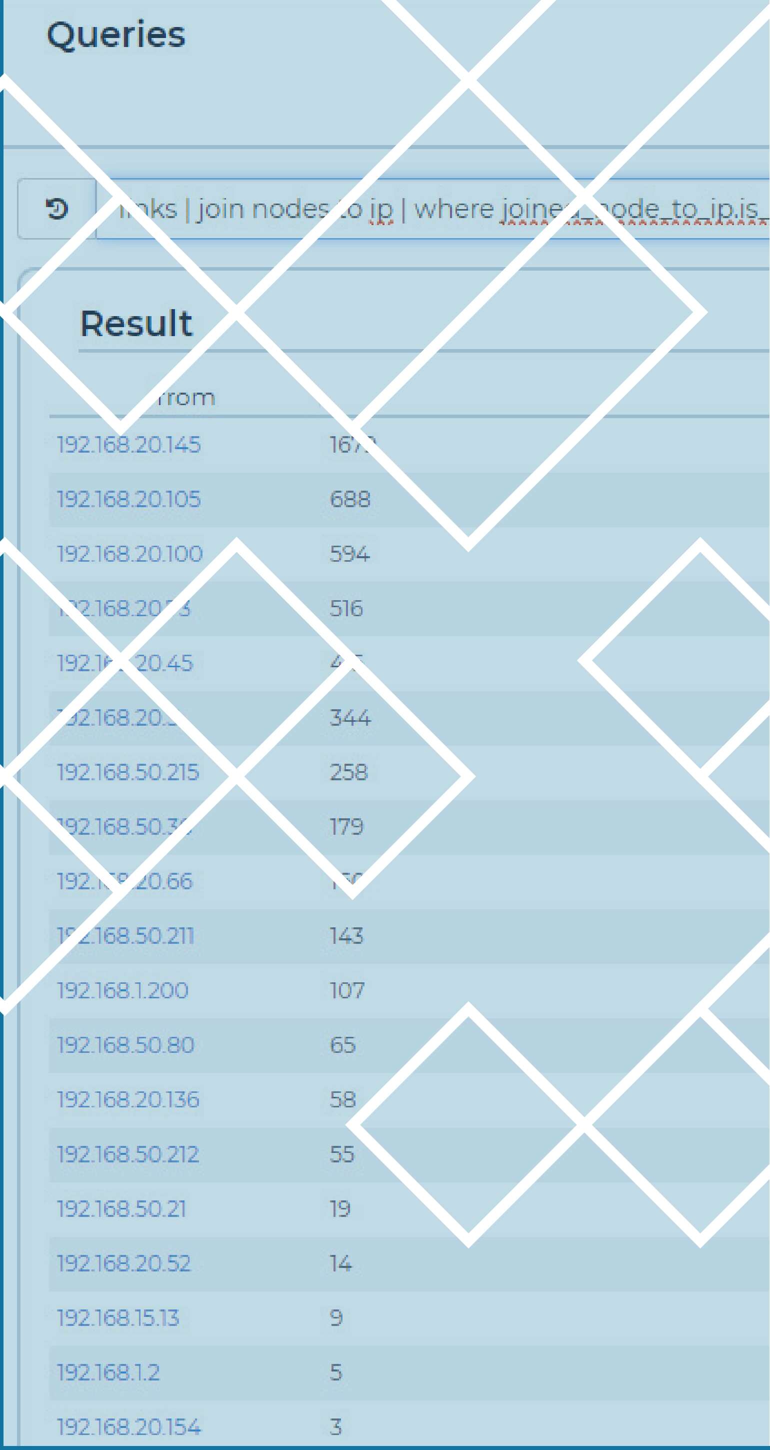
NOZOMI NETWORKS



Queries

links | join nodes_to_ip | where joined_node_to_ip.is_

Result

from	
192.168.20.145	[illegible]
192.168.20.105	688
192.168.20.100	594
[illegible]	516
192.168.20.45	[illegible]
[illegible]	344
192.168.50.215	258
[illegible]	179
192.168.20.66	[illegible]
192.168.50.211	143
192.168.1.200	107
192.168.50.80	65
192.168.20.136	58
192.168.50.212	55
192.168.50.21	19
192.168.20.52	14
192.168.15.13	9
192.168.1.2	5
192.168.20.154	3

WHITE PAPER

20 Nozomi Networks Queries: Unlock More Analytics on Assets, Traffic, and Alerts

Table of Contents

Introduction

One of the most popular Technical Support resources for Nozomi Networks customers has proven to be a knowledgebase article offering sample queries.

In this document, we have gathered a selection of 20 queries which can help build out your query library and enhance the value your Nozomi Networks system provides. Videos introducing these queries can also be found in the **Nozomi Networks Academy**.

These queries were chosen as much for their practical applications as for their examples of what's possible with Nozomi Network Query Language. Each query can be modified to expand its use, optimize it for your environment and ensure that you get the most benefit from the time you invest in understanding and applying the query.

If you need further assistance to understand or apply queries in the Nozomi Networks platform, please reach out to your Account Executive.

To easily copy and paste the queries from this White Paper, head to our **GitHub** link: **https://github.com/NozomiNetworks/query_library**

Get the queries here

Query 1

What are my top 10 alerts?

Why would you use this query?

This query is short and simple. Queries like this can help new users become familiar with the query language and rapidly produce results. This query can form the basis for many more complicated queries across a wide range of data sets.

When should you use this query?

To produce a quick, simple, high level result set for regular reporting.

What can you expect this query to produce?

This query returns a pie chart of results. It can be altered to return other display types.

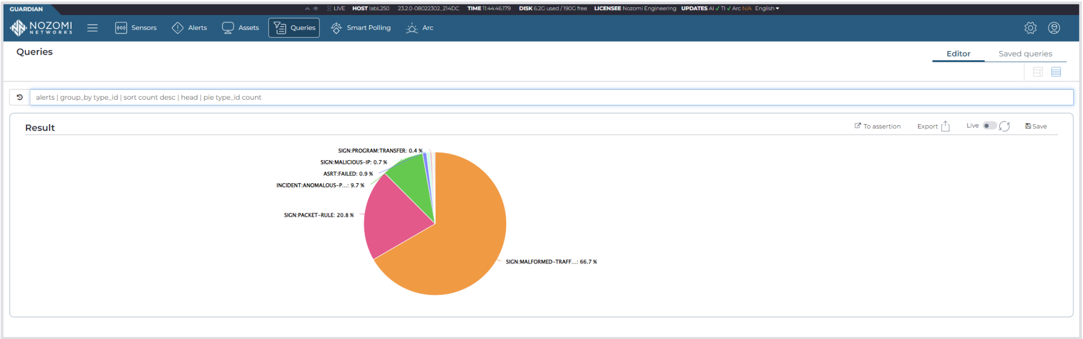
Where can this query be executed?

This can be run on a **Guardian**, **CMC**, or in **Vantage**.

What is the query format?

```
alerts | group_by type_id | sort count desc | head | pie type_id count
```

What should the results look like?



LIVE **HOST** labL250 23.2.0-08022302_214DC **TIME** 11:44:46.179 **DISK** 6.2G used / 190G free **LICENSEE** Nozomi Engineering **UPDATES** AI ✓ TI ✓ Arc N/A English

```
alerts | group_by type_id | sort count desc | head | pie type_id count
```

Query 2

What traffic exists between these two zones?

Why would you use this query?

This query can be used to monitor links between zones. This can be used to monitor existing links, or to identify new links.

What can you expect this query to produce?

The output will be a table containing details of links which have occurred between the two zones identified in the query.

When should you use this query?

You can use this query early in your security journey as you come to understand your network. It will also be valuable in the future to identify new links which are occurring.

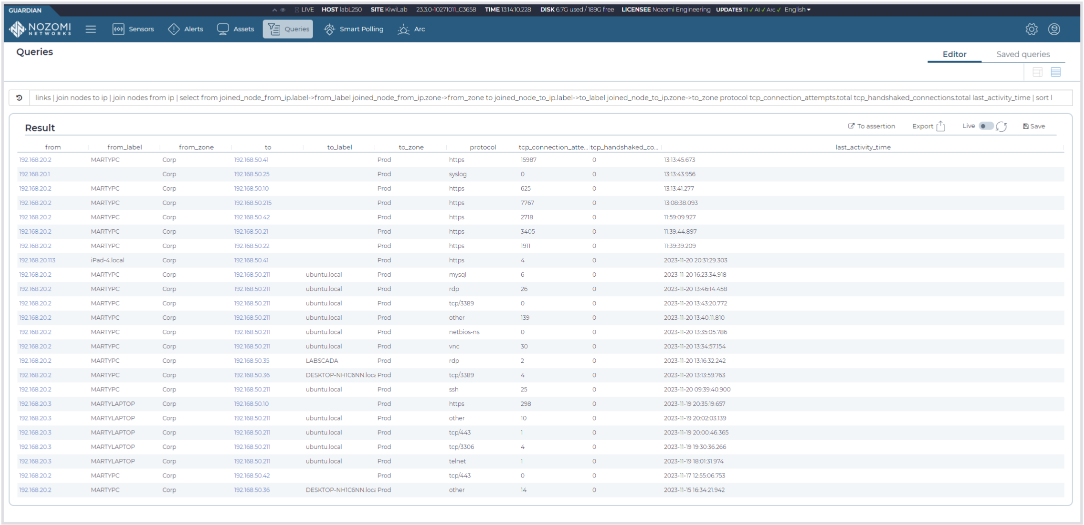
Where can this query be executed?

This query can be executed on a **Guardian** or **CMC**.

What is the query format?

```
links | join nodes to ip | join nodes from ip | select from joined_node_from_ip.label->from_label joined_node_from_ip.zone->from_zone to joined_node_to_ip.label->to_label joined_node_to_ip.zone->to_zone protocol tcp_connection_attempts.total tcp_handshaked_connections.total last_activity_time | sort last_activity_time desc | where from_zone include? Corp | where to_zone include? Prod
```

What should the results look like?


GUARDIAN
LIVE HOST labL250 SITE KiwiLab 23.3.0-10271011_C3658 TIME 13:14:10.228 DISK 6.7G used / 189G free LICENSEE Nozomi Engineering UPDATES TI ✓ AI ✓ Arc ✓ English
NOZOMI NETWORKS
Sensors
Alerts
Assets
Queries
Smart Polling
Arc
Editor
Saved queries
To assertion
Export
Live
Save

Queries

```
links | join nodes to ip | join nodes from ip | select from joined_node_from_ip.label->from_label joined_node_from_ip.zone->from_zone to joined_node_to_ip.label->to_label joined_node_to_ip.zone->to_zone protocol tcp_connection_attempts.total tcp_handshaked_connections.total last_activity_time | sort l
```

