# Monitor remote access vpn users

How-to troubleshoot or measure the service

This tcpstats command from netsh on windows gives you the state of tcp statistics with that you can identify connectivity issues between the vpn client and the server.
In the output we see different counters and gauges that shows us the network state on layer 4 level in numbers.

2023-06-25 23:09:20

```
PS C:\Users\killer> netsh interface ipv4 show tcpstats

TCP Statistics
----------------------------------------------------
Timeout Algorithm:                      Van Jacobson's Algorithm
Minimum Timeout:                        5
Maximum Timeout:                        4294967295
Maximum Connections:                    Dynamic
Active Opens:                           18721
Passive Opens:                          33
Attempts Failed:                        90
Established Resets:                     395
Currently Established:                  89
In Segments:                            1368161
Out Segments:                           542512
Retransmitted Segments:                 1106
In Errors:                              0
Out Resets:                             2176
Fastopen Active Opens:                  0
Fastopen Passive Opens:                 0
Fastopen Attempts Failed:               0
Retransmits Of First SYN:               102
Retransmits Of First SYN (Fastopen):    0
```

This command is polled in every 5 minutes, but the poll time can be customized for shorter or longer time as requested by the case

2023-06-25 23:09:20

```
PS C:\Users\killer> netsh interface ipv4 show tcpstats

TCP Statistics
----------------------------------------------------
Timeout Algorithm:                      Van Jacobson's Algorithm
Minimum Timeout:                        5
Maximum Timeout:                        4294967295
Maximum Connections:                    Dynamic
Active Opens:                           18721
Passive Opens:                          33
Attempts Failed:                        90
Established Resets:                     395
Currently Established:                  89
In Segments:                            1368161
Out Segments:                           542512
Retransmitted Segments:                 1106
In Errors:                              0
Out Resets:                             2176
Fastopen Active Opens:                  0
Fastopen Passive Opens:                 0
Fastopen Attempts Failed:               0
Retransmits Of First SYN:               102
Retransmits Of First SYN (Fastopen):    0
```

…run the command with 5 minutes poll time…

Issuing the command after 5 minutes we see the counters or gauges are changing

2023-06-25 23:09:20

```
PS C:\Users\killer> netsh interface ipv4 show tcpstats

TCP Statistics
----------------------------------------------------
Timeout Algorithm:                      Van Jacobson's Algorithm
Minimum Timeout:                        5
Maximum Timeout:                        4294967295
Maximum Connections:                    Dynamic
Active Opens:                           18721
Passive Opens:                          33
Attempts Failed:                        90
Established Resets:                     395
Currently Established:                  89
In Segments:                            1368161
Out Segments:                           542512
Retransmitted Segments:                 1106
In Errors:                              0
Out Resets:                             2176
Fastopen Active Opens:                  0
Fastopen Passive Opens:                 0
Fastopen Attempts Failed:               0
Retransmits Of First SYN:               102
Retransmits Of First SYN (Fastopen):    0
```

2023-06-25 23:14:20

```
PS C:\Users\killer> netsh interface ipv4 show tcpstats

TCP Statistics
----------------------------------------------------
Timeout Algorithm:                      Van Jacobson's Algorithm
Minimum Timeout:                        5
Maximum Timeout:                        4294967295
Maximum Connections:                    Dynamic
Active Opens:                           18830
Passive Opens:                          33
Attempts Failed:                        91
Established Resets:                     395
Currently Established:                  73
In Segments:                            1371417
Out Segments:                           544753
Retransmitted Segments:                 1152
In Errors:                              0
Out Resets:                             2186
Fastopen Active Opens:                  0
Fastopen Passive Opens:                 0
Fastopen Attempts Failed:               0
Retransmits Of First SYN:               103
Retransmits Of First SYN (Fastopen):    0
```

This outputs will be converted into a simple json that contains the numbers from the original output.
With powershell you can easily convert the unstructured data to a structured data format like json.

