

The background is a solid blue color with a complex, low-poly geometric pattern of various shades of blue, creating a textured, crystalline effect. The text is centered and rendered in a clean, white, sans-serif font.

NANOG 91 Hackathon Report

Hackathon In-Kind Sponsors



CONTAINERlab



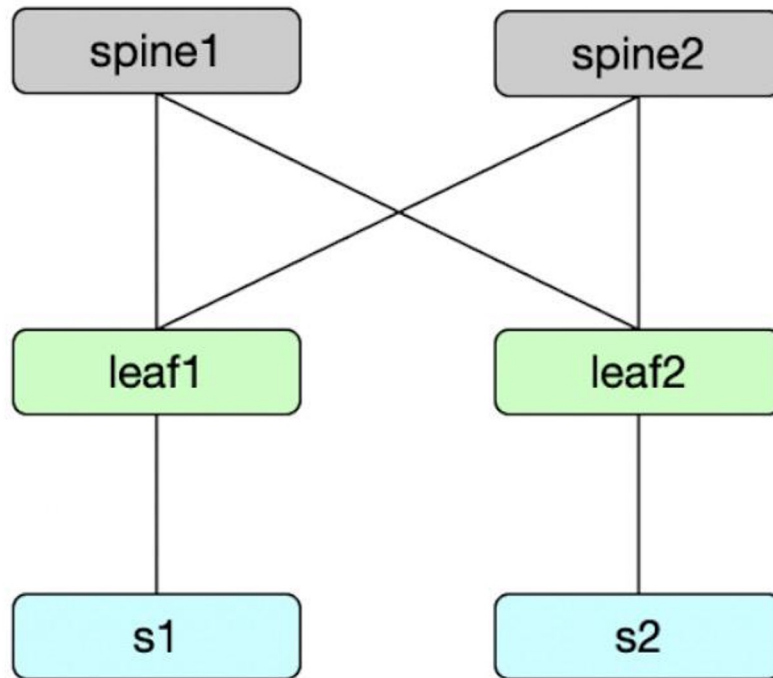
Honorable Mentions



Theme

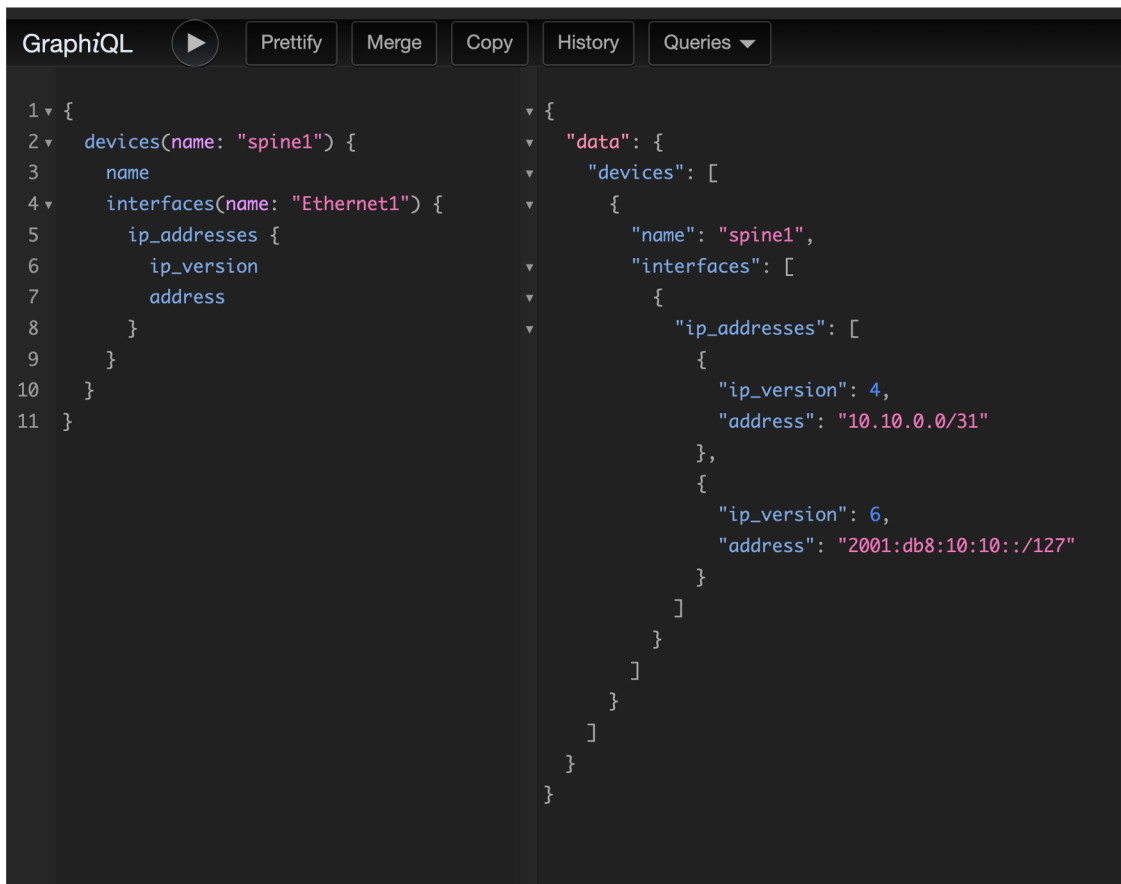
- CTF style!
- Resolve a variety of network problems
- Players were presented with a set of challenges
 - Challenges were to resolve a variety of network problems in order to capture a flag.
 - Each flag was equal to a certain amount of points. First to score most points wins.