Result

from	from_label	from_zone	to	to_label	to_zone	protocol	tcp_connection_atte...	tcp_handshaked_co...	last_activity_time
192.168.20.2	MARTYPC	Corp	192.168.50.41		Prod	https	15987	0	13:13:45.673
192.168.20.1		Corp	192.168.50.25		Prod	syslog	0	0	13:13:43.956
192.168.20.2	MARTYPC	Corp	192.168.50.10		Prod	https	625	0	13:13:41.277
192.168.20.2	MARTYPC	Corp	192.168.50.215		Prod	https	7767	0	13:08:38.093
192.168.20.2	MARTYPC	Corp	192.168.50.42		Prod	https	2718	0	11:59:09.927
192.168.20.2	MARTYPC	Corp	192.168.50.21		Prod	https	3405	0	11:39:44.897
192.168.20.2	MARTYPC	Corp	192.168.50.22		Prod	https	1911	0	11:39:39.209
192.168.20.113	iPad-4.local	Corp	192.168.50.41		Prod	https	4	0	2023-11-20 20:31:29.303
192.168.20.2	MARTYPC	Corp	192.168.50.211	ubuntu.local	Prod	mysql	6	0	2023-11-20 16:23:34.918
192.168.20.2	MARTYPC	Corp	192.168.50.211	ubuntu.local	Prod	rdp	26	0	2023-11-20 13:46:14.458
192.168.20.2	MARTYPC	Corp	192.168.50.211	ubuntu.local	Prod	tcp/3389	0	0	2023-11-20 13:43:20.772
192.168.20.2	MARTYPC	Corp	192.168.50.211	ubuntu.local	Prod	other	139	0	2023-11-20 13:40:11.810
192.168.20.2	MARTYPC	Corp	192.168.50.211	ubuntu.local	Prod	netbios-ns	0	0	2023-11-20 13:35:05.786
192.168.20.2	MARTYPC	Corp	192.168.50.211	ubuntu.local	Prod	vnc	30	0	2023-11-20 13:34:57.154
192.168.20.2	MARTYPC	Corp	192.168.50.35	LABSCADA	Prod	rdp	2	0	2023-11-20 13:16:32.242
192.168.20.2	MARTYPC	Corp	192.168.50.36	DESKTOP-NH1C6NN.loca	Prod	tcp/3389	4	0	2023-11-20 13:13:59.763
192.168.20.2	MARTYPC	Corp	192.168.50.211	ubuntu.local	Prod	ssh	25	0	2023-11-20 09:39:40.900
192.168.20.3	MARTYLAPTOP	Corp	192.168.50.10		Prod	https	298	0	2023-11-19 20:35:19.657
192.168.20.3	MARTYLAPTOP	Corp	192.168.50.211	ubuntu.local	Prod	other	10	0	2023-11-19 20:02:03.139
192.168.20.3	MARTYLAPTOP	Corp	192.168.50.211	ubuntu.local	Prod	tcp/443	1	0	2023-11-19 20:00:46.365
192.168.20.3	MARTYLAPTOP	Corp	192.168.50.211	ubuntu.local	Prod	tcp/3306	4	0	2023-11-19 19:30:36.266
192.168.20.3	MARTYLAPTOP	Corp	192.168.50.211	ubuntu.local	Prod	telnet	1	0	2023-11-19 18:01:31.974
192.168.20.2	MARTYPC	Corp	192.168.50.42		Prod	tcp/443	0	0	2023-11-17 12:55:06.753
192.168.20.2	MARTYPC	Corp	192.168.50.36	DESKTOP-NH1C6NN.loca	Prod	other	14	0	2023-11-15 16:34:21.942

Query 3

Do I have traffic to the public internet?

Why would you use this query?

Identifying and mitigating unwanted public internet communications can be a big step in improving security. If misconfiguration occurs in the future, monitoring and reporting on this activity will help identify potential problems before they can be exploited.

When should you use this query?

You can use this query early in your security journey as you come to understand your network. It will also be valuable in the future to identify if assets attempt new communications to the public internet.

What can you expect this query to produce?

The output will be a table containing details of assets which are attempting to communicate with public IP addresses. This query can be modified to focus on any IP ranges outside your production environment.

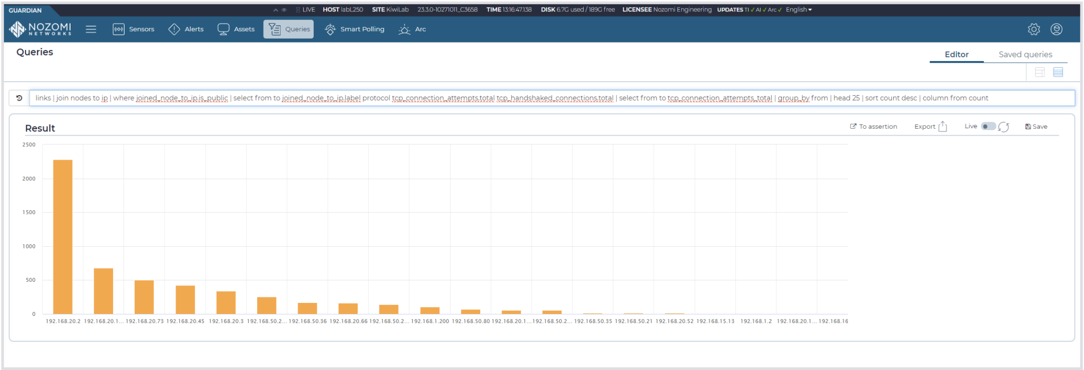
Where can this query be executed?

This query can be executed on a **Guardian** or **CMC**.

What is the query format?

```
links | join nodes to ip | where joined_node_to_ip.is_public | select from to joined_node_to_ip.label protocol tcp_connection_attempts.total tcp_handshaked_connections.total | select from to tcp_connection_attempts_total | group_by from | head 25 | sort count desc | column from count
```

What should the results look like?



```
links | join nodes to ip | where joined_node_to_ip.is_public | select from to joined_node_to_ip.label protocol tcp_connection_attempts.total tcp_handshaked_connections.total | select from to tcp_connection_attempts_total | group_by from | head 25 | sort count desc | column from count
```

Query 4

Do I have VNC and RDP traffic?

Why would you use this query?

Remote Access can be one of the largest risks to any environment. Monitoring and reporting on this activity will help identify potential problems before they can be exploited.

What can you expect this query to produce?

This query will output a column chart sorted by traffic throughput from highest to lowest.

When should you use this query?

This query is also useful early in your security journey as you come to understand your network. Further along the roadmap, it will also be valuable to identify remote access connections, both expected and unexpected. The query can easily be altered to work with almost any protocol.

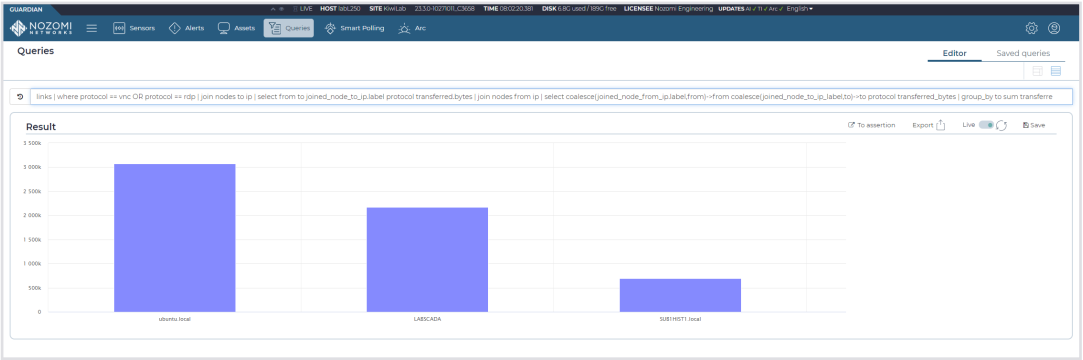
Where can this query be executed?

This query can be executed on a **Guardian** or **CMC**.

What is the query format?

```
links | where protocol == vnc OR protocol == rdp | join nodes to ip | select from to joined_node_to_ip.label protocol transferred.bytes | join nodes from ip | select coalesce(joined_node_from_ip.label,from)->from coalesce(joined_node_to_ip_label,to)->to protocol transferred_bytes | group_by to sum transferred_bytes | select to sum->dst_transferred_bytes | sort dst_transferred_bytes desc | head 25 | column to dst_transferred_bytes
```

What should the results look like?



```
links | where protocol == vnc OR protocol == rdp | join nodes to ip | select from to joined_node_to_ip.label protocol transferred.bytes | join nodes from ip | select coalesce(joined_node_from_ip.label,from)->from coalesce(joined_node_to_ip_label,to)->to protocol transferred_bytes | group_by to sum transferre
```

Query 5

How big is my traffic to the internet?

Why would you use this query?

If internet traffic is part of your operation, monitoring the volume of traffic may be a useful part of your risk management practice.

When should you use this query?

If you have an environment, or subnets and zones, where traffic to the internet may be expected (perhaps cloud-based IoT or IIoT devices) and this traffic is considered a manageable and acceptable risk.

What can you expect this query to produce?

This query will output a column chart highlighting links and sorted by traffic throughput from highest to lowest.

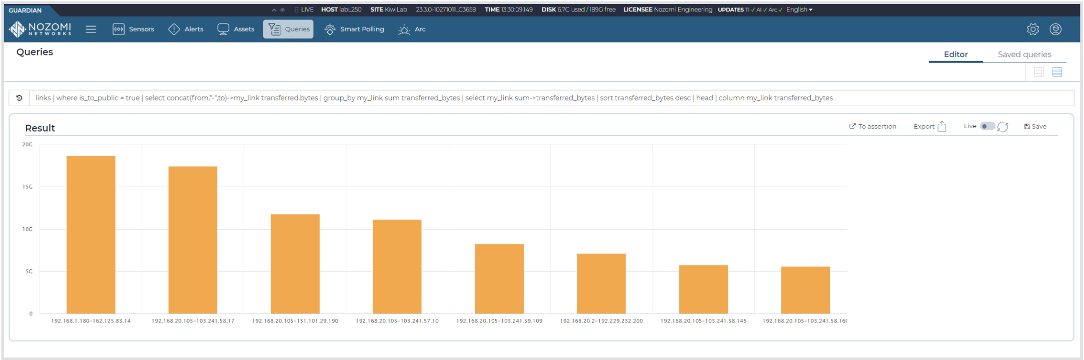
Where can this query be executed?

This query can be executed on a **Guardian** or **CMC**.

What is the query format?

```
links | where is_to_public = true | select concat(from,"-",to)->my_link transferred.bytes
| group_by my_link sum transferred_bytes | select my_link sum->transferred_bytes |
sort transferred_bytes desc | head | column my_link transferred_bytes
```

What should the results look like?



```
links | where is_to_public = true | select concat(from,"-",to)->my_link transferred.bytes | group_by my_link sum transferred_bytes | select my_link sum->transferred_bytes | sort transferred_bytes desc | head | column my_link transferred_bytes
```

Query 6

What CVEs are unresolved on this asset?

Why would you use this query?

This query can form a useful part in a Vulnerability Management program. You could alter this query to report on zones or subnets instead of single nodes.

When should you use this query?

This query will provide vulnerability data for assets. This data can be used to plan workloads ensuring that devices with the greatest risk are targeted at the most appropriate time.

What can you expect this query to produce?

This query will output a table of information sorted in descending order of risk. This query returns vulnerabilities that have been publicly exploited.

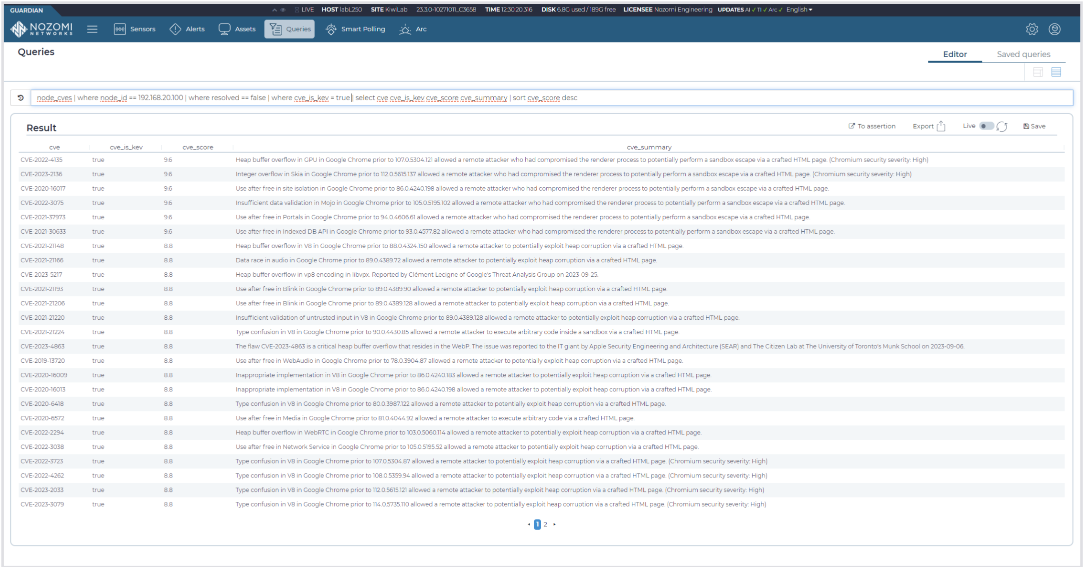
Where can this query be executed?

This query can be executed on a **Guardian** or **CMC**.

What is the query format?

```
node_cves | where node_id == 192.168.20.100 | where resolved == false | where cve_is_kev = true | select cve cve_is_kev cve_score cve_summary | sort cve_score desc
```

What should the results look like?


