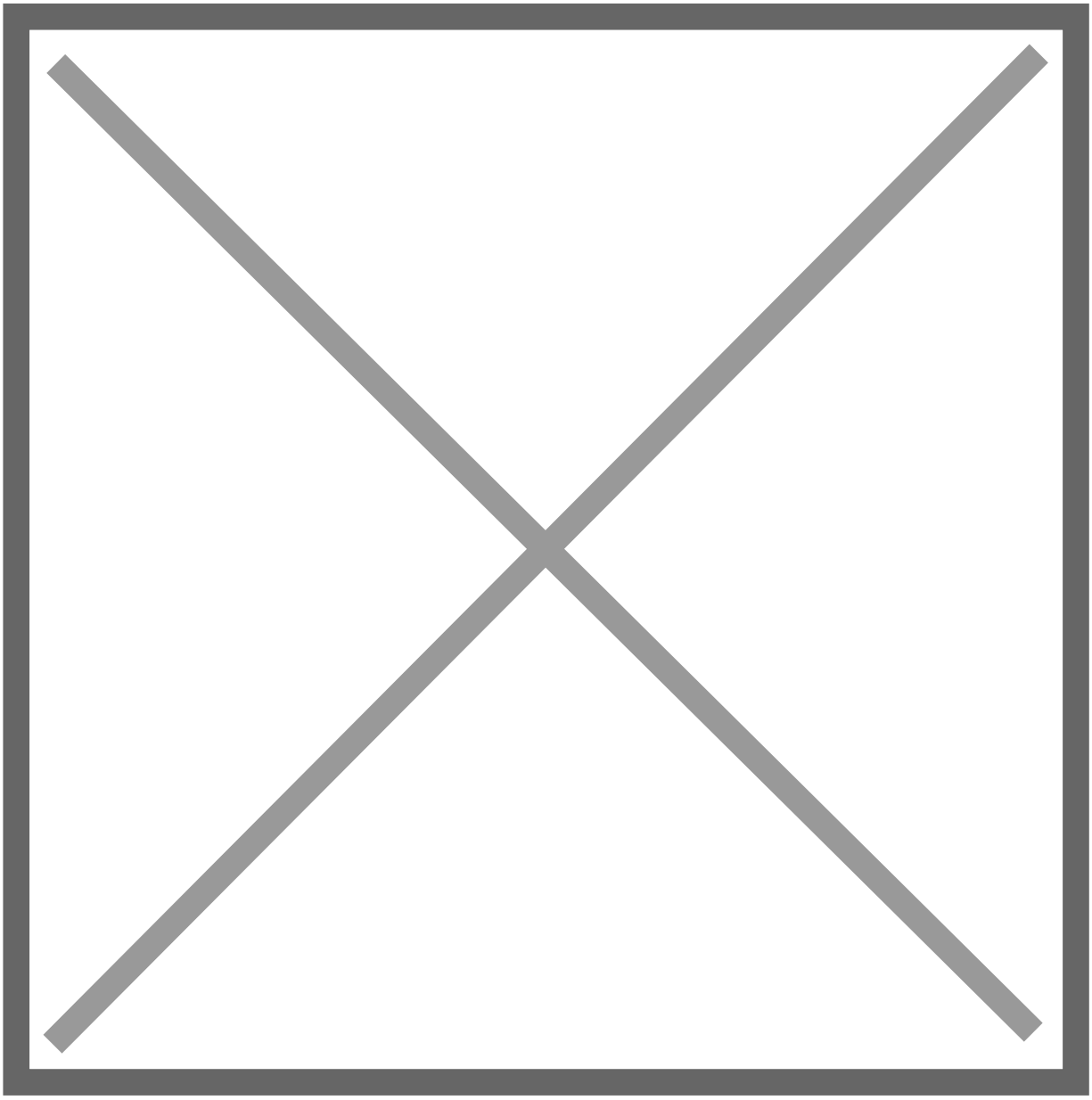
?????????? ????? ????????????? Windows

????????

При попадании в инфраструктуру обычно производится поиск административных учетных записей, привилегированных группы безопасности, наличие настроенных соглашений (trust) между контроллерами доменов с дочерними/родительскими организациями.

`net user /domain` выведет названия всех учетных записей в домене. Достаточна непривилегированная учетная запись в группе пользователей домена.



Данная команда журналируется только на локальном компьютере, на котором она была исполнена, ее обычно запускают на первом зараженном хосте. Событие EventID=4688 журнала Security или на событии EventID=1 журнала Sysmon.

Рассмотрим ниже пример события EventID=1 журнала Sysmon с разведкой в формате xml, где обратите внимание на поле

```
<Data Name="CommandLine">C:\Windows\system32\net1 user /domain</Data>
```

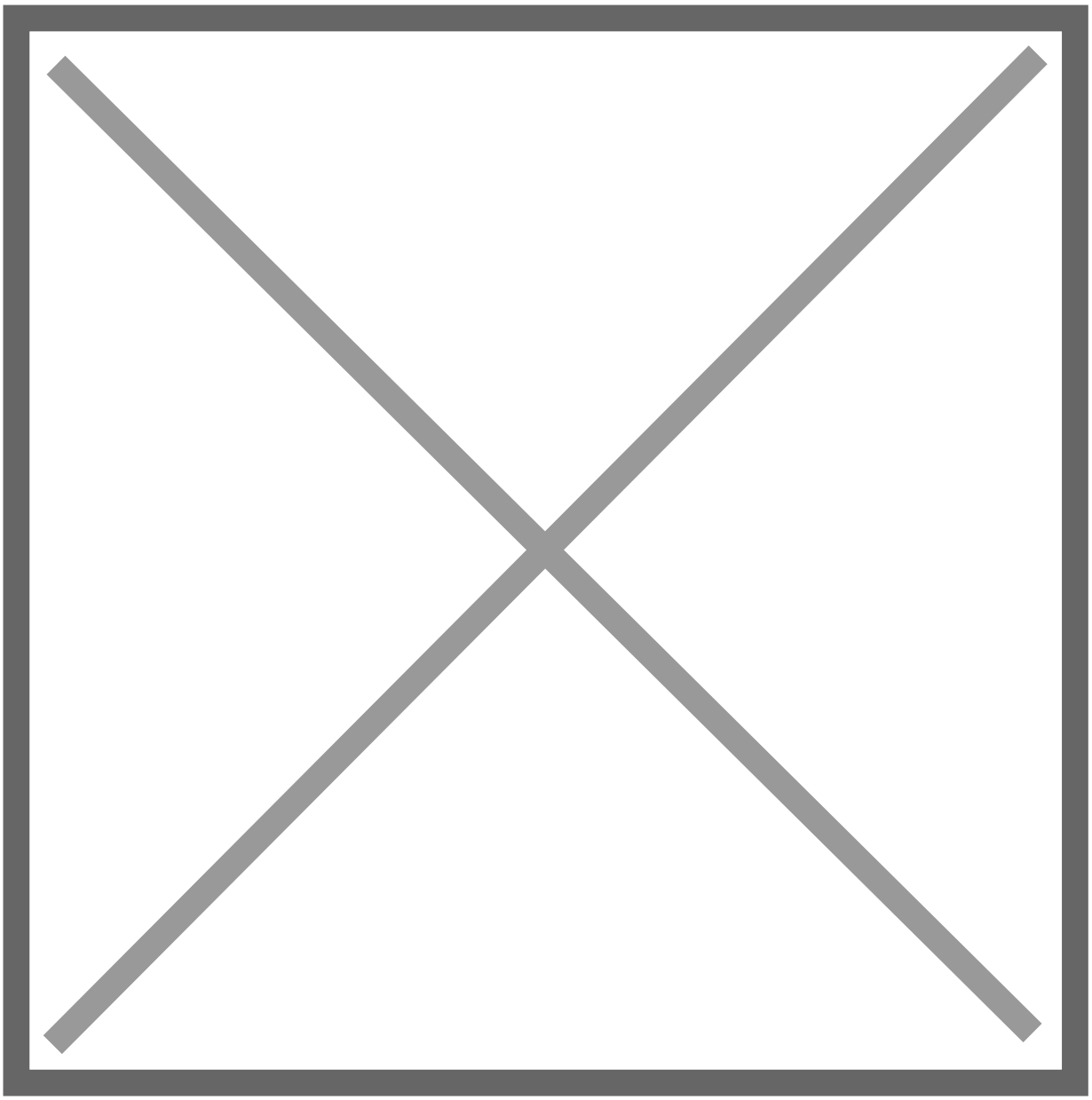
, в котором зафиксирована команда разведки. Как вы заметили в журнале команда `net` зафиксирована как `net1`, это сделано для обеспечения совместимости и исправления старой ошибки в команде `net` в 2000 году и осталась в системе до сих пор. Ситуации с различием команд при логировании и исполнении в командной строке не так часты, но их

надо учитывать при разработке корреляционных правил выявления атак и проверять, как журналирует система выполненные команды, если даже они кажутся очевидными.

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>
  <Provider Name="Microsoft-Windows-Sysmon" Guid="{5770385F-C22A-43E0-BF4C-06F5698FFBD9}" />
  <EventID>1</EventID>
  <Version>5</Version>
  <Level>4</Level>
  <Task>1</Task>
  <Opcode>0</Opcode>
  <Keywords>0x8000000000000000</Keywords>
  <TimeCreated SystemTime="2024-02-04T14:23:41.035418100Z" />
  <EventRecordID>6914807</EventRecordID>
  <Correlation />
  <Execution ProcessID="1604" ThreadID="2020" />
  <Channel>Microsoft-Windows-Sysmon/Operational</Channel>
  <Computer>WIN-O6QVDOTOFCA.lab.local</Computer>
  <Security UserID="S-1-5-18" />
  </System>
- <EventData>
  <Data Name="RuleName">-</Data>
  <Data Name="UtcTime">2024-02-04 14:23:41.019</Data>
  <Data Name="ProcessGuid">{460797FE-9DED-65BF-7D01-000000001800}</Data>
  <Data Name="ProcessId">976</Data>
  <Data Name="Image">C:\Windows\System32\net1.exe</Data>
  <Data Name="FileVersion">6.3.9600.17415 (winblue_r4.141028-1500)</Data>
  <Data Name="Description">Net Command</Data>
  <Data Name="Product">Microsoft® Windows® Operating System</Data>
  <Data Name="Company">Microsoft Corporation</Data>
  <Data Name="OriginalFileName">net1.exe</Data>
  <Data Name="CommandLine">C:\Windows\system32\net1 user /domain</Data>
  <Data Name="CurrentDirectory">C:\Windows\system32\</Data>
  <Data Name="User">LAB\Administrator</Data>
  <Data Name="LogonGuid">{460797FE-9671-65BE-4054-040000000000}</Data>
  <Data Name="LogonId">0x45440</Data>
  <Data Name="TerminalSessionId">1</Data>
  <Data Name="IntegrityLevel">High</Data>
  <Data
Name="Hashes">MD5=A3F48D90EE53FDF2547B41F87A7C8080,SHA256=D28BC8FA6E80316833C0EBB948B46511971B
```

96635892F40998A216A2DD5EC8AA,IMPHASH=EA59607831B13D45CEC2066C9436738F</Data>
<Data Name="ParentProcessGuid">{460797FE-9DEC-65BF-7C01-000000001800}</Data>
<Data Name="ParentProcessId">2372</Data>
<Data Name="ParentImage">C:\Windows\System32\net.exe</Data>
<Data Name="ParentCommandLine">net user /domain</Data>
<Data Name="ParentUser">LAB\Administrator</Data>
</EventData>
</Event>

`net group /domain`.



Так будет выглядить событие в формате xml по разведке доменных групп. Базируется выявление активности так же на событии EventID=4688 журнала Security или на событии EventID=1 журнала Sysmon. Обратите внимание на поле

