INE- Cyber Sec

OUTLINE

Table of Contents

Module 04 | Endpoint Analytics

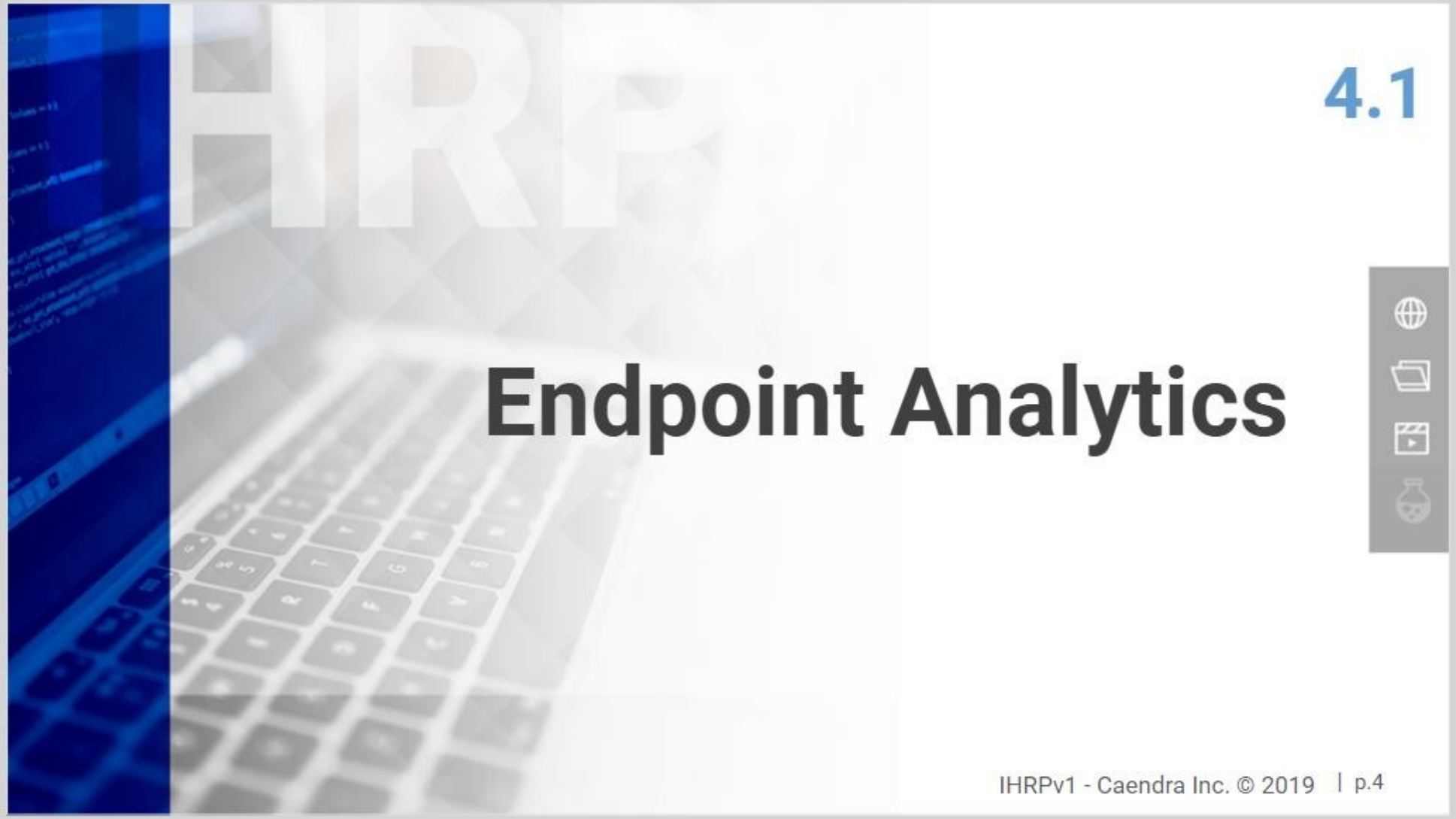
4.1 Endpoint Analytics

OUTLINE

Learning Objectives

By the end of this module, you should have a better understanding of:

- ✓ How to translate tactical threat intelligence into actionable SIEM queries



4.1

Endpoint Analytics

OUTLINE

4.1 Endpoint Analytics

It is about time we put what we have learned about ELK and Splunk to the test.

This module will be dedicated to translating attacker TTPs (a.k.a Tactical Threat intelligence) into actionable SIEM queries / searches.

Note: Covered TTPs will **not** follow the cyber kill chain's order of events.

