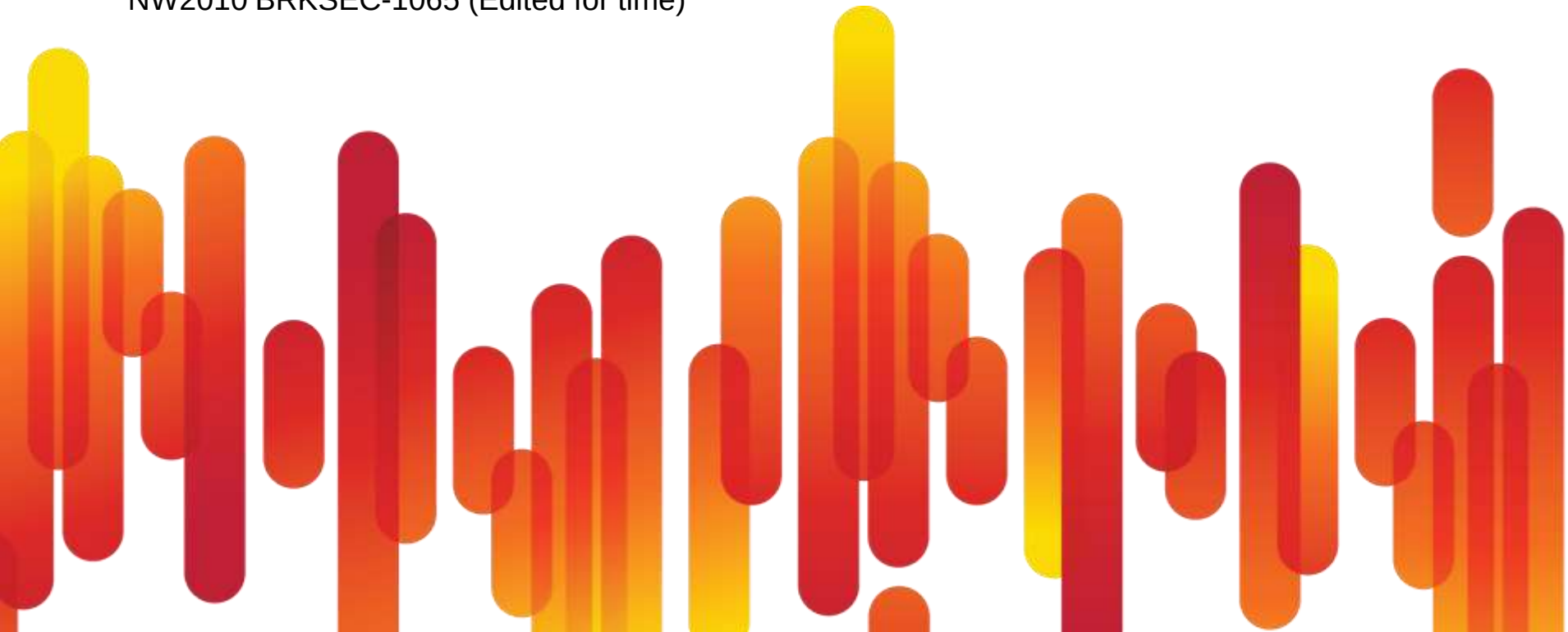
Automating Network Security Assessment

NW2010 BRKSEC-1065 (Edited for time)

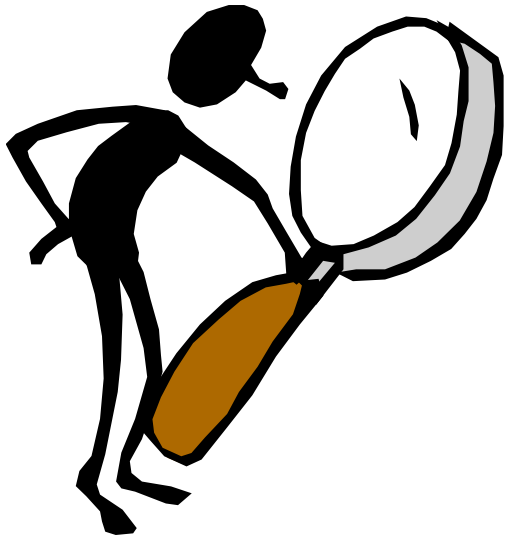


What we will cover

- Traditional approach
- What's new: Automation
- Case study: Network modeling
  - Cisco's global infrastructure
- Case study: Zone defense
  - Scrub down of border PoP's
- Case study: Defending critical assets
  - Isolating PKI
- Case study: "Surprise!"
  - Handling new infrastructure
- Case study: Managing change day to day
  - The Carnac moment

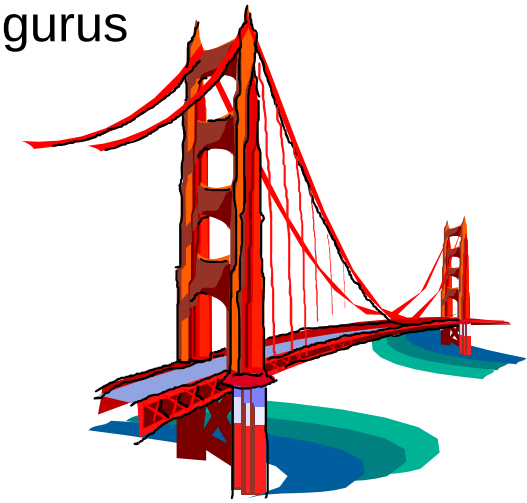
Today's network security audits

- Typically, network and hosts treated separately
- Network:
  - Elbow grease and eye strain
    - Gather configs; print configs; read configs
    - Similar to proof-reading the phone book
- Hosts:
  - Level 1: Leave the admins to patch
    - Problem: hope is not a strategy
  - Level 2: Scan for unpatched systems
    - Problem: more data than you can handle
  - Level 3: Drive cleanup based on risk
    - Problem: prioritization easier said than done



