



CASE STUDY

# ANÁLISE FORENSE & MALWARE

A UM ATAQUE DE RANSOMWARE  
IDENTIFICAÇÃO DA ORIGEM DO INCIDENTE

# ÍNDICE

- Contexto.....pág.2
- Análise.....pág.3
- Conclusão.....pág.8

# ÍNDICE DE FIGURAS

- **Figura 1** - Autoruns da máquina.....pág.3
- **Figura 2** - Ficheiros na pasta.....pág.3
- **Figura 3** - Del.cmd.....pág.4
- **Figura 4** - Log.cmd.....pág.4
- **Figura 5** - Data criação, na registry, da conta Administratr.....pág.6
- **Figura 6** - Vestígios de mimikatz e port scanner, no registo de ficheiros acedidos.pág.6
- **Figura 7** - Vestígios de execução de mimikatz e port scanner, na registry.....pág.6
- **Figura 8** - URL usado para explorar a vulnerabilidade.....pág.7
- **Figura 9** - URL usado aquando do crash.....pág.7
- **Figura 10** - Pedidos HTTP a explorar a vulnerabilidade.....pág.8



# CONTEXTO

A Hardsecure foi contactada pelo cliente X no seguimento de um comprometimento de um dos seus servidores, com um ataque de *ransomware*, onde o cliente pretendeu determinar a origem da intrusão.

Fase inicial da recolha de artefactos:

- Cliente: “Deixámos de conseguir aceder aos servidores, as contas da AD não funcionavam. Conseguimos mudar a password de admin e restaurar o acesso.”
- Foi detetado uma máquina que tinha os ficheiros encriptados, numa 1ª análise esta foi a única máquina com *ransomware*.
- Só ao analisar os *logs* da *firewall* é que se aperceberam de um alarme disparado pela mesma no dia 17 de maio às **23:48**, de referir que as datas dos sistemas analisados alteram por vezes em uma hora.
- Embora tenha sido efetuado um *snapshot* da máquina para análise, este *snapshot* já foi feito depois de feita uma "limpeza" de *malware* na máquina.
- Não tinha *logs* centralizados, apenas os localizados na *firewall*.
- Algo que intrigou o cliente foi o facto de esta máquina não ter serviços expostos para a internet.

Quando foi efetuado o contacto com a Hardsecure, o cliente nesta fase já tinha efetuado uma higienização da infraestrutura e das contas dos utilizadores.

Foi dado acesso à infraestrutura do cliente para a Hardsecure poder iniciar o processo de investigação.

# ANÁLISE

A máquina, **10.0.0.20 (Xserver)**, foi restaurada do *snapshot* para a Hardsecure poder analisar a mesma. Depois de terem sido extraídos os artefactos do sistema para análise forense, após uma análise live à máquina foi descoberto de imediato um ficheiro, **Del.cmd** que estava nos *autoruns* da máquina, estando, portanto, feito de maneira a ser executado assim que a máquina iniciasse.

Name	Path	Date Modified
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup		18-05-2021 00:51
CentralGest Atualização...	(Verified) CENTRALG... c:\program files (x86)\central...	11-02-2021 17:06
Del.cmd	c:\programdata\microsoft\win...	18-05-2021 00:51
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components		16-01-2021 23:04

Figura 1 - Autoruns da máquina.

Este mesmo ficheiro é um *script* que vai eliminar um executável no caminho: **C:\Users\Public\Videos\Sys.exe**

Durante o processo de procura de artefactos nesta pasta, também foram identificados outros ficheiros relativos ao *ransomware*:

