ROVER: Using the Reverse DNS to Secure BGP Route Origins

Presentation to DNS-OARC

March 21, 2012

SECURE64

Topics

SECURE64

- IP Hijacking
- What does this have to do with Reverse DNS?
- Rover Overview
- Reverse DNS Naming Convention for CIDR blocks
- New record types for BGP Origin information
- The ROVER Testbed
- Preliminary Study on DNS load from ROVER

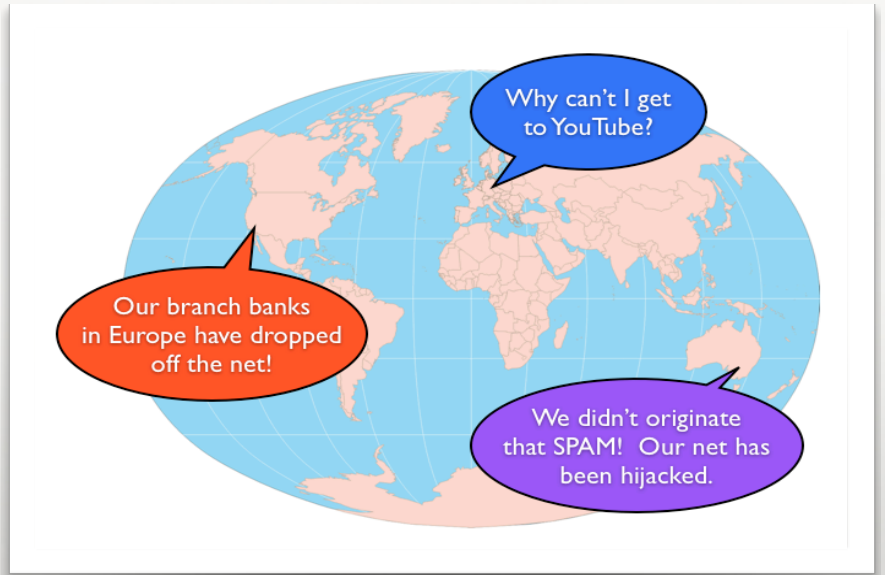
IP Hijacking in the News

SECURE64

- Web References: (click on hyperlink)
  - The Cyber-warfare Market 2012-2022 (Internet Re-routing)
  - A Chinese ISP momentarily hijacks the Internet
    - Traffic for 10 percent of the Internet, including to the sites of Dell, Apple, Starbucks and CNN, was redirected to China
  - IP route hijack prevention on tap at RSA Conference 2011
  - Dodo Explains National Telstra Outage
  - IP hijacking - Wikipedia, the free encyclopedia

Border Routers have no built-in security to block network hijacks.

Incidents are frequent. They are caused by *intentional* and *accidental* (fat-fingering) route leakages.

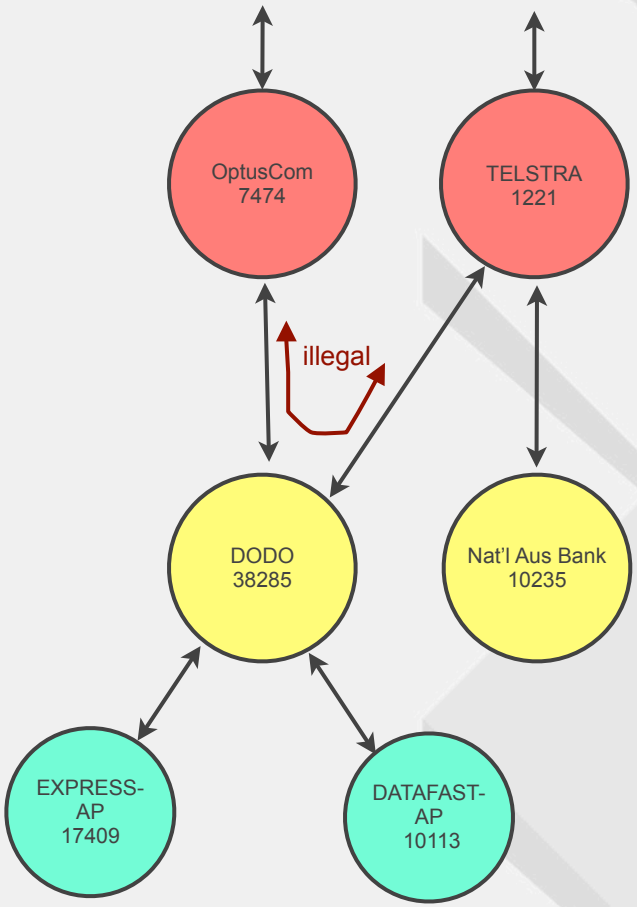


Typical IP Hijacks & Filter Errors

SECURE64

- Origin hijack
  - Rogue AS advertises a prefix with shorter path.
- Sub-Prefix Hijack
  - Example: YouTube hijacked by Pakistan in 2008
    - Youtube advertises 208.65.152.0/22
    - Pakistan Telecom advertised more specific prefix 208.65.153.0/24 to its provider, PCCW (AS 3491).
    - Route leaked out; 2/3 of internet couldn't reach Youtube for 2 hours.
- MITM - less typical; paths are usually short
- Failure to Filter - Telstra and Dodo in early March

Australia Telstra/Dodo



- As near as we can tell so far...
- DODO is multi-homed with 2 transit providers, Telstra and AS7474
- Lots of blame being passed around, from router failures to wrong filtering, etc.
- DODO announced routes it heard from AS7474 to Telstra
- Telstra sends LOTS of traffic to its customer, DODO, because customer path is preferred.
- This is like sending LOTS of highway traffic down a 2-lane road.
- Routers should never have a path containing TRANSIT-CUSTOMER-TRANSIT. Rover can filter this at DODO, or Telstra, or upstream to provide defense-in-depth.