GUARDIAN
LIVE HOST labL250 SITE KiwiLab 23.3.0-10271011_C3658 TIME 12:30:20.316 DISK 6.8G used / 189G free LICENSEE Nozomi Engineering UPDATES AI ✓ TI ✓ Arc ✓ English
NOZOMI NETWORKS
Sensors
Alerts
Assets
Queries
Smart Polling
Arc
Queries
Editor
Saved queries
To assertion
Export
Live
Save
1 2

```
node_cves | where node_id == 192.168.20.100 | where resolved == false | where cve_is_kev = true| select cve cve_is_kev cve_score cve_summary | sort cve_score desc
```

Result

cve	cve_is_kev	cve_score	cve_summary
CVE-2022-4135	true	9.6	Heap buffer overflow in GPU in Google Chrome prior to 107.0.5304.121 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: High)
CVE-2023-2136	true	9.6	Integer overflow in Skia in Google Chrome prior to 112.0.5615.137 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page. (Chromium security severity: High)
CVE-2020-16017	true	9.6	Use after free in site isolation in Google Chrome prior to 86.0.4240.198 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page.
CVE-2022-3075	true	9.6	Insufficient data validation in Mojo in Google Chrome prior to 105.0.5195.102 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page.
CVE-2021-37973	true	9.6	Use after free in Portals in Google Chrome prior to 94.0.4606.61 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page.
CVE-2021-30633	true	9.6	Use after free in Indexed DB API in Google Chrome prior to 93.0.4577.82 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page.
CVE-2021-21148	true	8.8	Heap buffer overflow in V8 in Google Chrome prior to 88.0.4324.150 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page.
CVE-2021-21166	true	8.8	Data race in audio in Google Chrome prior to 89.0.4389.72 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page.
CVE-2023-5217	true	8.8	Heap buffer overflow in vp8 encoding in libvpx. Reported by Clément Lecigne of Google's Threat Analysis Group on 2023-09-25.
CVE-2021-21193	true	8.8	Use after free in Blink in Google Chrome prior to 89.0.4389.90 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page.
CVE-2021-21206	true	8.8	Use after free in Blink in Google Chrome prior to 89.0.4389.128 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page.
CVE-2021-21220	true	8.8	Insufficient validation of untrusted input in V8 in Google Chrome prior to 89.0.4389.128 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page.
CVE-2021-21224	true	8.8	Type confusion in V8 in Google Chrome prior to 90.0.4430.85 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page.
CVE-2023-4863	true	8.8	The flaw CVE-2023-4863 is a critical heap buffer overflow that resides in the WebP. The issue was reported to the IT giant by Apple Security Engineering and Architecture (SEAR) and The Citizen Lab at The University of Toronto's Munk School on 2023-09-06.
CVE-2019-13720	true	8.8	Use after free in WebAudio in Google Chrome prior to 78.0.3904.87 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page.
CVE-2020-16009	true	8.8	Inappropriate implementation in V8 in Google Chrome prior to 86.0.4240.183 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page.
CVE-2020-16013	true	8.8	Inappropriate implementation in V8 in Google Chrome prior to 86.0.4240.198 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page.
CVE-2020-6418	true	8.8	Type confusion in V8 in Google Chrome prior to 80.0.3987.122 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page.
CVE-2020-6572	true	8.8	Use after free in Media in Google Chrome prior to 81.0.4044.92 allowed a remote attacker to execute arbitrary code via a crafted HTML page.
CVE-2022-2294	true	8.8	Heap buffer overflow in WebRTC in Google Chrome prior to 103.0.5060.114 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page.
CVE-2022-3038	true	8.8	Use after free in Network Service in Google Chrome prior to 105.0.5195.52 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page.
CVE-2022-3723	true	8.8	Type confusion in V8 in Google Chrome prior to 107.0.5304.87 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page. (Chromium security severity: High)
CVE-2022-4262	true	8.8	Type confusion in V8 in Google Chrome prior to 108.0.5359.94 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page. (Chromium security severity: High)
CVE-2023-2033	true	8.8	Type confusion in V8 in Google Chrome prior to 112.0.5615.121 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page. (Chromium security severity: High)
CVE-2023-3079	true	8.8	Type confusion in V8 in Google Chrome prior to 114.0.5735.110 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page. (Chromium security severity: High)

Query 7

What are the most used protocols in my network?

Why would you use this query?

If internet traffic is part of your operation, monitoring the volume of traffic may be a useful part of your risk management practice.

What can you expect this query to produce?

This query will output a column chart highlighting links and sorted by traffic throughput from highest to lowest.

When should you use this query?

If you have an environment, or subnets and zones, where traffic to the internet may be expected (perhaps cloud-based IoT or IIoT devices) and this traffic is considered a manageable and acceptable risk.

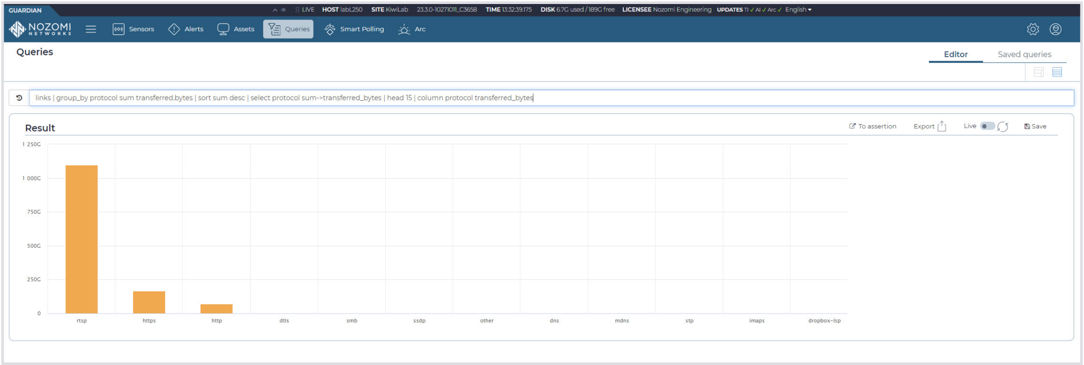
Where can this query be executed?

This query can be executed on a **Guardian** or **CMC**.

What is the query format?

```
links | group_by protocol sum transferred.bytes | sort sum desc | select protocol sum->transferred_bytes | head 15 | column protocol transferred_bytes
```

What should the results look like?



GUARDIAN LIVE **HOST** labL250 **SITE** KiwiLab 23.3.0-10271011_C3658 **TIME** 13:32:39.175 **DISK** 6.7G used / 189G free **LICENSEE** Nozomi Engineering **UPDATES** TI ✓ AI ✓ Arc ✓ English

Queries

```
links | group_by protocol sum transferred.bytes | sort sum desc | select protocol sum->transferred_bytes | head 15 | column protocol transferred_bytes
```

Result

Query 8

How are my vulnerabilities trending?

Why would you use this query?

This query makes use of the ‘bucket’ and ‘history’ features to display a line graph measuring quantity over time. This can be adapted to measure other metrics in a similar manner.

When should you use this query?

Monitoring trends over time is an integral part of good reporting. Vulnerability trending can be one metric to help track the overall performance of your security program when reporting to senior levels.

What can you expect this query to produce?

The line graph output from this query is very simple to understand and can be adapted to suit reporting periods by altering the time “buckets” to give hours, days, weeks and so on. In this case, if the line is trending up – more vulnerabilities exist. If it trends down, vulnerabilities are being mitigated or managed. Please note, this query will return an empty result when vulnerability assessment is performed on an upstream CMC or Vantage.

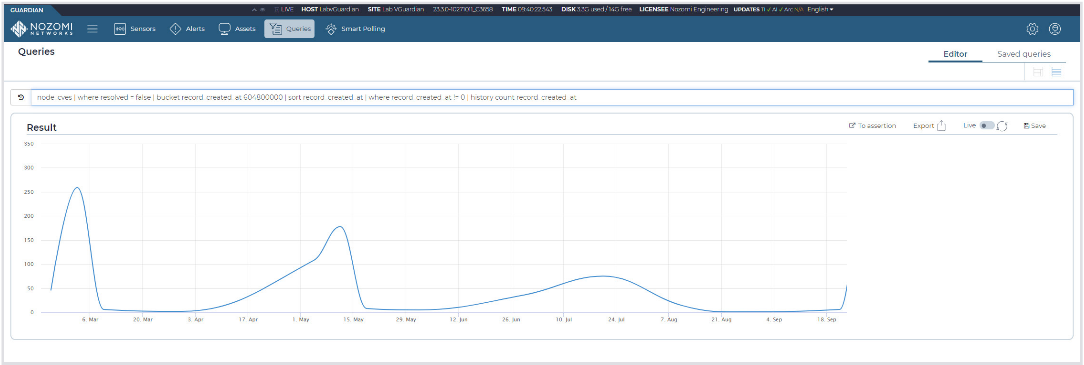
Where can this query be executed?

This query can be executed on a **Guardian** or **CMC**.

What is the query format?

```
node_cves | where resolved = false | bucket record_created_at 604800000 | sort
record_created_at | where record_created_at != 0 | history count record_created_at
```

What should the results look like?



GUARDIAN LIVE **HOST** LabvGuardian **SITE** Lab VGuardian 23.3.0-10271011_C3658 **TIME** 09:40:22.543 **DISK** 3.3G used / 14G free **LICENSEE** Nozomi Engineering **UPDATES** TI ✓ AI ✓ Arc N/A English

```
node_cves | where resolved = false | bucket record_created_at 604800000 | sort record_created_at | where record_created_at != 0 | history count record_created_at
```

Query 9

What are my greatest MITRE ATT&CK tactic risks?

Why would you use this query?

This query would be useful in an environment where MITRE tactics are used in risk assessment, or as a common language across IT and OT teams.

When should you use this query?

Assessing detected MITRE ATT&CK tactics may help you recognize patterns of weakness in areas of your network, and/or help you develop workstreams that return larger risk reductions or ROI for effort expended.

What can you expect this query to produce?

This query will output a column graph showing the number of assets which may be susceptible to a given attack tactic. This information will show where common improvements will affect the largest change.

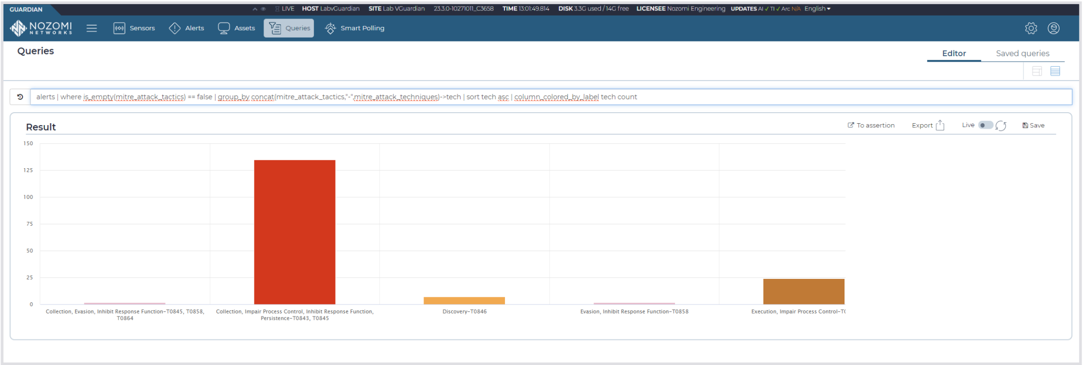
Where can this query be executed?

This query can be executed on a **Guardian** or **CMC**.

What is the query format?

```
alerts | where is_empty(mitre_attack_tactics) == false | group_by concat(mitre_attack_tactics,"-",mitre_attack_techniques)->tech | sort tech asc | column_colored_by_label tech count
```

What should the results look like?


GUARDIAN
LIVE HOST LabvGuardian SITE Lab VGuardian 23.3.0-10271011_C3658 TIME 13:01:49.814 DISK 3.3G used / 14G free LICENSEE Nozomi Engineering UPDATES AI ✓ TI ✓ Arc N/A English
NOZOMI NETWORKS
Sensors
Alerts
Assets
Queries
Smart Polling
Queries
Editor
Saved queries
Result
To assertion
Export
Live
Save
150
125
100
75
50
25
0
Collection, Evasion, Inhibit Response Function–T0845, T0858, T0864
Collection, Impair Process Control, Inhibit Response Function, Persistence–T0843, T0845
Discovery–T0846
Evasion, Inhibit Response Function–T0858
Execution, Impair Process Control–T(

```
alerts | where is_empty(mitre_attack_tactics) == false | group_by concat(mitre_attack_tactics,"-",mitre_attack_techniques)->tech | sort tech asc | column_colored_by_label tech count
```

Query 10

Is this protocol/function used in my network?