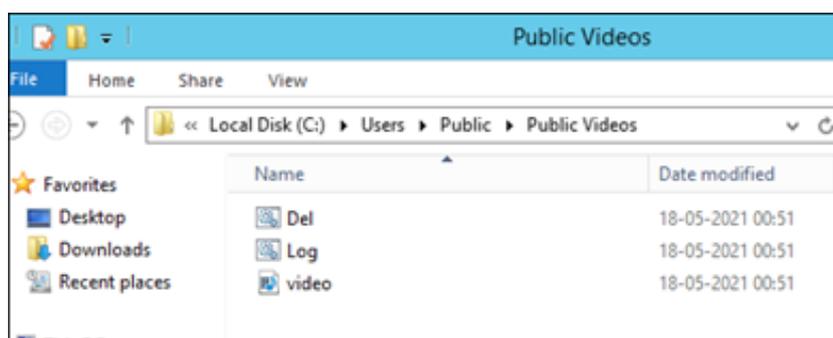
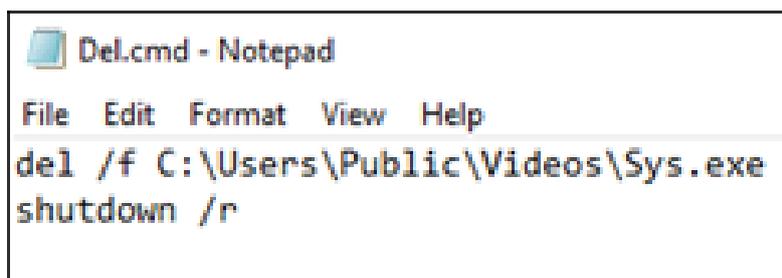


Figura 2 - Ficheiros na pasta

Portanto por aqui já temos algumas evidências que houve de facto, software malicioso executado na máquina.

O ficheiro **SYS.exe** já não se encontrava na pasta, o ficheiro **log.cmd** serve para apagar os logs da máquina, o ficheiro **del.cmd** é o mesmo que encontrado nos *autoruns* e o **video.mp4** é um vídeo a mostrar o *ransomware* em execução e como eles conseguem cifrar e decifrar os ficheiros.



```

Del.cmd - Notepad
File Edit Format View Help
del /f C:\Users\Public\Videos\Sys.exe
shutdown /r

```

Figura 3 - Del.cmd



```

Log.cmd - Notepad
File Edit Format View Help
@echo off
FOR /F "tokens=1,2*" %V IN ('bcdedit') DO SET adminTest=%V
IF (%adminTest%)==(Access) goto noAdmin
for /F "tokens=" %G in ('wevtutil.exe el') DO (call :do_clear "%G")
echo.
echo goto theEnd
:do_clear
echo clearing %1
wevtutil.exe cl %1
goto :eof
:noAdmin
exit

```

Figura 4 - Log.cmd

De seguida, tendo esta informação, a Hardsecure executou a ferramenta KAPE, produzida por Eric Zimmerman (@EricRZimmerman), ferramenta esta que irá extrair uma vasta quantidade de artefactos do sistema para posterior análise.

Depois de extraídos os artefactos, a ferramenta permite, para além de outras tarefas, a criação de uma linha temporal global juntando todos os *Event Logs* do sistema, o que permite uma melhor compreensão do que se está a passar no sistema em cada momento.

Foi então nesta análise que foram identificados dois eventos importantes:

- 17/05/2021 11:52:03 pm
- » *Logs cleared by user batchaccount*
  
- 17/05/2021 11:54:15 pm
- » *Remote Desktop Services: Session has been disconnected for user DOMAIN\batchaccount address 10.0.0.21*

Portanto os *logs* foram eliminados às **23:52**, não permitindo ver o que aconteceu antes, contudo posteriormente, foi identificado às **23:54** o *logoff* de uma conta e um IP de origem, no qual pressupõe-se que foi este utilizador que se ligou e eliminou os *logs*, escondendo parcialmente as suas ações.