SECURE64

2009 Large-Scale Route Leaks (collected by U. of Arizona)

SECURE64

"Large-scale" = 1/3 or more of internet

Date	Duration	Origin of Leak
02/14	1.96 Hours	Saudi Arabia
04/07	9.98 minutes	Nigeria
05/05	3.06 hours	Argentina
07/12	23.45 minutes	Romania
07/22	59 seconds	Russia
08/12	32 seconds	Indonesia
08/13	7.82 hours	Indonesia
12/04	68 seconds	Russia
12/15	62 seconds	Saudi Arabia

BGP Security Techniques

SECURE64

- Currently at the IETF: RPKI
- A Complementary Technology: ROVER -- Route Origin Verification
- Other Methods:
  - S-BGP et al, BGPMON

BGP Security Techniques

SECURE64

- RPKI
  - uses a chain of signed certificates with route-origin-authentication (ROA) data.
  - Uses an external PKI to distribute the data to be used by routers
  - Requires software and policy changes to be incorporated into the routers

BGP Security Techniques

SECURE64

- An Complementary Technology: ROVER --"Route Origin Verification"
- 2 Basic Components:
  - **Publish** route origin data in the reverse-DNS and authenticated via DNSSEC signatures
    - an authoritative distributed naming structure managed at each level by the proper owner of IP address block
  - **Verify** -- *Verify and Adjust* or *Verify and Advise* route announcements as they arrive at your routers using a 'helper' software appliance
    - use EXISTING routers and EXISTING policies -- no changes to router software or router policy configurations
- Proposed to private group at Quebec IETF (attendees included Level3, ARIN, RIPE, NLNetLabs, Cisco) and to various backbone and tier-1 ISP's to determine viability and support
- intention is to formally propose ROVER at the March IETF, Paris

ROVER Design Objectives

prevent BGP origin and sub-prefix hijacks

SECURE64

- *Hippocratic Oath* - "First, I will do no harm"
  - the system must not break what is working today!
  - it must fail-safe
- A viable solution must be publicly checkable
  - Anyone must be able to ask "who owns these IPs?"
- Should align operational costs with benefits
  - There has to be incentives to keeping this resource certification up to date
- Owners must be able to maintain their own authorization information
  - If someone owns IP addresses, they must be able to authorize origin ASN, next-hop, and other route security information
  - if someone assigns IP addresses to a 3rd party, they can act as an agent or delegate publishing authority information to that 3rd party
- Use DNS as an out-of-band advisory mechanism to advise BGP
  - Avoids cyclic dependencies
  - Rewards early movers without any flag days

Route Publishing: CSU at 129.82.0.0/16

SECURE64

129.82	/16	/17	/18	/19	/20/	/21	/22	/23	/24
0	Colorado State University		Colorado State University						
8	129.82/16		129.82.0/18						
16	AS 12145		AS 12145						
24									
32									
40									
48									
56									
64			Colorado State University						
72			129.82.64/18						
80			AS 12145						
88									
96									
104									
112									
120									
128			Colorado State University						
136			129.82.128/18						
144			AS 12145						
152									
160									
168									
176									.177 16496
184									
192			Colorado State University						
200			129.82.192/18						
208			AS 12145						
216									
224									
232									
240									
248									

Zone file: (uses CIDR reverse-DNS naming convention)

```
$ORIGIN 82.129.in-addr.arpa
$TTL 3600

@      IN  RLOCK  ; secure entire zone
m      IN  SRO 12145  ;129.82.0.0/16
0.0.m  IN  SRO 12145  ;129.82.0.0/18
1.0.m  IN  SRO 12145  ;129.82.64.0/18
0.1.m  IN  SRO 12145  ;129.82.128.0/18
1.1.m  IN  SRO 12145  ;129.82.192.0/18

; can now directly add /24 SROs
; or can let the lower octet do it

; existing delegations

0    IN   NS    rush.colostate.edu
1    IN   NS    rush.colostate.edu
;....
255  IN   NS    rush.colostate.edu
```

RLOCK = Route LOCK
SRO = Secure Route Origin
Automated provisioning tools have been written

ROVER Verification

SECURE64

- The published records in the REVERSE DNS can be used to:
  - create route filters on a periodic basis for loading into a router
  - or perform real-time verifications using a device that listens to announcements arriving at a router. Bogus announcements can either
    - send a notification to an operator
    - interact with a router to re-announce a competing route that blocks the bogus one

Route Classification

SECURE64

- ROVER listens to announcements, does a reverse-DNS query, and classifies the route as
  - *VALID - a matching origin was found*
  - *VIABLE - nothing found, can't say whether it is good or not*
  - *BOGUS - announced origins do not match data in SR or RSON protects the zone*

- CSU Example
  - 129.82.0.0/18 origin AS 12145 --> SECURE
  - 129.82.0.0/18 origin AS 666 --> BOGUS
  - 129.82.0.0/19 origin AS 666 --> BOGUS due to RLOCK
  - 129.83.0.0/16 origin 666 --> VIABLE (no data found)

Integrating Rover With Routers

SECURE64

- We do not want to change router code
- We also don't want to change any policy configurations

- Instead, we want BGP to operate as it always does, but we want to do out-of-band route verification
- We want a way to promote routes that are secure

Implications to Existing Routers

SECURE64

- What needs to change on my router?
  - NOTHING: Don't mess with policies; Don't change the IOS
  - Decision Process is Already Complex Process
    - Don't muck with it!
  - Some Simple Security Cases
    - If all routes were secure, should cause no change in you existing decision process
    - If a route clearly invalid, should never be chosen
  - But more complex mix of secure/uncertain
    - Claim Can Solve With the Three Buckets described earlier

How this would Work

SECURE64

- We create an “offload” box (Rover) to sit next to routers
- This box looks at Adj-RIB-Ins for routes and classifies them
  - We will use BMP or other methods to get Adj-RIB-Ins from routers
  - When a route is received, we count on the MRAI timer to give us ~30 seconds to verify a route
- The offload box will check routes against DNS
- We will re-announce valid route from ROVER if a conflict is found; higher local-pref and community strings will make the valid route the preferred route.