What needs to change

- Typical teams:
  - Host exploit gurus
    - Working without network or business context
  - A few network specialists
    - Critical “how’s & why’s” in the heads of a few gurus
- Audit treadmill
  - Like painting more bridges than you have crews
- Need to:
  - Finish each audit in less time
  - Increase accuracy
  - Capture the rules for next time
  - Integrate across specialties – put issues in context



Why network assessment is different

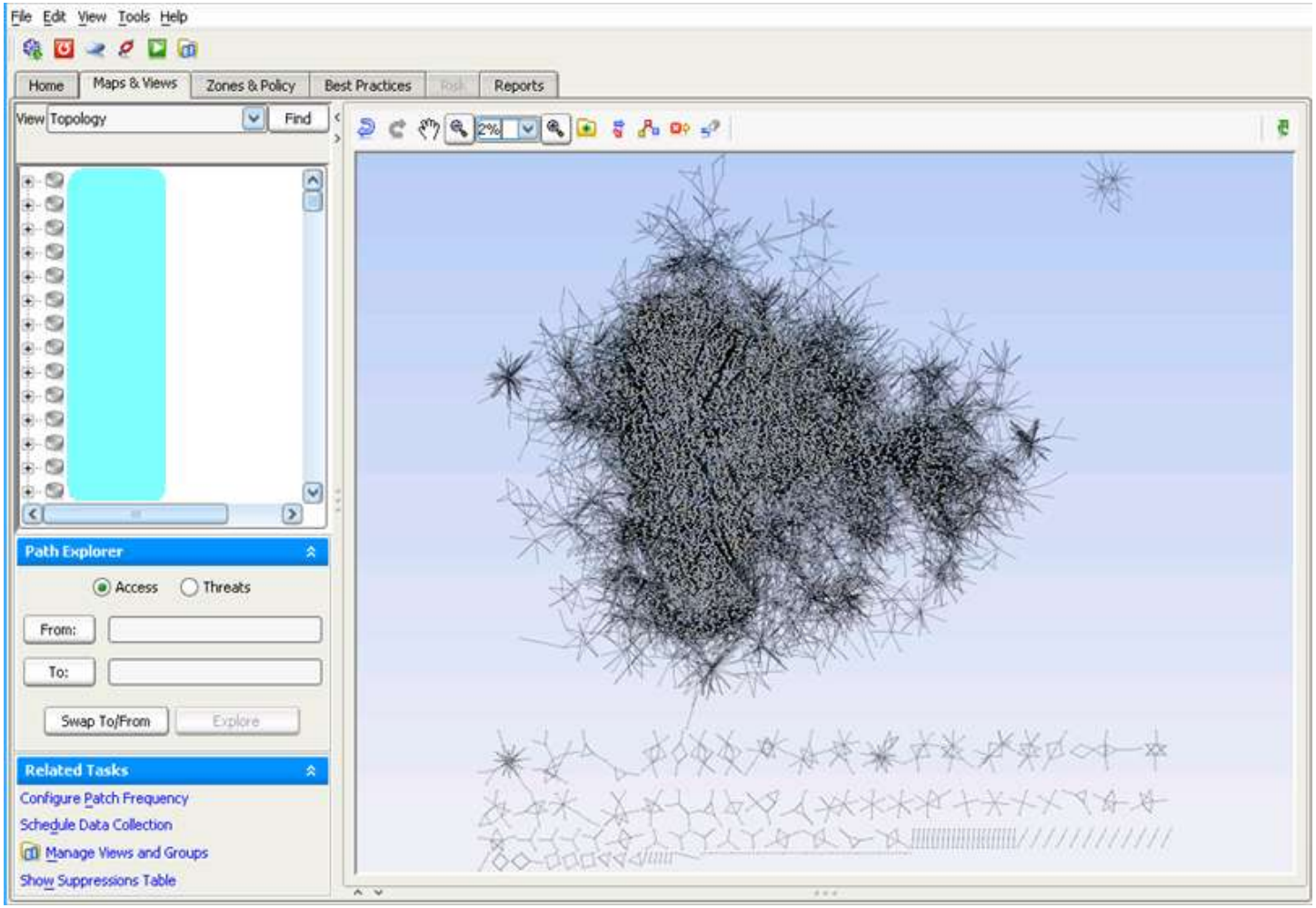
You can't detect a route around the firewall by reading the firewall

Case study: “Project Atlas”

- Objective:
  - Map the entire global Cisco environment
  - Review major site interconnections
  - Audit access to sensitive locations
- Resources:
  - Installed RedSeal software
  - Two weeks
  - 27,000 configuration files