```
<Data Name="CommandLine">C:\Windows\system32\net1 group /domain</Data>
```


```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>
  <Provider Name="Microsoft-Windows-Sysmon" Guid="{5770385F-C22A-43E0-BF4C-06F5698FFBD9}" />
  <EventID>1</EventID>
  <Version>5</Version>
  <Level>4</Level>
  <Task>1</Task>
  <Opcode>0</Opcode>
  <Keywords>0x8000000000000000</Keywords>
  <TimeCreated SystemTime="2024-02-04T15:15:07.129601400Z" />
  <EventRecordID>6921609</EventRecordID>
  <Correlation />
  <Execution ProcessID="1604" ThreadID="2020" />
  <Channel>Microsoft-Windows-Sysmon/Operational</Channel>
  <Computer>WIN-O6QVDOTOFCA.lab.local</Computer>
  <Security UserID="S-1-5-18" />
  </System>
- <EventData>
  <Data Name="RuleName">-</Data>
  <Data Name="UtcTime">2024-02-04 15:15:07.129</Data>
  <Data Name="ProcessGuid">{460797FE-A9FB-65BF-9201-000000001800}</Data>
  <Data Name="ProcessId">2988</Data>
  <Data Name="Image">C:\Windows\System32\net1.exe</Data>
  <Data Name="FileVersion">6.3.9600.17415 (winblue_r4.141028-1500)</Data>
  <Data Name="Description">Net Command</Data>
  <Data Name="Product">Microsoft® Windows® Operating System</Data>
  <Data Name="Company">Microsoft Corporation</Data>
  <Data Name="OriginalFileName">net1.exe</Data>
  <Data Name="CommandLine">C:\Windows\system32\net1 group /domain</Data>
  <Data Name="CurrentDirectory">C:\Windows\system32\</Data>
  <Data Name="User">LAB\Administrator</Data>
  <Data Name="LogonGuid">{460797FE-9671-65BE-4054-040000000000}</Data>
  <Data Name="LogonId">0x45440</Data>
```

```
  <Data Name="TerminalSessionId">1</Data>
  <Data Name="IntegrityLevel">High</Data>
  <Data
Name="Hashes">MD5=A3F48D90EE53FDF2547B41F87A7C8080,SHA256=D28BC8FA6E80316833C0EBB948B46511971B
96635892F40998A216A2DD5EC8AA,IMPHASH=EA59607831B13D45CEC2066C9436738F</Data>
  <Data Name="ParentProcessGuid">{460797FE-A9FB-65BF-9101-000000001800}</Data>
  <Data Name="ParentProcessId">2908</Data>
  <Data Name="ParentImage">C:\Windows\System32\net.exe</Data>
  <Data Name="ParentCommandLine">net group /domain</Data>
  <Data Name="ParentUser">LAB\Administrator</Data>
  </EventData>
  </Event>
```

Отслеживать каждую команду разведки по отдельности в большой инфраструктуре может быть черевато большим количеством ложных срабатываний, поэтому стоит рассмотреть вариант выявления активнсти по массовому запуску команд разведки за короткий промежуток времени.

Примеры корреляционных правил на Sigma:

- Правило 1

```
title: Suspicious Reconnaissance Activity
id: d95de845-b83c-4a9a-8a6a-4fc802ebf6c0
status: experimental
description: Detects suspicious command line activity on Windows systems
author: Florian Roth
date: 2019/01/16
tags:
    - attack.discovery
    - attack.t1087
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        CommandLine:
            - net group "domain admins" /domain
            - net localgroup administrators
```

```
    condition: selection
fields:
    - CommandLine
    - ParentCommandLine
falsepositives:
    - Inventory tool runs
    - Penetration tests
    - Administrative activity
analysis:
    recommendation: Check if the user that executed the commands is suspicious (e.g. service accounts, LOCAL_SYSTEM)
level: medium
```

- Правило 2

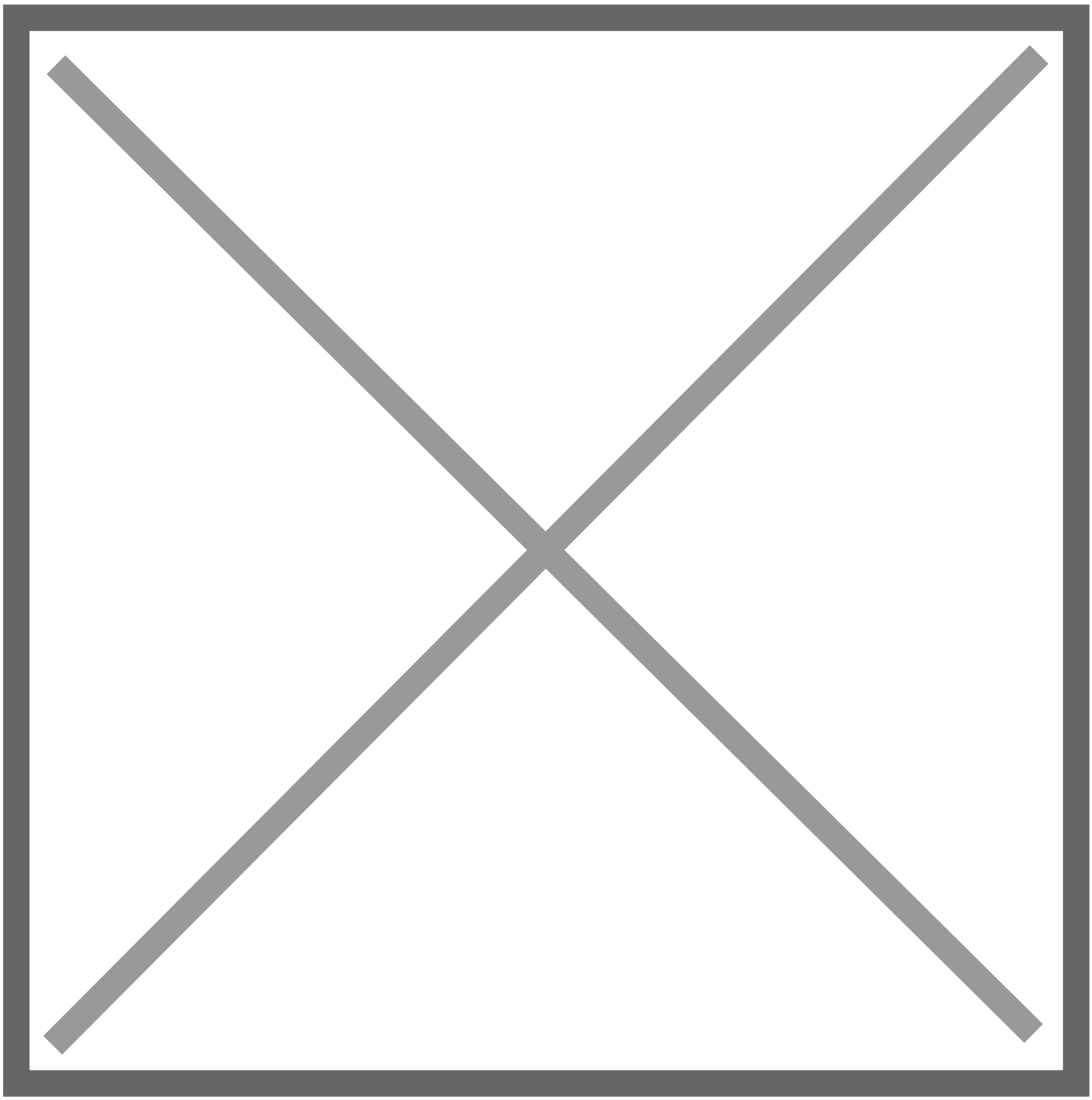
```
title: Suspicious Group And Account Reconnaissance Activity Using Net.EXE
id: d95de845-b83c-4a9a-8a6a-4fc802ebf6c0
status: test
description: Detects suspicious reconnaissance command line activity on Windows systems using Net.EXE
references:
    - https://redcanary.com/blog/how-one-hospital-thwarted-a-ryuk-ransomware-outbreak/
    - https://thedfirreport.com/2020/10/18/ryuk-in-5-hours/
    - https://research.nccgroup.com/2022/08/19/back-in-black-unlocking-a-lockbit-3-0-ransomware-attack/
author: Florian Roth (Nextron Systems), omkar72, @svch0st, Nasreddine Bencherchali (Nextron Systems)
date: 2019/01/16
modified: 2023/03/02
tags:
    - attack.discovery
    - attack.t1087.001
    - attack.t1087.002
logsource:
    category: process_creation
    product: windows
detection:
    selection_img:
```

```
        - Image|endswith:
              - '\net.exe'
              - '\net1.exe'
        - OriginalFileName:
              - 'net.exe'
              - 'net1.exe'
    # Covers group and localgroup flags
    selection_group_root:
        CommandLine|contains:
            - ' group '
            - ' localgroup '
    selection_group_flags:
        CommandLine|contains:
            # Add more groups for other languages
            - 'domain admins'
            - ' administrator' # Typo without an 'S' so we catch both
            - ' administrateur' # Typo without an 'S' so we catch both
            - 'enterprise admins'
            - 'Exchange Trusted Subsystem'
            - 'Remote Desktop Users'
            - 'Utilisateurs du Bureau à distance' # French for "Remote Desktop Users"
            - 'Usuarios de escritorio remoto' # Spanish for "Remote Desktop Users"
            - ' /do' # short for domain
    filter_group_add:
        # This filter is added to avoid the potential case where the point is not recon but
addition
        CommandLine|contains: ' /add'
    # Covers 'accounts' flag
    selection_accounts_root:
        CommandLine|contains: ' accounts '
    selection_accounts_flags:
        CommandLine|contains: ' /do' # short for domain
    condition: selection_img and ((all of selection_group_* and not filter_group_add) or all
of selection_accounts_*)
fields:
    - CommandLine
    - ParentCommandLine
falsepositives:
    - Inventory tool runs
    - Administrative activity
```