Blackhole Through Competition

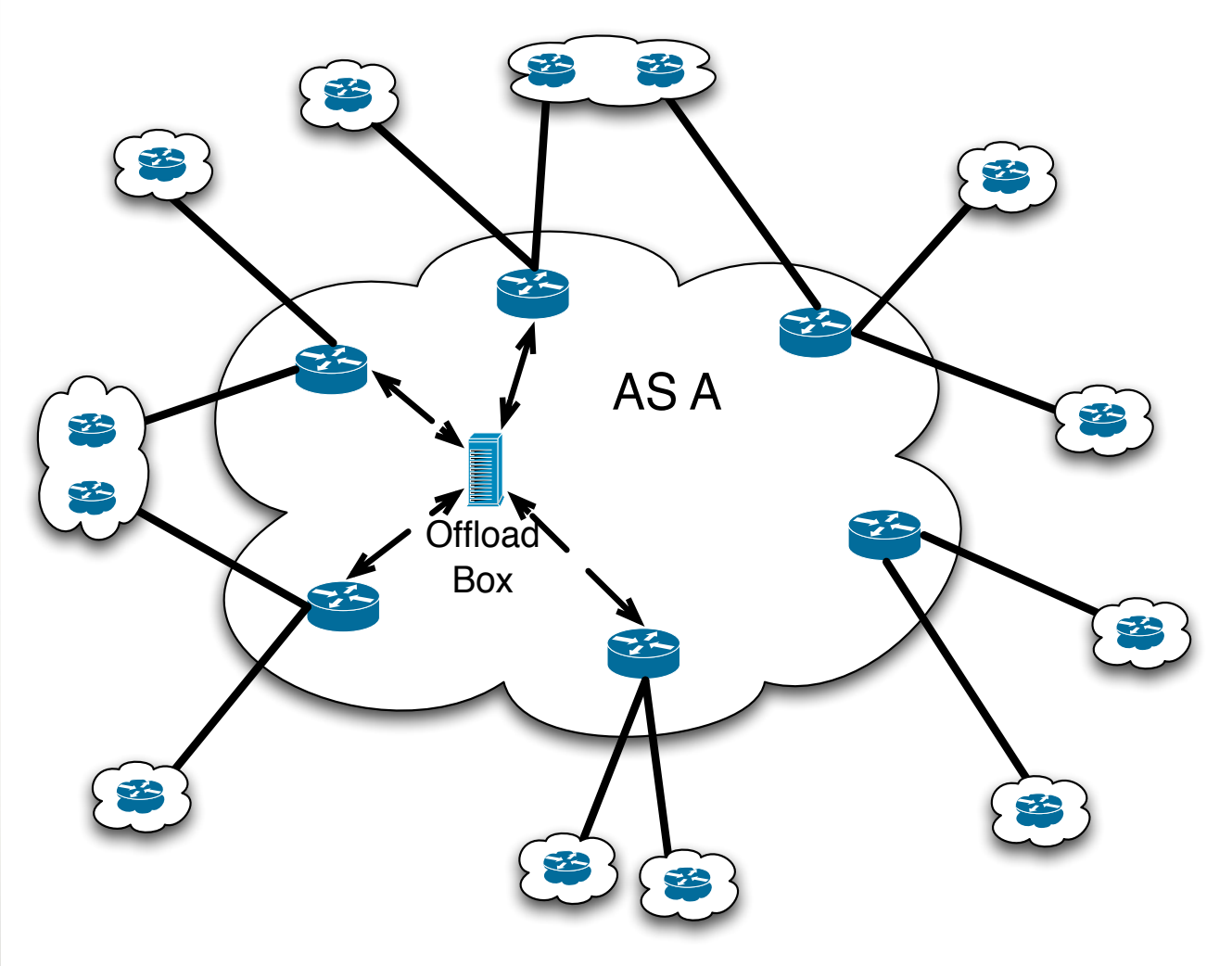
SECURE64

Objective: block bogus route to 129.82/16

- Case 1: Competing Already Route Exists
  - Competing 129.82/16 appears in Viable or Preferred.
  - If bogus route arrives at router, ROVER sends announcement with higher local pref configured.
- Case 2: Covering Route Exists
  - Have route to 129.82/15 in Viable or Preferred
  - But packets will follow more specific bogus 129.82/16 route
  - Announce a new Viable route to 129.82/16 using same attributes as valid 129.82/15.
  - Viable route wins and follows same path covering prefix
- Case 3: Non Routable Space (e.g. BOA)
  - Without bogus route, packets would be dropped
  - Add route to 129.82/16 next hop /dev/null to Viable Routes
  - Or just don't care….

Picture of offload box

SECURE64



Avoids a Cyclic Dependency

SECURE64

- Since this is an out-of-band mechanism failure to reach a zone reverts it to “viable” but not blocked
- Further, if zones have secondaries in other networks, they will still be reachable
- Further further, if a covering prefix provides transit, the queries may still flow over that route

Proposal

SECURE64

- publish your AS origins in the ROVER test-bed
  - to build a large set of test cases
  - to prove feasibility of the ROVER concept
  - to gain feedback to evolve the concept.

- optionally use a ROVER advisor in a 6-month trial (availability TBD)

CIDR Address Names in Reverse-DNS

Quick Overview of Naming Convention

More details can be found in the SATIN paper

SECURE64

Converting CIDR to reverse-DNS

SECURE64

- Invert the address per the usual reverse-DNS method. Remove any trailing zeroes.
  - 129.82.0.0/16 --> 82.129.in-addr.arpa
- Calculate `N = prefix-length mod 8.`
- if N = 0, you are at an octet boundary and are done.
- Otherwise:
  - add an “m” character to indicate “mask”
  - convert the least significant octet to binary, separate with “.” characters
  - truncate to the “N” significant binary characters for this prefix length
  - reverse the string per reverse DNS
- Examples: (showing step 1: “convert to binary”, and step 2: “truncate and reverse”)
  - 129.82.64.0/18 --> 129.82.m.0.1.0.0.0.0.0.0 --> 1.0.m.82.129.in-addr.arpa.
  - 129.82.64.0/20 --> 129.82.m. 0.1.0.0.0.0.0.0 --> 0.0.1.0.m.82.129.in-addr.arpa.
  - 129.82.160.0/20 --> 129.82.m.1.0.1.0.0.0.0.0 --> 0.1.0.1.m.82.129.in-addr.arpa.
  - 129.82.160.0/23 --> 129.82.m.1.0.1.0.0.0.0.0 --> 0.0.0.0.1.0.1.m.82.129.in-addr.arpa.