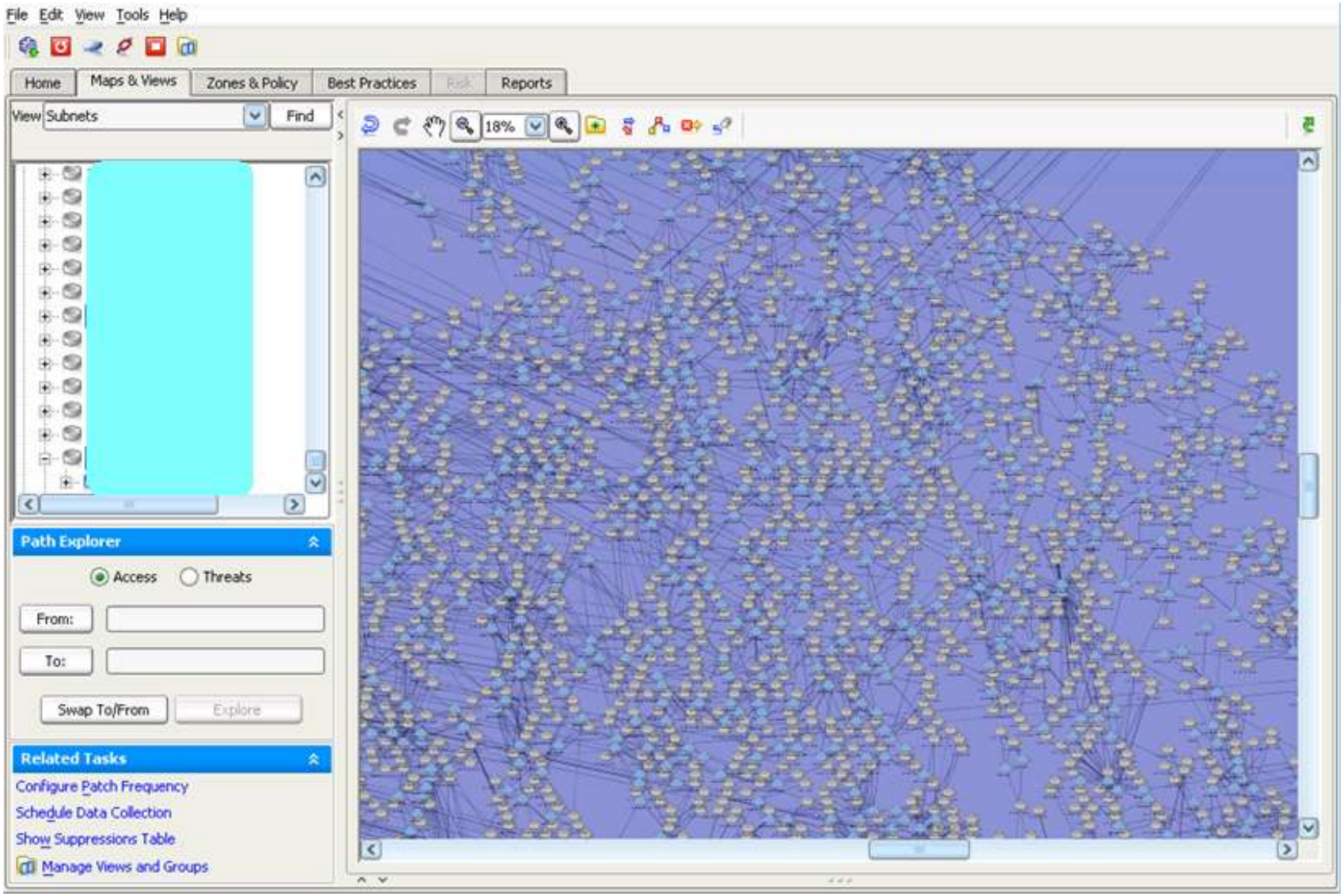
- Originally on ~$5K server (quad core, 32G RAM)
- Now running on Cisco UCS – much faster!

Raw network (aka “The Bug Splat”)

