INGRESS AND EGRESS FILTERING FOR SERVICE PROVIDERS

WHOAMI

- Brian Knight
- Engineering Director at Nitel [AS 53828]
- Based in Chicago, IL
- Business eyeballs
  - Internet DIA, MPLS VPNs, L2 Ethernet, SD-WAN
- brian at knight-networks dot. com
- https://www.linkedin.com/in/brian-knight-94394021/

OUTLINE

- How We Got Started
- Decisions Made
- Scenarios
- Implementation
- Observations
- Recommendations

HOW WE GOT STARTED

- Oct 2020: Casey Deccio's DSAV team from BYU notified us of DNS spoofing vulnerabilities
  - Team was measuring vulnerability to DNS amplification attacks
- We found that:
  - In fact, DNS spoofing was possible because no anti-spoofing existed on network edge ports
- Could not find easy howto for configuring anti-spoofing on an SP network
- Posted to ML in Oct 2020

HOW WE GOT STARTED

- Many eyeball networks in the NANOG community did ingress filtering
- Many also:
  - Performed egress filtering
  - Blocked invalid IPs (bogon filtering)
  - Blocked invalid ports
  - Blocked traffic to critical infrastructure
- We still struggled to find a clear howto / template
  - That's what this preso strives to be

HOW WE GOT STARTED

- Ingress filtering covered in depth by **BCP 84** / RFC 3704
  - Outlines 5 techniques for anti-spoofing measures
  - Discusses implementation recommendations
- Has the same overarching goal as BCP 38
  - Filtering traffic from single-homed sites to SP's
- But BCP 84's audience includes any multi-homed network
- Deals mainly with security at data plane
  - Don't throw out your BGP route maps and prefix lists

DECISIONS MADE

- Static ACLs seemed to be the only way
  - Loose RPF:
    - If route exists in RIB, accept pkt
    - Too little granularity
  - Loose RPF ignoring default route:
    - Same as above, just doesn't consider default route
  - Strict RPF:
    - If route exists in RIB, and pkt came from best destination interface, accept
    - Too strict – would block legitimate traffic
  - Feasible RPF:
    - If route exists in RIB, and pkt came in on any valid destination interface
    - Not currently an option on $VENDOR_C_XR

DECISIONS MADE

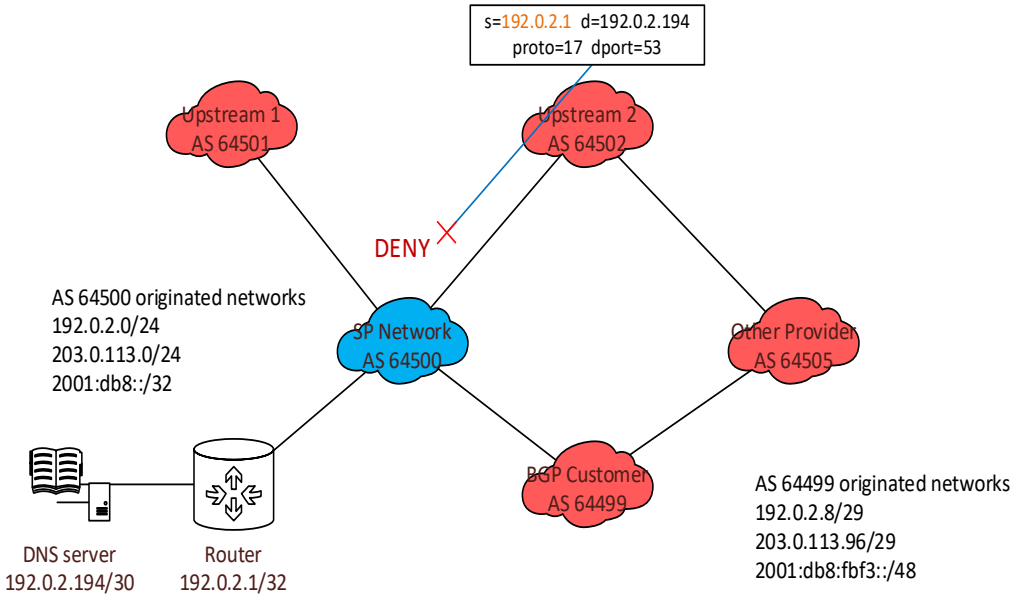
- Block invalid traffic on egress as well as ingress
- Block ”Bogon” traffic in and out
- Block multicast
- Invalid services or ports
  - UDP and TCP port 0 should never be seen or used
  - UDP and TCP port 445 should not be used
- Infrastructure
  - Except for ICMP and traceroute, nothing should be permitted to hit our router loopbacks or internal point-to-point links