Converting Reverse-DNS Name to CIDR

- Mask length = 8*octets + number of binary digits
- Reverse the string. Add up the values of the binary digits to calculate the final octet. Append the "/" and mask length.
  - 1.0.m.82.129.in-addr.arpa --> 129.82.64.0/18
    - example has 2 octets + 2 binary digits, so mask length = 18
  - 0.0.1.0.m.82.129.in-addr.arpa --> 129.82.64.0/20
    - example has 2 octets + 4 binary digits, so mask length = 20
  - 0.0.0.1.0.1.m.129.in-addr.arpa --> 129.160.0/14
    - example has 1 octet + 6 binary digits, so mask length = 14

ROVER Zone Data: the new record types

Overview

SECURE64

Two New DNS Record Types

SECURE64

- **RLOCK**
  - *Purpose*: specify OPT-IN to route security for this zone. Prevents sub-prefix hijacks.
  - Placed at the zone apex to indicate secure routing is enabled for that zone. All route announcements that map to this zone will be denied as BOGUS unless an SRO record exists that specifically authorizes the announcement
  - May also be placed at domain names in the bottom level of the zone. (Rationale described later in this document).

- **SRO**
  - *Purpose*: authorize a route announcement by declaring a route origin and and optional next-hop
  - Placed at the domain name corresponding to the CIDR address block.

RLOCK Record

SECURE64

- has no RDATA fields
- Temporary implementation until standardization and IANA numbering:
  - `TYPE65400 \# 0`
  - 0 indicates data length, so no RDATA fields

SRO Record

SECURE64

- Has 1 mandatory field, 1 optional field
  - Mandatory: ORIGIN AS
  - Optional: TRANSIT AS
    - note, this is an experimental extension not mentioned in the IETF draft

- Temporary implementation until standardization and IANA numbering:
  - `TYPE65401 \# 4 xxxxxxxx`
  - `TYPE65401 \# 8 xxxxxxxxxxxxxxxx`
  - 4 byte data length indicates ORIGIN AS only, no TRANSIT AS
  - 8 byte length indicates both ORIGIN AS and TRANSIT AS are specified.
  - each data field consists of 4-byte
  - the data is entered as hexadecimal digits
  - This is able to handle both 2 and 4-byte AS numbers

Locking the next-level zones

SECURE64

- Zone cuts normally occur at octet (or nibble) boundaries, but can actually occur at any delegation point to a CIDR block.

- There are 2 places to lock a zone with RLOCK:
  - The zone apex of the delegated child zone
  - or, the bottom nodes of the parent zone (a nice effect of the CIDR naming convention). This can save a lot of effort.

- So... you have a choice. As an example, consider the 256 possible /24 children of a /16 zone.
  - You can either create 256 zones and provision each of them with RLOCKS at each zone apex (and possible SRO records).
  - Or, you can put in 256 RLOCK records at the 256 possible 0m24 to 255m24 records in the parent zone.

- The child zone, if present, takes precedence over the parent zone.

Authorizing Route Announcements

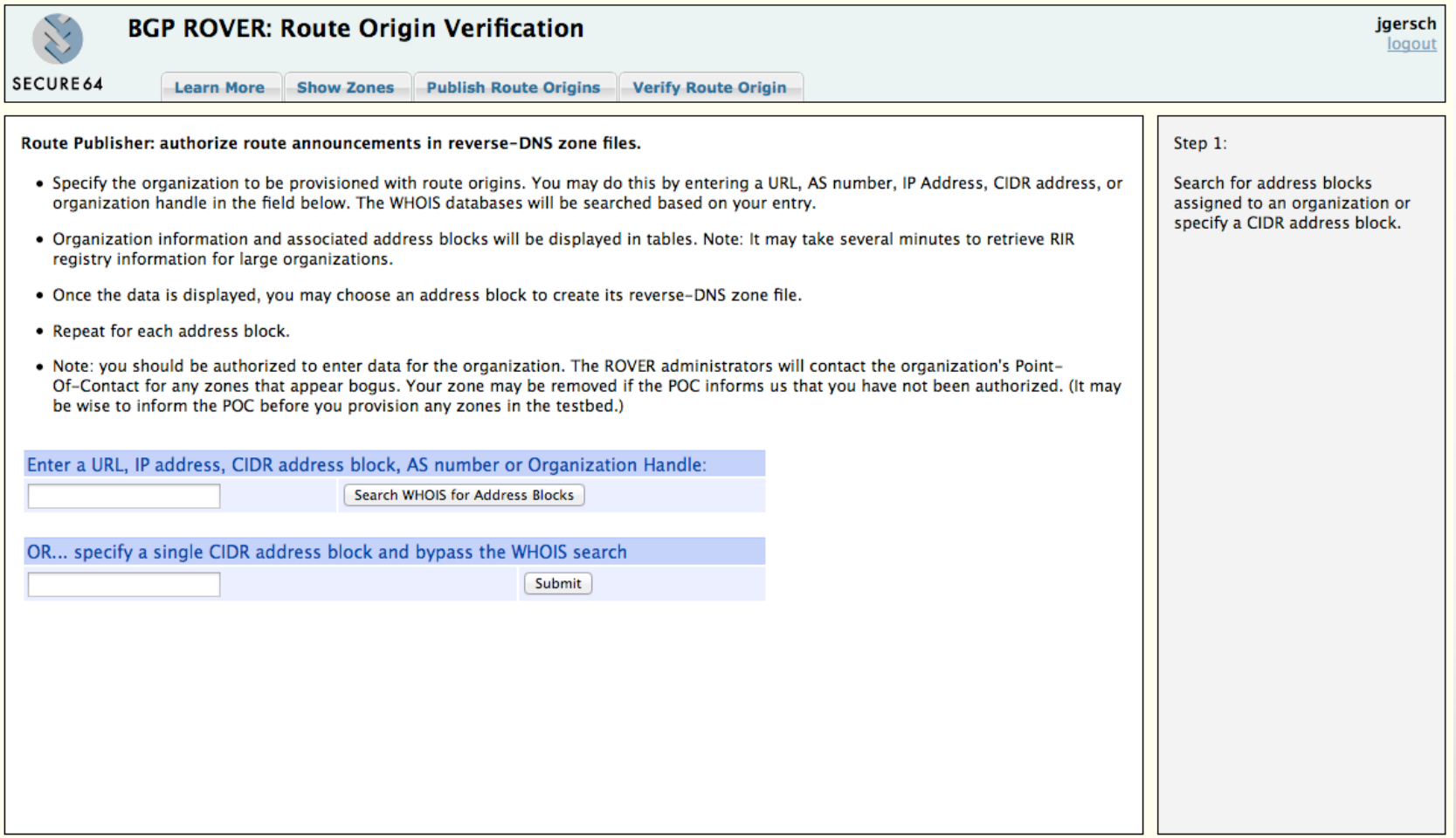
Step-by-Step Instructions
for
Provisioning a Reverse-DNS Zone
in the ROVER Testbed