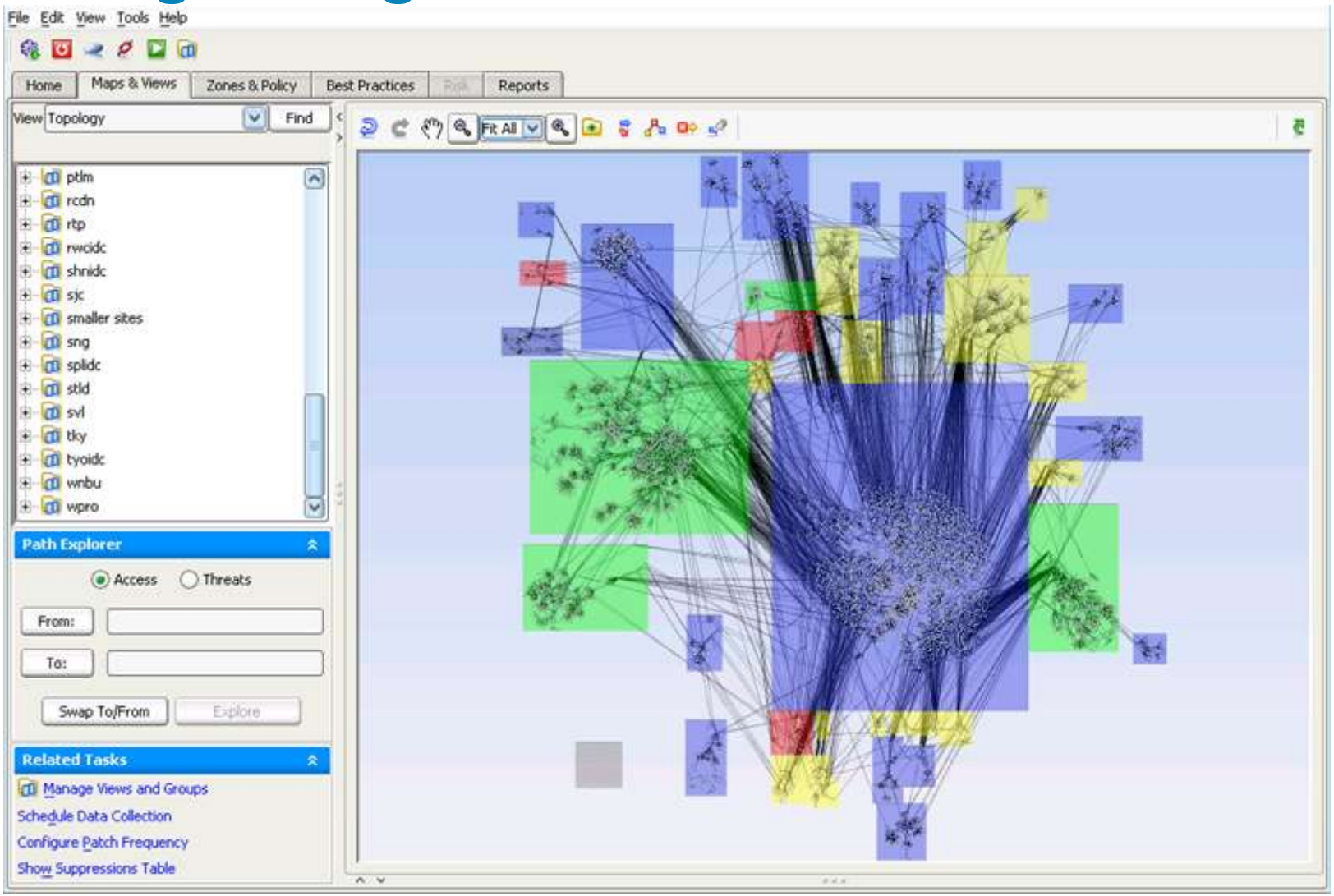


Lesson #1: You need a config repository

Complexity level is high

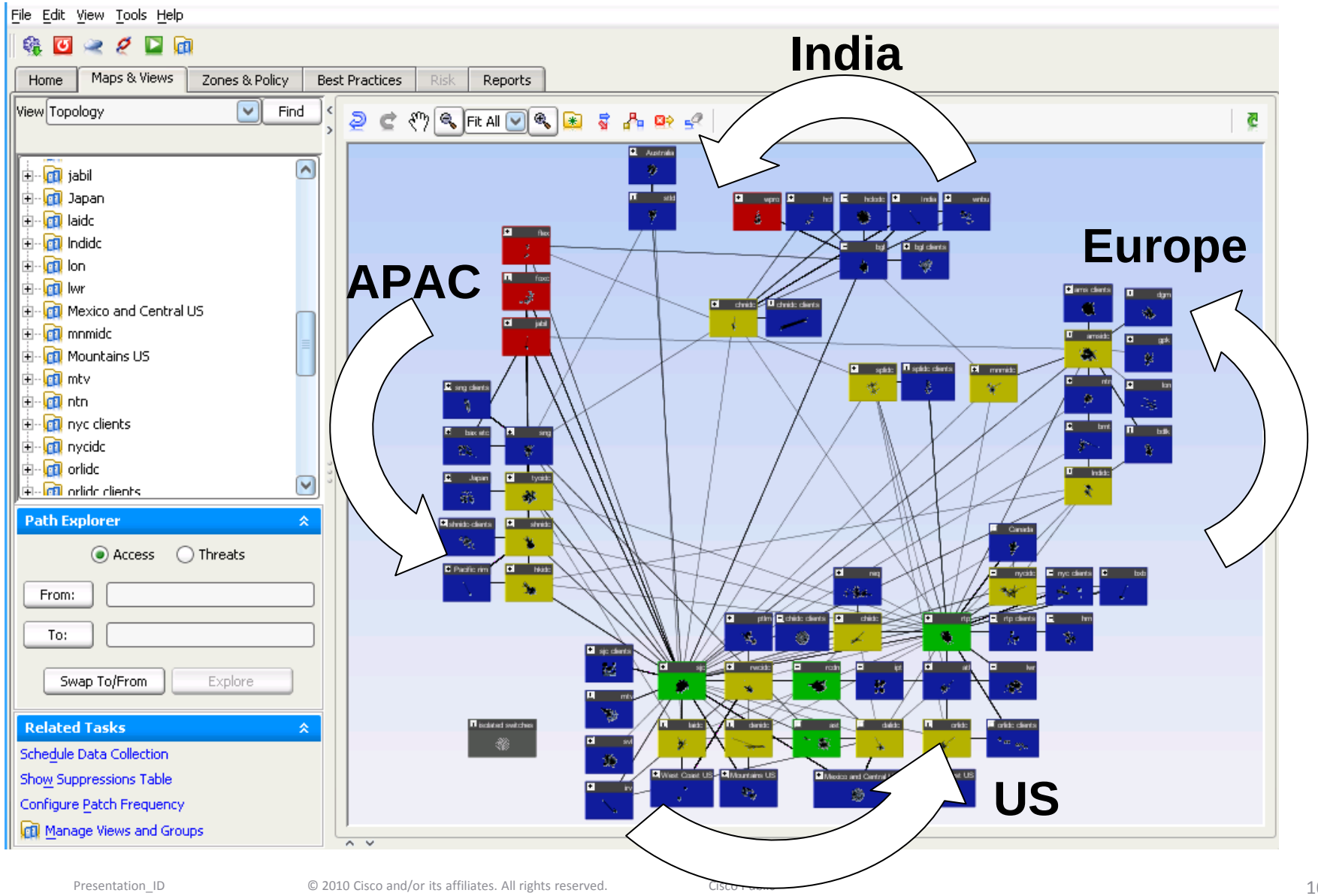


Organizing Cisco's worldwide network