2023-06-25 23:09:20


```
{
  "Minimum_Timeout": 5,
  "Maximum_Timeout": 4294967295,
  "Active_Opens": 18721,
  "Passive_Opens": 33,
  "Attempts_Failed": 90,
  "Established_Resets": 395,
  "Currently_Established": 89,
  "In_Segments": 1368161,
  "Out_Segments": 542512,
  "Retransmitted_Segments": 1106,
  "In_Errors": 0,
  "Out_Resets": 2176,
  "Fastopen_Active_Opens": 0,
  "Fastopen_Passive_Opens": 0,
  "Fastopen_Attempts_Failed": 0,
  "Retransmits_Of_First_SYN": 102,
  "Retransmits_Of_First_SYN_(Fastopen)": 0
}
```


2023-06-25 23:14:20


```
{
  "Minimum_Timeout": 5,
  "Maximum_Timeout": 4294967295,
  "Active_Opens": 18830,
  "Passive_Opens": 33,
  "Attempts_Failed": 91,
  "Established_Resets": 395,
  "Currently_Established": 73,
  "In_Segments": 1371417,
  "Out_Segments": 544753,
  "Retransmitted_Segments": 1152,
  "In_Errors": 0,
  "Out_Resets": 2186,
  "Fastopen_Active_Opens": 0,
  "Fastopen_Passive_Opens": 0,
  "Fastopen_Attempts_Failed": 0,
  "Retransmits_Of_First_SYN": 103,
  "Retransmits_Of_First_SYN_(Fastopen)": 0
}
```

Once the conversion is ready the json data will be extended with additional informations.
We add the vpn client ip address, the username, the domain name, the vpn gateway address and the timestamp

2023-06-25 23:09:20


```
{
  "gpipv4address": "10.47.174.170",
  "currentuser": "akdaniel",
  "currentdomain": "PALOALTONETWORK",
  "gpgatewayaddress": "134.238.71.129",
  "timestamp": "2023-06-25 23:09:20",
  "subtype": "tcpstats",
  "Minimum_Timeout": 5,
  "Maximum_Timeout": 4294967295,
  "Active_Opens": 18721,
  "Passive_Opens": 33,
  "Attempts_Failed": 90,
  "Established_Resets": 395,
  "Currently_Established": 89,
  "In_Segments": 1368161,
  "Out_Segments": 542512,
  "Retransmitted_Segments": 1106,
  "In_Errors": 0,
  "Out_Resets": 2176,
  "Fastopen_Active_Opens": 0,
  "Fastopen_Passive_Opens": 0,
  "Fastopen_Attempts_Failed": 0,
  "Retransmits_Of_First_SYN": 102,
  "Retransmits_Of_First_SYN_(Fastopen)": 0
}
```

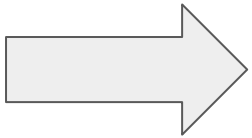

2023-06-25 23:14:20


```
{
  "gpipv4address": "10.47.174.170",
  "currentuser": "akdaniel",
  "currentdomain": "PALOALTONETWORK",
  "gpgatewayaddress": "134.238.71.129",
  "timestamp": "2023-06-25 23:14:20",
  "subtype": "tcpstats",
  "Minimum_Timeout": 5,
  "Maximum_Timeout": 4294967295,
  "Active_Opens": 18830,
  "Passive_Opens": 33,
  "Attempts_Failed": 91,
  "Established_Resets": 395,
  "Currently_Established": 73,
  "In_Segments": 1371417,
  "Out_Segments": 544753,
  "Retransmitted_Segments": 1152,
  "In_Errors": 0,
  "Out_Resets": 2186,
  "Fastopen_Active_Opens": 0,
  "Fastopen_Passive_Opens": 0,
  "Fastopen_Attempts_Failed": 0,
  "Retransmits_Of_First_SYN": 103,
  "Retransmits_Of_First_SYN_(Fastopen)": 0
}
```