Why would you use this query?

This query may return this information faster than other tools such as Wireshark. Identifying this kind of traffic may not be possible using proprietary OT tools, so this becomes a way to perform deeper investigations into traffic.

What can you expect this query to produce?

This query will output a table showing the link where the protocol/function was detected and the last time it was detected.

When should you use this query?

This query can be useful for quickly and easily identifying if certain protocol functions are in use in production. This may even extend to alerting if certain function codes are detected in traffic – an added layer of protection. You can refine the query by specifying only the protocol you wish to investigate.

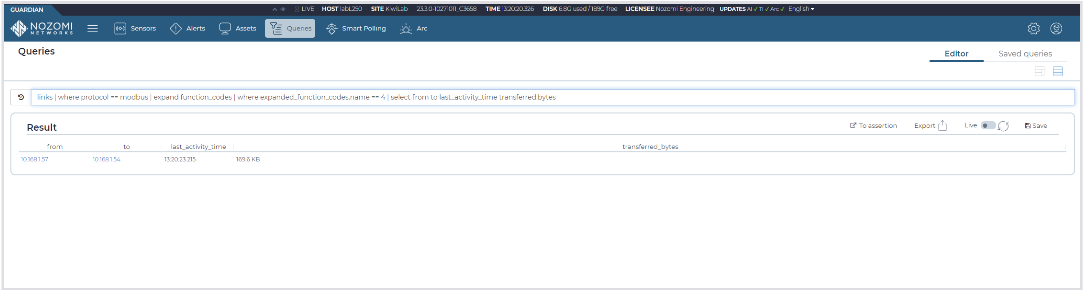
Where can this query be executed?

This query can be executed on a **Guardian** or **CMC**.

What is the query format?

```
links | expand function_codes | select from to expanded_function_codes.name last_activity_time transferred.bytes
```

What should the results look like?



Queries

```
links | where protocol == modbus | expand function_codes | where expanded_function_codes.name == 4 | select from to last_activity_time transferred.bytes
```

Result

from	to	last_activity_time	transferred_bytes
10.168.1.57	10.168.1.54	13:20:23.215	169.6 KB

Query 11

What alerts happened on this day?

Why would you use this query?

This query can be customized down to the millisecond level using the epoch timestamps. With minimal alteration, the query could narrow in on specific alert types, network zones or subnets.

When should you use this query?

You may find this query useful during general security operations, forensic activities or threat hunting.

What can you expect this query to produce?

The table this query outputs can be configured to deliver a lot of different information depending on your needs.

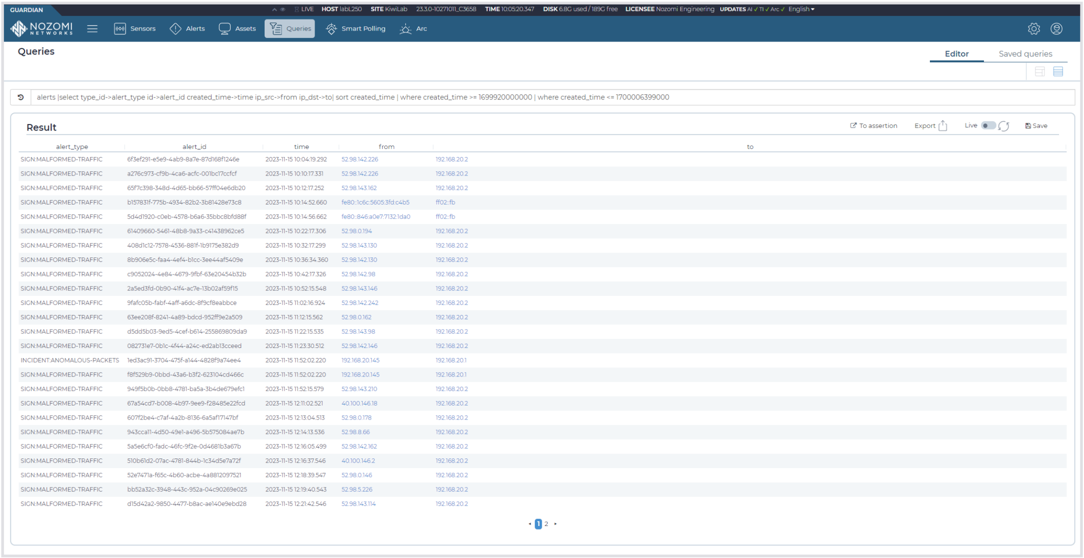
Where can this query be executed?

This query can be executed on a **Guardian** or **CMC**. With minimal alteration, it can be used on **Vantage**.

What is the query format?

```
alerts |select type_id->alert_type id->alert_id created_time->time ip_src->source ip_dst->destination | sort created_time | where created_time >= 1699920000000 | where created_time <= 1700006399000
```

What should the results look like?


GUARDIAN
LIVE HOST labL250 SITE KiwiLab 23.3.0-10271011_C3658 TIME 10:05:20.347 DISK 6.8G used / 189G free LICENSEE Nozomi Engineering UPDATES AI ✓ TI ✓ Arc ✓ English
NOZOMI NETWORKS
Sensors
Alerts
Assets
Queries
Smart Polling
Arc
Editor
Saved queries
To assertion
Export
Live
Save
1 2

Queries

```
alerts |select type_id->alert_type id->alert_id created_time->time ip_src->from ip_dst->to| sort created_time | where created_time >= 1699920000000 | where created_time <= 1700006399000
```

Result

alert_type	alert_id	time	from	to
SIGN:MALFORMED-TRAFFIC	6f3ef291-e5e9-4ab9-8a7e-87d168f1246e	2023-11-15 10:04:19.292	52.98.142.226	192.168.20.2
SIGN:MALFORMED-TRAFFIC	a276c973-cf9b-4ca6-acfc-001bc17ccfcf	2023-11-15 10:10:17.331	52.98.142.226	192.168.20.2
SIGN:MALFORMED-TRAFFIC	65f7c398-348d-4d65-bb66-57ff04e6db20	2023-11-15 10:12:17.252	52.98.143.162	192.168.20.2
SIGN:MALFORMED-TRAFFIC	b157831f-775b-4934-82b2-3b81428e73c8	2023-11-15 10:14:52.660	fe80::1c6c:5605:3fd:c4b5	ff02::fb
SIGN:MALFORMED-TRAFFIC	5d4d1920-c0eb-4578-b6a6-35bbc8bfd88f	2023-11-15 10:14:56.662	fe80::846:a0e7:7132:1da0	ff02::fb
SIGN:MALFORMED-TRAFFIC	61409660-5461-48b8-9a33-c41438962ce5	2023-11-15 10:22:17.306	52.98.0.194	192.168.20.2
SIGN:MALFORMED-TRAFFIC	408d1c12-7578-4536-881f-1b9175e382d9	2023-11-15 10:32:17.299	52.98.143.130	192.168.20.2
SIGN:MALFORMED-TRAFFIC	8b906e5c-faa4-4ef4-b1cc-3ee44af5409e	2023-11-15 10:36:34.360	52.98.142.130	192.168.20.2
SIGN:MALFORMED-TRAFFIC	c9052024-4e84-4679-9fbf-63e20454b32b	2023-11-15 10:42:17.326	52.98.142.98	192.168.20.2
SIGN:MALFORMED-TRAFFIC	2a5ed3fd-0b90-41f4-ac7e-13b02af59f15	2023-11-15 10:52:15.548	52.98.143.146	192.168.20.2
SIGN:MALFORMED-TRAFFIC	9fafc05b-fabf-4aff-a6dc-8f9cf8eabbce	2023-11-15 11:02:16.924	52.98.142.242	192.168.20.2
SIGN:MALFORMED-TRAFFIC	63ee208f-8241-4a89-bdcd-952ff9e2a509	2023-11-15 11:12:15.562	52.98.0.162	192.168.20.2
SIGN:MALFORMED-TRAFFIC	d5dd5b03-9ed5-4cef-b614-255869809da9	2023-11-15 11:22:15.535	52.98.143.98	192.168.20.2
SIGN:MALFORMED-TRAFFIC	082731e7-0b1c-4f44-a24c-ed2ab13cceed	2023-11-15 11:23:30.512	52.98.142.146	192.168.20.2
INCIDENT:ANOMALOUS-PACKETS	1ed3ac91-3704-475f-a144-4828f9a74ee4	2023-11-15 11:52:02.220	192.168.20.145	192.168.20.1
SIGN:MALFORMED-TRAFFIC	f8f529b9-0bbd-43a6-b3f2-623104cd466c	2023-11-15 11:52:02.220	192.168.20.145	192.168.20.1
SIGN:MALFORMED-TRAFFIC	949f5b0b-0bb8-4781-ba5a-3b4de679efc1	2023-11-15 11:52:15.579	52.98.143.210	192.168.20.2
SIGN:MALFORMED-TRAFFIC	67a54cd7-b008-4b97-9ee9-f28485e22fcd	2023-11-15 12:11:02.521	40.100.146.18	192.168.20.2
SIGN:MALFORMED-TRAFFIC	607f2be4-c7af-4a2b-8136-6a5af17147bf	2023-11-15 12:13:04.513	52.98.0.178	192.168.20.2
SIGN:MALFORMED-TRAFFIC	943cca11-4d50-49e1-a496-5b575084ae7b	2023-11-15 12:14:13.536	52.98.8.66	192.168.20.2
SIGN:MALFORMED-TRAFFIC	5a5e6cf0-fadc-46fc-9f2e-0d4681b3a67b	2023-11-15 12:16:05.499	52.98.142.162	192.168.20.2
SIGN:MALFORMED-TRAFFIC	510b61d2-07ac-4781-844b-1c34d5e7a72f	2023-11-15 12:16:37.546	40.100.146.2	192.168.20.2
SIGN:MALFORMED-TRAFFIC	52e7471a-f65c-4b60-acbe-4a8812097521	2023-11-15 12:18:39.547	52.98.0.146	192.168.20.2
SIGN:MALFORMED-TRAFFIC	bb52a32c-3948-443c-952a-04c90269e025	2023-11-15 12:19:40.543	52.98.5.226	192.168.20.2
SIGN:MALFORMED-TRAFFIC	d15d42a2-9850-4477-b8ac-ae140e9ebd28	2023-11-15 12:21:42.546	52.98.143.114	192.168.20.2

Query 12

What assets appeared in the last 7 days?

Why would you use this query?

While not a particularly complicated query, the output of this query would make a good discussion point during weekly team meetings and can provide a way to open discussion between Security and OT teams.

When should you use this query?

This query should form part of your regular monitoring and reporting. Regular alerting will highlight new assets on your network, so this query becomes a summary of that activity.

What can you expect this query to produce?

The table and information produced by this query are a good starting point to customize to the needs of your environment and how you intend to implement the information returned. Start with this, and expand to suit your needs.

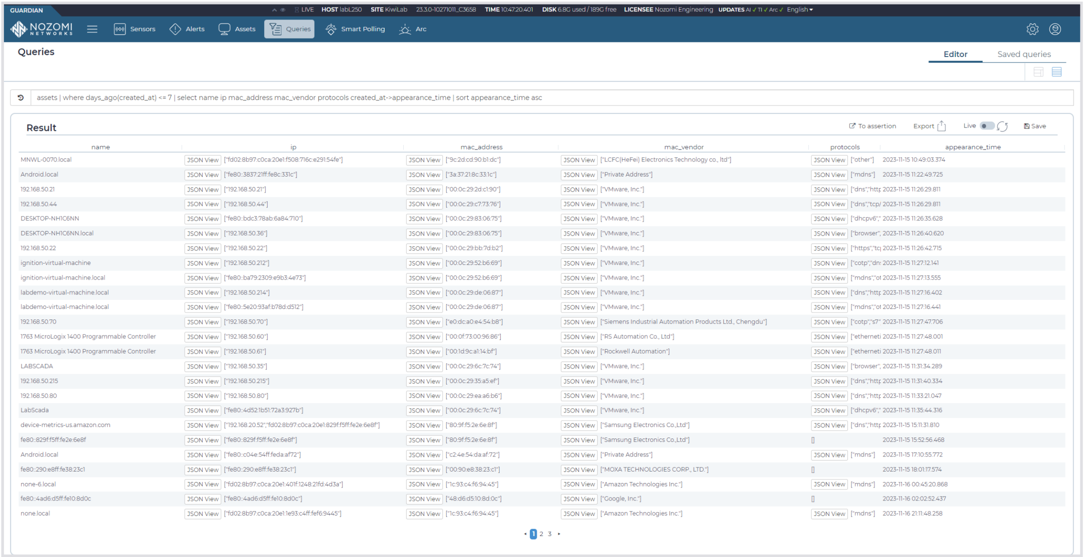
Where can this query be executed?