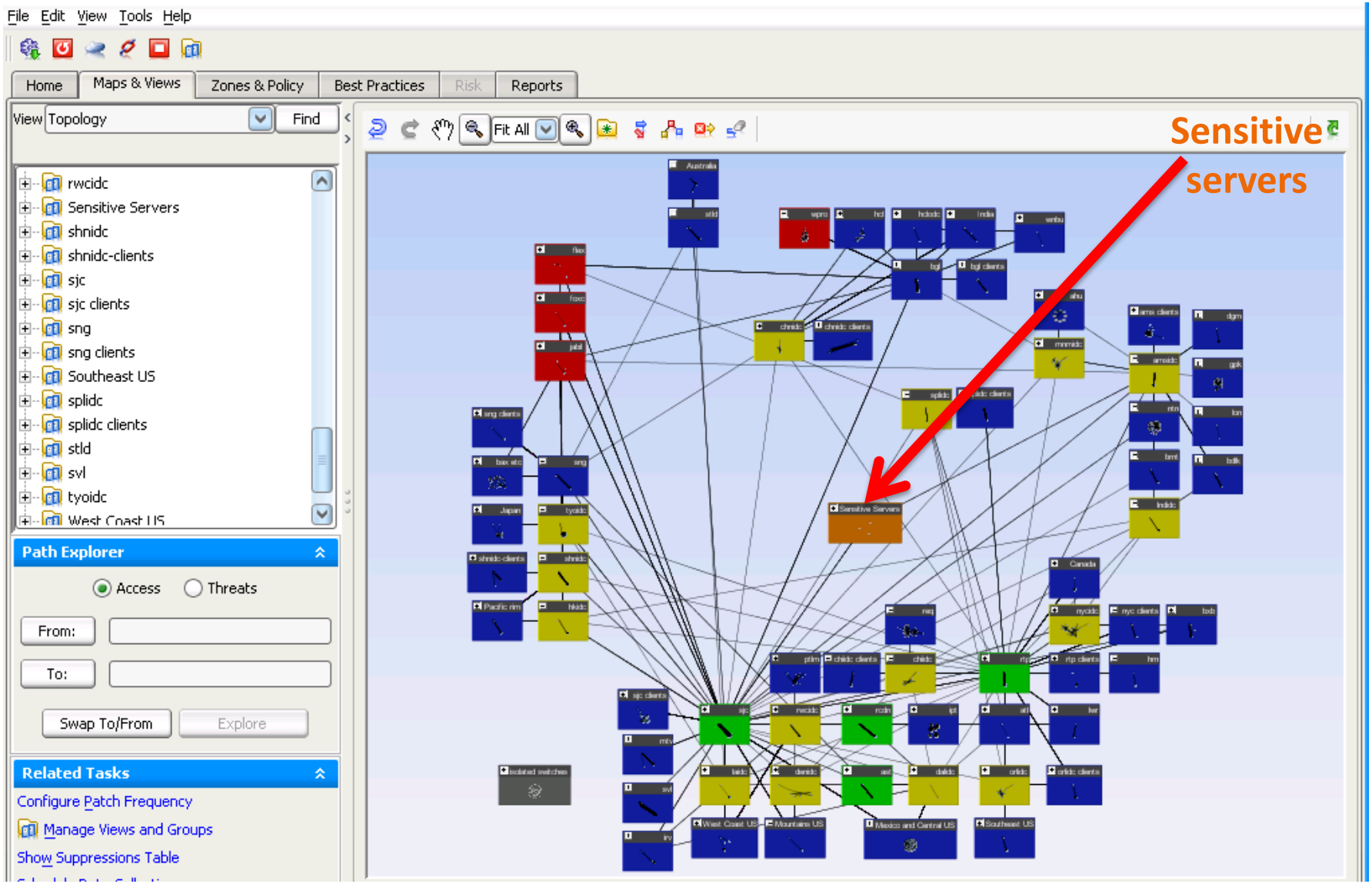


Lesson #2: Naming conventions are your friend

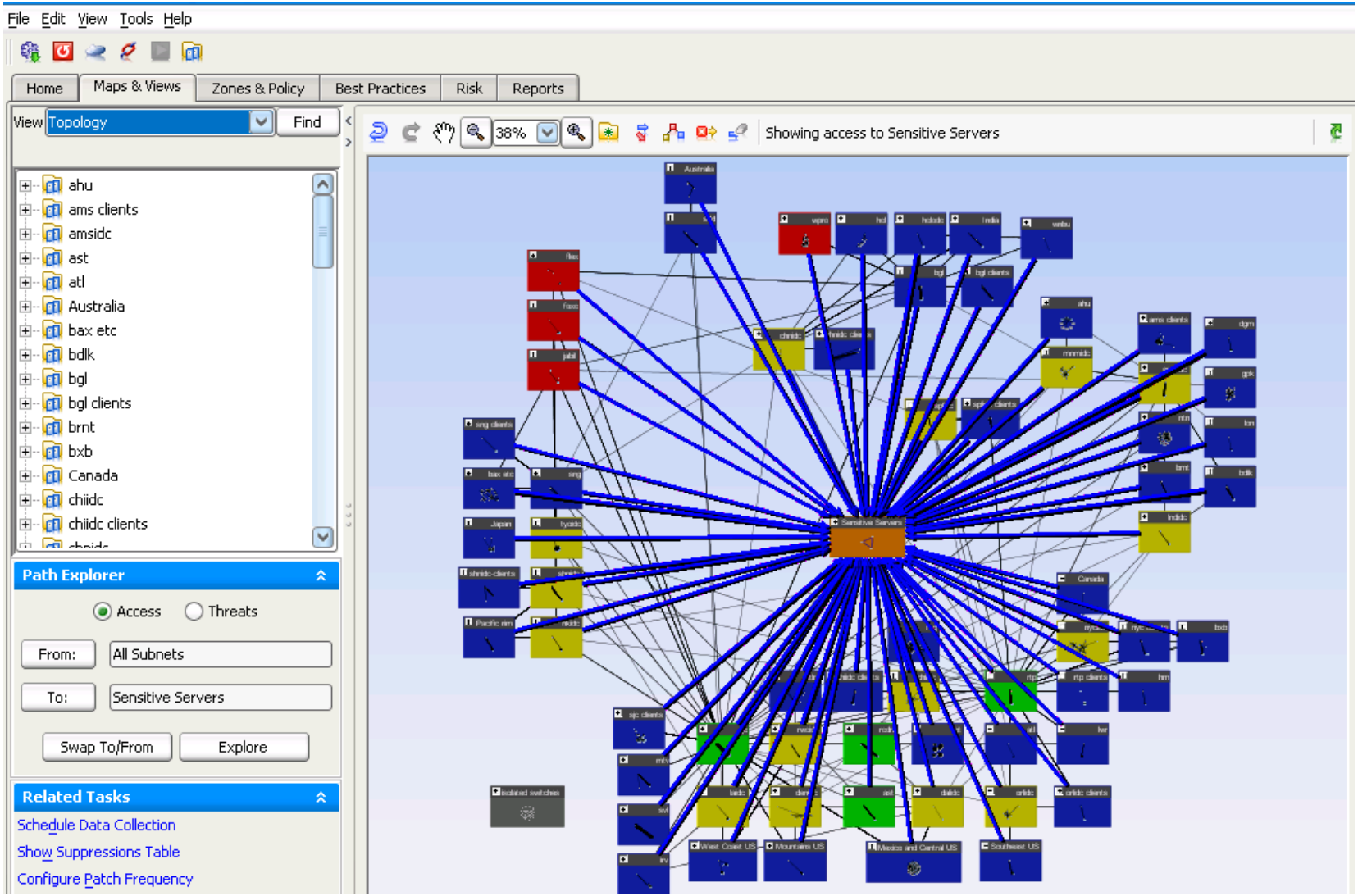
Final “circumpolar” zoned view



Connectivity to six sensitive servers