```
level: medium
analysis:
    recommendation: Check if the user that executed the commands is suspicious (e.g. service
accounts, LOCAL_SYSTEM)
```

???????? ? ??????? ??????????? (cmdlets) PowerShell

Выполним команду `Get-ADForest`. Данная команда выдает информацию о лесе AD. **Лесом** называют полностью самостоятельную организацию Active Directory, которая имеет определенный набор атрибутов и является периметром безопасности организации. В состав леса могут входить как один, так и несколько доменов. В лесу у каждого домена есть своя база данных и свои собственные контроллеры домена. Однако пользователи домена в лесу также могут получить доступ к другим доменам леса. Это означает, что даже если домен может быть автономным (без необходимости взаимодействия с другими доменами), он не изолирован с точки зрения безопасности, поскольку пользователь из одного домена по умолчанию может получить доступ к ресурсам других доменов в том же лесу. Однако пользователи леса по умолчанию не могут получить доступ к ресурсам из других лесов, поэтому лес является логической структурой, которая может обеспечить изоляцию с точки зрению безопасности.



Ниже представлено событие выполнения скрипт-блоков с **EventID = 4104** из журнала **Powershell**. Обратите внимание на строку

```
<Data Name="ScriptBlockText">get-adforest</Data>
```

, в которой зафиксировано выполнение команды.

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>
  <Provider Name="Microsoft-Windows-PowerShell" Guid="{A0C1853B-5C40-4B15-8766-3CF1C58F985A}"
```

```
/>
  <EventID>4104</EventID>
  <Version>1</Version>
  <Level>5</Level>
  <Task>102</Task>
  <Opcode>15</Opcode>
  <Keywords>0x0</Keywords>
  <TimeCreated SystemTime="2024-02-04T20:00:41.649279800Z" />
  <EventRecordID>4667</EventRecordID>
  <Correlation ActivityID="{6487D210-56D8-0000-D6F8-8764D856DA01}" />
  <Execution ProcessID="4092" ThreadID="3048" />
  <Channel>Microsoft-Windows-PowerShell/Operational</Channel>
  <Computer>WIN-O6QVDOTOFCA.lab.local</Computer>
  <Security UserID="S-1-5-21-1043167210-2633990363-2710869231-500" />
  </System>
- <EventData>
  <Data Name="MessageNumber">1</Data>
  <Data Name="MessageTotal">1</Data>
  <Data Name="ScriptBlockText">get-adforest</Data>
  <Data Name="ScriptBlockId">07651e4c-bb49-4563-8b55-1454584112d7</Data>
  </EventData>
  </Event>
```

Количество командлетов PowerShell для разведки не малое количество, выявить их запуск можно так же по событию с EventID = 4104.

Пример корреляционного правила на Sigma для выявления команд разведки данных AD и не только.

```
title: PowerView PowerShell Cmdlets - ScriptBlock
id: dcd74b95-3f36-4ed9-9598-0490951643aa
related:
    - id: b2317cfa-4a47-4ead-b3ff-297438c0bc2d
      type: similar
status: test
description: Detects Cmdlet names from PowerView of the PowerSploit exploitation framework.
references:
    - https://powersploit.readthedocs.io/en/stable/Recon/README
```

```
    - https://github.com/PowerShellMafia/PowerSploit/tree/master/Recon
    - https://thedfirreport.com/2020/10/08/ryuks-return
    - https://adsecurity.org/?p=2277
author: Bhabesh Raj
date: 2021/05/18
modified: 2023/11/22
tags:
    - attack.execution
    - attack.t1059.001
logsource:
    product: windows
    category: ps_script
    definition: 'Requirements: Script Block Logging must be enabled'
detection:
    selection:
        ScriptBlockText|contains:
            - 'Export-PowerViewCSV'
            - 'Find-DomainLocalGroupMember'
            - 'Find-DomainObjectPropertyOutlier'
            - 'Find-DomainProcess'
            - 'Find-DomainShare'
            - 'Find-DomainUserEvent'
            - 'Find-DomainUserLocation'
            - 'Find-ForeignGroup'
            - 'Find-ForeignUser'
            - 'Find-GPOComputerAdmin'
            - 'Find-GPOLocation'
            - 'Find-InterestingDomain' # Covers: Find-InterestingDomainAcl, Find-
InterestingDomainShareFile
            - 'Find-InterestingFile'
            - 'Find-LocalAdminAccess'
            - 'Find-ManagedSecurityGroups'
            - 'Get-CachedRDPConnection'
            - 'Get-DFSshare'
            - 'Get-DomainDFSShare'
            - 'Get-DomainDNSRecord'
            - 'Get-DomainDNSZone'
            - 'Get-DomainFileServer'
            - 'Get-DomainGPOComputerLocalGroupMapping'
```

```
            - 'Get-DomainGPOLocalGroup'
            - 'Get-DomainGPOUserLocalGroupMapping'
            - 'Get-LastLoggedOn'
            - 'Get-LoggedOnLocal'
            - 'Get-NetFileServer'
            - 'Get-NetForest' # Covers: Get-NetForestCatalog, Get-NetForestDomain, Get-
NetForestTrust
            - 'Get-NetGPOGroup'
            - 'Get-NetProcess'
            - 'Get-NetRDPSession'
            - 'Get-RegistryMountedDrive'
            - 'Get-RegLoggedOn'
            - 'Get-WMIRegCachedRDPConnection'
            - 'Get-WMIRegLastLoggedOn'
            - 'Get-WMIRegMountedDrive'
            - 'Get-WMIRegProxy'
            - 'Invoke-ACLScanner'
            - 'Invoke-CheckLocalAdminAccess'
            - 'Invoke-EnumerateLocalAdmin'
            - 'Invoke-EventHunter'
            - 'Invoke-FileFinder'
            - 'Invoke-Kerberoast'
            - 'Invoke-MapDomainTrust'
            - 'Invoke-ProcessHunter'
            - 'Invoke-RevertToSelf'
            - 'Invoke-ShareFinder'
            - 'Invoke-UserHunter'
            - 'Invoke-UserImpersonation'
            - 'Remove-RemoteConnection'
            - 'Request-SPNTicket'
            - 'Resolve-IPAddress'
            # - 'Get-ADObject'  # prone to FPs
            # - 'Get-Domain'  # too many FPs  # Covers Cmdlets like: DomainComputer,
DomainController, DomainDFSShare, DomainDNSRecord, DomainGPO, etc.
            # - 'Add-DomainGroupMember'
            # - 'Add-DomainObjectAcl'
            # - 'Add-ObjectAcl'
            # - 'Add-RemoteConnection'
            # - 'Convert-ADName'
```

```
            # - 'Convert-NameToSid'
            # - 'ConvertFrom-UACValue'
            # - 'ConvertTo-SID'
            # - 'Get-DNSRecord'
            # - 'Get-DNSZone'
            # - 'Get-DomainComputer'
            # - 'Get-DomainController'
            # - 'Get-DomainGroup'
            # - 'Get-DomainGroupMember'
            # - 'Get-DomainManagedSecurityGroup'
            # - 'Get-DomainObject'
            # - 'Get-DomainObjectAcl'
            # - 'Get-DomainOU'
            # - 'Get-DomainPolicy'
            # - 'Get-DomainSID'
            # - 'Get-DomainSite'
            # - 'Get-DomainSPNTicket'
            # - 'Get-DomainSubnet'
            # - 'Get-DomainUser'
            # - 'Get-DomainUserEvent'
            # - 'Get-Forest' # Covers: Get-ForestDomain, Get-ForestGlobalCatalog, Get-
ForestTrust
            # - 'Get-IPAddress'
            # - 'Get-NetComputer' # Covers: Get-NetComputerSiteName
            # - 'Get-NetDomain' # Covers: Get-NetDomainController, Get-NetDomainTrust
            # - 'Get-NetGroup' # Covers: Get-NetGroupMember
            # - 'Get-NetLocalGroup' # Covers: NetLocalGroupMember
            # - 'Get-NetLoggedon'
            # - 'Get-NetOU'
            # - 'Get-NetSession'
            # - 'Get-NetShare'
            # - 'Get-NetSite'
            # - 'Get-NetSubnet'
            # - 'Get-NetUser'
            # - 'Get-ObjectAcl'
            # - 'Get-PathAcl'
            # - 'Get-Proxy'
            # - 'Get-SiteName'
            # - 'Get-UserEvent'
```

```
            # - 'Get-WMIProcess'
            # - 'New-DomainGroup'
            # - 'New-DomainUser'
            # - 'Set-ADObject'
            # - 'Set-DomainObject'
            # - 'Set-DomainUserPassword'
    condition: selection
falsepositives:
    - Unknown
level: high
```

????????? ???????? ? ???????? ??????? AD

Для наиболее эффективной разведки могут воспользоваться утилитой SharpHound и результаты выполнения, потом, для удобства, визуализировать с помощью утилиты BloodHound. Аналогичные инструменты атакующих для разведки данных в AD, это, ADFind, PowerView и ldapsearch.