This query can be executed on a **Guardian** or **CMC**.

What is the query format?

```
assets | where days_ago(created_at) <= 7 | select name ip mac_address mac_vendor
protocols created_at->appearance_time | sort appearance_time asc
```

What should the results look like?



```
assets | where days_ago(created_at) <= 7 | select name ip mac_address mac_vendor protocols created_at->appearance_time | sort appearance_time asc
```

Result

name	ip	mac_address	mac_vendor	protocols	appearance_time
MNWL-0070.local	JSON View ["fd02:8b97:c0ca:20e1:f508:716c:e291:54fe"]	JSON View ["9c:2d:cd:90:b1:dc"]	JSON View ["LCFC(HeFei) Electronics Technology co., ltd"]	JSON View ["other"]	2023-11-15 10:49:03.374
Android.local	JSON View ["fe80::3837:21ff:fe8c:331c"]	JSON View ["3a:37:21:8c:33:1c"]	JSON View ["Private Address"]	JSON View ["mdns"]	2023-11-15 11:22:49.725
192.168.50.21	JSON View ["192.168.50.21"]	JSON View ["00:0c:29:2d:c1:90"]	JSON View ["VMware, Inc."]	JSON View ["dns","http	2023-11-15 11:26:29.811
192.168.50.44	JSON View ["192.168.50.44"]	JSON View ["00:0c:29:c7:73:76"]	JSON View ["VMware, Inc."]	JSON View ["dns","tcp/	2023-11-15 11:26:29.811
DESKTOP-NH1C6NN	JSON View ["fe80::bdc3:78ab:6a84:710"]	JSON View ["00:0c:29:83:06:75"]	JSON View ["VMware, Inc."]	JSON View ["dhcpv6","	2023-11-15 11:26:35.628
DESKTOP-NH1C6NN.local	JSON View ["192.168.50.36"]	JSON View ["00:0c:29:83:06:75"]	JSON View ["VMware, Inc."]	JSON View ["browser",	2023-11-15 11:26:40.620
192.168.50.22	JSON View ["192.168.50.22"]	JSON View ["00:0c:29:bb:7d:b2"]	JSON View ["VMware, Inc."]	JSON View ["https","tc	2023-11-15 11:26:42.715
ignition-virtual-machine	JSON View ["192.168.50.212"]	JSON View ["00:0c:29:52:b6:69"]	JSON View ["VMware, Inc."]	JSON View ["cotp","dn:	2023-11-15 11:27:12.141
ignition-virtual-machine.local	JSON View ["fe80::ba79:2309:e9b3:4e73"]	JSON View ["00:0c:29:52:b6:69"]	JSON View ["VMware, Inc."]	JSON View ["mdns","ot	2023-11-15 11:27:13.555
labdemo-virtual-machine.local	JSON View ["192.168.50.214"]	JSON View ["00:0c:29:de:06:87"]	JSON View ["VMware, Inc."]	JSON View ["dns","http	2023-11-15 11:27:16.402
labdemo-virtual-machine.local	JSON View ["fe80::5e20:93af:b78d:d512"]	JSON View ["00:0c:29:de:06:87"]	JSON View ["VMware, Inc."]	JSON View ["mdns","ot	2023-11-15 11:27:16.441
192.168.50.70	JSON View ["192.168.50.70"]	JSON View ["e0:dc:a0:e4:54:b8"]	JSON View ["Siemens Industrial Automation Products Ltd., Chengdu"]	JSON View ["cotp","s7",	2023-11-15 11:27:47.706
1763 MicroLogix 1400 Programmable Controller	JSON View ["192.168.50.60"]	JSON View ["00:0f:73:00:96:86"]	JSON View ["RS Automation Co., Ltd"]	JSON View ["etherneti	2023-11-15 11:27:48.001
1763 MicroLogix 1400 Programmable Controller	JSON View ["192.168.50.61"]	JSON View ["00:1d:9c:a1:14:bf"]	JSON View ["Rockwell Automation"]	JSON View ["etherneti	2023-11-15 11:27:48.011
LABSCADA	JSON View ["192.168.50.35"]	JSON View ["00:0c:29:6c:7c:74"]	JSON View ["VMware, Inc."]	JSON View ["browser",	2023-11-15 11:31:34.289
192.168.50.215	JSON View ["192.168.50.215"]	JSON View ["00:0c:29:35:a5:ef"]	JSON View ["VMware, Inc."]	JSON View ["dns","http	2023-11-15 11:31:40.334
192.168.50.80	JSON View ["192.168.50.80"]	JSON View ["00:0c:29:ea:a6:b6"]	JSON View ["VMware, Inc."]	JSON View ["dns","http	2023-11-15 11:33:21.047
LabScada	JSON View ["fe80::4d52:1b51:72a3:927b"]	JSON View ["00:0c:29:6c:7c:74"]	JSON View ["VMware, Inc."]	JSON View ["dhcpv6","	2023-11-15 11:35:44.316
device-metrics-us.amazon.com	JSON View ["192.168.20.52","fd02:8b97:c0ca:20e1:829f:f5ff:fe2e:6e8f"]	JSON View ["80:9f:f5:2e:6e:8f"]	JSON View ["Samsung Electronics Co.,Ltd"]	JSON View ["dns","http	2023-11-15 15:11:31.810
fe80::829f:f5ff:fe2e:6e8f	JSON View ["fe80::829f:f5ff:fe2e:6e8f"]	JSON View ["80:9f:f5:2e:6e:8f"]	JSON View ["Samsung Electronics Co.,Ltd"]	[]	2023-11-15 15:52:56.468
Android.local	JSON View ["fe80::c04e:54ff:feda:af72"]	JSON View ["c2:4e:54:da:af:72"]	JSON View ["Private Address"]	JSON View ["mdns"]	2023-11-15 17:10:55.772
fe80::290:e8ff:fe38:23c1	JSON View ["fe80::290:e8ff:fe38:23c1"]	JSON View ["00:90:e8:38:23:c1"]	JSON View ["MOXA TECHNOLOGIES CORP., LTD."]	[]	2023-11-15 18:01:17.574
none-6.local	JSON View ["fd02:8b97:c0ca:20e1:401f:1248:21fd:4d3a"]	JSON View ["1c:93:c4:f6:94:45"]	JSON View ["Amazon Technologies Inc."]	JSON View ["mdns"]	2023-11-16 00:45:20.868
fe80::4ad6:d5ff:fe10:8d0c	JSON View ["fe80::4ad6:d5ff:fe10:8d0c"]	JSON View ["48:d6:d5:10:8d:0c"]	JSON View ["Google, Inc."]	[]	2023-11-16 02:02:52.437
none.local	JSON View ["fd02:8b97:c0ca:20e1:1e93:c4ff:fef6:9445"]	JSON View ["1c:93:c4:f6:94:45"]	JSON View ["Amazon Technologies Inc."]	JSON View ["mdns"]	2023-11-16 21:11:48.258

◂ 1 2 3 ▸

Query 13

Do I have open CVEs more than a month old?

Why would you use this query?

This query differs from the earlier query “What CVEs are unresolved on this asset?” by not focusing on a single asset. This query can be modified to use the ‘is_kev’ flag to extract only vulnerabilities that have been exploited in public.

When should you use this query?

As your program matures and you get on top of vulnerability management, this query can help ensure you stay ahead. At other times, adjusting the period of reporting can also help.

What can you expect this query to produce?

The table produced by this query is sorted by CVE score. After that, you should pay most attention to the ‘record_created_at’ field and use this information to plan remediation efforts.

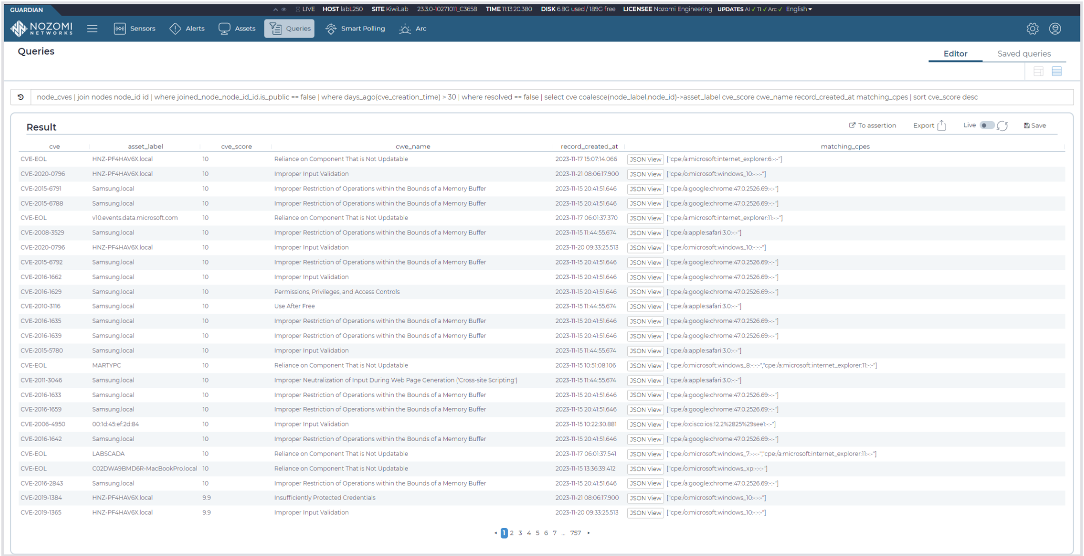
Where can this query be executed?

This query can be executed on a **Guardian** or **CMC**.

What is the query format?

```
node_cves | join nodes node_id id | where joined_node_node_id_id.is_public == false | where days_ago(cve_creation_time) > 30 | where resolved == false | select cve coalesce(node_label,node_id)->asset_label cve_score cwe_name record_created_at matching_cpes | sort cve_score desc
```

What should the results look like?