Antes de ser continuada a investigação ao servidor **10.0.0.21 (WebServer)**, procedemos ao acesso à máquina **10.0.0.10 (DCServer)** de forma a extrair os artefactos da mesma, para procura de evidências de atividade maliciosa, onde identificamos esta entrada nos *logs*:

- 17/05/2021 11:16:59 pm
- » *Remote Desktop Services: Session logon succeeded for account **DOMAIN\batchaccount** address **10.0.0.20**.*

Através da análise ao evento, verificamos que a ligação foi feita a partir da primeira máquina analisada, a **10.0.0.20 (Xserver)**, mas numa data anterior à que temos nos *logs*, dado terem sido eliminados, portanto partimos para a próxima máquina a ser analisada, a **10.0.0.21 (WebServer)**.

Extraímos então, novamente, os artefactos desta máquina, **10.0.0.21 (WebServer)**, e dado ser um servidor web, extraímos também os *logs* do IIS.

Através da análise aos *logs* de IIS pela data do alarme, não foi possível isolar nenhum pedido, ou conjunto de pedidos, em particular.

Procedemos então para a análise da linha temporal dos eventos e foi nesta máquina que encontramos os eventos mais perto da altura do alarme disparado pela *firewall*, pelas **22:45**, e foi a sequência dos próximos eventos que revelou o que aconteceu:

- 17/05/2021 10:45:52 pm
- » *ASP.NET application crash report, IIS APPPOOL\XApp, failed to load file **file:///C:/Windows/Temp/1621291550.8233094.dll***
- 17/05/2021 10:46:53 pm
- » *FW rule added to exception list, \*: \*, : RDP Port 3389, Direction: 1, ModifyingApplication: C:\\Windows\\System32\\netsh.exe*
- 17/05/2021 10:48:54 pm
- » *Remote Desktop Services: Session logon succeeded, account **WEBSERVER\Administratr**, address ::%**16777216***

Ora nesta sequência vemos que houve um *crash* da aplicação ao tentar carregar uma biblioteca, pouco depois é criada uma exceção na *firewall* a permitir todo o tráfego **RDP** e um *login* na conta **Administratr** com um endereço ao que aparenta, corrompido/desconhecido.

O primeiro crash ao carregar a biblioteca não foi imediatamente evidente, e portanto, começámos por investigar se a regra de *firewall* existia, e de facto estava ativa. Após isso, analisamos a conta **Administatr**, tão parecida em nome, com uma conta legítima de administrador, e verificámos que a mesma foi criada, de acordo com os registos na *registry*, às **22:46**, pouco tempo antes do primeiro *login*, com o endereço desconhecido:

CreatedOn	LastLoginTime	LastPasswordChange	LastIncorrectPassword	ExpiresOn	UserName	FullName	Password	Groups
17/05/2021 22:46:52	17/05/2021 23:10:46	17/05/2021 22:46:52			Administatr			Administrators, Users

**Figura 5** - Data criação, na *registry*, da conta *Administatr*

Na pasta de *Desktop* desta conta foram encontrados vestígios de utilização do **mimikatz**. Para além de um *port scanner*, o **mimikatz** permite a extração de *passwords* guardadas na máquina ou na memória da mesma e o *port scanner* permite encontrar portas abertas noutras máquinas, e foi nos resultados do **mimikatz** que encontramos a conta **batchaccount** com a *password* em claro, esta foi a conta que identificamos nos *logs* da primeira máquina analisada a fazer *logoff* e a eliminar os *logs* de sistema.

file:///C:/Users/Administatr/Desktop/KPortScan%203.0/results.txt	17/05/2021 23:52:56	1
file:///C:/Users/Administatr/Desktop/KPortScan%203.0[sentorion].zip	17/05/2021 23:52:15	1
file:///C:/Users/Administatr/Desktop/mimikatz_trunk.zip	17/05/2021 23:49:50	1