Запустим ADFind и посмотрим какие останутся следы в жураналах событий. Предварительно должен быть настроен аудит журналирования запросов для генерации событий EventID=1644 журнала Directory Service. Для настройки аудита необходимо с помощью PowerShell на контроллере домена выполнить команды:

```
Set-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Services\NTDS\Diagnostics' -Name '15 Field Engineering' -Value "5"
```

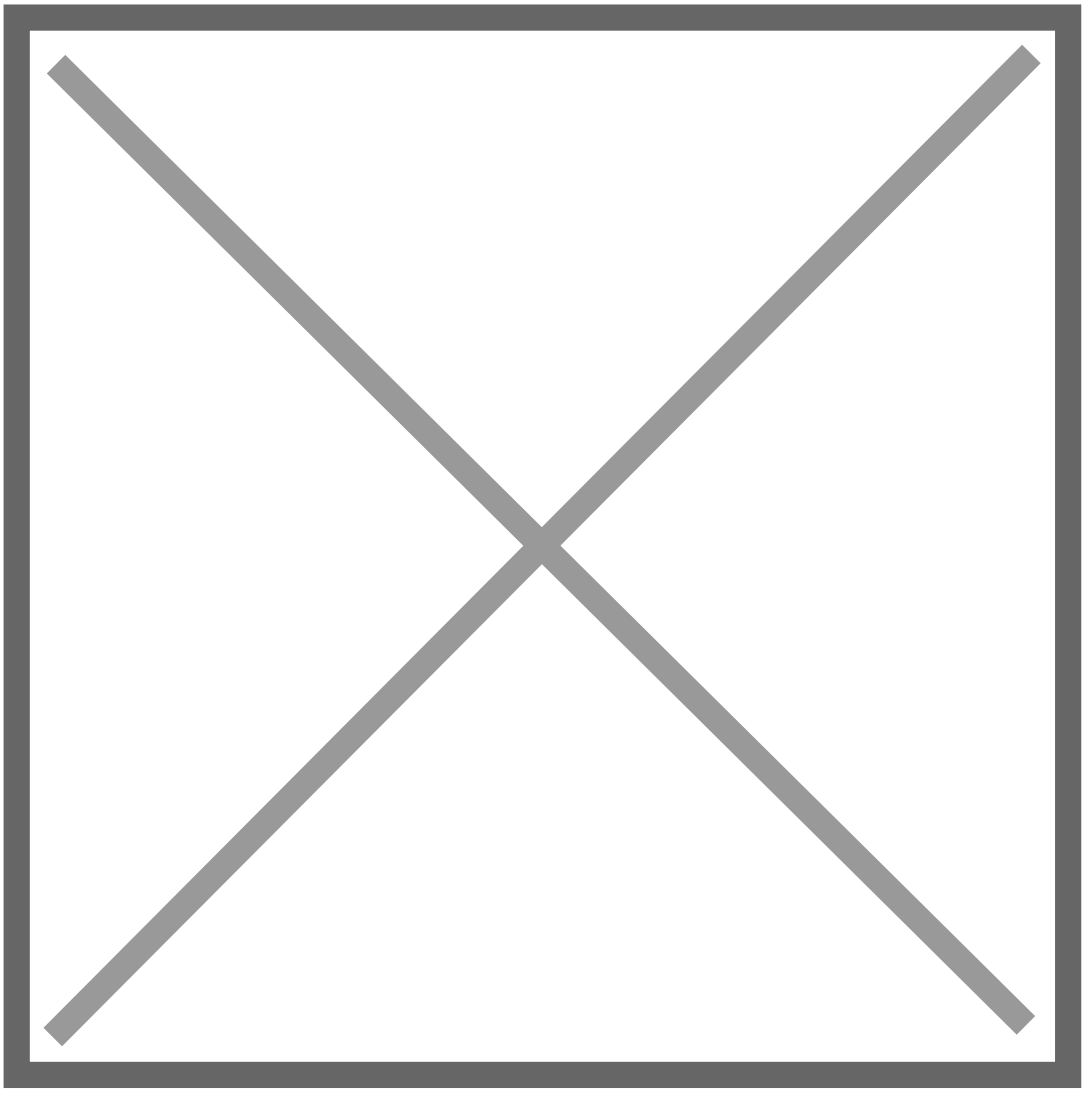
```
Set-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Services\NTDS\Parameters' -Name 'Expensive Search Results Threshold' -Value "10"
```

```
Set-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Services\NTDS\Parameters' -Name 'Inefficient Search Results Threshold' -Value "10"
```

```
Set-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Services\NTDS\Parameters' -Name 'Search Time Threshold (msecs)' -Value "80"
```

Данные команды изначально были предназначены отслеживания медленных запросов к AD. Настройку надо выполнять с осторожностью, поскольку повышается нагрузка на контроллере домена. В данном случае уже мы можем отслеживать аномальную активность на самом контроллере домена, это значит, что выявить атаку вероятность выше по сравнению с подходом выявления атаки при исполнении на конечном узле, т.к. события с контроллера домена всегда должны собираться в SIEM, как от критичного узла в инфраструктуре.
Выполним команду:

```
adfind -b dc=lab,dc=local -f "( |(sAMAccountName=Bill) )"
```



Обратите внимание на строку в событии ниже с идентификатором 1644 со значением `<Data>( | (sAMAccountName=Bill) )</Data>` , это зафиксирована команда с запросом от утилиты **ADFind** для получения информации о пользователе **Bill**.

В поле `<Data>192.168.0.5:51653</Data>` указан IP-адрес с которого был выполнен запрос.

В поле `<Data>LAB\Nick</Data>` указано под какой учетной записью был выполнен запрос.

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>
  <Provider Name="Microsoft-Windows-ActiveDirectory_DomainService" Guid="{0e8478c5-3605-4e8c-
8497-1e730c959516}" EventSourceName="NTDS General" />
  <EventID Qualifiers="16384">1644</EventID>
  <Version>0</Version>
  <Level>4</Level>
  <Task>15</Task>
  <Opcode>0</Opcode>
  <Keywords>0x8080000000000000</Keywords>
  <TimeCreated SystemTime="2024-02-04T17:56:12.804105800Z" />
  <EventRecordID>309</EventRecordID>
  <Correlation />
  <Execution ProcessID="480" ThreadID="1088" />
  <Channel>Directory Service</Channel>
  <Computer>WIN-O6QVDOTOFCA.lab.local</Computer>
  <Security UserID="S-1-5-21-1043167210-2633990363-2710869231-1003" />
  </System>
- <EventData>
  <Data>CN=Schema,CN=Configuration,DC=lab,DC=local</Data>
  <Data>( | (sAMAccountName=Bill) )</Data>
  <Data>1630</Data>
  <Data>50</Data>
  <Data>192.168.0.5:51653</Data>
  <Data>subtree</Data>
  <Data>uid,sAMAccountName</Data>
  <Data />
  <Data>DNT_index:1070:N;</Data>
  <Data>14809</Data>
  <Data>4</Data>
  <Data>0</Data>
  <Data>0</Data>
  <Data>0</Data>
```