GUARDIAN
LIVE HOST labL250 SITE KiwiLab 23.3.0-10271011_C3658 TIME 11:13:20.380 DISK 6.8G used / 189G free LICENSEE Nozomi Engineering UPDATES AI ✓ TI ✓ Arc ✓ English
NOZOMI NETWORKS
Sensors
Alerts
Assets
Queries
Smart Polling
Arc
Queries
Editor
Saved queries
Result
To assertion
Export
Live
Save

```
node_cves | join nodes node_id id | where joined_node_node_id_id.is_public == false | where days_ago(cve_creation_time) > 30 | where resolved == false | select cve coalesce(node_label,node_id)->asset_label cve_score cwe_name record_created_at matching_cpes | sort cve_score desc
```

cve	asset_label	cve_score	cwe_name	record_created_at	matching_cpes
CVE-EOL	HNZ-PF4HAV6X.local	10	Reliance on Component That is Not Updatable	2023-11-17 15:07:14.066	JSON View ["cpe:/a:microsoft:internet_explorer:6:-:-"]
CVE-2020-0796	HNZ-PF4HAV6X.local	10	Improper Input Validation	2023-11-21 08:06:17.900	JSON View ["cpe:/o:microsoft:windows_10:-:-:-"]
CVE-2015-6791	Samsung.local	10	Improper Restriction of Operations within the Bounds of a Memory Buffer	2023-11-15 20:41:51.646	JSON View ["cpe:/a:google:chrome:47.0.2526.69:-:-"]
CVE-2015-6788	Samsung.local	10	Improper Restriction of Operations within the Bounds of a Memory Buffer	2023-11-15 20:41:51.646	JSON View ["cpe:/a:google:chrome:47.0.2526.69:-:-"]
CVE-EOL	v10.events.data.microsoft.com	10	Reliance on Component That is Not Updatable	2023-11-17 06:01:37.370	JSON View ["cpe:/a:microsoft:internet_explorer:11:-:-"]
CVE-2008-3529	Samsung.local	10	Improper Restriction of Operations within the Bounds of a Memory Buffer	2023-11-15 11:44:55.674	JSON View ["cpe:/a:apple:safari:3.0:-:-"]
CVE-2020-0796	HNZ-PF4HAV6X.local	10	Improper Input Validation	2023-11-20 09:33:25.513	JSON View ["cpe:/o:microsoft:windows_10:-:-:-"]
CVE-2015-6792	Samsung.local	10	Improper Restriction of Operations within the Bounds of a Memory Buffer	2023-11-15 20:41:51.646	JSON View ["cpe:/a:google:chrome:47.0.2526.69:-:-"]
CVE-2016-1662	Samsung.local	10	Improper Input Validation	2023-11-15 20:41:51.646	JSON View ["cpe:/a:google:chrome:47.0.2526.69:-:-"]
CVE-2016-1629	Samsung.local	10	Permissions, Privileges, and Access Controls	2023-11-15 20:41:51.646	JSON View ["cpe:/a:google:chrome:47.0.2526.69:-:-"]
CVE-2010-3116	Samsung.local	10	Use After Free	2023-11-15 11:44:55.674	JSON View ["cpe:/a:apple:safari:3.0:-:-"]
CVE-2016-1635	Samsung.local	10	Improper Restriction of Operations within the Bounds of a Memory Buffer	2023-11-15 20:41:51.646	JSON View ["cpe:/a:google:chrome:47.0.2526.69:-:-"]
CVE-2016-1639	Samsung.local	10	Improper Restriction of Operations within the Bounds of a Memory Buffer	2023-11-15 20:41:51.646	JSON View ["cpe:/a:google:chrome:47.0.2526.69:-:-"]
CVE-2015-5780	Samsung.local	10	Improper Input Validation	2023-11-15 11:44:55.674	JSON View ["cpe:/a:apple:safari:3.0:-:-"]
CVE-EOL	MARTYPC	10	Reliance on Component That is Not Updatable	2023-11-15 10:51:08.106	JSON View ["cpe:/o:microsoft:windows_8:-:-:-","cpe:/a:microsoft:internet_explorer:11:-:-"]
CVE-2011-3046	Samsung.local	10	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	2023-11-15 11:44:55.674	JSON View ["cpe:/a:apple:safari:3.0:-:-"]
CVE-2016-1633	Samsung.local	10	Improper Restriction of Operations within the Bounds of a Memory Buffer	2023-11-15 20:41:51.646	JSON View ["cpe:/a:google:chrome:47.0.2526.69:-:-"]
CVE-2016-1659	Samsung.local	10	Improper Restriction of Operations within the Bounds of a Memory Buffer	2023-11-15 20:41:51.646	JSON View ["cpe:/a:google:chrome:47.0.2526.69:-:-"]
CVE-2006-4950	00:1d:45:ef:2d:84	10	Improper Input Validation	2023-11-15 10:22:30.881	JSON View ["cpe:/o:cisco:ios:12.2%2825%29see1:-:-"]
CVE-2016-1642	Samsung.local	10	Improper Restriction of Operations within the Bounds of a Memory Buffer	2023-11-15 20:41:51.646	JSON View ["cpe:/a:google:chrome:47.0.2526.69:-:-"]
CVE-EOL	LABSCADA	10	Reliance on Component That is Not Updatable	2023-11-17 06:01:37.541	JSON View ["cpe:/o:microsoft:windows_7:-:-:-","cpe:/a:microsoft:internet_explorer:11:-:-"]
CVE-EOL	C02DWA9BMD6R-MacBookPro.local	10	Reliance on Component That is Not Updatable	2023-11-15 13:36:39.412	JSON View ["cpe:/o:microsoft:windows_xp:-:-:-"]
CVE-2016-2843	Samsung.local	10	Improper Restriction of Operations within the Bounds of a Memory Buffer	2023-11-15 20:41:51.646	JSON View ["cpe:/a:google:chrome:47.0.2526.69:-:-"]
CVE-2019-1384	HNZ-PF4HAV6X.local	9.9	Insufficiently Protected Credentials	2023-11-21 08:06:17.900	JSON View ["cpe:/o:microsoft:windows_10:-:-:-"]
CVE-2019-1365	HNZ-PF4HAV6X.local	9.9	Improper Input Validation	2023-11-20 09:33:25.513	JSON View ["cpe:/o:microsoft:windows_10:-:-:-"]

◂ 1 2 3 4 5 6 7 … 757 ▸

Query 14

What traffic is being blocked by my firewall?

Why would you use this query?

This query extracts link attempts which did not successfully complete TCP handshaking. From this you could make the assumption that the handshake was blocked by a firewall. When combined with a firewall integration on a Guardian sensor, this could be a useful test.

When should you use this query?

This query may be helpful when testing, or confirming operation of, firewall rules.

What can you expect this query to produce?

This query will output a table of link connection attempts which did not complete handshaking. This query may also capture events which are not the result of a firewall block. This information in itself may be useful in optimizing network performance.

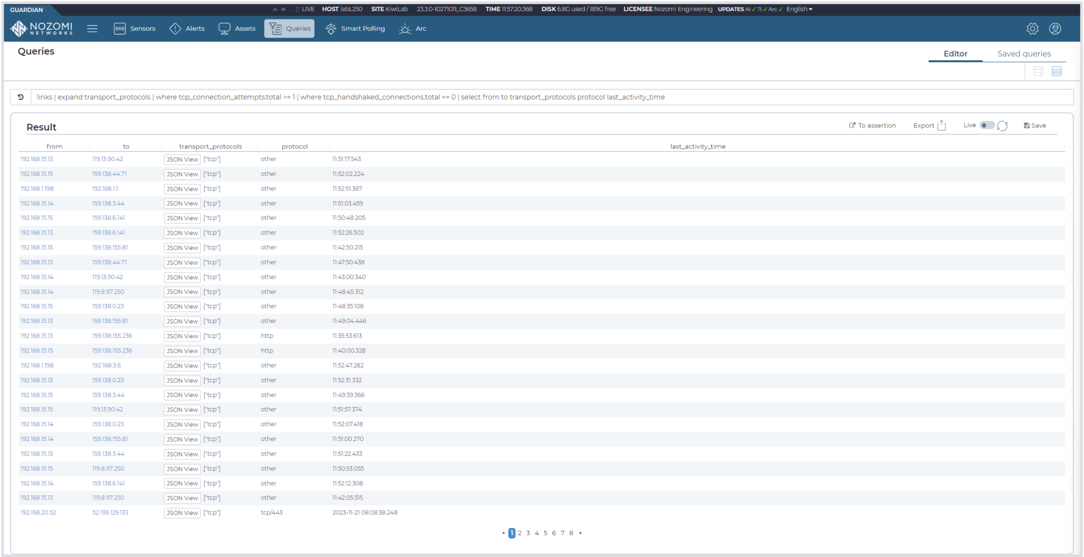
Where can this query be executed?

This query can be executed on a **Guardian** or **CMC**.

What is the query format?

```
links | expand transport_protocols | where tcp_connection_attempts.
total >= 1 | where tcp_handshaked_connections.total == 0 | select from to
```

What should the results look like?


GUARDIAN
LIVE HOST labL250 SITE KiwiLab 23.3.0-10271011_C3658 TIME 11:57:20.368 DISK 6.8G used / 189G free LICENSEE Nozomi Engineering UPDATES AI ✓ TI ✓ Arc ✓ English
NOZOMI NETWORKS
Sensors
Alerts
Assets
Queries
Smart Polling
Arc
Editor
Saved queries
To assertion
Export
Live
Save
‹ 1 2 3 4 5 6 7 8 ›

Queries

```
links | expand transport_protocols | where tcp_connection_attempts.total >= 1 | where tcp_handshaked_connections.total == 0 | select from to transport_protocols protocol last_activity_time
```

Result

from	to	transport_protocols	protocol	last_activity_time
192.168.15.13	119.13.90.42	JSON View ["tcp"]	other	11:51:17.543
192.168.15.15	159.138.44.71	JSON View ["tcp"]	other	11:52:02.224
192.168.1.198	192.168.1.1	JSON View ["tcp"]	other	11:52:51.367
192.168.15.14	159.138.3.44	JSON View ["tcp"]	other	11:51:03.459
192.168.15.15	159.138.6.141	JSON View ["tcp"]	other	11:50:48.205
192.168.15.13	159.138.6.141	JSON View ["tcp"]	other	11:52:26.502
192.168.15.15	159.138.155.81	JSON View ["tcp"]	other	11:42:50.215
192.168.15.13	159.138.44.71	JSON View ["tcp"]	other	11:47:50.438
192.168.15.14	119.13.90.42	JSON View ["tcp"]	other	11:43:00.340
192.168.15.14	119.8.97.250	JSON View ["tcp"]	other	11:48:45.312
192.168.15.15	159.138.0.23	JSON View ["tcp"]	other	11:48:35.108
192.168.15.13	159.138.155.81	JSON View ["tcp"]	other	11:49:04.446
192.168.15.13	159.138.155.236	JSON View ["tcp"]	http	11:35:53.613
192.168.15.15	159.138.155.236	JSON View ["tcp"]	http	11:40:00.328
192.168.1.198	192.168.3.6	JSON View ["tcp"]	other	11:52:47.282
192.168.15.13	159.138.0.23	JSON View ["tcp"]	other	11:52:31.332
192.168.15.15	159.138.3.44	JSON View ["tcp"]	other	11:49:39.366
192.168.15.15	119.13.90.42	JSON View ["tcp"]	other	11:51:57.374
192.168.15.14	159.138.0.23	JSON View ["tcp"]	other	11:52:07.418
192.168.15.14	159.138.155.81	JSON View ["tcp"]	other	11:51:00.270
192.168.15.13	159.138.3.44	JSON View ["tcp"]	other	11:51:22.433
192.168.15.15	119.8.97.250	JSON View ["tcp"]	other	11:50:53.055
192.168.15.14	159.138.6.141	JSON View ["tcp"]	other	11:52:12.308
192.168.15.13	119.8.97.250	JSON View ["tcp"]	other	11:42:05.515
192.168.20.52	52.193.129.133	JSON View ["tcp"]	tcp/443	2023-11-21 08:08:38.248

Query 15

What traffic is crossing Purdue levels?

Why would you use this query?

This query could form part of a regular review of traffic looking for an increase or decrease in allowed traffic between levels. This query could easily be modified to assert and alert on new links.

When should you use this query?

While the Purdue model itself is not the panacea to OT/ICS security issues, accepting that traffic should only cross the production levels laid out in the Purdue model at the correct places and times is a sensible approach to quick security wins. This query highlights active links and allows you to triage those and mitigate unnecessary risks.

What can you expect this query to produce?

This query outputs a table of links showing the origin and destination layers with respect to the Purdue model. Attention should be paid to those links which do not have layers assigned, and to those where the distance between layers is greater than 1. You will need to have assigned, or confirmed the detected, Purdue level to each of your assets for this query to produce best results.

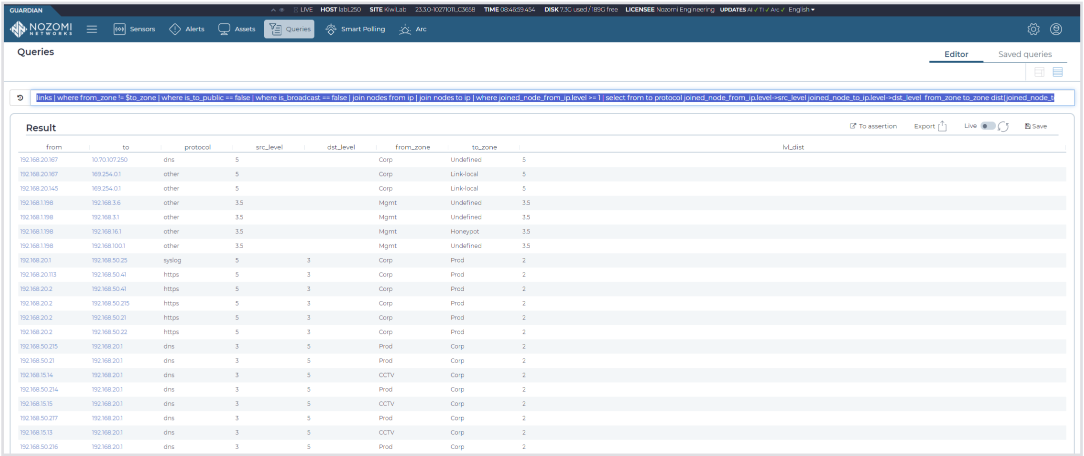
Where can this query be executed?