SECURE64

1) Search WHOIS -- enter a URL or AS

SECURE64

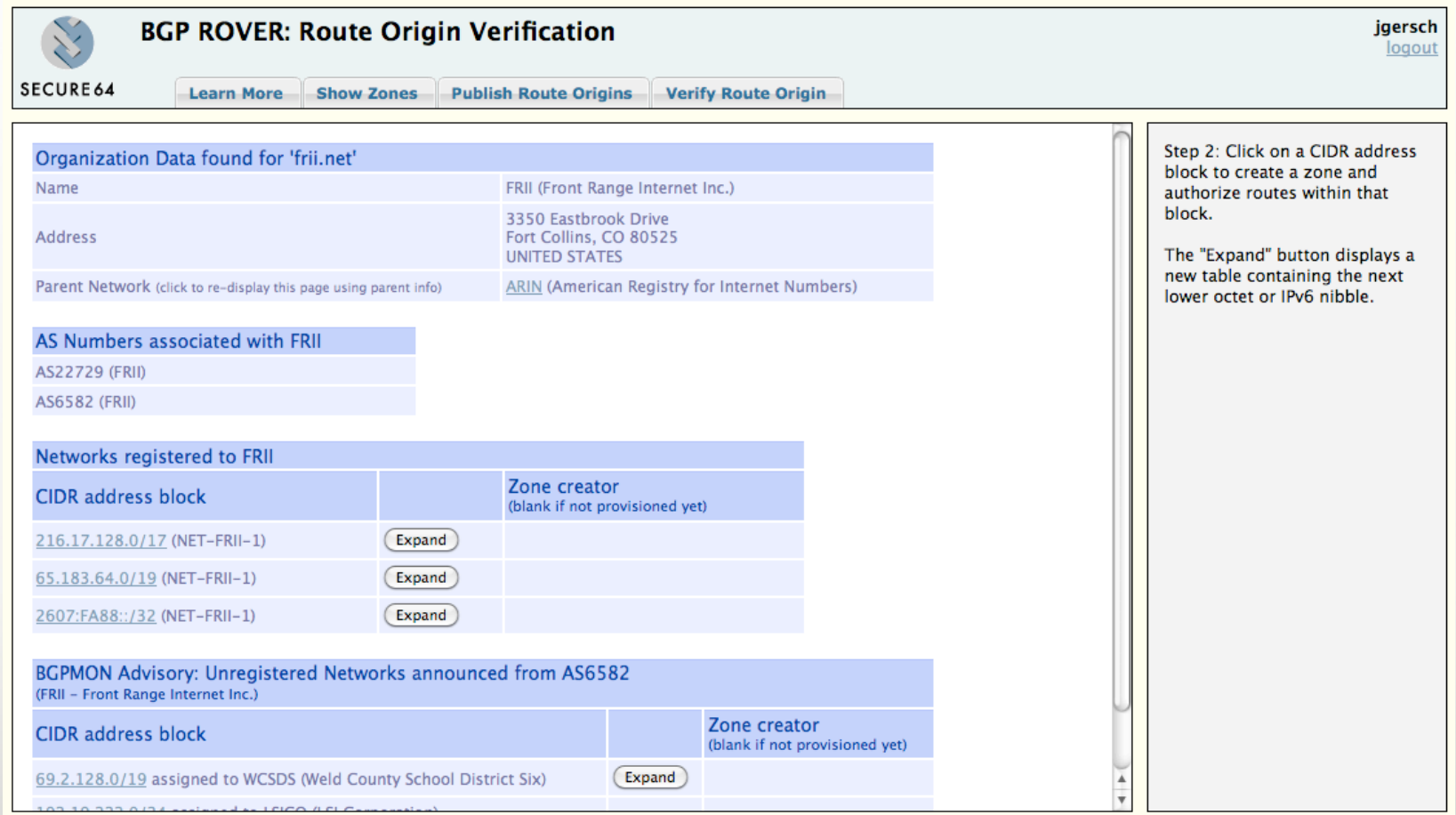
- This will search for relevant CIDR address blocks



2) Examine the information

SECURE64

- In some cases you may need to click the parent organization to display relevant CIDR blocks.
- You will see registered address blocks and “extra” blocks found by BGPMON that were announced from your AS numbers. These may or may not be legitimate connections.