From now on the json is ready to processed by the logstash or telegraf services, depending on your needs.
For demonstration purposes we use the elasticsearch logstash and kibana stack


```
{
  "gpipv4address": "10.47.174.170",
  "currentuser": "akdaniel",
  "currentdomain": "PALOALTONETWORK",
  "gpgatewayaddress": "134.238.71.129",
  "timestamp": "2023-06-25 23:09:20",
  "subtype": "tcpstats",
  "Minimum_Timeout": 5,
  "Maximum_Timeout": 4294967295,
  "Active_Opens": 18721,
  "Passive_Opens": 33,
  "Attempts_Failed": 90,
  "Established_Resets": 395,
  "Currently_Established": 89,
  "In_Segments": 1368161,
  "Out_Segments": 542512,
  "Retransmitted_Segments": 1106,
  "In_Errors": 0,
  "Out_Resets": 2176,
  "Fastopen_Active_Opens": 0,
  "Fastopen_Passive_Opens": 0,
  "Fastopen_Attempts_Failed": 0,
  "Retransmits_Of_First_SYN": 102,
  "Retransmits_Of_First_SYN_(Fastopen)": 0
}
```

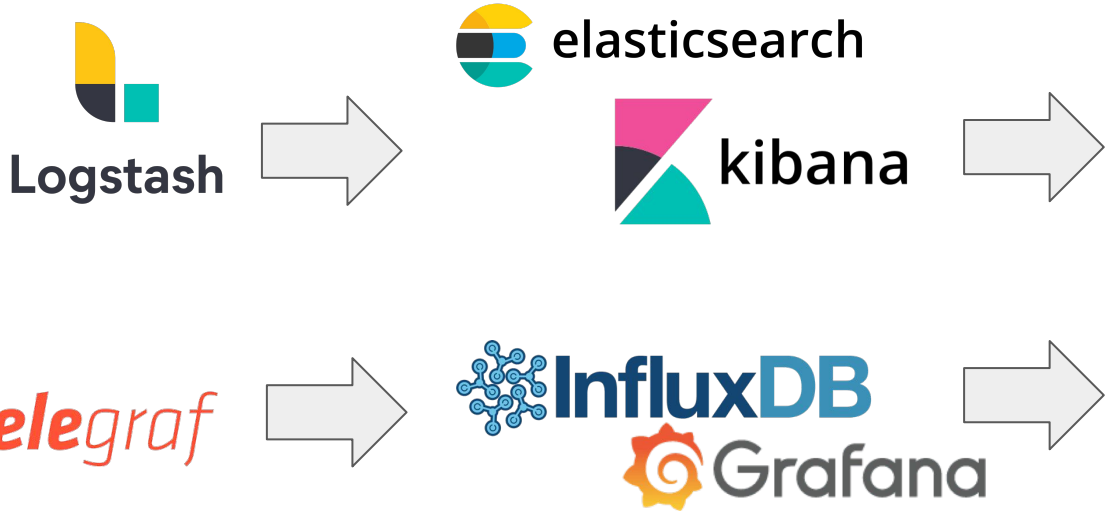
```
{
  "gpipv4address": "10.47.174.170",
  "currentuser": "akdaniel",
  "currentdomain": "PALOALTONETWORK",
  "gpgatewayaddress": "134.238.71.129",
  "timestamp": "2023-06-25 23:14:20",
  "subtype": "tcpstats",
  "Minimum_Timeout": 5,
  "Maximum_Timeout": 4294967295,
  "Active_Opens": 18830,
  "Passive_Opens": 33,
  "Attempts_Failed": 91,
  "Established_Resets": 395,
  "Currently_Established": 73,
  "In_Segments": 1371417,
  "Out_Segments": 544753,
  "Retransmitted_Segments": 1152,
  "In_Errors": 0,
  "Out_Resets": 2186,
  "Fastopen_Active_Opens": 0,
  "Fastopen_Passive_Opens": 0,
  "Fastopen_Attempts_Failed": 0,
  "Retransmits_Of_First_SYN": 103,
  "Retransmits_Of_First_SYN_(Fastopen)": 0
}
```