This query can be executed on a **Guardian** or **CMC**.

What is the query format?

```
links | where from_zone != $to_zone | where to != 0.0.0.0 | where to exclude? "224.0.0" | where to exclude? "255.255.255.255" | join nodes from ip | join nodes to ip | select from to protocol joined_node_from_ip.level->src_level joined_node_to_ip.level->dst_level  from_zone to_zone | select from to protocol dst_level src_level dist(dst_level,src_level) from_zone to_zone |  where dst_level_src_level_dist > 1 | sort dst_level_src_level_dist desc
```

What should the results look like?


GUARDIAN
LIVE HOST labL250 SITE KiwiLab 23.3.0-10271011_C3658 TIME 08:46:59.454 DISK 7.3G used / 189G free LICENSEE Nozomi Engineering UPDATES AI ✓ TI ✓ Arc ✓ English
NOZOMI NETWORKS
Sensors
Alerts
Assets
Queries
Smart Polling
Arc
Queries
Editor
Saved queries
Result
To assertion
Export
Live
Save

```
links | where from_zone != $to_zone | where is_to_public == false | where is_broadcast == false | join nodes from ip | join nodes to ip | where joined_node_from_ip.level >= 1 | select from to protocol joined_node_from_ip.level->src_level joined_node_to_ip.level->dst_level  from_zone to_zone dist(joined_node_t
```

from	to	protocol	src_level	dst_level	from_zone	to_zone	lvl_dist
192.168.20.167	10.70.107.250	dns	5		Corp	Undefined	5
192.168.20.167	169.254.0.1	other	5		Corp	Link-local	5
192.168.20.145	169.254.0.1	other	5		Corp	Link-local	5
192.168.1.198	192.168.3.6	other	3.5		Mgmt	Undefined	3.5
192.168.1.198	192.168.3.1	other	3.5		Mgmt	Undefined	3.5
192.168.1.198	192.168.16.1	other	3.5		Mgmt	Honeypot	3.5
192.168.1.198	192.168.100.1	other	3.5		Mgmt	Undefined	3.5
192.168.20.1	192.168.50.25	syslog	5	3	Corp	Prod	2
192.168.20.113	192.168.50.41	https	5	3	Corp	Prod	2
192.168.20.2	192.168.50.41	https	5	3	Corp	Prod	2
192.168.20.2	192.168.50.215	https	5	3	Corp	Prod	2
192.168.20.2	192.168.50.21	https	5	3	Corp	Prod	2
192.168.20.2	192.168.50.22	https	5	3	Corp	Prod	2
192.168.50.215	192.168.20.1	dns	3	5	Prod	Corp	2
192.168.50.21	192.168.20.1	dns	3	5	Prod	Corp	2
192.168.15.14	192.168.20.1	dns	3	5	CCTV	Corp	2
192.168.50.214	192.168.20.1	dns	3	5	Prod	Corp	2
192.168.15.15	192.168.20.1	dns	3	5	CCTV	Corp	2
192.168.50.217	192.168.20.1	dns	3	5	Prod	Corp	2
192.168.15.13	192.168.20.1	dns	3	5	CCTV	Corp	2
192.168.50.216	192.168.20.1	dns	3	5	Prod	Corp	2

Query 16

Do we have ICMP traffic? Can we alert on that?

Why would you use this query?

This query could be used to detect changes in normal network operations, particularly in network environments where legacy devices cannot detect or protect against this type of attack.

What can you expect this query to produce?

This query outputs a table of links showing the origin and destination layers with respect to the Purdue model. Attention should be paid to those links which do not have layers assigned, and to those where the distance between layers is greater than 1. With some small modification, this query can be configured to assert/alert on this traffic.

When should you use this query?

Depending on your policies, ICMP traffic may, or may not, be allowed in your environment. Using this query can either detect unauthorized ICMP usage, or detect attempts at improper ICMP use, such as ICMP tunnelling.

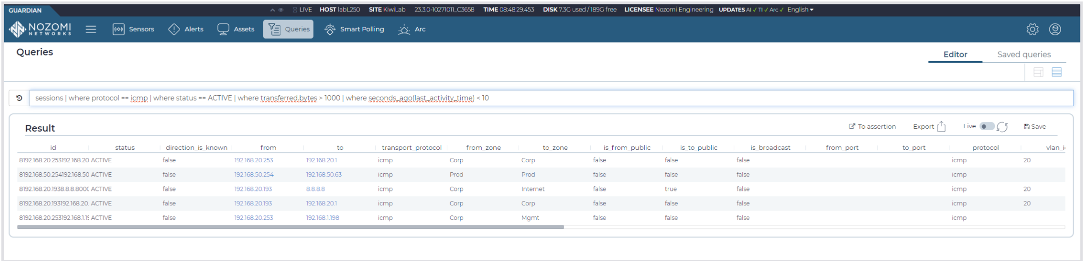
Where can this query be executed?

This query can be executed on a **Guardian** or **CMC**.

What is the query format?

```
sessions | where protocol == icmp | where status == ACTIVE | where transferred.bytes >
1000 | where seconds_ago(last_activity_time) < 10
```

What should the results look like?



Queries

```
sessions | where protocol == icmp | where status == ACTIVE | where transferred.bytes > 1000 | where seconds_ago(last_activity_time) < 10
```

Result

id	status	direction_is_known	from	to	transport_protocol	from_zone	to_zone	is_from_public	is_to_public	is_broadcast	from_port	to_port	protocol	vlan_i
8192.168.20.253192.168.20	ACTIVE	false	192.168.20.253	192.168.20.1	icmp	Corp	Corp	false	false	false			icmp	20
8192.168.50.254192.168.50	ACTIVE	false	192.168.50.254	192.168.50.63	icmp	Prod	Prod	false	false	false			icmp	
8192.168.20.1938.8.8.800(	ACTIVE	false	192.168.20.193	8.8.8.8	icmp	Corp	Internet	false	true	false			icmp	20
8192.168.20.193192.168.20.	ACTIVE	false	192.168.20.193	192.168.20.1	icmp	Corp	Corp	false	false	false			icmp	20
8192.168.20.253192.168.1.1	ACTIVE	false	192.168.20.253	192.168.1.198	icmp	Corp	Mgmt	false	false	false			icmp	

Query 17

What vulnerabilities exist on this device?

Why would you use this query?

This query would be useful when a deeper level of reporting is required across your entire asset inventory. The query can be refined to target assets, zones, or subnets with the same level of detail.

When should you use this query?

This query is intended to provide more detailed results than previous queries around vulnerabilities.

What can you expect this query to produce?

The table of results for this query contains a lot more information which may be of use when used as part of an audit. The results can be further refined by adding or removing fields as needed.

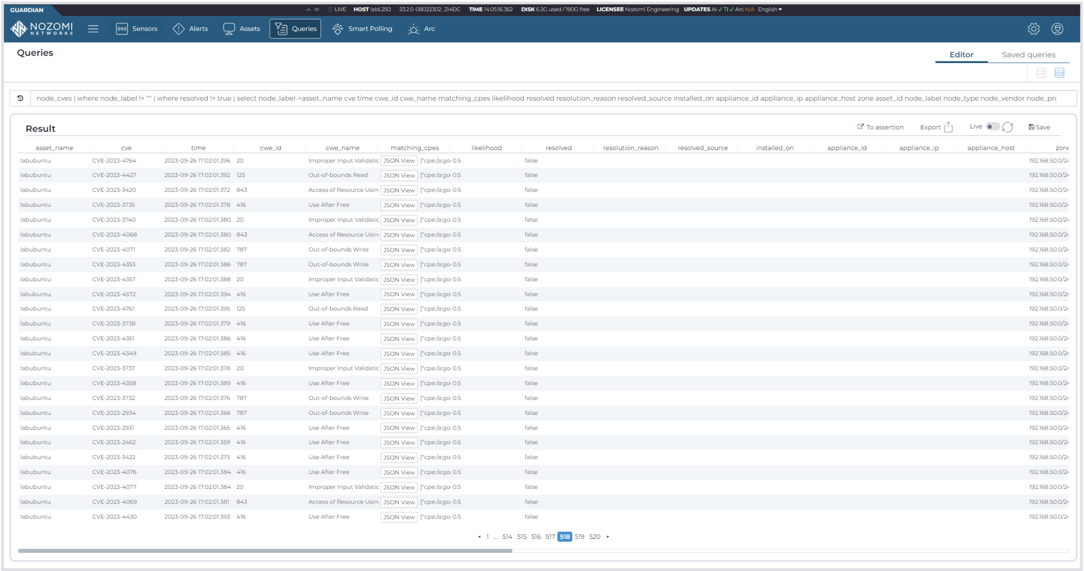
Where can this query be executed?

This query can be executed on a **Guardian** or **CMC**.

What is the query format?

```
node_cves | where node_label != "" | where resolved != true | select node_label->asset_name cve time cwe_id cwe_name matching_cpes likelihood resolved resolution_reason resolved_source installed_on appliance_id appliance_ip appliance_host zone asset_id node_label node_type node_vendor node_product_name node_firmware_version node_os resolution_status minimum_hotfix latest_hotfix cve_summary cve_references cve_score cve_creation_time cve_update_time cve_source | sort asset_name  asc
```

What should the results look like?


GUARDIAN
LIVE HOST labL250 23.2.0-08022302_214DC TIME 14:05:16.362 DISK 6.2G used / 190G free LICENSEE Nozomi Engineering UPDATES AI ✓ TI ✓ Arc N/A English
NOZOMI NETWORKS
Sensors
Alerts
Assets
Queries
Smart Polling
Arc
Editor
Saved queries
To assertion
Export
Live
Save

Queries

```
node_cves | where node_label != "" | where resolved != true | select node_label->asset_name cve time cwe_id cwe_name matching_cpes likelihood resolved resolution_reason resolved_source installed_on appliance_id appliance_ip appliance_host zone asset_id node_label node_type node_vendor node_prc
```

Result

asset_name	cve	time	cwe_id	cwe_name	matching_cpes	likelihood	resolved	resolution_reason	resolved_source	installed_on	appliance_id	appliance_ip	appliance_host	zone
labubuntu	CVE-2023-4764	2023-09-26 17:02:01.396	20	Improper Input Validatic	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-4427	2023-09-26 17:02:01.392	125	Out-of-bounds Read	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-3420	2023-09-26 17:02:01.372	843	Access of Resource Usin	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-3735	2023-09-26 17:02:01.378	416	Use After Free	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-3740	2023-09-26 17:02:01.380	20	Improper Input Validatic	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-4068	2023-09-26 17:02:01.380	843	Access of Resource Usin	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-4071	2023-09-26 17:02:01.382	787	Out-of-bounds Write	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-4353	2023-09-26 17:02:01.386	787	Out-of-bounds Write	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-4357	2023-09-26 17:02:01.388	20	Improper Input Validatic	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-4572	2023-09-26 17:02:01.394	416	Use After Free	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-4761	2023-09-26 17:02:01.395	125	Out-of-bounds Read	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-3738	2023-09-26 17:02:01.379	416	Use After Free	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-4351	2023-09-26 17:02:01.386	416	Use After Free	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-4349	2023-09-26 17:02:01.385	416	Use After Free	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-3737	2023-09-26 17:02:01.378	20	Improper Input Validatic	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-4358	2023-09-26 17:02:01.389	416	Use After Free	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-3732	2023-09-26 17:02:01.376	787	Out-of-bounds Write	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-2934	2023-09-26 17:02:01.366	787	Out-of-bounds Write	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-2931	2023-09-26 17:02:01.365	416	Use After Free	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-2462	2023-09-26 17:02:01.359	416	Use After Free	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-3422	2023-09-26 17:02:01.373	416	Use After Free	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-4076	2023-09-26 17:02:01.384	416	Use After Free	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-4077	2023-09-26 17:02:01.384	20	Improper Input Validatic	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-4069	2023-09-26 17:02:01.381	843	Access of Resource Usin	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2
labubuntu	CVE-2023-4430	2023-09-26 17:02:01.393	416	Use After Free	JSON View ["cpe:/a:go	0.5	false							192.168.50.0/2

Query 18

What Nozomi Networks licenses do I have?