Scenario - GraphQL



- A network topology was modeled in Nautobot with hardware roles, addressing, and interface connections.

Scenario - GraphQL



The screenshot shows a GraphQL IDE interface with a query on the left and its JSON response on the right. The query is:

```
1 {  
2   devices(name: "spine1") {  
3     name  
4     interfaces(name: "Ethernet1") {  
5       ip_addresses {  
6         ip_version  
7         address  
8       }  
9     }  
10  }  
11 }
```

The JSON response is:

```
{  
  "data": {  
    "devices": [  
      {  
        "name": "spine1",  
        "interfaces": [  
          {  
            "ip_addresses": [  
              {  
                "ip_version": 4,  
                "address": "10.10.0.0/31"  
              },  
              {  
                "ip_version": 6,  
                "address": "2001:db8:10:10::/127"  
              }  
            ]  
          }  
        ]  
      }  
    ]  
  }  
}
```

- Participants were challenged to craft GraphQL queries to retrieve specific information about the network

Scenario - GraphQL

```
caw@bananastand ~/git/github.com/chriswoodfield/n91 main • ? ./gql_bgp_config_gen.py leaf2
Gathering facts on leaf2 from https://n91-nautobot.hackathon.nanog.org/api/graphql/...

Rendering template bgp_template.j2...

router bgp 65011
  no bgp default ipv4-unicast
  maximum-paths 8 ecmp 64
  neighbor SPINE peer group
  neighbor SPINE send-community extended
  neighbor SPINE_IP6 peer group
  neighbor SPINE_IP6 send-community extended
  neighbor 10.10.0.2 peer group SPINE
  neighbor description spine1
  neighbor 10.10.0.2 remote-as 65000
  neighbor 10.10.1.2 peer group SPINE
  neighbor description spine2
  neighbor 10.10.1.2 remote-as 65000
  neighbor 2001:db8:10:10::2 peer group SPINE_IP6
  neighbor description spine1
  neighbor 2001:db8:10:10::2 remote-as 65000
  neighbor 2001:db8:10:10:1::2 peer group SPINE_IP6
  neighbor description spine2
  neighbor 2001:db8:10:10:1::2 remote-as 65000
caw@bananastand ~/git/github.com/chriswoodfield/n91 main • ?
```

- At the final stage, participants generated a router configuration from GraphQL facts in NB

Scenario: Network Modeling

- Participants were given a pre-seeded Nautobot instance.
- The first set of challenges led them through finding and entering network information in the Nautobot web interface.
- The second set of challenges involved performing similar activities programmatically using the API.
- Goal: learn and demonstrate the basics of modeling a network with a tool like Nautobot.

Scenario: Kubernetes

- Participants were guided through deploying a Kubernetes cluster
- Challenged to perform requested operations on cluster, report back results.