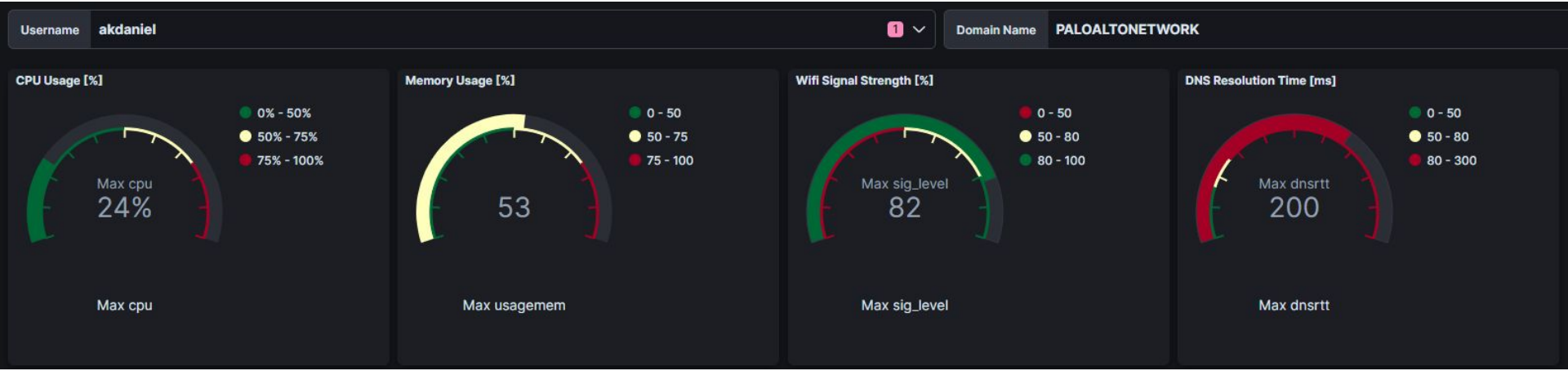
Logstash

telegraf

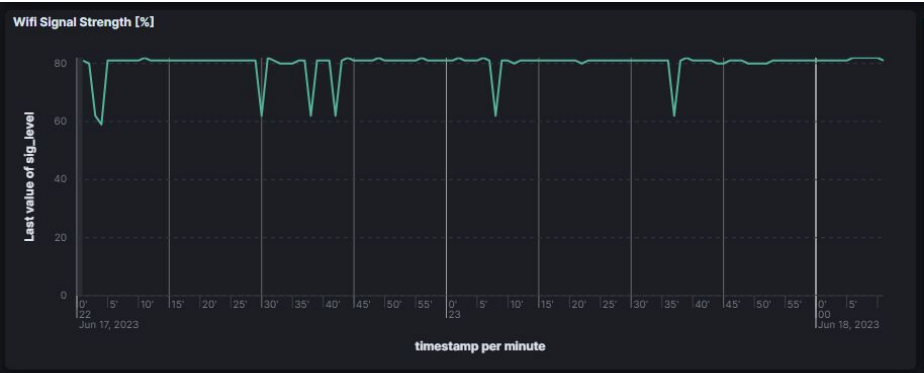
The logstash or telegraf writes the data to their databases as selected.
In case of logstash it writes the data or measurement to elasticsearch.
The telegraf writes the measurements to influx database.
The visualisation task belongs to kibana in case of logstash.
From the influx database we visualize the graphs with grafana