Why would you use this query?

When the time comes to renew your licenses or provide evidence for an audit, this query will be more efficient than checking all sensors (physical, virtual or container) individually.

When should you use this query?

This query can save you time when auditing your license status.

What can you expect this query to produce?

This query will return a table of information showing the license status of each Nozomi Networks sensor connected to the CMC where the query was executed. You may need to run this query on multiple CMCs to capture your entire environment.

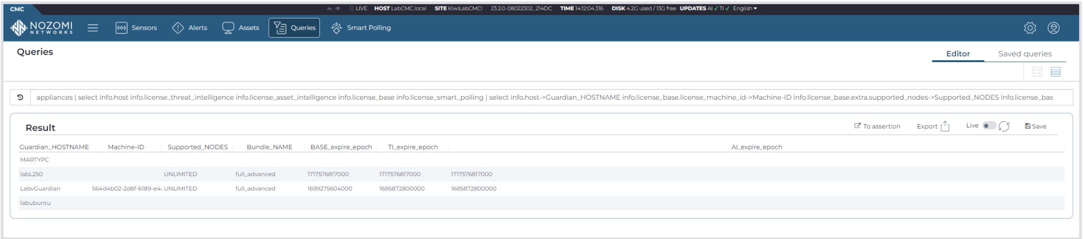
Where can this query be executed?

This query can only be executed on a **CMC**.

What is the query format?

```
appliances | select info.host info.license_threat_intelligence info.license_asset_intelligence info.license_base info.license_smart_polling | select info.host->Guardian_HOSTNAME info.license_base.license_machine_id->Machine-ID info.license_base.extra.supported_nodes->Supported_NODES info.license_base.bundle_name->Bundle_NAME info.license_base.extra.expire_date->Base_Expiry_date info.license_threat_intelligence.extra.expire_date->TI_expiry_date info.license_asset_intelligence.extra.expire_date->AI_expiry_date
```

What should the results look like?


CMC
LIVE HOST LabCMC.local SITE KiwiLabCMC! 23.2.0-08022302_214DC TIME 14:12:04.316 DISK 4.2G used / 13G free UPDATES AI ✓ TI ✓ English
NOZOMI NETWORKS
Sensors
Alerts
Assets
Queries
Smart Polling
Queries
Editor
Saved queries
Result
To assertion
Export
Live
Save

```
appliances | select info.host info.license_threat_intelligence info.license_asset_intelligence info.license_base info.license_smart_polling | select info.host->Guardian_HOSTNAME info.license_base.license_machine_id->Machine-ID info.license_base.extra.supported_nodes->Supported_NODES info.license_bas
```

Guardian_HOSTNAME	Machine-ID	Supported_NODES	Bundle_NAME	BASE_expire_epoch	TI_expire_epoch	AI_expire_epoch
MARTYPC						
labL250		UNLIMITED	full_advanced	1717576817000	1717576817000	1717576817000
LabvGuardian	564d4b02-2d8f-6189-e4	UNLIMITED	full_advanced	1699275604000	1685872800000	1685872800000
labubuntu						

Query 19

How can I tell when a multihomed asset is present?

Why would you use this query?

Use this query to find or report on multihomed assets. You could modify the query to assert/alert if there's an unexpected change.

What can you expect this query to produce?

This query will return a table of information showing the label/name of assets with more than one node associated. This will be a good indication of a multihomed asset. If the results of this report change over time, there is a likelihood that a configuration change has happened in your environment.

When should you use this query?

This query would be useful for reporting at any point in your security maturity journey or during an audit.

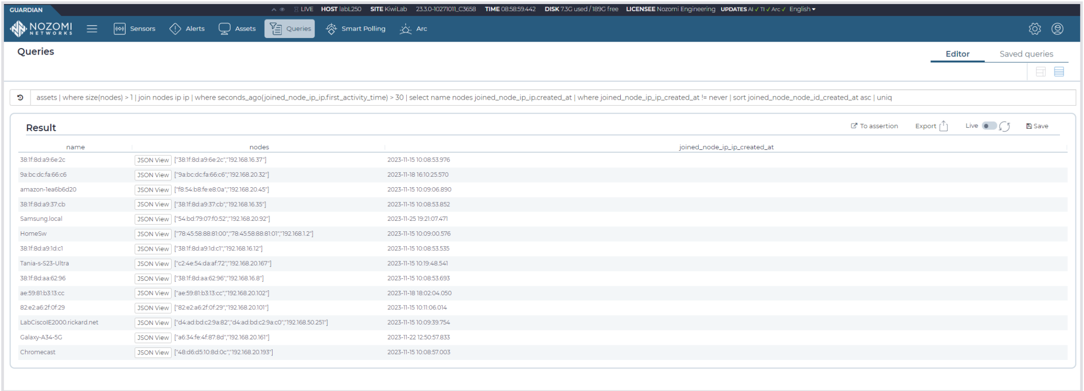
Where can this query be executed?

This query can be executed on a **Guardian** or **CMC**.

What is the query format?

```
assets | where size(nodes) > 1 | join nodes ip ip | where seconds_ago(joined_node_ip_ip.first_activity_time) > 30 | select name nodes joined_node_ip_ip.created_at | where joined_node_ip_ip_created_at != never | sort joined_node_node_id_created_at asc | uniq
```

What should the results look like?



LIVE **HOST** labL250 **SITE** KiwiLab 23.3.0-10271011_C3658 **TIME** 08:58:59.442 **DISK** 7.3G used / 189G free **LICENSEE** Nozomi Engineering **UPDATES** AI ✓ TI ✓ Arc ✓ English

Queries

```
assets | where size(nodes) > 1 | join nodes ip ip | where seconds_ago(joined_node_ip_ip.first_activity_time) > 30 | select name nodes joined_node_ip_ip.created_at | where joined_node_ip_ip_created_at != never | sort joined_node_node_id_created_at asc | uniq
```

Result

name	nodes	joined_node_ip_ip_created_at
38:1f:8d:a9:6e:2c	JSON View ["38:1f:8d:a9:6e:2c","192.168.16.37"]	2023-11-15 10:08:53.976
9a:bc:dc:fa:66:c6	JSON View ["9a:bc:dc:fa:66:c6","192.168.20.32"]	2023-11-18 16:10:25.570
amazon-1ea6b6d20	JSON View ["f8:54:b8:fe:e8:0a","192.168.20.45"]	2023-11-15 10:09:06.890
38:1f:8d:a9:37:cb	JSON View ["38:1f:8d:a9:37:cb","192.168.16.35"]	2023-11-15 10:08:53.852
Samsung.local	JSON View ["54:bd:79:07:f0:52","192.168.20.92"]	2023-11-25 19:21:07.471
HomeSw	JSON View ["78:45:58:88:81:00","78:45:58:88:81:01","192.168.1.2"]	2023-11-15 10:09:00.576
38:1f:8d:a9:1d:c1	JSON View ["38:1f:8d:a9:1d:c1","192.168.16.12"]	2023-11-15 10:08:53.535
Tania-s-S23-Ultra	JSON View ["c2:4e:54:da:af:72","192.168.20.167"]	2023-11-15 10:19:48.541
38:1f:8d:aa:62:96	JSON View ["38:1f:8d:aa:62:96","192.168.16.8"]	2023-11-15 10:08:53.693
ae:59:81:b3:13:cc	JSON View ["ae:59:81:b3:13:cc","192.168.20.102"]	2023-11-18 18:02:04.050
82:e2:a6:2f:0f:29	JSON View ["82:e2:a6:2f:0f:29","192.168.20.101"]	2023-11-15 10:11:06.014
LabCiscoIE2000.rickard.net	JSON View ["d4:ad:bd:c2:9a:82","d4:ad:bd:c2:9a:c0","192.168.50.251"]	2023-11-15 10:09:39.754
Galaxy-A34-5G	JSON View ["a6:34:fe:4f:87:8d","192.168.20.161"]	2023-11-22 12:50:57.833
Chromecast	JSON View ["48:d6:d5:10:8d:0c","192.168.20.193"]	2023-11-15 10:08:57.003

Query 20

How many assets are in each of my subnets?

Why would you use this query?

This query provides a quick answer and could be expanded to provide reporting and alerting.

When should you use this query?

Like many of the queries so far, this query would be useful for reporting at any point in your security maturity journey or during an audit.

What can you expect this query to produce?

This query will return a simple table listing subnets and the quantity of assets found in each.

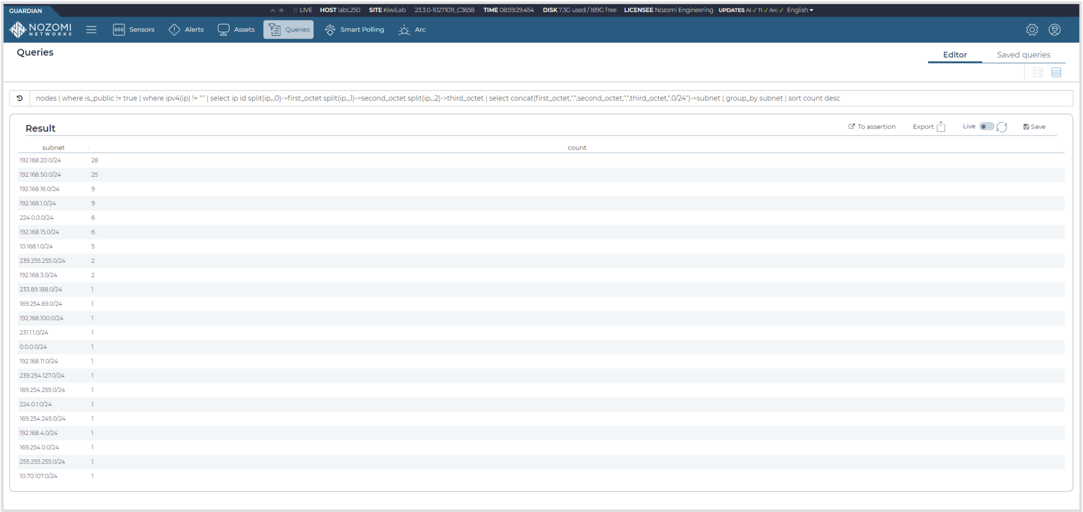
Where can this query be executed?

This query can be executed on a **Guardian** or **CMC**.

What is the query format?

```
nodes | where is_public != true | where is_broadcast != true | where ipv4(ip) != "" | select ip id split(ip,.,0)->first_octet split(ip,.,1)->second_octet split(ip,.,2)->third_octet | select concat(first_octet,".",second_octet,".",third_octet,".0/24")->subnet | group_by subnet | sort count desc
```

What should the results look like?



Queries

```
nodes | where is_public != true | where ipv4(ip) != "" | select ip id split(ip,.,0)->first_octet split(ip,.,1)->second_octet split(ip,.,2)->third_octet | select concat(first_octet,".",second_octet,".",third_octet,".0/24")->subnet | group_by subnet | sort count desc
```

Result

subnet	count
192.168.20.0/24	28
192.168.50.0/24	25
192.168.16.0/24	9
192.168.1.0/24	9
224.0.0.0/24	6
192.168.15.0/24	6
10.168.1.0/24	5
239.255.255.0/24	2
192.168.3.0/24	2
233.89.188.0/24	1
169.254.69.0/24	1
192.168.100.0/24	1
231.1.1.0/24	1
0.0.0.0/24	1
192.168.11.0/24	1
239.254.127.0/24	1
169.254.255.0/24	1
224.0.1.0/24	1
169.254.245.0/24	1
192.168.4.0/24	1
169.254.0.0/24	1
255.255.255.0/24	1
10.70.107.0/24	1

Cybersecurity for OT, IoT and Critical Infrastructure

Nozomi Networks protects the world's critical infrastructure from cyber threats. Our platform uniquely combines network and endpoint visibility, threat detection, and AI-powered analysis for faster, more effective incident response. Customers rely on us to minimize risk and complexity while maximizing operational resilience.

nozominetworks.com