OUTLINE

4.1.1 Attackers Leveraging Native Windows Binaries

Detection

A Splunk search that can identify Windows binaries being used to execute malicious code is the following.

You can try this command on the “Effectively Using Splunk” lab.

```
index=botsv1 source="WinEventLog:Microsoft-
Windows-Sysmon/Operational"
Image="*\\powershell.exe" OR
Image="*\\msbuild.exe" OR
Image="*\\psexec.exe" OR Image="*\\at.exe" OR
Image="*\\schtasks.exe" OR Image="*\\net.exe"
OR Image="*\\vssadmin.exe" OR
Image="*\\utilman.exe" OR Image="*\\wmic.exe"
OR Image="*\\mshta.exe" OR
Image="*\\wscript.exe" OR
Image="*\\cscript.exe" OR Image="*\\cmd.exe"
OR Image="*\\whoami.exe" OR Image="*\\mmc.exe"
OR Image="*\\systeminfo.exe" OR
Image="*\\csvde.exe" OR
Image="*\\certutil.exe" | stats
values(CommandLine) by Image
```

4.1.2 Remote Privileged User Enumeration

Attackers are known for performing remote privileged user enumeration through *net.exe*.

4.1.2 Remote Privileged User Enumeration

Detection

A Splunk search that can identify remote privileged user enumeration through *net.exe* is the following.

```
index=your_index
sourcetype="xmlwineventlog:microsoft-windowssysmon/operational"
process="*\\net.exe"
(CommandLine="*net  group*" OR
CommandLine="*net  localgroup*")
| stats count by
Computer,CommandLine
```


OUTLINE

4.1.3 PowerShell Executing An Encoded Script

Attackers are known for concealing their malicious PowerShell scripts through Base64 encoding. These scripts are loaded and executed into the target's memory using commands such as

```
powershell.exe -enc <long_Base64_string>
```

4.1.3 PowerShell Executing An Encoded Script

Detection

A Splunk search that can identify PowerShell executing an encoded script is the following.

```
index=your_index
sourcetype="xmlwineventlog:microsoft-windowssysmon/operational"
process="*\\powershell.exe"
(CommandLine="*-encodedcommand*" OR CommandLine="*-enc*" OR CommandLine="-e" OR CommandLine="-ec" OR CommandLine="-encodedcomman" OR CommandLine="-encodedcomma" OR CommandLine="-encodedcomm" OR CommandLine="-encodedcom" OR CommandLine="-encodedco" OR CommandLine="-encodedc" OR CommandLine="-encoded" OR CommandLine="-encode" OR CommandLine="-encod" OR CommandLine="-enco" OR CommandLine="-en")| stats count by CommandLine | top CommandLine
```

4.1.4 Mimikatz (Binary)

If you recall, we have talked about attackers using Mimikatz to perform credential theft and reuse.

OUTLINE

4.1.4 Mimikatz (Binary)

Detection

A Splunk search that can identify Mimikatz's binary being executed on an endpoint is the following.

```
index=your_index
sourcetype="xmlwineventlog:microsoft-
windowssysmon/operational"
CommandLine="*privileges::debug*" OR
CommandLine="*sekurlsa::*" OR
CommandLine="*kerberos::*" OR
CommandLine="*crypto::*" OR
CommandLine="*lsadump::*" OR
CommandLine="*process::*"
```

4.1.5 PSExec

As we have already mentioned Microsoft's *PSExec* can be used for lateral movement purposes.

4.1.5 PSExec

Detection

A Splunk search that can identify *PSExec* being executed on an endpoint (through its IMPHASH) is the following.

```
index=your_index
sourcetype="xmlwineventlog:microsoft-
windowssysmon/operational"
Hashes="*IMPHASH=B18A1401FF8F444056D29
450FBC0A6CE*" NOT
process="*PsExec.exe"
```

This search assumes PSExec has been renamed.

https://www.fireeye.com/blog/threat-research/2014/01/tracking-malware-import-hashing.html

4.1.6 rundll32

Attackers have been abusing Windows *rundll32* for years. Specifically, they are using the *rundll32* binary to load malicious DLLs.

OUTLINE

4.1.6 rundll32

Detection

A Splunk search that can identify *rundll32* being executed on an endpoint is the following.

This search also covers the case of Office (or other) binaries calling *rundll32*.

```
index=your_index
sourcetype="xmlwineventlog:microsoft-
windowssysmon/operational" 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")
```

4.1.7 Beaconing Malware

A large percentage of the malware in the wild perform some kind of beaconing.

Beaconing usually occurs by malware during initial “check in”, or when malware is expecting an update.