**Figura 6** - Vestígios de *mimikatz* e *port scanner*, no registo de ficheiros acedidos

HiveType	Description	Category	KeyPath	ValueName	ValueType	ValueData	ValueData2
NTUser	UserAssist	Program Execution	CsiTool-CreateHive-Zvpebfbsg.Nhgb1 (plugin)			Microsoft.AutoGenerated.(Unmapped GUID: 923DD477-5846-686)	Last executed:
NTUser	UserAssist	Program Execution	CsiTool-CreateHive-Zvpebfbsg.Jvaqbj (plugin)			Microsoft.Windows.Explorer	Last executed:
NTUser	UserAssist	Program Execution	CsiTool-CreateHive-P:\Hfref\Ngzvavf (plugin)			C:\Users\Administatr\Desktop\x64\mimikatz.exe	Last executed: 2021-05-17 22:49:58.3700000
NTUser	UserAssist	Program Execution	CsiTool-CreateHive-[1NP14R77-02R7- (plugin)			[System32]wuauclt.exe	Last executed:
NTUser	UserAssist	Program Execution	CsiTool-CreateHive-P:\Hfref\Ngzvavf (plugin)			C:\Users\Administatr\Desktop\KPortScan 3.0\KPortScan3.exe	Last executed: 2021-05-17 22:52:22.6760000

**Figura 7** - Vestígios de execução de *mimikatz* e *port scanner*, na *registry*

Pressupomos então que foi a partir desta máquina e com as credenciais obtidas pelo **mimikatz** que conseguiram acesso às outras máquinas.

Após recolha deste artefacto, fomos à procura do que foi executado no qual permitiu a criação de uma conta local no grupo dos **Administrators** e a criação da exceção na *firewall*. Ao analisarmos o *log* acerca do *crash* ao carregar a biblioteca **.dll**, fomos ver se o ficheiro **.dll** estava presente na pasta onde a aplicação tentou carregar o mesmo, e assim que efetuamos o acesso à pasta **C:\Windows\Temp**, o antivírus de imediato detetou o ficheiro **1621291550.8233094.dll** como **malicioso**.

Chegado a este ponto, podemos presumir duas coisas, que o atacante conseguiu fazer **upload de um ficheiro** e conseguiu **execução de código** no sistema, agora a pergunta é, como?

De início foi suposto que poderia ser uma ou duas vulnerabilidades na aplicação web, e que para tal, teria de ser feito um *pentest* à mesma para tentar encontrar as mesmas.

Mais tarde tivemos uma surpresa, e é aqui que se destaca a importância de ir registando tudo o que se vai analisando aquando de uma investigação, mesmo quando parece não ter relação com o resto, pois ao ler uma notícia, que era referente às vulnerabilidades mais utilizadas, vemos uma referência a uma vulnerabilidade no **Telerik**, o CVE-2019-18935. O **Telerik** é uma solução que permite criar de forma acelerada aplicações para diversos sistemas operativos, e foi então que verificámos que o **WebServer** também utilizava **Telerik**.

Analisando mais em detalhe a vulnerabilidade, [CVE-2019-18935](#), vemos que existem duas vulnerabilidades associadas à mesma, uma de **file upload** e outra de **execução de código remoto**, ora, precisamente o que tínhamos suposto que necessitava acontecer para validar a sequência de eventos.

Procedemos à procura de *exploits* públicos para este **CVE** e ao analisar um *exploit*, verificámos qual o *URL* que é utilizado:

```
"\n" +
"Decrypt a plaintext:      -d ciphertext\n" +
"Decrypt rauPostData:    -D rauPostData\n" +
"Encrypt a plaintext:     -e plaintext\n" +
"Gen rauPostData:        -E TempTargetFolder Version\n" +
"Gen rauPostData (quiet): -Q TempTargetFolder Version\n" +
"Version in HTTP response: -v url\n" +
"Generate a POST payload: -p TempTargetFolder Version c:\\\\folder\\\\filename\n" +
"Upload a payload:       -P TempTargetFolder Version c:\\\\folder\\\\filename url\n\n"
"Example URL:           http://target/Telerik.Web.UI.WebResource.axd?type=rau"
```

Figura 8 - URL usado para explorar a vulnerabilidade

Relacionando isto com os *logs* de erro de quando a aplicação *crashou* ao carregar a biblioteca maliciosa, podemos reparar que o mesmo URL foi usado:

```
Evidence, byte[] mimeType, Assembly mimeTypeAlgorithm, Boolean forInspection, Boolean suppressSecurity
, https://443/Telerik.Web.UI.WebResource.axd?type=rau, /Telerik.Web.UI.WebResource.axd,
```