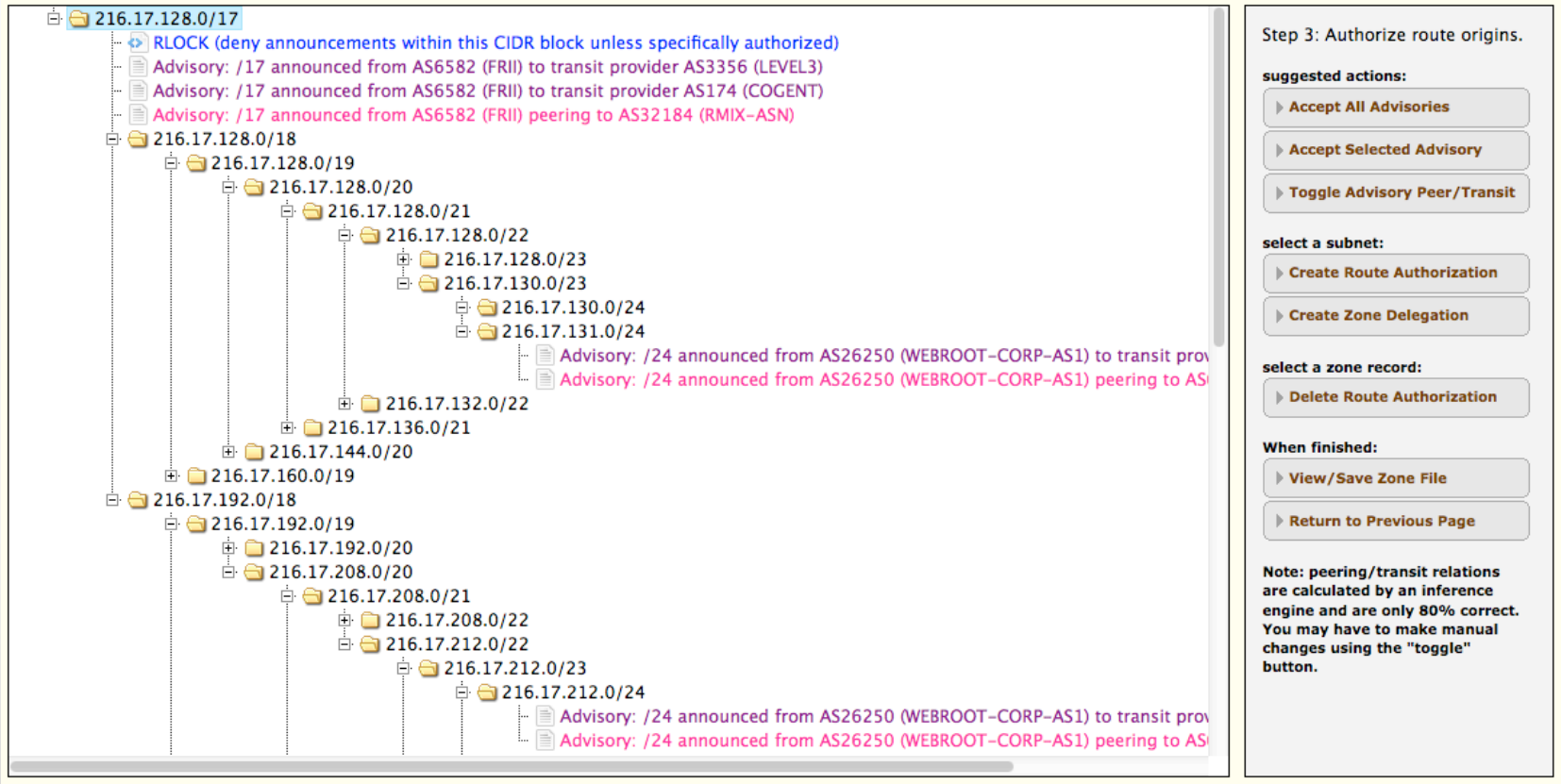
3) Select a CIDR block to provision

- Click on a CIDR block.
- or, expand the block to display the set of CIDR address blocks in the next lower octet or nibble. Then select one of those blocks.

4) Examine the CIDR block

SECURE64

- The Address Block is displayed as a tree (you can expand/collapse sub-blocks)
- A set of ADVISORY announcements are displayed. These may or may not be legitimate or complete. Examine carefully, and toggle transit/peer relationships if necessary. SRO statements that are generated will include origins and transits, or origins only, but no peers.
- Duplicate SRO statements will be eliminated in the final zone file.



5) Authorize Announcements

SECURE64

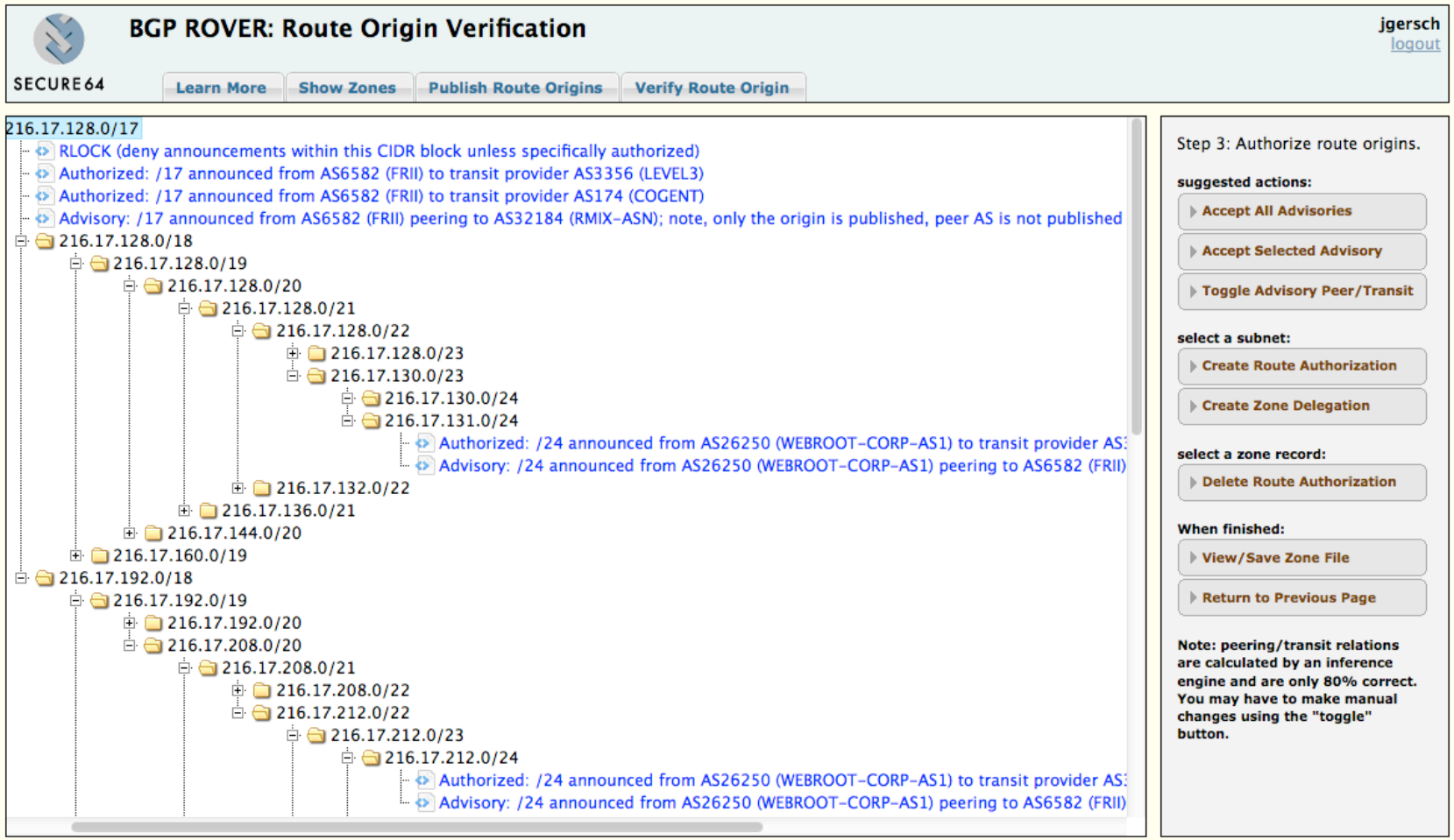
- You can authorize individual or all advisories
- You can delete records
- You can create a new announcement at any /xx in the tree.