4.1.7 Beaconing Malware

Detection

A Splunk search that can identify a malware beaconing is the following.

This search leverages ingested DNS information.

```
index=botsv1 source="stream:dns"
message_type="QUERY"
| fields _time, query
| streamstats current=f last(_time) as
last_time by query
| eval gap=last_time - _time
| stats count avg(gap) AS
AverageBeaconTime var(gap) AS
VarianceBeaconTime BY query
| eval
AverageBeaconTime=round(AverageBeaconTime,
3),
VarianceBeaconTime=round(VarianceBeaconTim
e,3)
| sort -count
| where VarianceBeaconTime < 60 AND count
> 2 AND AverageBeaconTime>1.000
| table  query VarianceBeaconTime  count
AverageBeaconTime
```

https://www.splunk.com/blog/2018/03/20/hunting-your-dns-dragons.html

4.1.8 Malicious PowerShell Activity

We remind you of Symantec's excellent report on PowerShell attacks. Let's try to detect some of that activity.

https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/increased-use-of-powershell-in-attacks-16-en.pdf

OUTLINE

4.1.8 Malicious PowerShell Activity

Detection

A Splunk search that can identify malicious PowerShell activity is the following.

```
index=botsv1 EventID=4688
(BaseFileName=powershell.exe OR
BaseFileName=powershell_ise.exe OR
BaseFileName=cmd.exe) (Copy-Item OR .CopyHere OR
New-Object OR WebClient OR DownloadFile OR
downloadstring OR WebRequest OR restmethod)
(CommandLine="*Copy-Item*" OR
CommandLine="*CopyHere*" OR CommandLine="*New-
Object*" OR CommandLine="*WebClient*" OR
CommandLine="*DownloadFile*" OR
CommandLine="*downloadstring*" OR
CommandLine="*WebRequest*" OR
CommandLine="*restmethod*" OR
CommandLine="*iex*" OR
CommandLine="*comobject*InternetExplorer*" OR
CommandLine="*Msxml2.XMLHTTP*" OR
CommandLine="*WinHttp*" OR
CommandLine="*bitstransfer*" | table _time,
Computer, SubjectDomainName, SubjectUserName,
BaseFileName, CommandLine, CreatorProcessName,
NewProcessName, FileDescription, FileVersion,
MD5
```

4.1.8 Malicious PowerShell Activity

Detection

Similarly we can check if an Office or Adobe application called PowerShell as follows

```
index=botsv1 EventID=4688
(CommandLine="*powershell")
(CreatorProcessName="WINWORD" OR
CreatorProcessName="POWERPNT" OR
CreatorProcessName="EXCEL" OR
CreatorProcessName="Adobe*") | table _time,
host, SubjectUserName, CreatorProcessName,
BaseFileName, CommandLine
```

4.1.9 Unauthorized DNS Server Interactions

During DNS tunneling activities attackers utilize a DNS server under their control.

OUTLINE

4.1.9 Unauthorized DNS Server Interactions

Detection

A Splunk search that can identify unauthorized DNS servers is the following.

This search assumes your intranet range is 10.0.0.0/8.

```
index=your_index
sourcetype=stream:dns
dest_port=53  dest_ip!=
10.0.0.0/8 | stats dc(src_ip)
values(src_ip) by dest_i
```

4.1.9 Unauthorized DNS Server Interactions

Detection

A Splunk search that can identify unauthorized DNS servers is the following.

This search assumes your intranet range is 10.0.0.0/8.

```
index=your_index
sourcetype=stream:dns
dest_port=53  dest_ip!=
10.0.0.0/8 | stats dc(src_ip)
values(src_ip) by dest_i
```

4.1.10 SQL Injection

SQL injection attacks leverage weak user input sanitization to execute arbitrary SQL (or even OS) commands on the vulnerable server.

https://www.acunetix.com/blog/articles/exploiting-sql-injection-example/

OUTLINE

4.1.10 SQL Injection

Detection

A Splunk search that can identify SQL injection attempts (against an IIS server) is the following.

This search assumes Splunk has ingested IIS logs.

You can try this command on the "Effectively Using Splunk" lab.

```
index=botsv1 sourcetype="iis"  | regex
cs_uri_query="(?i)(?:--
|\;|\/\*|\@|\@\@version|char|alter|begi
n|cast|create|cursor|declare|delete|dro
p|end|exec|fetch|insert|kill|open|selec
t|sys|table|update)"
| stats count by host c_ip cs_uri_stem
cs_uri_query
| rex field=cs_uri_query
"(?i)(?<suspect>--
|\;|\/\*|\@|\@\@version|char|alter|begi
n|cast|create|cursor|declare|delete|dro
p|end|exec|fetch|insert|kill|open|selec
t|sys|table|update)" max_match=0
```

4.1.11 WMI Persistence

We have already covered that WMI-based persistence could be detected through Sysmon Event IDs 19, 20 and 21. In addition, Events like 5858 could also help in detecting WMI persistence.