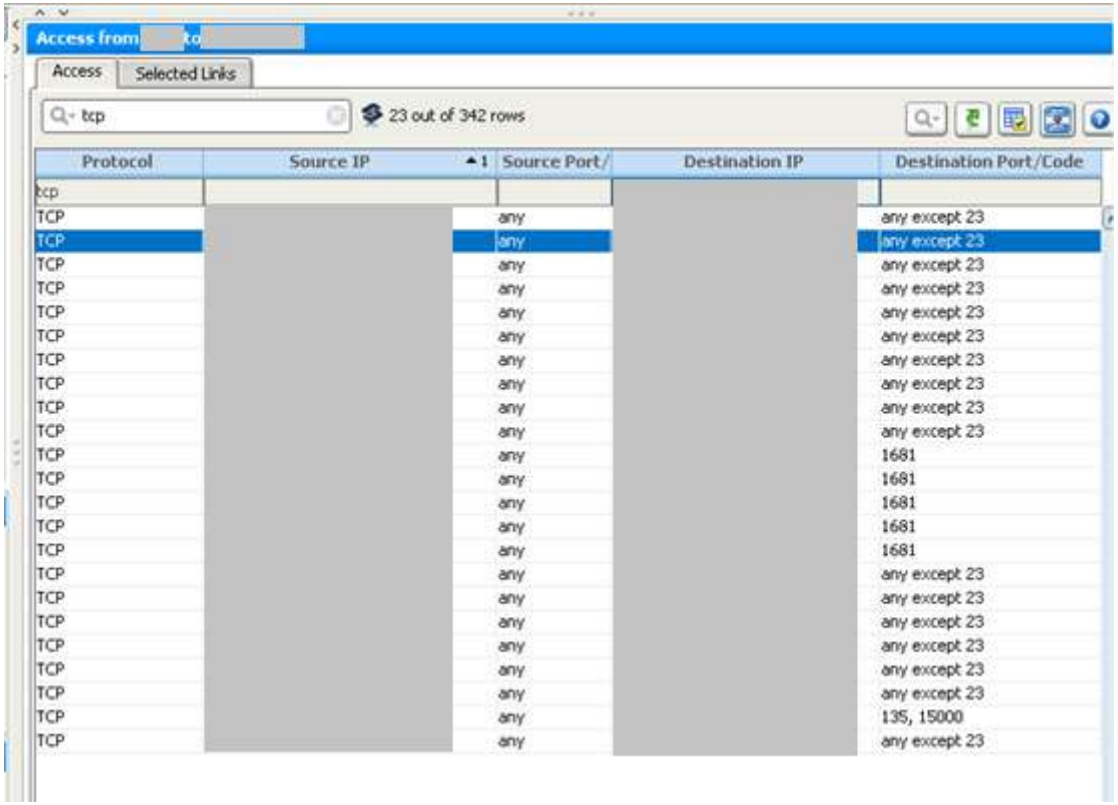


Automatic calculation of connectivity



Lesson #3: Pictures easily explain difficult concepts

Access specifics – “Is it just ping?”