- You may also create a delegation to another zone.
  - This will delete all levels of the tree below that cut point.
  - If you create a delegation at a leaf node (for example, at a /24 node in a /16 tree) it will also create an RLOCK statement to automatically lock the /24 zone. You may manually delete this lock if you prefer.
  - You should then create the new zone(s) by using the EXPAND button shown in the previous screens.

6) Double-check

SECURE64

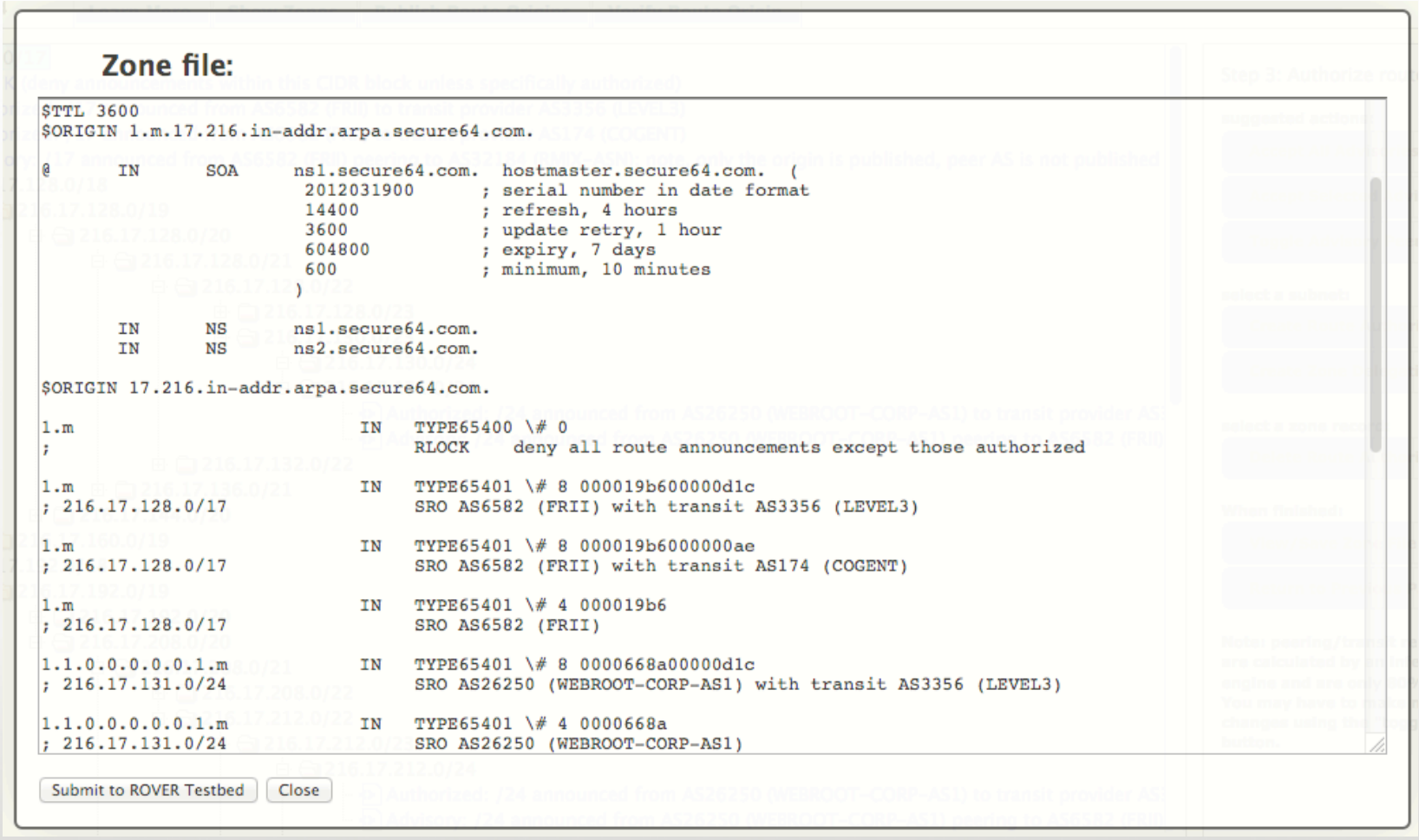
- In this case we selected all advisories.