DECISIONS MADE

- Ease of administration was key
- Implementation should be the same on every device
  - Same set of ACLs, configured the same way
    - The same ACL set is used on all { transit, IX, direct peer } ports on all routers
  - The same object groups are used on all routers
    - The same prefixes are in the same object groups across the entire router fleet

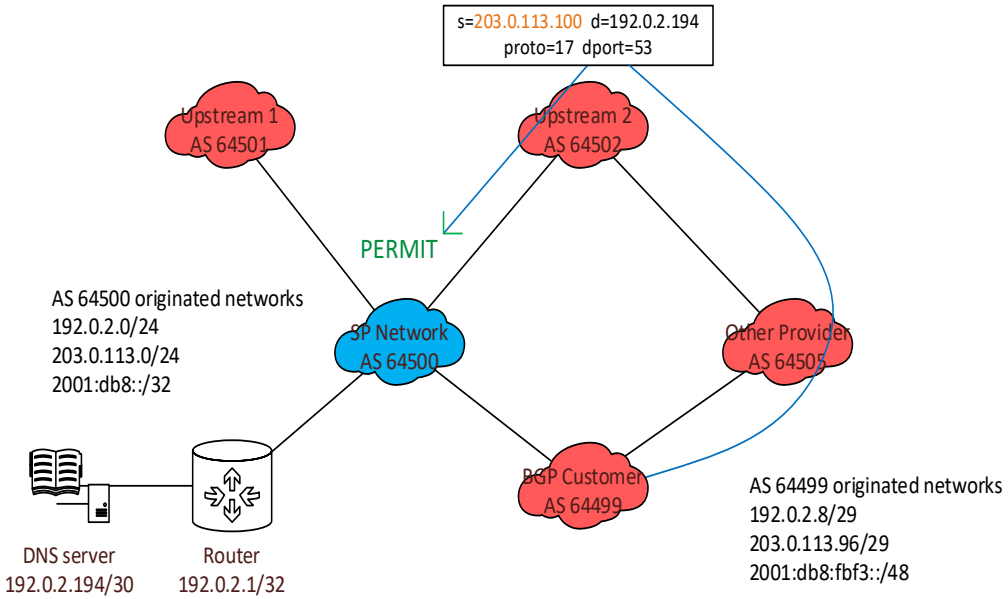
SCENARIOS – INGRESS AGGREGATE

- Source is part of aggregate block
- DENY



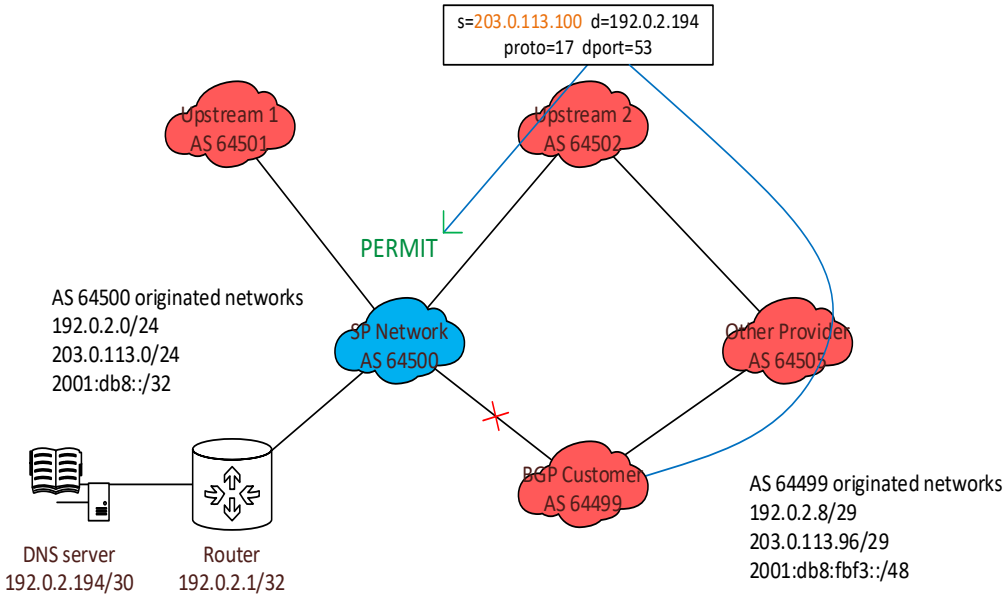
SCENARIOS – INGRESS CUSTOMER

- Source = customer PI/PA block
- PERMIT



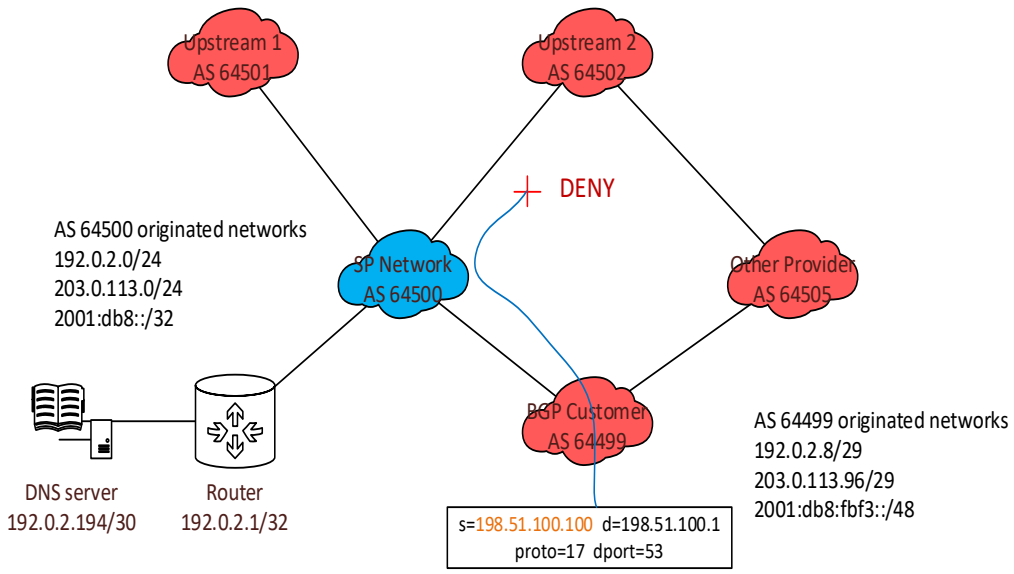
SCENARIOS – INGRESS CUSTOMER

- Source = customer PI/PA block
- PERMIT



SCENARIOS – EGRESS

- Egress
- Source is not from customer or aggregate
- DENY



IMPLEMENTATION

- Static ACL approach
- Inbound and outbound ACLs, for IPv4 and IPv6
- Object groups
- Content of ACLs and groups is exactly the same across every router