4.1.12 UAC Bypass Through Windows Event Viewer

A UAC bypass has been discovered some years ago, that uses the Windows Event Viewer and a technique known as registry hijacking. For more information refer to the resource below:
https://enigma0x3.net/2016/08/15/fileless-uac-bypass-using-eventvwr-exe-and-registry-hijacking/

Let's see how we can detect this through an ELK search.

https://docs.microsoft.com/en-us/windows/security/identity-protection/user-account-control/how-user-account-control-works
https://enigma0x3.net/2016/08/15/fileless-uac-bypass-using-eventvwr-exe-and-registry-hijacking/

4.1.12 UAC Bypass Through Windows Event Viewer

Detection

An ELK search that can identify this UAC bypass is the following.

```
( event_id:("1" "4688") AND
event_data.ParentImage:"*\\eve
ntvwr.exe" AND -
event_data.Image:"*\\mmc.exe"
) OR ( event_id:13 AND
event_data.TargetObject:"*\\ms
cfile\\shell\\open\\command")
```


OUTLINE

4.1.13 net.exe Accessing an Administrative Share

net.exe can be used by attackers to remotely access an administrative share (if valid credentials have been obtained).

4.1.13 net.exe Accessing an Administrative Share

Detection

An ELK search that can identify *net.exe* accessing an administrative share is the following.

```
event_data.CommandLine:*net*
AND
event_data.CommandLine:"*use
*" AND
event_data.CommandLine.keyword
:*$*
```


OUTLINE

4.1.14 Lateral Movement via Scheduled Tasks

Attackers abuse Windows scheduled tasks not only for persistence but also for lateral movement.

https://ired.team/offensive-security/t1053-schtask

OUTLINE

4.1.14 Lateral Movement via Scheduled Tasks

Detection

An ELK search that can identify lateral movement through scheduled tasks is the following.

```
event_id:("4698" "4702") AND
event_data.LogonType:3
```

4.1 Endpoint Analytics

After studying the ELK and Splunk labs as well as the 13 examples above you should have a good idea of how attacker TTPs can be translated to actionable SIEM queries.

We encourage you to keep doing this for every new TTP you come across…

VIDEO: Osquery Fundamentals and Endpoint Analysis

Osquery Fundamentals and Endpoint Analysis

In this video, we are going to cover osquery fundamentals, how to interrogate an endpoint through osquery, how to scale osquery and finally, some of osquery's advanced functionalities and applications. During the video osquery will be leveraged to detect stealthy malware, registry persistence, fileless malware, suspicious kernel modules etc.

**Videos are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the resources drop-down in the appropriate module line.*

OUTLINE

References

References

IMPHASH

https://www.fireeye.com/blog/threat-research/2014/01/tracking-malware-import-hashing.html

Hunting Your DNS Dragons

https://www.splunk.com/blog/2018/03/20/hunting-your-dns-dragons.html

The Increased Use Of Powershell In Attack

https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/increased-use-of-powershell-in-attacks-16-en.pdf

Exploiting SQL Injection: a Hands-on Example

https://www.acunetix.com/blog/articles/exploiting-sql-injection-example/

References

UAC

https://docs.microsoft.com/en-us/windows/security/identity-protection/user-account-control/how-user-account-control-works

“Fileless” UAC Bypass Using eventvwr.exe and Registry Hijacking

https://enigma0x3.net/2016/08/15/fileless-uac-bypass-using-eventvwr-exe-and-registry-hijacking/

T1053: Schtask

https://ired.team/offensive-security/t1053-schtask

Videos

Osquery Fundamentals and Endpoint Analysis

In this video, we are going to cover osquery fundamentals, how to interrogate an endpoint through osquery, how to scale osquery and finally, some of osquery's advanced functionalities and applications. During the video osquery will be leveraged to detect stealthy malware, registry persistence, fileless malware, suspicious kernel modules etc.

**Videos are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the resources drop-down in the appropriate module line.*

Videos

Osquery Fundamentals and Endpoint Analysis

In this video, we are going to cover osquery fundamentals, how to interrogate an endpoint through osquery, how to scale osquery and finally, some of osquery's advanced functionalities and applications. During the video osquery will be leveraged to detect stealthy malware, registry persistence, fileless malware, suspicious kernel modules etc.

**Videos are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the resources drop-down in the appropriate module line.*