```
<Data>47</Data>
<Data>none</Data>
<Data>LAB\Nick</Data>
</EventData>
</Event>
```

Далее необходимо выявить хост и проанализировать с него журналы событий.

????? Kerberoasting

В Active Directory любой пользователь может запросить сервисный билет — **Service Ticket** для любой зарегистрированной службы в домене и имеющий Service Principal Name (SPN), независимо от статуса службы. Service Ticket частично зашифрован ключом Kerberos, полученным из пароля пользователя сервиса, что позволяет подобрать оффлайн пароль, расшифровывая Service Ticket.

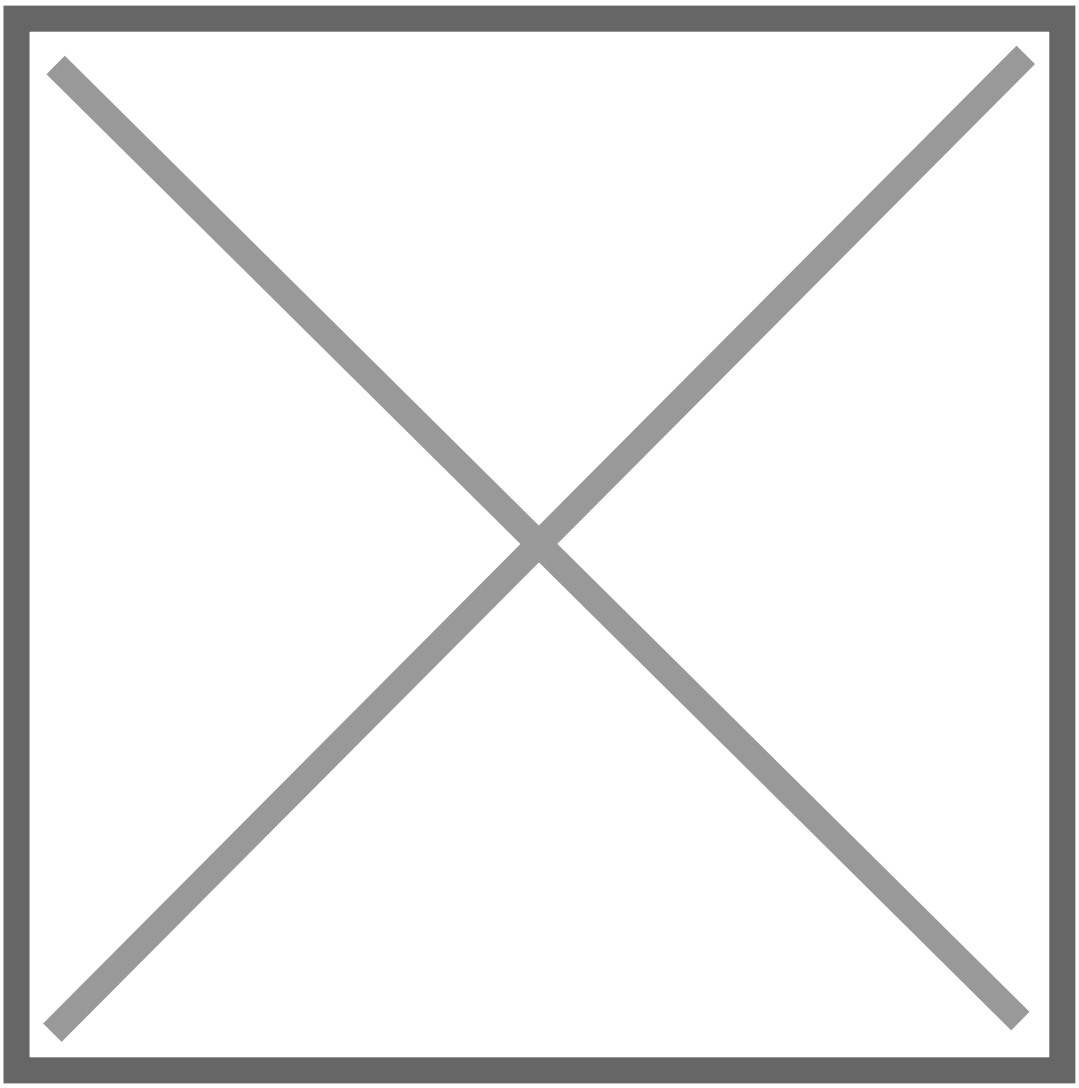
Большинство служб регистрируются в учетных записях компьютеров с автоматически генерируемыми паролями длиной 120 символов, меняющимися ежемесячно, что делает взлом Service Ticket невозможным. Однако иногда службы привязываются к обычным учетным записям пользователей со слабыми паролями, что может быть использовано для взлома и получения паролей пользователей.

Атака Kerberoasting заключается в запросе Service Ticket для обычных учетных записей пользователей служб и последующей попытке взлома для получения паролей пользователей, которые обычно имеют высокие привилегии и далее используются на следующих этапах атак.

Проверим учетные записи пользователей с именами участников-служб с помощью ADFind, выполнив команду.

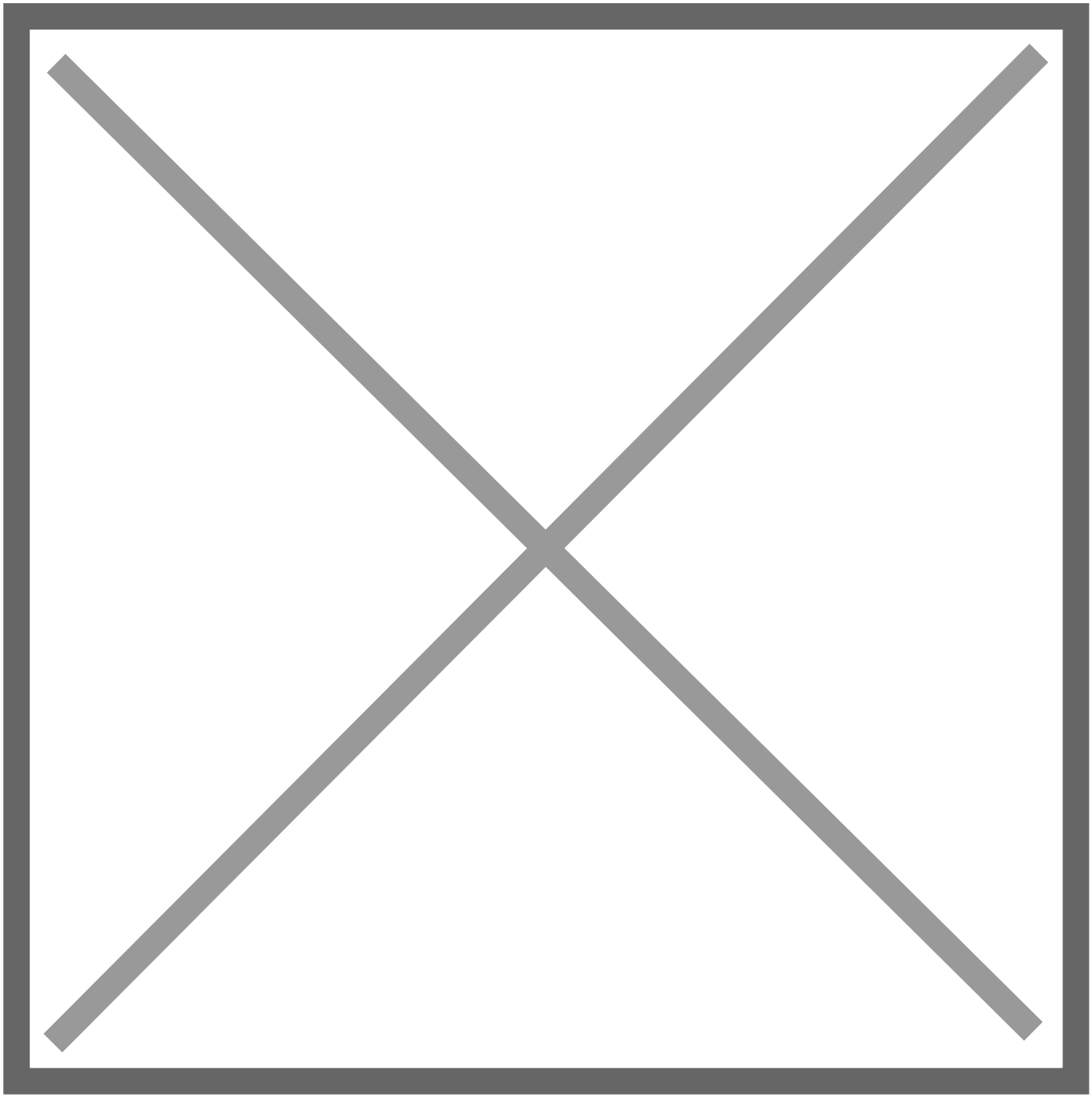
```
adfind -b dc=lab,dc=local -f "(&(samAccountType=805306368)(servicePrincipalName=*))"
```

В результате получаем две учетный записи с SPN`ами — это **krbtgt**, который не попал на скриншот в связи с тем, что его не взломать подбором. Вторая учетная запись с SPN выделена на скриншоте **sqlservice** — это как раз тот самый лакомый кусочек для злоумышленника. Запросив Servcie Ticket для учетной записи sqlservice, злоумышленник может взламывать его у себя на хосте с помощью hashcat.



Для получения Service Ticket и дальнейшего оффлайн взлома, злоумышленник можете использовать следующие скрипты impacket GetUserSPNs.py, команду Rubeus kerberoast или сценарий Invoke-Kerberoast.ps1.

Проведем атаку с помощью утилиты Rubeus используя команду `Rubeus.exe kerberoast` и результатом получаем хэши от пароля для учетной записи sqlservice на скриншоте ниже, которые можем дальше взламывать.



Посмотрим как это будет зафиксировано в журналах событий на AD. При запросе Service Ticket сгенерировалось событие с **EventID = 4769** в журнале **Security**. Обратите внимание на следующие важные поля:

- `<Data Name="TargetUserName">Nick@LAB.LOCAL</Data>` — учетная запись которая выполнила запрос.
- `<Data Name="ServiceName">sqlservice</Data>` — наша учетная запись с SPN.
- `<Data Name="TicketEncryptionType">0x17</Data>` — 0x17 означает шифрование AES256, но это не помеха для злоумышленника, потому что уже есть разработанные скрипты для взлома.
- `<Data Name="IpAddress">192.168.0.5</Data>` — IP-адрес выполнившего запрос Service Ticket.

- `<Data Name="IpPort">58334</Data>` — порты выполнившего запрос Service Ticket.

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>
  <Provider Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-A5BA-
3E3B0328C30D}" />
  <EventID>4769</EventID>
  <Version>0</Version>
  <Level>0</Level>
  <Task>14337</Task>
  <Opcode>0</Opcode>
  <Keywords>0x8020000000000000</Keywords>
  <TimeCreated SystemTime="2024-02-04T23:59:03.715787300Z" />
  <EventRecordID>94453</EventRecordID>
  <Correlation />
  <Execution ProcessID="480" ThreadID="1780" />
  <Channel>Security</Channel>
  <Computer>WIN-O6QVDOTOFCA.lab.local</Computer>
  <Security />
  </System>
- <EventData>
  <Data Name="TargetUserName">Nick@LAB.LOCAL</Data>
  <Data Name="TargetDomainName">LAB.LOCAL</Data>
  <Data Name="ServiceName">sqlservice</Data>
  <Data Name="ServiceSid">S-1-5-21-1043167210-2633990363-2710869231-1118</Data>
  <Data Name="TicketOptions">0x40800000</Data>
  <Data Name="TicketEncryptionType">0x17</Data>
  <Data Name="IpAddress">192.168.0.5</Data>
  <Data Name="IpPort">58334</Data>
  <Data Name="Status">0x0</Data>
  <Data Name="LogonGuid">{9B1A0512-6224-531F-E2B8-C5A47A332D75}</Data>
  <Data Name="TransmittedServices">-</Data>
  </EventData>
  </Event>
```

Важное замечание: подобных событий с **EventID = 4769** будет огромное множество в инфраструктуре. Это обычная активность.

Для повышения эффективности атаки злоумышленник может запросить несколько Service Ticket за короткий промежуток времени, чтобы получить для дальнейшего брутфорса как можно больше данных, так как он заведомо не знает, у какой сервисной или пользовательской УЗ пароль будет менее криптостойким. Подобную активность можно попытаться выявить корреляционным правилом, которое отслеживает множественные запросы Service Ticket от одного источника за короткий промежуток времени.

Для выявления атаки также можно отслеживать запросы Service Ticket с шифрованием RC4, но может быть большое количество ложных срабатываний, если в инфраструктуре есть сервисы, которые используют устаревшее шифрование.

Пример правила для выявления запроса Service Ticket с шифрованием RC4 на Sigma:

```
title: Suspicious Kerberos RC4 Ticket Encryption
id: 496a0e47-0a33-4dca-b009-9e6ca3591f39
status: test
description: Detects service ticket requests using RC4 encryption type
references:
    - https://adsecurity.org/?p=3458
    - https://www.trimarcsecurity.com/single-post/TrimarcResearch/Detecting-Kerberoasting-
Activity
author: Florian Roth (Nextron Systems)
date: 2017/02/06
modified: 2022/06/19
tags:
    - attack.credential_access
    - attack.t1558.003
logsource:
    product: windows
    service: security
detection:
    selection:
        EventID: 4769
        TicketOptions: '0x40810000'
        TicketEncryptionType: '0x17'
    reduction:
        ServiceName|endswith: '$'
    condition: selection and not reduction
falsepositives:
    - Service accounts used on legacy systems (e.g. NetApp)
    - Windows Domains with DFL 2003 and legacy systems
```

```
level: medium
```

Самый оптимальный способ выявления атаки kerberoasting — использование SPN HoneyToken . HoneyToken — это в данном случае подделанная учетная запись приманка с SPN, которая потом отслеживается на возникающие к ней запросы, т.к. в легитимных целях она не используется. Дополнительный материал об атаке Kerberoasting.

????? DCSync

DCSync — это атака, позволяющая злоумышленнику выдавать себя за контроллер домена (DC, domain controller) с целью получения базы учетных данных пользователей для последующего горизонтального перемещения в сети и/или доступа к конфиденциальной информации. В основе атаки лежит механизм, предусмотренный для выполнения репликации данных между контроллерами домена (DC).

Механизм репликации данных архитектурно заложен в операционной системе Windows. Службы Active Directory и протокол MS-DRSR отвечают за взаимодействие между контроллерами домена и осуществляют репликацию. Сама компания Microsoft рекомендует изначально устанавливать как минимум два и более контроллера для одного домена в корпоративной сети, чтобы обеспечить отказоустойчивость доменной инфраструктуры. В процессе репликации данных между контроллерами помимо обычных атрибутов об объекте (имя, отчество, списка групп и так далее) передается и чувствительная информация, например, хеши паролей пользователей, поскольку каждый контроллер выступает как точка для аутентификации и авторизации в домене.

Как уже можно было догадаться, отчасти именно из-за наличия такого механизма возможна реализация атаки типа DCSync. Атакующий, имея необходимый набор привилегий, может отправить одному из контроллеров домена организации запрос на выполнение репликации. Запросив при этом информацию по одному или нескольким объектам в домене. Таким образом злоумышленник удаленно собирает хеши паролей пользователей и другую полезную информацию в домене без выполнения какого-либо вредоносного кода на самих контроллерах домена организации.