IMPLEMENTATION – OBJECT GROUPS

IPV4 GROUP	IPV6 GROUP	PURPOSE
IPV4-IX	IPV6-IX	All IX prefixes at all locations
IPV4-PEER-WAN	IPV6-PEER-WAN	My direct peer /30's or /126's
IPV4-TRAN-WAN	IPV6-TRAN-WAN	My transit /30's or /126's
IPV4-CUST	IPV6-CUST	Customer PI / PA prefixes
IPV4-BOGON	IPV6-BOGON	Non-unique or invalid prefixes
IPV4-INFRA	IPV6-INFRA	Router loopbacks and point-to-points
IPV4-BGP-AGG	IPV6-BGP-AGG	My aggregate prefixes
IPV4-BACKDOOR-HOSTS		Other hosts found to be sending traffic into my network Treated similar to IPV4-CUST

IMPLEMENTATION

- Four static ACLs
  - IPV4-INET-IN
  - IPV4-INET-OUT
  - IPV6-INET-IN
  - IPV6-INET-OUT

IMPLEMENTATION

- Ingress ACLs IPV4-INET-IN and IPV6-INET-IN
  - Deny bogon IPs
  - Deny invalid port 0
  - Permit all traffic to/from transit, IX, and peering WAN blocks
  - Deny multicast
  - Permit where source = BGP customer IP space
  - Deny where source = Agg IP space
  - Permit where dest = Agg + Customer IP space
  - Deny any any