- Detailed drill-down from one blue arrow
- Well, at least we blocked telnet

  (Specifics hidden, for obvious reasons)

Before vs. After

- Before:

  No way to visualize global infrastructure

- After:

  Map of record in an “Atlas”

  Has become a working platform for further projects

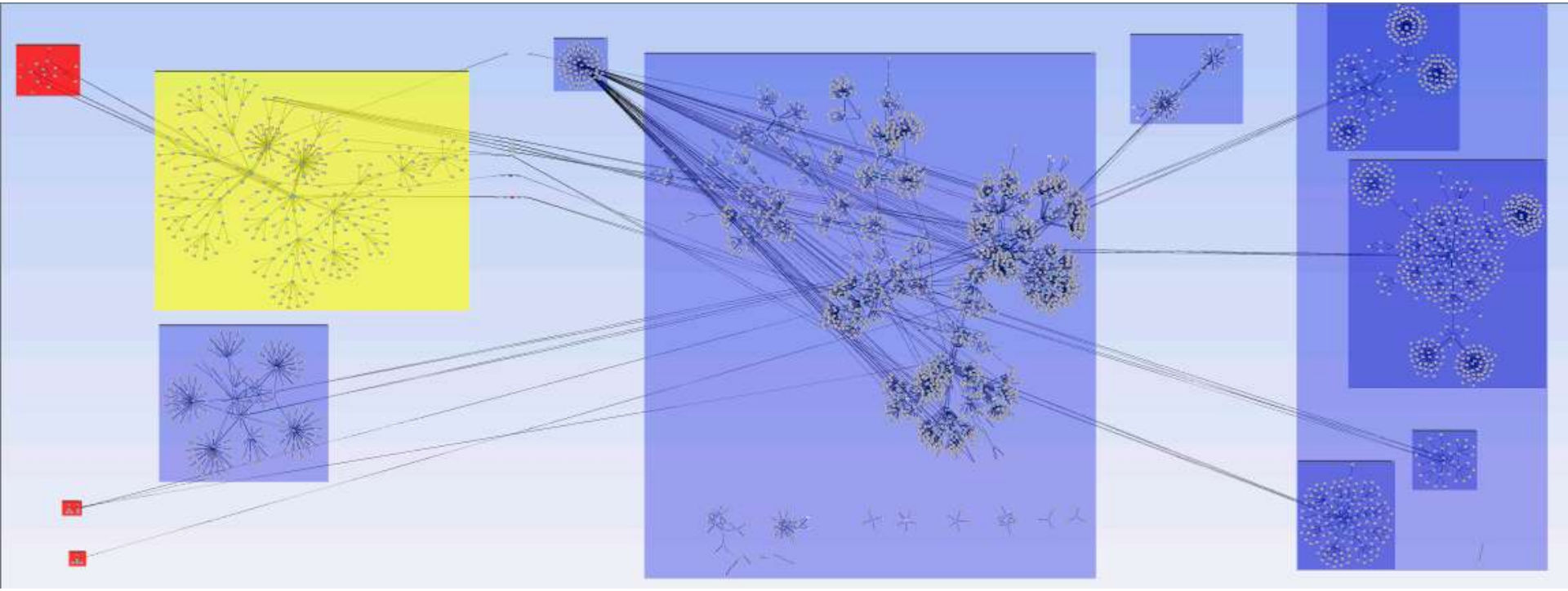
  Graphics to explain security issues to non-experts

Case Study: Zone defense

- Cisco has 15 major PoP's for external connections
- Typical manual assessment: 90 days per PoP
- Target:
  1. Build map
  2. Record major zones
     - Internet, DMZ, Inside, Labs, etc
  3. Analyze for Best Practice violations
  4. Add host vulnerabilities from scans
  5. Run penetration test

