation «-enc» is also used, which would also match the «-encoding» parameter.
This one is filtered out by the purple replace command.
For alerting there may be some filtering and tuning needed.
For hunting this should be very useful.

Slide 110

To wrap up just a quick conclusion.
I've shown you examples how to search and alert for known malicious activity by
- Image names and paths like svchost and java (adwind rat)
- Command line parameters like stext, delete shadows and qwerty
- parent- child-process relationships for certain infection vectors
- Process injection for keylogging

Slide 111

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

Slide 112

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

Slide 113

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