7) Display and save the zone file

SECURE64

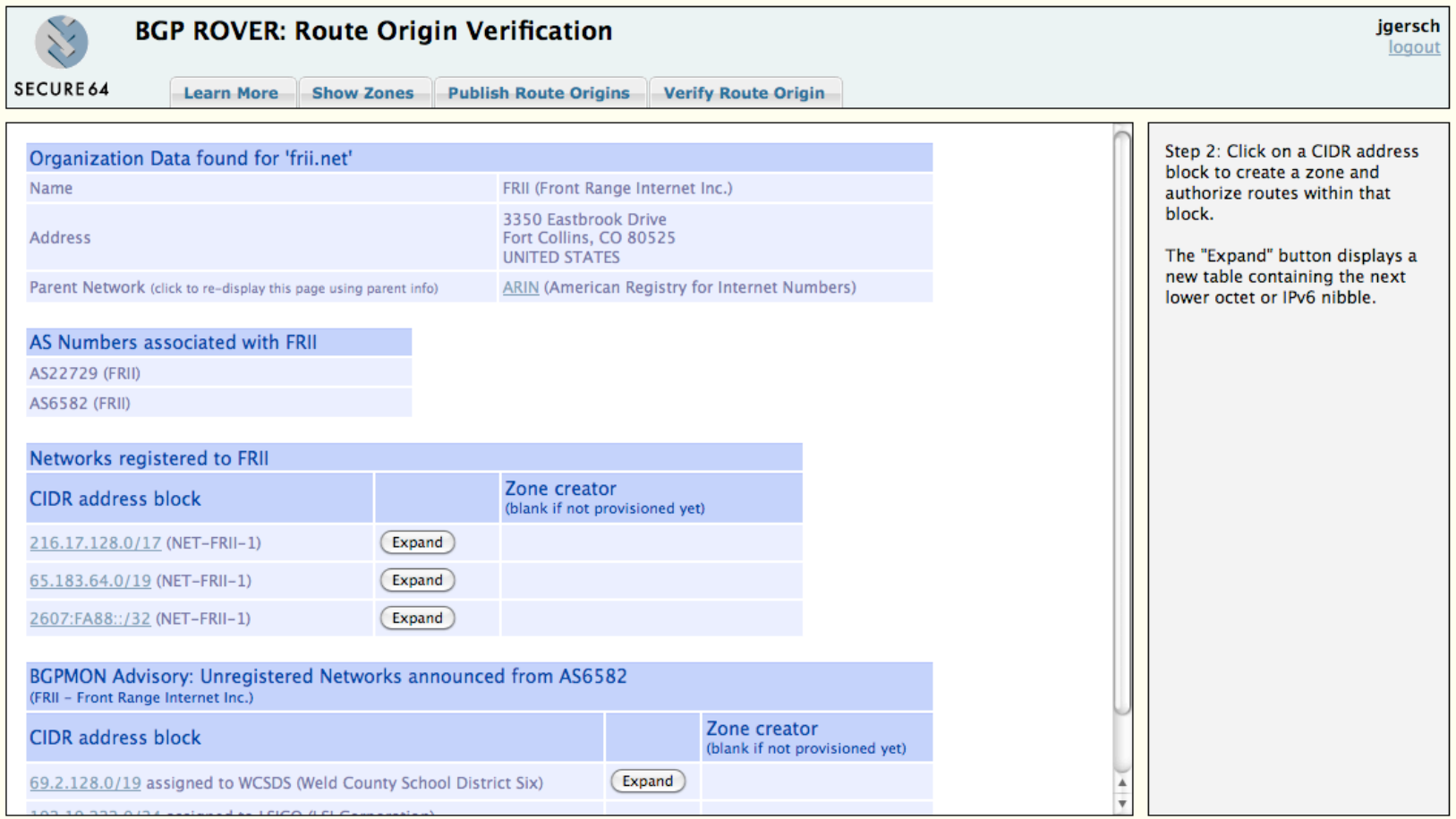
- Once submitted, it will be placed in the queue for live publication in the public shadow zone.



8) Go back and do another

SECURE64

- Provision all relevant blocks to authorize route announcements



DNS Load due to ROVER

Preliminary Investigation

SECURE64

What happens when a complete routing table needs to be verified?

- 401,970 prefixes to verify; today there is no ROVER data in the reverse-DNS.
  - Expected 2:1 ratio of queries (1 for SRO, 1 for RLOCK)
  - Actual: 754,567 queries; rate-limited to 1500 QPS
    - (fewer queries than expected due to 34k SERVFAILS and other timeouts removed need to do 2nd query for RLOCK. DNSSEC on.

	Cold Cache	Warm Cache
Cache Hits	201,868 (27%)	696,495 (92%)
Cache Misses	552,686 (73%)	58,958 (8%)
Outbound Queries	1,206,038 Fanned out to 50k servers	354,643 (6:1 due to SERVFAIL retries)

Query Fanout

54,694 authoritative servers queried

IP Address	Name	# queries
199.212.0.53	ARIN (tinnie.arin.net)	195,439
192.42.93.32	Verisign GTLD (g3.nstld.com)	43,909
199.71.0.63	ARIN (x.arin.net)	37,736
199.212.0.63	ARIN (z.arin.net)	36,118
192.5.4.1	sns-pb.isc.org - for RIPE	16,578
63.243.194.2	ISC (v.arin.net)	15,443
200.219.154.10	LACNIC (d.dns.br)	11,930
202.31.190.1	APNIC (g.dns.kr)	11,564
72.52.71.2	ISC (w.arin.net)	10,607
216.136.95.2	ns1.twtelecom.net	9,991
64.132.94.250	ns2.twtelecom.net	9,772
204.61.215.62	Woodynet (ns3.afrinic.net) (AFRINIC)	9,226
199.253.249.63	ARIN-SERVICES (t.arin.net)	9,220
202.106.196.234	APNIC (ddns2.bta.net.cn)	7,710
202.106.196.233	APNIC (ddns.bta.net.cn)	7,700
202.96.0.133	APNIC (ns.bta.net.cn)	7235
202.106.196.28	APNIC (ns2.bta.net.cn)	7235
199.212.0.73	ARIN (a.in-addr-servers.arpa)	5628
130.114.200.6	USAISC (ns03.army.mil)	3913
192.82.113.7	USAISC (ns02.army.mil)	3852
202.56.230.5	APNIC (dnsdel.mantraonline.com)	3803
140.153.43.44	USAISC (ns02.army.mil)	3798

SECURE64