*See appendix for an example of these ACLs

IMPLEMENTATION

- Egress ACLs IPV4-INET-OUT and IPV6-INET-OUT
  - Deny bogon IPs
  - Deny invalid port 0
  - Permit all traffic to/from transit, IX, and peering WAN blocks
  - Deny multicast
  - Permit where dest = Customer IP space
  - Permit where source = Nitel Agg + Customer IP space
  - Deny any any

*See appendix for an example of these ACLs

IMPLEMENTATION

The Plan:

- Create the initial ACLs (IPv4 and IPv6)
- Apply to Internet transits, IXes, and direct peers
- Enable blocking for bogon prefixes immediately
- Other rules that were intended to be “deny”, make them permit / log rules for the testing period
- Analyze logs for matching traffic
- Refine the ACL & object groups
- Repeat until only unwanted traffic is being logged
- Switch “permit” to “deny”

OBSERVATIONS

- Traffic review
  - ACLs logged denied traffic to our syslog server
  - Logs were processed into CSV
  - CSV imported into Excel
  - Used a PivotTable to summarize and view the data
  - IPs were cross-referenced as needed with router configs, RIB/FIB, and BSS/OSS to determine if traffic was legit
  - 3 weeks' worth of logs were analyzed

OBSERVATIONS

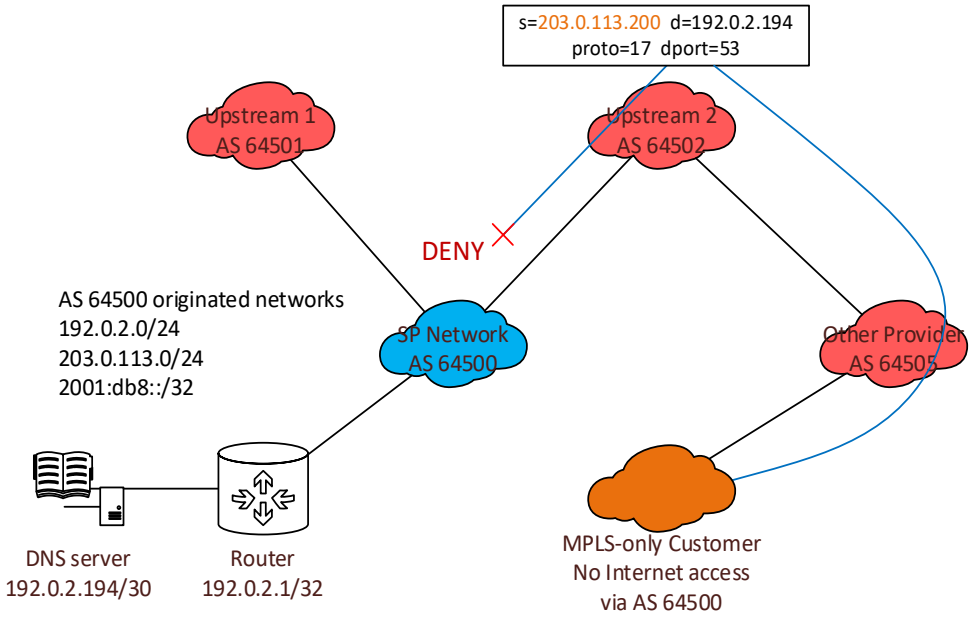
- Bogon prefixes
  - Blocked from the start
  - No issues observed
- Invalid ports
  - Port 0
    - Initially saw a lot of random traffic – $VENDOR_C_XR logs fragments with port = 0!
    - Plenty of scanning and abuse, no real traffic observed
    - Many network tools don’t permit the use of port 0
  - Port 445
    - We did observe traffic that appeared to be bona fide
    - We decided to abandon port 445 blocking

OBSERVATIONS

- Infrastructure
  - A lot of abuse hit our infrastructure – all different ports
  - None of it was valid traffic
  - .. Except for that IPsec tunnel terminating to an old concentrator…

OBSERVATIONS

- Found many MPLS VPN IPs coming back in via transits
  - IPs were globally unique but not routed across public Internet
  - Customers likely had firewalls that did not NAT for globally unique IPs