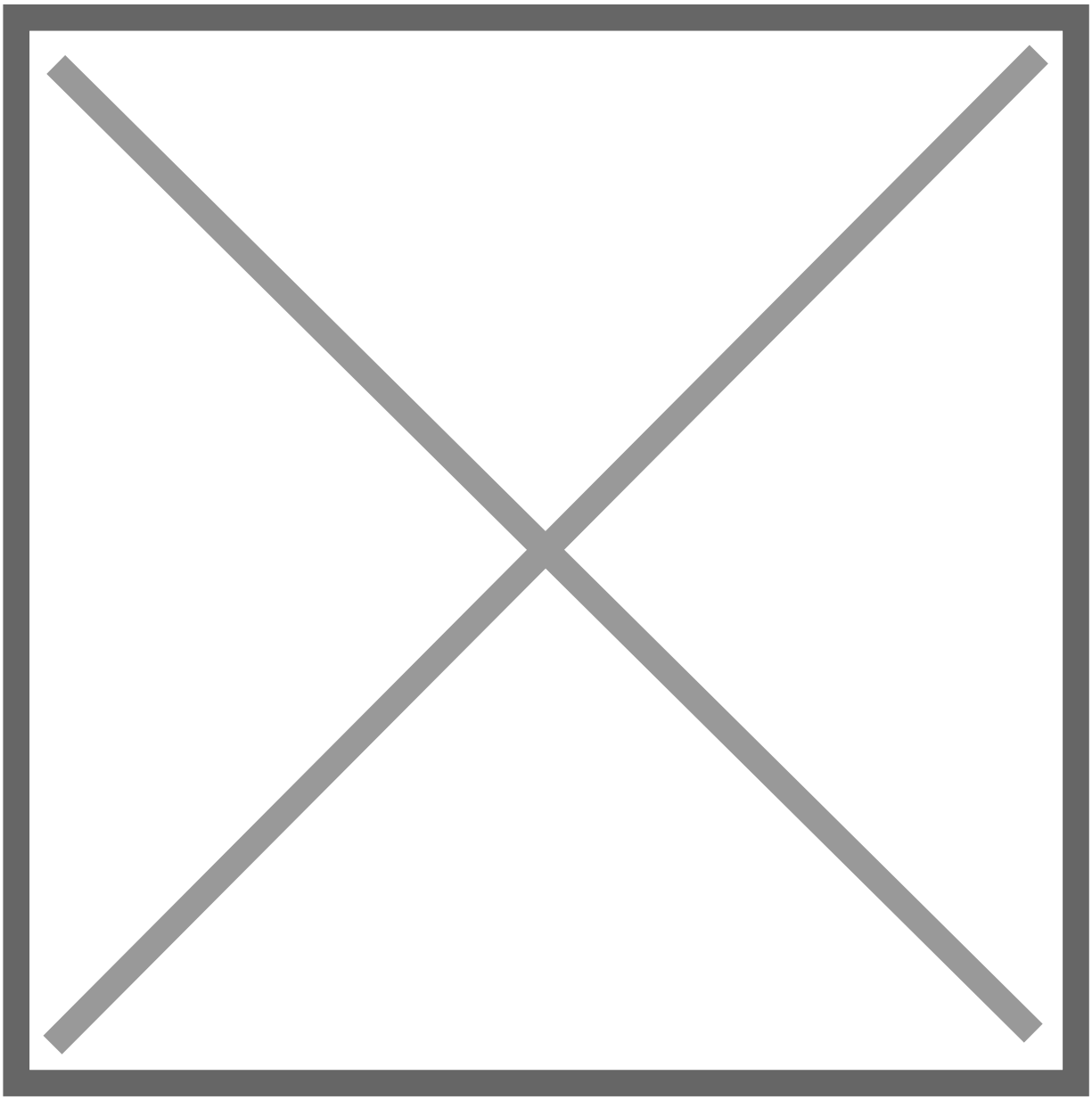
Необходимые права доступа для проведения атаки DCSync.

Наименование	Common Name(общее имя)	Rights-GUID(идентификатор прав)
Replicating Directory Changes	DRS-Replication-Get-Changes	1131f6aa-9c07–11d1-f79f-00c04fc2dcd2
Replicating Directory Changes All	DRS-Replication-Get-Changes-All	1131f6ad-9c07–11d1-f79f-00c04fc2dcd2

Атаку DCSync можно выполнить с помощью инструментов, таких как **secretsdump** из набора impacket и широко известной утилитой **mimikatz**.

Например злоумышленник выполняет атаку DCSync с помощью команды

```
lsadump::dcsync /dc:$DomainController /domain:$DOMAIN /all /csv
```



Выявить атаку на контроллере домена можно с помощью события **EventID=4662** журнала Security. Важно понимать, что включение аудита события 4662 может повлечь за собой генерацию большого количества событий, особенно при неаккуратной настройке SACL. Настройка происходит в групповой политике: *Computer configurations* > *Policies* > *Windows Settings* > *Security Settings* > *Local Policies* > *Audit Policy* > *Audit Directory Service Access* > *Enable Success*. При настройке этого параметра в журналах будут генерироваться два новых

идентификатора события: 4661 и 4662.
Предварительно должен быть так же настроен SACL для отслеживания доступа к объектам AD на контроллере домена связанные с репликацией:

```
AD Users and Computers >
[Domain] >
properties >
security >
advanced >
auditing > add:
Principal: Everyone
Type: Success
Applies to: This Object Only
Permissions: Replicating Directory Changes; Replicating Directory Changes All
```

В результате атаки фиксируется событие, в котором необходимо обратить внимание на поле:

```
<Data Name="Properties">%%7688 {1131f6aa-9c07-11d1-f79f-00c04fc2dcd2} {19195a5b-6da0-11d0-afd3-00c04fd930c9}</Data>
```

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>
  <Provider Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-A5BA-3E3B0328C30D}" />
  <EventID>4662</EventID>
  <Version>0</Version>
  <Level>0</Level>
  <Task>14080</Task>
  <Opcode>0</Opcode>
  <Keywords>0x8020000000000000</Keywords>
  <TimeCreated SystemTime="2024-02-10T14:41:31.872715100Z" />
  <EventRecordID>111044</EventRecordID>
  <Correlation />
  <Execution ProcessID="480" ThreadID="3808" />
  <Channel>Security</Channel>
  <Computer>WIN-O6QVDOTOFCA.lab.local</Computer>
```

```
  <Security />
  </System>
- <EventData>
  <Data Name="SubjectUserSid">S-1-5-21-1043167210-2633990363-2710869231-500</Data>
  <Data Name="SubjectUserName">administrator</Data>
  <Data Name="SubjectDomainName">LAB</Data>
  <Data Name="SubjectLogonId">0x45440</Data>
  <Data Name="ObjectServer">DS</Data>
  <Data Name="ObjectType">%{19195a5b-6da0-11d0-afd3-00c04fd930c9}</Data>
  <Data Name="ObjectName">%{83fb1e47-6d25-4d4b-a710-e244aca1c5e8}</Data>
  <Data Name="OperationType">Object Access</Data>
  <Data Name="HandleId">0x0</Data>
  <Data Name="AccessList">%%7688</Data>
  <Data Name="AccessMask">0x100</Data>
  <Data Name="Properties">%%7688 {1131f6aa-9c07-11d1-f79f-00c04fc2dcd2} {19195a5b-6da0-11d0-
afd3-00c04fd930c9}</Data>
  <Data Name="AdditionalInfo">-</Data>
  <Data Name="AdditionalInfo2" />
  </EventData>
  </Event>
```

Пример правила для выявления атаки DCSync и в том числе DCShadow на Sigma.

Дополнительный материал про DCSync.

????????? ???? ????? NTDS.dit

Контроллер домена отвечает за хранение базы данных домена **NTDS.dit** со всей информацией об объектах домена и обслуживает службы Active Directory, такие как аутентификация, авторизация, разрешение имен и т.д.

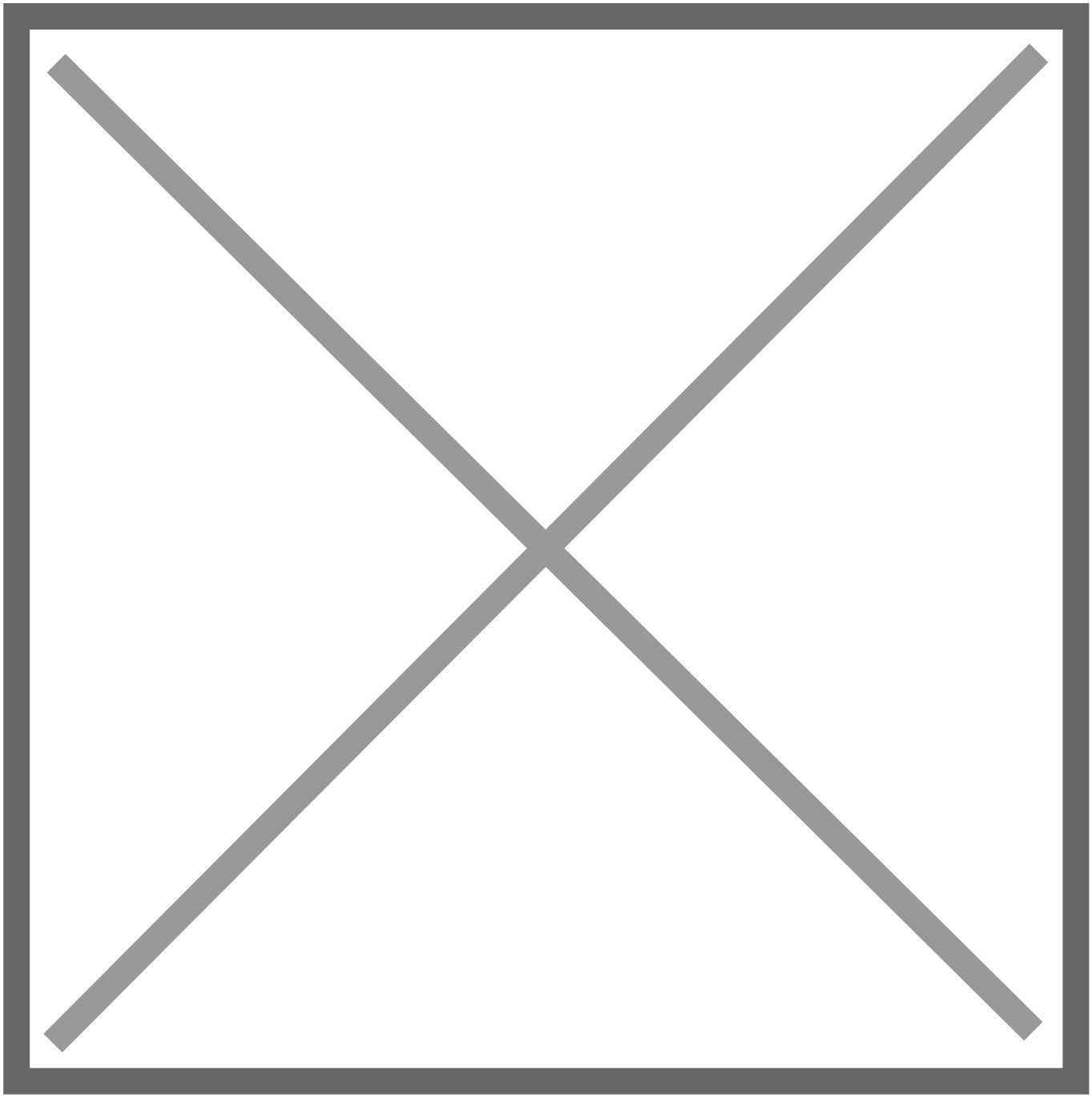
База данных хранится в файле `C:\Windows\NTDS\ntds.dit` на контроллере домена. Поэтому, если кто-то украдет этот файл, он сможет получить доступ ко всей информации об объектах домена (компьютерах, пользователях, группах, политиках и т.д.), включая учетные данные и хэши пользователей. Следовательно, доступ к этому файлу и к контроллерам домена должен быть ограничен администраторами домена и отслеживаться корреляционными правилам командой мониторинга SOC.

При получении доступа к учетной записи администратора домена, можно сделать дамп содержимого базы данных контроллера домена, чтобы прочитать некоторые

конфиденциальные данные, такие как `krbtgt` учетные данные пользователя, для создания золотого билета (Golden Ticket) и дальнейшего свободного продвижения по всей Windows инфраструктуре.
Чтобы извлечь содержимое базы данных, злоумышленник может войти в систему на контроллере домена и локально выгрузить файл NTDS.dit с помощью встроенных в системе утилит ntdsutil или vssadmin. Данные утилиты нужны, потому что просто так взять и скопировать файл `C:\Windows\NTDS\ntds.dit` не получится так, как он используется всегда системой. Есть еще наиболее изящный для злоумышленников способов заполучить базу NTDS.dit, злоумышленник может заполучить административный доступ к системе виртуализации, где у него будет возможность сделать мгновенный снимок (SnapShot) виртуальной машины с сервером Active Directory, далее он выгружает к себе снимок виртуальной машины и делает с ним что хочет, в нашем случае, разбирает базу NTDS.dit. При этом система мониторинга уже это не сможет выявить, если только отслеживать на более раннем этапе действия учтеных записей по созданию снапшотов критичных серверов и их выгрузке из системы виртуализации. Детальнее об этом рассмотрим далее в уроке 4.4 текущего курса.

Рассмотрим вариант дампа базы NTDS.dit с помощью утилиты ntdsutil. Классическая команда выглядит следующим образом

