

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 следующую команду:

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

Вот как это выглядит в работе:

```
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 writer
>>     $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:

События

SELECT * FROM 'events' WHERE ServiceID = '35d7598d-217c-4e0e-b8a8-35a608ec6e93' ORDER BY Timestamp DESC LIMIT 250

TenantID	Timestamp ↓	Name	DeviceProduct	DeviceVendor	DestinationAddress
Main	30.10.2023 15:13:16				

Информация о событии

TenantName	Main
Timestamp	30.10.2023 15:13:16:492
EndTime	30.10.2023 15:13:16:492
DeviceAddress	10.16.58.37
DeviceReceiptTime	30.10.2023 15:13:16:492
DeviceTimeZone	+03:00
Service	BORIS TEST 2 (TCP/5578)
Type	Base
Исходное событие	
KUMA the best SIEM !	