Аналог netcat с помощью PowerShell на Windows

Часто для проверки поступления события требуется специальные утилиты для отправки тестового сообщения с Windows машины на коллектор, в рамках этой статьи будет представлен скрипт на PowerShell , с помощью которого можно будет отправить тестовое сообщение по IP:PORT по протоколу TCP.

Код скрипта:

```
Function Send-TCPMessage {
    Param (
        [Parameter(Mandatory=$true, Position=0)]
        [ValidateNotNullOrEmpty()]
        [string]
        $EndPoint
      ,
        [Parameter(Mandatory=$true, Position=1)]
        [int]
        $Port
      ,
        [Parameter(Mandatory=$true, Position=2)]
        [string]
        $Message
    )
    Process {
      # Setup connection
      $IP = [System.Net.Dns]::GetHostAddresses($EndPoint)
      $Address = [System.Net.IPAddress]::Parse($IP)
      $Socket = New-Object System.Net.Sockets.TCPClient($Address,$Port)

      # Setup stream wrtier
      $Stream = $Socket.GetStream()
      $Writer = New-Object System.IO.StreamWriter($Stream)

      # Write message to stream
```

```
        $Message | % {
            $Writer.WriteLine($_)
            $Writer.Flush()
        }

        # Close connection and stream
        $Stream.Close()
        $Socket.Close()
    }
}
```

Для отправки тестового сообщения нужно выполнить слледующую команду:

```
Send-TCPMessage -Port 5578 -Endpoint 10.68.85.125 -message "KUMA the best SIEM !"
```

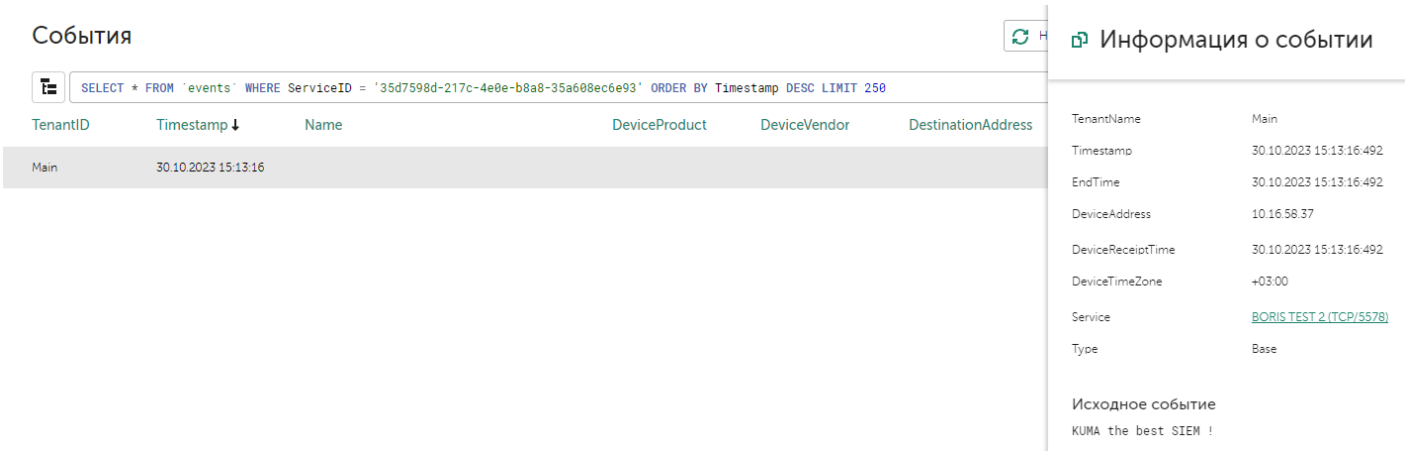
Вот как это выглядит в работе:

```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\windows\system32> Function Send-TCPMessage {
>>     Param (
>>             [Parameter(Mandatory=$true, Position=0)]
>>             [ValidateNotNullOrEmpty()]
>>             [string]
>>             $EndPoint
>>         ,
>>             [Parameter(Mandatory=$true, Position=1)]
>>             [int]
>>             $Port
>>         ,
>>             [Parameter(Mandatory=$true, Position=2)]
>>             [string]
>>             $Message
>>     )
>>     Process {
>>         # Setup connection
>>         $IP = [System.Net.Dns]::GetHostAddresses($EndPoint)
>>         $Address = [System.Net.IPAddress]::Parse($IP)
>>         $Socket = New-Object System.Net.Sockets.TCPClient($Address,$Port)
>>
>>         # Setup stream wrtier
>>         $Stream = $Socket.GetStream()
>>         $Writer = New-Object System.IO.StreamWriter($Stream)
>>
>>         # Write message to stream
>>         $Message | % {
>>             $Writer.WriteLine($_)
>>             $Writer.Flush()
>>         }
>>
>>         # Close connection and stream
>>         $Stream.Close()
>>         $Socket.Close()
>>     }
>> }
PS C:\windows\system32> Send-TCPMessage -Port 5578 -Endpoint 10.68.85.125 -message "KUMA the best SIEM !"
PS C:\windows\system32>
```

На стороне KUMA:



Revision #1
Created 30 October 2023 12:08:46 by Boris Rzr
Updated 26 June 2024 14:59:17 by Boris Rzr