```
ntdsutil "activate instance ntds" "ifm" "create full C:\Windows\Temp\NTDS" quit quit
```



Выполнение вышеуказанной команды можно зафиксировать с помощью события создания процесса **EventID=1** журнала Symon, **EventID=4688** журнала Security и с помощью события запуска скриптблоков **EventID=4104** журнала Powershell на контроллере домена.

Обратите ниже внимание на строку в событии с EventID=1(Sysmon)

```
<Data Name="CommandLine">ntdsutil "activate instance ntds" "ifm" "create full
C:\Windows\Temp\NTDS" quit quit</Data>
```

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>
  <Provider Name="Microsoft-Windows-Sysmon" Guid="{5770385F-C22A-43E0-BF4C-06F5698FFBD9}" />
  <EventID>1</EventID>
  <Version>5</Version>
  <Level>4</Level>
  <Task>1</Task>
  <Opcode>0</Opcode>
  <Keywords>0x8000000000000000</Keywords>
  <TimeCreated SystemTime="2024-02-10T09:42:40.633769700Z" />
  <EventRecordID>8119972</EventRecordID>
  <Correlation />
  <Execution ProcessID="1604" ThreadID="2020" />
  <Channel>Microsoft-Windows-Sysmon/Operational</Channel>
  <Computer>WIN-O6QVDOTOFCA.lab.local</Computer>
  <Security UserID="S-1-5-18" />
  </System>
- <EventData>
  <Data Name="RuleName">-</Data>
  <Data Name="UtcTime">2024-02-10 09:42:40.602</Data>
  <Data Name="ProcessGuid">{460797FE-4510-65C7-AF07-000000001800}</Data>
  <Data Name="ProcessId">3856</Data>
  <Data Name="Image">C:\Windows\System32\ntdsutil.exe</Data>
  <Data Name="FileVersion">6.3.9600.16384 (winblue_rtm.130821-1623)</Data>
  <Data Name="Description">NT5DS</Data>
  <Data Name="Product">Microsoft® Windows® Operating System</Data>
  <Data Name="Company">Microsoft Corporation</Data>
  <Data Name="OriginalFileName">ntdsutil.exe</Data>
  <Data Name="CommandLine">ntdsutil "activate instance ntds" "ifm" "create full
C:\Windows\Temp\NTDS" quit quit</Data>
  <Data Name="CurrentDirectory">C:\Temp\</Data>
  <Data Name="User">LAB\Administrator</Data>
  <Data Name="LogonGuid">{460797FE-9671-65BE-4054-040000000000}</Data>
  <Data Name="LogonId">0x45440</Data>
  <Data Name="TerminalSessionId">1</Data>
  <Data Name="IntegrityLevel">High</Data>
  <Data
Name="Hashes">MD5=0741B31AF51B150DF84BFEFD4A15C624,SHA256=D2C7BD14D91124401AAC6F19DD2D2EDDA0EA
AC55CFFB654583444137960EEDCA,IMPHASH=6D8CC7C1C74B6AA69C6C1F189D5781D9</Data>
  <Data Name="ParentProcessGuid">{460797FE-9696-65BE-3A00-000000001800}</Data>
```

```
<Data Name="ParentProcessId">2056</Data>
<Data Name="ParentImage">C:\Windows\System32\cmd.exe</Data>
<Data Name="ParentCommandLine">"C:\Windows\system32\cmd.exe"</Data>
<Data Name="ParentUser">LAB\Administrator</Data>
</EventData>
</Event>
```

Так же сгенерировалось событие создания файла с **EventID=11** журнала Sysmon. Обратите внимание на следующие поля:

- `<Data Name="Image">C:\Windows\system32\ntdsutil.exe</Data>` — процес который создал файл
- `<Data Name="TargetFilename">C:\Windows\Temp\NTDS\Active Directory\ntds.dit</Data>` — этот файл уже может спокойно себе скачать злоумышленник, например, для дальнейшей генерации Golden Ticket.

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>
<Provider Name="Microsoft-Windows-Sysmon" Guid="{5770385F-C22A-43E0-BF4C-06F5698FFBD9}" />
<EventID>11</EventID>
<Version>2</Version>
<Level>4</Level>
<Task>11</Task>
<Opcode>0</Opcode>
<Keywords>0x8000000000000000</Keywords>
<TimeCreated SystemTime="2024-02-10T09:42:51.306219100Z" />
<EventRecordID>8120158</EventRecordID>
<Correlation />
<Execution ProcessID="1604" ThreadID="2020" />
<Channel>Microsoft-Windows-Sysmon/Operational</Channel>
<Computer>WIN-O6QVDOTOFCA.lab.local</Computer>
<Security UserID="S-1-5-18" />
</System>
- <EventData>
<Data Name="RuleName">Suspicious</Data>
<Data Name="UtcTime">2024-02-10 09:42:51.306</Data>
<Data Name="ProcessGuid">{460797FE-4510-65C7-AF07-000000001800}</Data>
<Data Name="ProcessId">3856</Data>
<Data Name="Image">C:\Windows\system32\ntdsutil.exe</Data>
```

```
  <Data Name="TargetFilename">C:\Windows\Temp\NTDS\Active Directory\ntds.dit</Data>
  <Data Name="CreationUtcTime">2024-02-10 09:42:51.306</Data>
  <Data Name="User">LAB\Administrator</Data>
  </EventData>
  </Event>
```

Пример правила на Sigma для выявления атак связанных с кражей ntds.dit ниже.

```
title: Suspicious Process Patterns NTDS.DIT Exfil
id: 8bc64091-6875-4881-aaf9-7bd25b5dda08
status: test
description: Detects suspicious process patterns used in NTDS.DIT exfiltration
references:
    - https://www.ired.team/offensive-security/credential-access-and-credential-
dumping/ntds.dit-enumeration
    - https://www.n00py.io/2022/03/manipulating-user-passwords-without-mimikatz/
    - https://pentestlab.blog/tag/ntds-dit/
    -
https://github.com/samratashok/nishang/blob/414ee1104526d7057f9adaeee196d91ae447283e/Gather/Co
py-VSS.ps1
    - https://github.com/zcgonvh/NTDSDumpEx
    - https://github.com/rapid7/metasploit-
framework/blob/d297adcebb5c1df6fe30b12ca79b161deb71571c/data/post/powershell/NTDSgrab.ps1
    - https://blog.talosintelligence.com/2022/08/recent-cyber-attack.html?m=1
author: Florian Roth (Nextron Systems)
date: 2022/03/11
modified: 2022/11/10
tags:
    - attack.credential_access
    - attack.t1003.003
logsource:
    product: windows
    category: process_creation
detection:
    selection_tool:
        # https://github.com/zcgonvh/NTDSDumpEx
        - Image|endswith:
              - '\NTDSDump.exe'
```

```
            - '\NTDSDumpEx.exe'
        - CommandLine|contains|all:
              # ntdsdumpex.exe -d ntds.dit -o hash.txt -s system.hiv
              - 'ntds.dit'
              - 'system.hiv'
        - CommandLine|contains: 'NTDSgrab.ps1'
    selection_oneliner_1:
        # powershell "ntdsutil.exe 'ac i ntds' 'ifm' 'create full c:\temp' q q"
        CommandLine|contains|all:
            - 'ac i ntds'
            - 'create full'
    selection_onliner_2:
        # cmd.exe /c copy z:\windows\ntds\ntds.dit c:\exfil\ntds.dit
        CommandLine|contains|all:
            - '/c copy '
            - '\windows\ntds\ntds.dit'
    selection_onliner_3:
        # ntdsutil "activate instance ntds" "ifm" "create full c:\windows\temp\data\" "quit"
"quit"
        CommandLine|contains|all:
            - 'activate instance ntds'
            - 'create full'
    selection_powershell:
        CommandLine|contains|all:
            - 'powershell'
            - 'ntds.dit'
    set1_selection_ntds_dit:
        CommandLine|contains: 'ntds.dit'
    set1_selection_image_folder:
        - ParentImage|contains:
              - '\apache'
              - '\tomcat'
              - '\AppData\'
              - '\Temp\'
              - '\Public\'
              - '\PerfLogs\'
        - Image|contains:
              - '\apache'
              - '\tomcat'
              - '\AppData\'
```