OBSERVATIONS

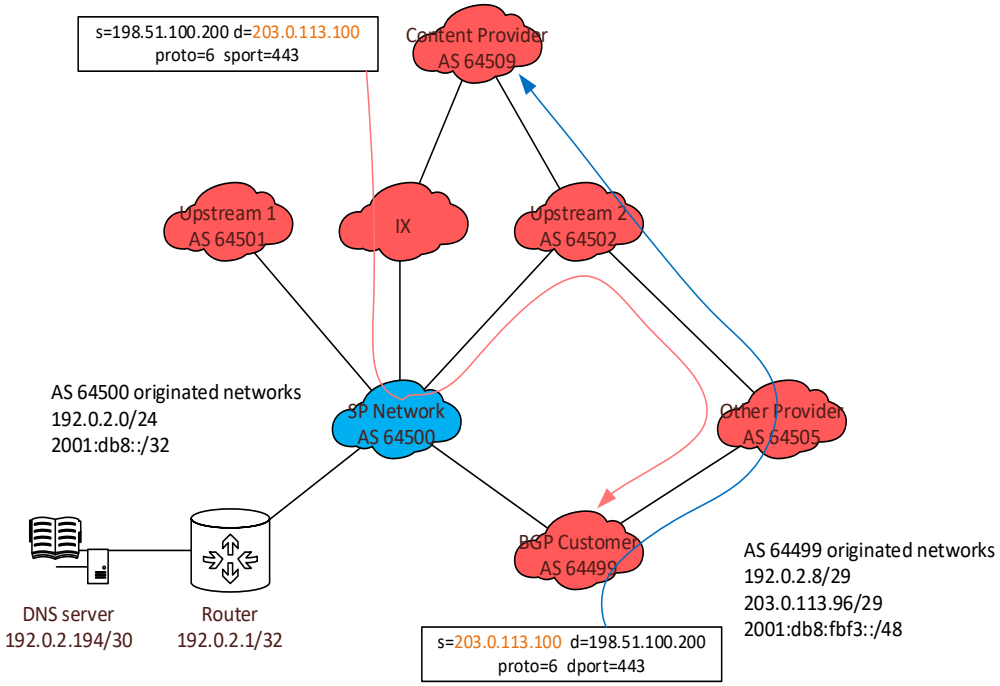
- Inbound anti-spoofing
  - Found a few Internet WAN IPs with back-door routing
  - In the interest of not breaking things that have been working, these were explicitly permitted in a separate ACL entry
- Inbound catch-all deny rule
  - No significant traffic found

OBSERVATIONS

- Outbound anti-spoofing
  - Other carriers' MPLS VRFs where WAN IP was not assigned by us
    - These VRFs were used to provide Internet access through us
  - Lots of back-door connections
    - Customer router receives traffic on other provider's WAN IP, reply sent through us
  - These were blocked

OBSERVATIONS

- Outbound anti-spoofing
  - Source = Internet
  - Dest = customer IPs
  - Routing policy from customer did not make it to the remote end
  - So remote side would send reply to us, then we would forward back out transits
  - Can't assume BGP customer traffic will be routed to them directly



OBSERVATIONS

- Process implications
  - Any BGP customer must have their prefixes added to IPV4-CUST or IPV6-CUST on all routers
  - Any new aggregate blocks must be added to IPV4-AGG or IPV6-AGG
  - If a customer calls in with an issue where traffic is not getting through, we may need to look at those filters
- In practice, this has not cost us in terms of admin time
  - Had two issues with this since implementation, both resolved quickly
- After implementation, the DSAV team showed that we were no longer vulnerable to spoofed DNS traffic

RECOMMENDATIONS

- Agree internally what you plan to block
- The usual: Get all stakeholders involved early, communicate often
- The usual: Test, test, test
  - Start with rules that permit undesired traffic, but log
  - Then run down everything in your logs before switching the rules to deny

FUTURE WORK

- Automation of object-group IP block management
- Further security analysis and lockdown of point-to-point subnets
  - These are open wide from any subnet to the /30 or /126
  - A better practice may be to block all protocols except ICMP, traceroute, and BGP
- Further characterization of infrastructure subnets

Q&A

THANK YOU!