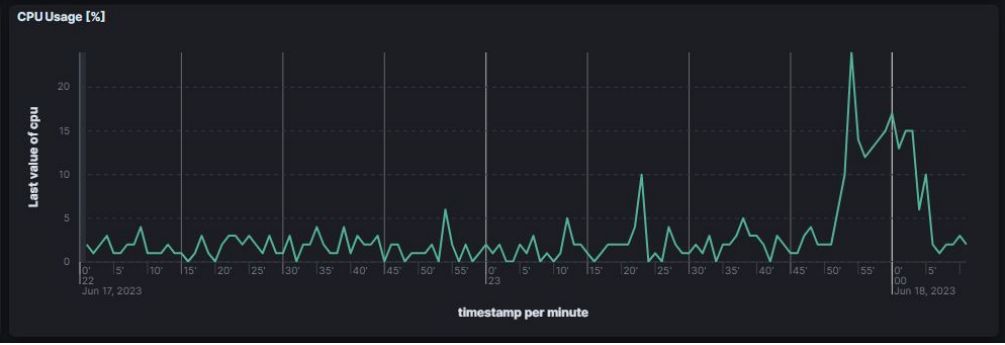

Apart from tcp statistics we can collect data about the cpu usage and the memory usage and the wifi signal strength or about the dns resolution time.



Apart from tcp statistics we can collect data about the cpu usage and the memory usage and the wifi signal strength or about the dns resolution time.





We can measure the round trip time to servers at different location to locate possible latency issues.
The psping utility gives you the ability to measure the round trip time with icmp or even with tcp if you set the destination port.