Scenario: Rain Cloud

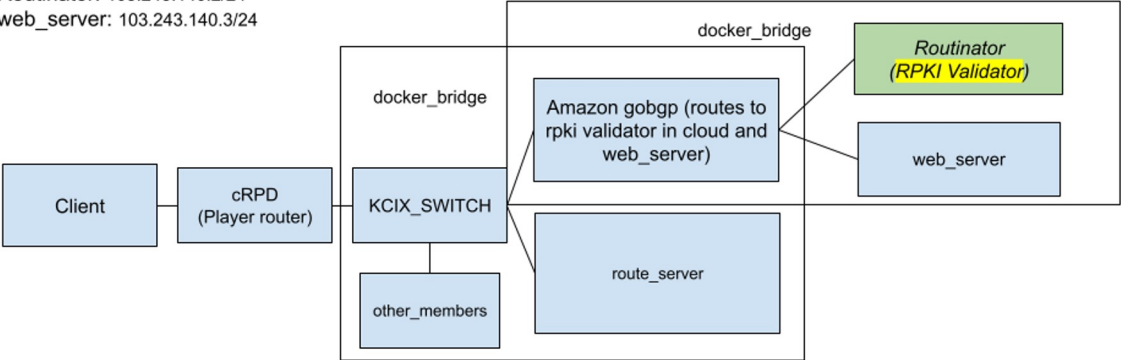
- Participants given a clab topology where they are a hired consultant for a network here in KC.
- The network is having issues reaching their RPKI validator and web server behind AWS.
- The network connects to KCIX and it is assumed that something has gone wrong at the IX causing these issues. KCIX is willing to accept the consultant's help as well.
- Players were tasked with diagnosing and resolving these issues.

Scenario: Rain Cloud

Client: 192.168.0.0/31
cRPD_client_int: 192.168.0.1/31

IX Network: 192.168.100.0/24
-cRPD_IX_link: 192.168.100.32/24
-AWS: 192.168.100.33/24
-IX Route Server: 192.168.100.34/24

AWS Network: 103.243.140.0/24
-Routinator: 103.243.140.2/24
-web_server: 103.243.140.3/24



Scenario: Rain Cloud

- First, find out why their web server is only intermittently reachable? It seems this may be contributed due to a flapping IX RS bgp session.

```
root@2_player_crpd> show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 1 Peers: 2 Down peers: 1
Table      Tot Paths  Act Paths  Suppressed  History  Damp State  Pending
inet.0
Peer      AS      InPkt    OutPkt    OutQ    Flaps  Last  Up/Dwn  State|#Active/Received/Accepted/Damped...
192.168.100.33  16509    74      83      0      0      1:12  Establ
  inet.0: 1/1/1/0
192.168.100.34  65002    0        0        0      3      12    Connect
```

Scenario: Rain Cloud

- After flapping is resolved the bgp session with the IX RS comes up, but now the web server and rpki validator are not reachable at all due to a new best-active route taken. The player's network is AS59209. This new path is originated by AS24342.

```
root@2_player_crpd> show route 103.243.140.0/24

inet.0: 7 destinations, 8 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

103.243.140.0/24  *[BGP/170] 00:00:07, localpref 100
                  AS path: 65002 24342 I, validation-state: unverified
                  > to 192.168.100.34 via eth2
                  [BGP/170] 00:09:17, localpref 100
                  AS path: 16509 59209 59209 I, validation-state: unverified
                  > to 192.168.100.33 via eth2

root@2_player_crpd> show validation session
Session          State  Flaps    Uptime #IPv4/IPv6 records
103.243.140.2    Connect  1         0/0
```

Scenario: Rain Cloud

- It is discovered that this new path is an rpk invalid route, the goal is to no longer take this route and capture a flag from the web server via the client behind the cRPD router.

VALIDATION

Results for 103.243.140.0/24 - AS24342 INVALID AS

At least one VRP Covers the Route Prefix, but no VRP ASN matches the route origin ASN