Figura 9 - URL usado aquando do crash

Com esta informação, fomos então aos *logs* do IIS confirmar se existiam pedidos nesta altura, e confirmamos que a vulnerabilidade foi utilizada, precisamente na altura do *crash* da aplicação web:

```
2021-05-17 22:45:51 10.0.50.21 POST /Telerik.Web.UI.WebResource.axd type=rau 443 -
2021-05-17 22:45:52 10.0.50.21 POST /Telerik.Web.UI.WebResource.axd type=rau 443 -
2021-05-17 22:45:57 10.0.50.21 POST /Telerik.Web.UI.WebResource.axd type=rau 443 -
2021-05-17 22:45:58 10.0.50.21 POST /Telerik.Web.UI.WebResource.axd type=rau 443 -
2021-05-17 22:46:37 10.0.50.21 POST /Telerik.Web.UI.WebResource.axd type=rau 443 -
2021-05-17 22:46:38 10.0.50.21 POST /Telerik.Web.UI.WebResource.axd type=rau 443 -
```

Figura 10 - Pedidos HTTP a explorar a vulnerabilidade

Verificámos também que existe um módulo do *metasploit* para esta vulnerabilidade. O **metasploit** é uma ferramenta ofensiva que, entre outras coisas, facilita a exploração de vulnerabilidades, e se tal foi usada, podemos presumir que o atacante usou uma *reverse shell* de **meterpreter**. O **meterpreter** é uma *reverse shell* que possui muitas funcionalidades para ajudar um atacante após acesso ao sistema, e usou a sua funcionalidade de *port-forwarding* para criar um túnel que lhe permitiu ligar-se por **RDP** à máquina e daí termos a criação de uma regra na *firewall* a permitir o acesso **RDP**, e aquele primeiro *login* na conta **Administatr** com uma origem que não é um endereço IP, pois o mesmo está a surgir da máquina para ela mesma através do túnel.

Desta forma, foi detetada a origem da intrusão, o uso de uma *framework* desatualizada e com *exploits* públicos disponíveis.

## CONCLUSÃO

Durante o processo de análise de *malware* & forense, é fundamental criar notas acerca do que se vai observando para mais tarde, depois de tudo visto pelo menos uma vez, podermos começar a correlacionar informação.

A análise forense passa por uma metodologia de recolha de dados, análise dos mesmos, criação de pressuposições, confirmação da existência ou inexistência de evidências que confirmem o pressuposto, repetindo todo o processo.

O objetivo desta análise não era saber o que tinha sido afetado ou extraído, pois o cliente já tinha feito uma higienização à sua infraestrutura, e, portanto, contaminado as evidências, mas sim descobrir e compreender a origem do problema.

Verificamos que o incidente não começou na máquina infetada com *ransomware*, mas antes, esta foi o culminar do ataque à infraestrutura que começou na exploração de vulnerabilidades de uma *framework* desatualizada que estava instalada numa máquina exposta à Internet.

É importante descobrir e compreender o que aconteceu num caso de intrusão e comprometimento da infraestrutura para que a mesma falha não possa ser utilizada de novo por atacantes, o facto de termos descoberto esta falha no **Telerik** do cliente, permitiu o mesmo atualizar o sistema e colmatar a falha, bloqueando a entrada de novos atacantes na sua infraestrutura. Se apenas tivesse sido feita uma higienização da infraestrutura, o que é sempre aconselhável, seria uma questão de tempo até acontecer de novo, pois foram verificados nos *logs*, à data da investigação, pedidos frequentes ao URL vulnerável, não num contexto de explorar a falha, mas possivelmente apenas uma verificação por parte de terceiros para manter um registo de máquinas/infraestruturas vulneráveis para posterior utilização.