ACLS

```
object-group network ipv4 IPV4-BOGON
  # Contains non-unique or invalid prefixes
 0.0.0.0/8
 10.0.0.0/8
 100.64.0.0/10
 127.0.0.0/8
 172.16.0.0/12
 198.18.0.0/15
 240.0.0.0/4
 169.254.0.0/16
 192.0.0.0/24
 192.0.2.0/24
 192.168.0.0/16
 198.51.100.0/24
 203.0.113.0/24
 description Invalid IPV4 networks
```

```
object-group network ipv6 IPV6-BOGON
  # Contains non-unique or invalid prefixes
 ::/3
 4000::/3
 6000::/3
 8000::/3
 a000::/3
 c000::/3
 e000::/4
 f000::/5
 f800::/6
 fc00::/7
 fe00::/9
 fec0::/10
 2001::/23
 2001:2::/48
 2001:10::/28
 2001:db8::/32
 2002::/16
 3ffe::/16
 description Invalid IPV6 networks
```

ACLS

```
ipv4 access-list IPV4-INET-IN
 10 remark BCP 84 for transits, IX, and peering
 101 remark *** Block bogon networks as src or dest ***
 110 deny ipv4 net-group IPV4-BOGON any
 111 deny ipv4 any net-group IPV4-BOGON
 201 remark *** Blocked protocols ***
 210 deny udp any port-group TCPUDP-BLOCKED any log
 211 deny udp any any port-group TCPUDP-BLOCKED log
 212 deny tcp any port-group TCPUDP-BLOCKED any log
 213 deny tcp any any port-group TCPUDP-BLOCKED log
 301 remark *** Transit, IX, peer connected networks ***
 310 permit ipv4 net-group IPV4-PEER-WAN any
 311 permit ipv4 any net-group IPV4-PEER-WAN
 312 permit ipv4 net-group IPV4-TRAN-WAN any
 313 permit ipv4 any net-group IPV4-TRAN-WAN
 314 permit ipv4 net-group IPV4-IX any
 315 permit ipv4 any net-group IPV4-IX
 401 remark *** Block multicast ***
 410 deny ipv4 224.0.0.0 15.255.255.255 any
 411 deny ipv4 any 224.0.0.0 15.255.255.255
 501 remark *** Protect infrastructure subnets ***
 510 deny icmp any net-group IPV4-INFRA fragments log
 511 permit icmp any net-group IPV4-INFRA
 512 permit udp any range 1024 65535 net-group IPV4-INFRA range 33435
33535
 513 permit udp any range 33435 33535 net-group IPV4-INFRA range 1024
65535
 514 deny ipv4 any net-group IPV4-INFRA
 601 remark *** Customer Inet BGP Announced Prefixes ***
 620 permit ipv4 net-group IPV4-CUST any
 640 permit ipv4 net-group IPV4-BACKDOOR-HOSTS any
 701 remark *** Block originated networks ***
 710 deny ipv4 net-group IPV4-BGP-AGG any log
 801 remark *** Permit traffic only to networks we announce ***
 820 permit ipv4 any net-group IPV4-BGP-AGG
 840 permit ipv4 any net-group IPV4-CUST
 901 remark *** Deny all other traffic ***
 910 deny ipv4 any any log
```

ACLS

```
ipv6 access-list IPV6-INET-IN
 10 remark BCP 84 for transits, IX, and peering
 101 remark *** Block bogon networks as src or dest ***
 110 deny ipv6 net-group IPV6-BOGON any
 111 deny ipv6 any net-group IPV6-BOGON
 201 remark *** Blocked protocols ***
 210 deny udp any port-group TCPUDP-BLOCKED any log
 211 deny udp any any port-group TCPUDP-BLOCKED log
 212 deny tcp any port-group TCPUDP-BLOCKED any log
 213 deny tcp any any port-group TCPUDP-BLOCKED log
 301 remark *** Transit, IX, peer connected networks ***
 310 permit ipv6 fe80::/10 any
 311 permit ipv6 net-group IPV6-PEER-WAN any
 312 permit ipv6 any net-group IPV6-PEER-WAN
 313 permit ipv6 net-group IPV6-TRAN-WAN any
 314 permit ipv6 any net-group IPV6-TRAN-WAN
 315 permit ipv6 net-group IPV6-IX any
 316 permit ipv6 any net-group IPV6-IX
 401 remark *** Block multicast ***
 410 deny ipv6 ff00::/8 any
 411 deny ipv6 any ff00::/8
 501 remark *** Protect infrastructure subnets ***
 510 deny icmpv6 any net-group IPV6-INFRA fragments log
 511 permit icmpv6 any net-group IPV6-INFRA
 512 permit udp any range 1024 65535 net-group IPV6-INFRA range 33435
33535
 513 permit udp any range 33435 33535 net-group IPV6-INFRA range 1024
65535
 514 deny ipv6 any net-group IPV6-INFRA
 601 remark *** Customer Inet BGP Announced Prefixes ***
 620 permit ipv6 net-group IPV6-CUST any
 701 remark *** Block networks we originate ***
 710 deny ipv6 net-group IPV6-BGP-AGG any log
 801 remark *** Permit traffic only to networks we announce ***
 820 permit ipv6 any net-group IPV6-BGP-AGG
 840 permit ipv6 any net-group IPV6-CUST
 901 remark *** Deny all other traffic ***
 910 deny ipv6 any any log
```