```
            - '\Temp\'
            - '\Public\'
            - '\PerfLogs\'
    condition: 1 of selection* or all of set1*
falsepositives:
    - Unknown
level: high
```

Рассмотрим следующий вариант дампа базы ntds.dit с помощью утилиты vssadmin. Классическая команда выглядит следующим образом `vssadmin create shadow /for=C:`. Выявить активность можно с помощью событий **EventID=4688** журнала Security, **EventID=1** журнала Sysmon.

Обратите внимание на зафиксированную командную строку в событии `<Data Name="CommandLine">vssadmin create shadow /for=C:</Data>`

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>
  <Provider Name="Microsoft-Windows-Sysmon" Guid="{5770385F-C22A-43E0-BF4C-06F5698FFBD9}" />
  <EventID>1</EventID>
  <Version>5</Version>
  <Level>4</Level>
  <Task>1</Task>
  <Opcode>0</Opcode>
  <Keywords>0x8000000000000000</Keywords>
  <TimeCreated SystemTime="2024-02-10T12:14:55.042278500Z" />
  <EventRecordID>8140166</EventRecordID>
  <Correlation />
  <Execution ProcessID="1604" ThreadID="2020" />
  <Channel>Microsoft-Windows-Sysmon/Operational</Channel>
  <Computer>WIN-O6QVDOTOFCA.lab.local</Computer>
  <Security UserID="S-1-5-18" />
  </System>
- <EventData>
  <Data Name="RuleName">-</Data>
  <Data Name="UtcTime">2024-02-10 12:14:55.026</Data>
  <Data Name="ProcessGuid">{460797FE-68BF-65C7-EC07-000000001800}</Data>
  <Data Name="ProcessId">860</Data>
  <Data Name="Image">C:\Windows\System32\vssadmin.exe</Data>
```

```
  <Data Name="FileVersion">6.3.9600.17415 (winblue_r4.141028-1500)</Data>
  <Data Name="Description">Command Line Interface for Microsoft® Volume Shadow Copy
Service</Data>
  <Data Name="Product">Microsoft® Windows® Operating System</Data>
  <Data Name="Company">Microsoft Corporation</Data>
  <Data Name="OriginalFileName">VSSADMIN.EXE</Data>
  <Data Name="CommandLine">vssadmin create shadow /for=C:</Data>
  <Data Name="CurrentDirectory">C:\Temp\</Data>
  <Data Name="User">LAB\Administrator</Data>
  <Data Name="LogonGuid">{460797FE-9671-65BE-4054-040000000000}</Data>
  <Data Name="LogonId">0x45440</Data>
  <Data Name="TerminalSessionId">1</Data>
  <Data Name="IntegrityLevel">High</Data>
  <Data
Name="Hashes">MD5=D9EE4ACBA0FD5AF721EC2CE5226B5E2E,SHA256=AF08DA2358D55665FAE06AE694129B5F3778
989E93F5F369E0B594E1A2BC521E,IMPHASH=E29ADBD24C814ABA83B2027E5BB6C452</Data>
  <Data Name="ParentProcessGuid">{460797FE-9696-65BE-3A00-000000001800}</Data>
  <Data Name="ParentProcessId">2056</Data>
  <Data Name="ParentImage">C:\Windows\System32\cmd.exe</Data>
  <Data Name="ParentCommandLine">"C:\Windows\system32\cmd.exe"</Data>
  <Data Name="ParentUser">LAB\Administrator</Data>
  </EventData>
  </Event>
```

Соответственно мы обнаружим создание файла `C:\Windows\Temp\ntds.dit.save` в событие **EventID=11** журнала Sysmon.

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>
  <Provider Name="Microsoft-Windows-Sysmon" Guid="{5770385F-C22A-43E0-BF4C-06F5698FFBD9}" />
  <EventID>11</EventID>
  <Version>2</Version>
  <Level>4</Level>
  <Task>11</Task>
  <Opcode>0</Opcode>
  <Keywords>0x8000000000000000</Keywords>
  <TimeCreated SystemTime="2024-02-10T12:19:54.526660200Z" />
  <EventRecordID>8140853</EventRecordID>
  <Correlation />
```

```
  <Execution ProcessID="1604" ThreadID="2020" />
  <Channel>Microsoft-Windows-Sysmon/Operational</Channel>
  <Computer>WIN-O6QVDOTOFCA.lab.local</Computer>
  <Security UserID="S-1-5-18" />
  </System>
- <EventData>
  <Data Name="RuleName">Executable</Data>
  <Data Name="UtcTime">2024-02-10 12:19:54.526</Data>
  <Data Name="ProcessGuid">{460797FE-9696-65BE-3A00-000000001800}</Data>
  <Data Name="ProcessId">2056</Data>
  <Data Name="Image">C:\Windows\system32\cmd.exe</Data>
  <Data Name="TargetFilename">C:\Windows\Temp\ntds.dit.save</Data>
  <Data Name="CreationUtcTime">2024-02-10 12:19:54.526</Data>
  <Data Name="User">LAB\Administrator</Data>
  </EventData>
  </Event>
```

Для заметания следов следующим шагом злоумышленник может выполнить команду:

```
vssadmin delete shadows /shadow={e1f24f05-c919-4f96-ac60-fad4bfb07459}
```

Обратите внимание на зафиксированную команду в событии:

```
<Data Name="CommandLine">vssadmin delete shadows /shadow={e1f24f05-c919-4f96-ac60-
fad4bfb07459}</Data>
```

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>
  <Provider Name="Microsoft-Windows-Sysmon" Guid="{5770385F-C22A-43E0-BF4C-06F5698FFBD9}" />
  <EventID>1</EventID>
  <Version>5</Version>
  <Level>4</Level>
  <Task>1</Task>
  <Opcode>0</Opcode>
  <Keywords>0x8000000000000000</Keywords>
```

<TimeCreated SystemTime="2024-02-10T12:25:31.136231200Z" />
<EventRecordID>8141617</EventRecordID>
<Correlation />
<Execution ProcessID="1604" ThreadID="2020" />
<Channel>Microsoft-Windows-Sysmon/Operational</Channel>
<Computer>WIN-O6QVDOTOFCA.lab.local</Computer>
<Security UserID="S-1-5-18" />
</System>
- <EventData>
<Data Name="RuleName">-</Data>
<Data Name="UtcTime">2024-02-10 12:25:31.136</Data>
<Data Name="ProcessGuid">{460797FE-6B3B-65C7-F207-000000001800}</Data>
<Data Name="ProcessId">3332</Data>
<Data Name="Image">C:\Windows\System32\vssadmin.exe</Data>
<Data Name="FileVersion">6.3.9600.17415 (winblue_r4.141028-1500)</Data>
<Data Name="Description">Command Line Interface for Microsoft® Volume Shadow Copy Service</Data>
<Data Name="Product">Microsoft® Windows® Operating System</Data>
<Data Name="Company">Microsoft Corporation</Data>
<Data Name="OriginalFileName">VSSADMIN.EXE</Data>
<Data Name="CommandLine">vssadmin delete shadows /shadow={e1f24f05-c919-4f96-ac60-fad4bfb07459}</Data>
<Data Name="CurrentDirectory">C:\Temp\</Data>
<Data Name="User">LAB\Administrator</Data>
<Data Name="LogonGuid">{460797FE-9671-65BE-4054-040000000000}</Data>
<Data Name="LogonId">0x45440</Data>
<Data Name="TerminalSessionId">1</Data>
<Data Name="IntegrityLevel">High</Data>
<Data Name="Hashes">MD5=D9EE4ACBA0FD5AF721EC2CE5226B5E2E,SHA256=AF08DA2358D55665FAE06AE694129B5F3778989E93F5F369E0B594E1A2BC521E,IMPHASH=E29ADBD24C814ABA83B2027E5BB6C452</Data>
<Data Name="ParentProcessGuid">{460797FE-9696-65BE-3A00-000000001800}</Data>
<Data Name="ParentProcessId">2056</Data>
<Data Name="ParentImage">C:\Windows\System32\cmd.exe</Data>
<Data Name="ParentCommandLine">"C:\Windows\system32\cmd.exe"</Data>
<Data Name="ParentUser">LAB\Administrator</Data>
</EventData>
</Event>

Далее злоумышленник любым из возможных способов может забрать к себе файл `C:\Windows\Temp\ntds.dit.save`.

Действия при компроментации системы

1. Сбросьте все пароли учетных записей пользователей:
   - сбросьте дважды (с интервалом не менее восьми часов) пароль пользователя KRBTGT;
   - сбросить все пароли администратора;
   - сбросить пароли всех сервисных учетных записей;
   - сбросить все пароли учетных записей компьютеров.
2. Проверьте значение параметра срока жизни пароля для компьютеров. Злоумышленники могут изменить этот срок, чтобы предоставить себе доступ с использованием машинных хэшей на более длительный срок.
3. Сбросить все пароли LAPS.
4. Сбросить разрешения для объекта AdminSDHolders.
5. Удалить у всех администраторов домена и администраторов систем все данные из атрибута msDS-KeyCredentialLink.
6. Занести всех администраторов домена и администраторов систем в группу Protected Users.
7. Отозвать и перевыпустить все сертификаты ADCS.
8. Проверьте наличие вредоносных запланированных задач.
9. Проверьте наличие вредоносных автозапусков или других механизмов сохранения на основе реестра.
10. Проверьте наличие бэкдоров в стиле utilman.
11. Проверьте наличие вредоносных принтеров/драйверов принтеров.
12. Просмотрите права делегированного доступа Active Directory (RBCD Bakdoors).
13. Ротация сертификатов подписи токенов ADFS и сертификатов расшифровки токенов.
14. Проверьте дескрипторы безопасности Service Control Manager (SCM).
15. Проверьте наличие изменений объекта в соответствии с первоначальными временными рамками доступа/событий.
16. Проверка членства в группах на соответствие известным базовым показателям.
17. Просмотрите сценарии входа в GPO и SYSVOL.
18. Просмотрите домены и доверительные отношения Active Directory.
19. Установить все последние обновления безопасности.

Revision #2
Created 27 October 2025 14:46:08 by Admin
Updated 27 October 2025 16:24:03 by Admin