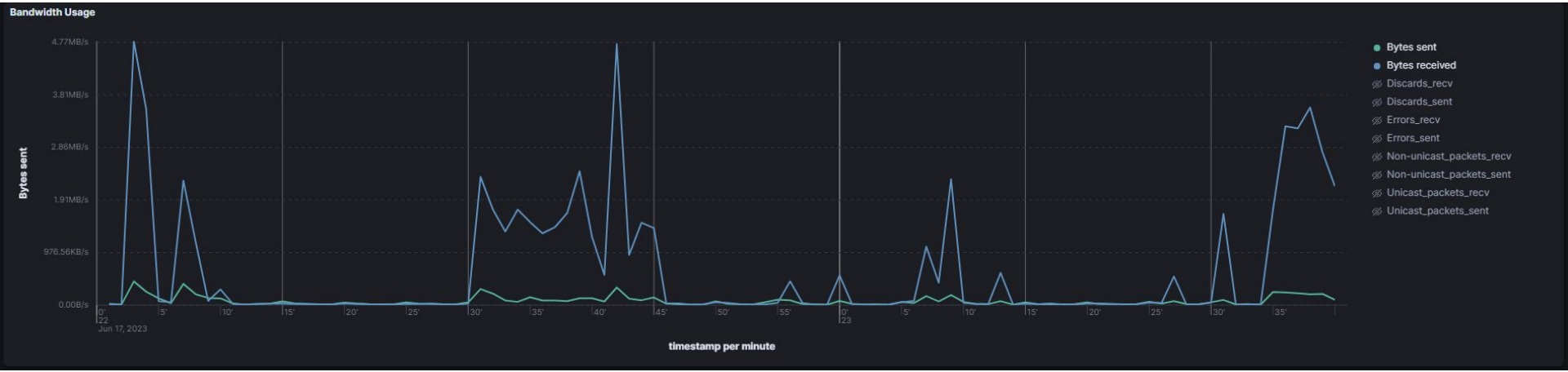
**DataCenter connectivity tests:**

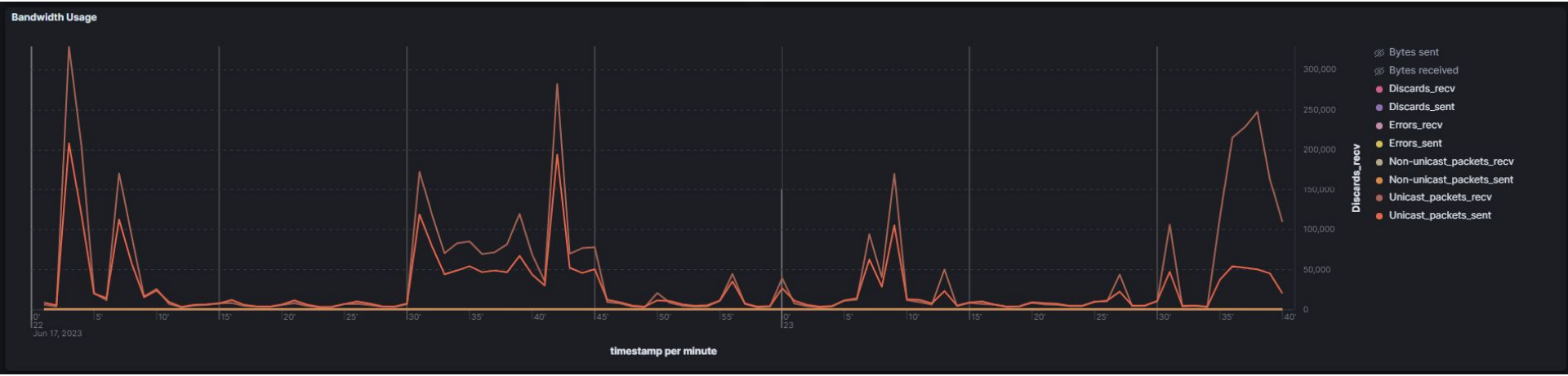
- confluence_server
- jira_server
- intranet

**Internet connectivity tests:**

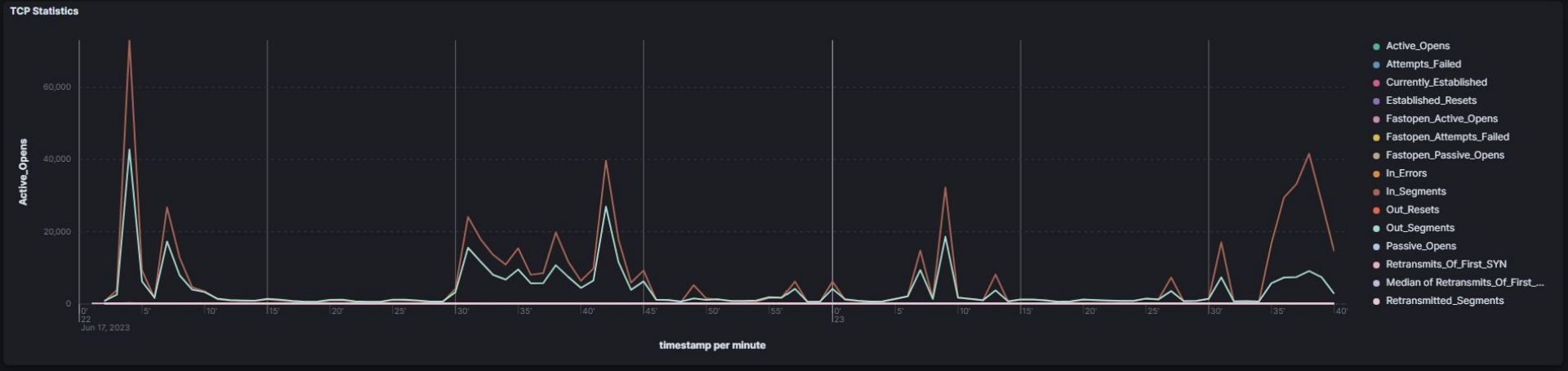
- google_root_dns
- cnn
- gp_gateway
- df_gateway

The bandwidth usage can be measured the same way with a different netsh command as well



Everything, that a netsh command or other command in powershell gives you back as a counter in their output,  can be measured the same way like the tcp statistics from netsh.

For demonstration purposes we use the following measurements with powershell on Windows:

- cpu usage
- memory usage
- dns resolution time
- ip statistics
- net statistics
- different round trip time measurement with psping
- tcp statistics
- wifi signal strength

For demonstration purposes we use the following measurements with powershell on Windows:

- cpu usage
- memory usage
- dns resolution time
- ip statistics
- net statistics
- different round trip time measurement with psping
- tcp statistics
- wifi signal strength


clean view

visibility everywhere