ACLS

```
ipv4 access-list IPV4-INET-OUT
 10 remark BCP 84 for transits, IX, and peering
 101 remark *** Block bogon networks as src or dest ***
 110 deny ipv4 net-group IPV4-BOGON any
 111 deny ipv4 any net-group IPV4-BOGON
 201 remark *** Blocked protocols ***
 210 deny udp any port-group TCPUDP-BLOCKED any log
 211 deny udp any any port-group TCPUDP-BLOCKED log
 212 deny tcp any port-group TCPUDP-BLOCKED any log
 213 deny tcp any any port-group TCPUDP-BLOCKED log
 301 remark *** Transit, IX, peer networks ***
 310 permit ipv4 net-group IPV4-PEER-WAN any
 311 permit ipv4 any net-group IPV4-PEER-WAN
 312 permit ipv4 net-group IPV4-TRAN-WAN any
 313 permit ipv4 any net-group IPV4-TRAN-WAN
 314 permit ipv4 net-group IPV4-IX any
 315 permit ipv4 any net-group IPV4-IX
 401 remark *** Block multicast ***
 410 deny ipv4 224.0.0.0 15.255.255.255 any
 411 deny ipv4 any 224.0.0.0 15.255.255.255
 601 remark *** Customer Inet BGP Announced Prefixes ***
 620 permit ipv4 any net-group IPV4-CUST
 640 permit ipv4 any net-group IPV4-BACKDOOR-HOSTS
 801 remark *** Permit locally sourced traffic ***
 820 permit ipv4 net-group IPV4-BGP-AGG any
 840 permit ipv4 net-group IPV4-CUST any
 901 remark *** Deny all other traffic ***
 910 deny ipv4 any any log
```

ACLS

```
ipv6 access-list IPV6-INET-OUT
 10 remark BCP 84 for transits, IX, and peering
 101 remark *** Block bogon networks as src or dest ***
 110 deny ipv6 net-group IPV6-BOGON any
 111 deny ipv6 any net-group IPV6-BOGON
 201 remark *** Blocked protocols ***
 210 deny udp any port-group TCPUDP-BLOCKED any log
 211 deny udp any any port-group TCPUDP-BLOCKED log
 212 deny tcp any port-group TCPUDP-BLOCKED any log
 213 deny tcp any any port-group TCPUDP-BLOCKED log
 301 remark *** Transit, IX, peer networks ***
 310 permit ipv6 fe80::/10 any
 311 permit ipv6 net-group IPV6-PEER-WAN any
 312 permit ipv6 any net-group IPV6-PEER-WAN
 313 permit ipv6 net-group IPV6-TRAN-WAN any
 314 permit ipv6 any net-group IPV6-TRAN-WAN
 315 permit ipv6 net-group IPV6-IX any
 316 permit ipv6 any net-group IPV6-IX
 401 remark *** Block multicast ***
 410 deny ipv6 ff00::/8 any
 411 deny ipv6 any ff00::/8
 601 remark *** Customer Inet BGP Announced Prefixes ***
 620 permit ipv6 any net-group IPV6-CUST
 801 remark *** Permit locally sourced traffic ***
 820 permit ipv6 net-group IPV6-BGP-AGG any
 840 permit ipv6 net-group IPV6-CUST any
 901 remark *** Deny all other traffic ***
 910 deny ipv6 any any log
```

REFERENCES

- BCP 84: https://tools.ietf.org/search/bcp84
- Post to mailing list: https://mailman.nanog.org/pipermail/nanog/2020-October/210030.html