Unmatched VRPs - ASN		
Prefix	Max Length	ASN
103.243.140.0/24	24	AS59209

Unmatched VRPs - Length		
Prefix	Max Length	ASN
103.243.140.0/22	22	AS59209

Scenario: Rain Cloud

- Multiple ways to accomplish that goal. Once the invalid route is no longer best-active, the player can retrieve the flag. The next steps are to find a contact from the router originating the rpki-invalid route and find out when they started announcing it.

```
root@1_player_client:~# wget 103.243.140.3
--2024-06-12 16:40:12-- http://103.243.140.3/
Connecting to 103.243.140.3:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 69 [text/html]
Saving to: 'index.html'

index.html                               100%[=====>]          69 --.-KB/s   in 0s

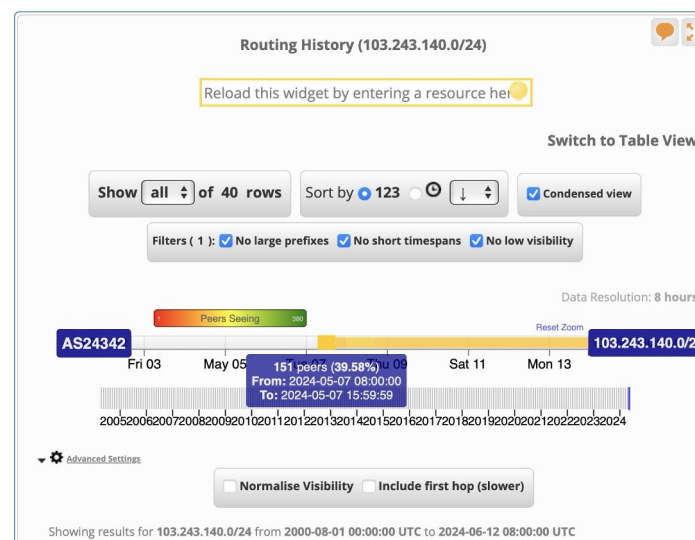
2024-06-12 16:40:12 (11.0 MB/s) - 'index.html' saved [69/69]

root@1_player_client:~# cat index.html
as long as you're not made of sugar, what's the problem if it rains?
root@1_player_client:~#
```

Scenario: Rain Cloud

- Run whois AS24342 and find a contact. Use RIPEStat routing history to see when the route was first announced.

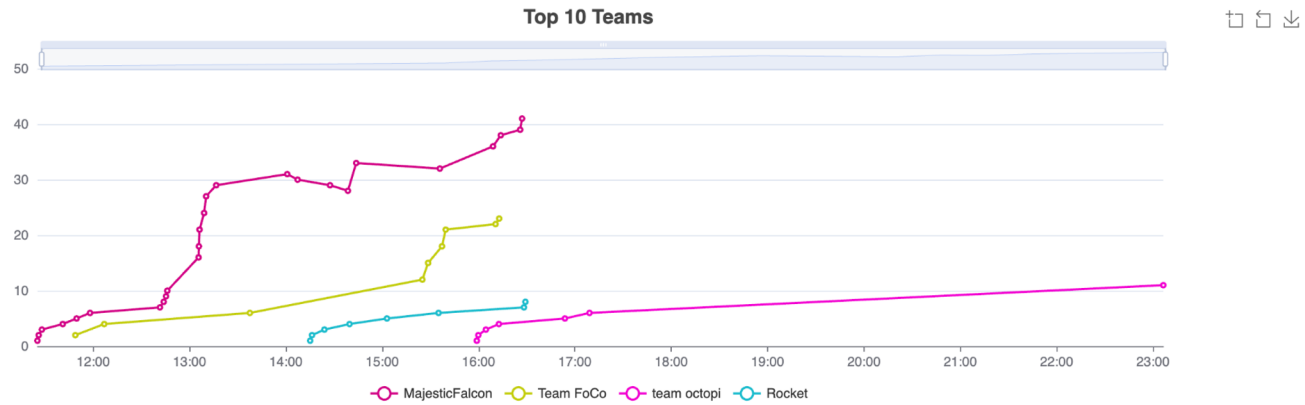
```
person:      Nizamul Islam
address:    16,Mohakhali, Navana Yusuf Infinity,7th Floor
country:    BD
phone:      +88-01614169844
e-mail:     nizamul.islam@bracmail.net
nic-hdl:    NI93-AP
mnt-by:     MAINT-BD-BBN
last-modified: 2020-01-27T10:23:58Z
source:     APNIC
```



What We've Learned

- We had high engagement in the room.
- Had a wider range of smaller challenges this time, teams kept occupied
- Learned more about CTFd features, avoided glitches (flags were strings or md5sums generated by validation scripts)
- Looking forward to building on this!

Results



Place	Team	Score
1	MajesticFalcon	41
2	Team FoCo	23
3	team octopi	11
4	Rocket	8

Next up... Hackathon at NANOG 92 in Toronto

- We will continue the competition format
- A virtual kick-off in the week of 14 October
- Sunday, 20 October 2024 is Hybrid Competition Day
- Registration to open mid-July