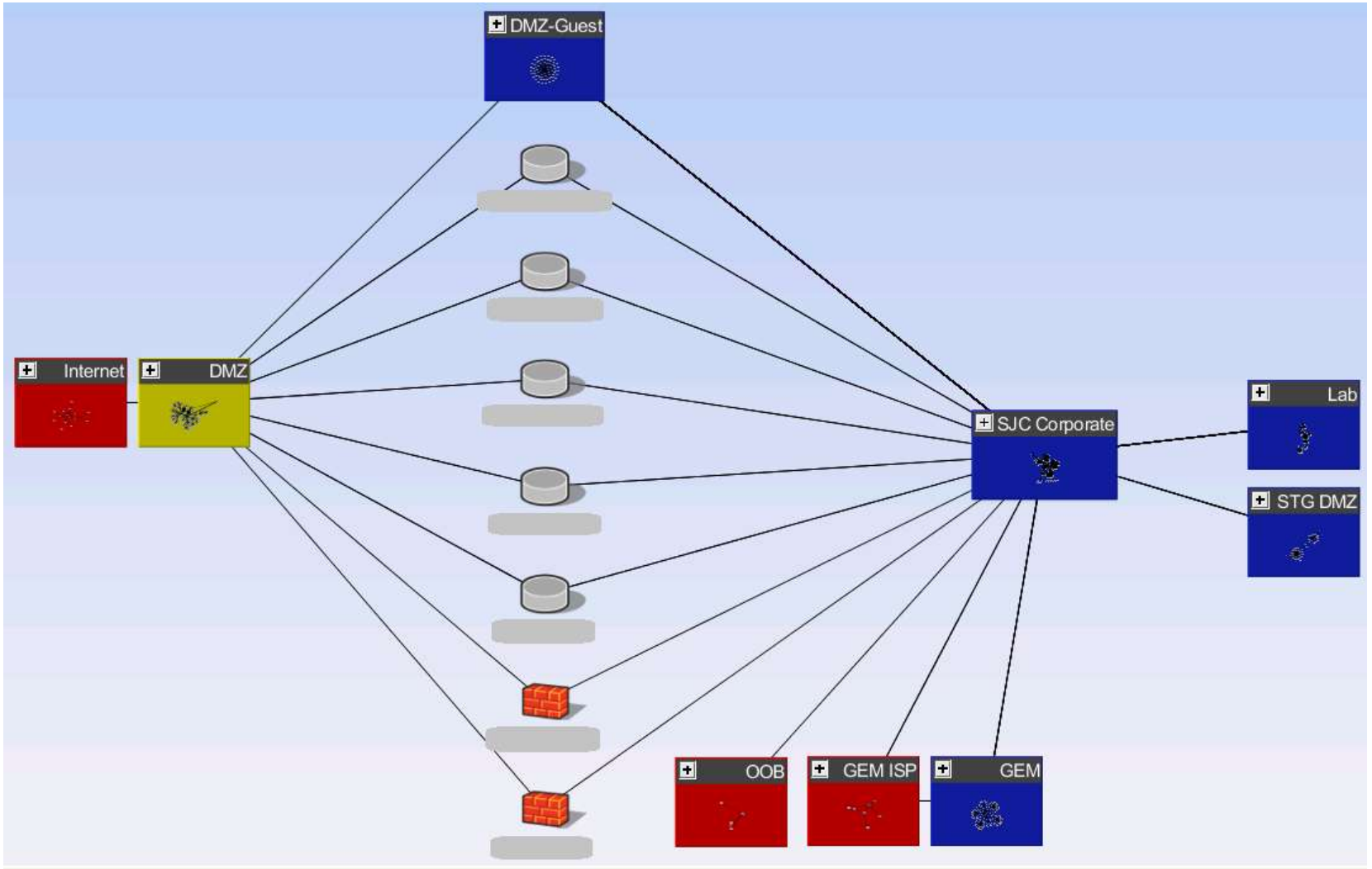
San Jose Campus Network Map

- Map of one PoP
- Zoning done “semi-automatically”



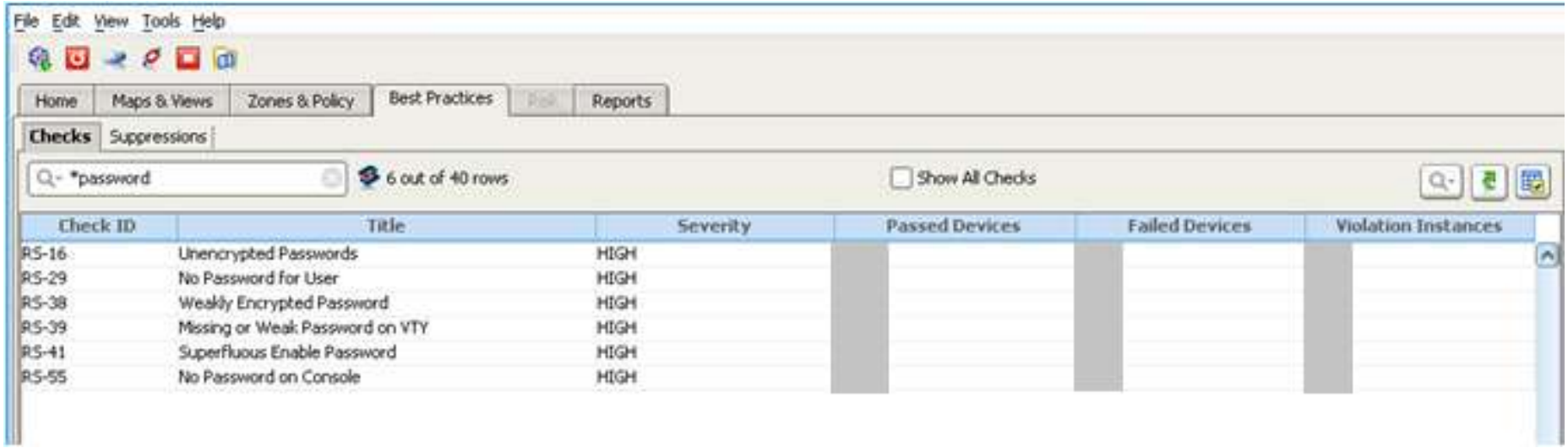
Internet DMZ Main Site Labs

San Jose Campus Network Map



Example of Best Practice Checks

- Automatic evaluation of 100+ rules
- Weak or missing passwords, redundant rules, etc

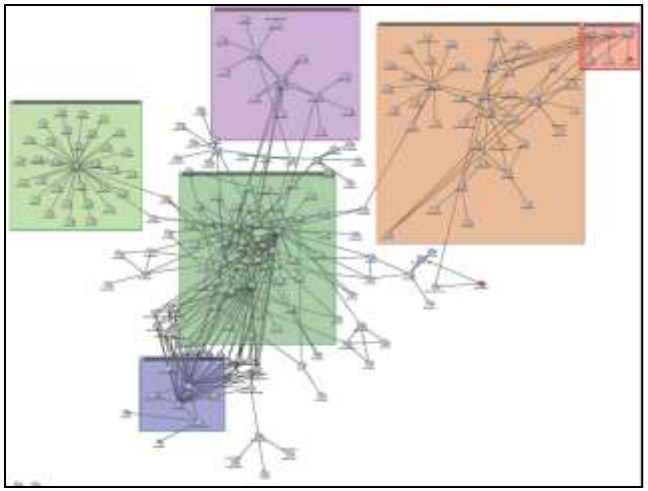


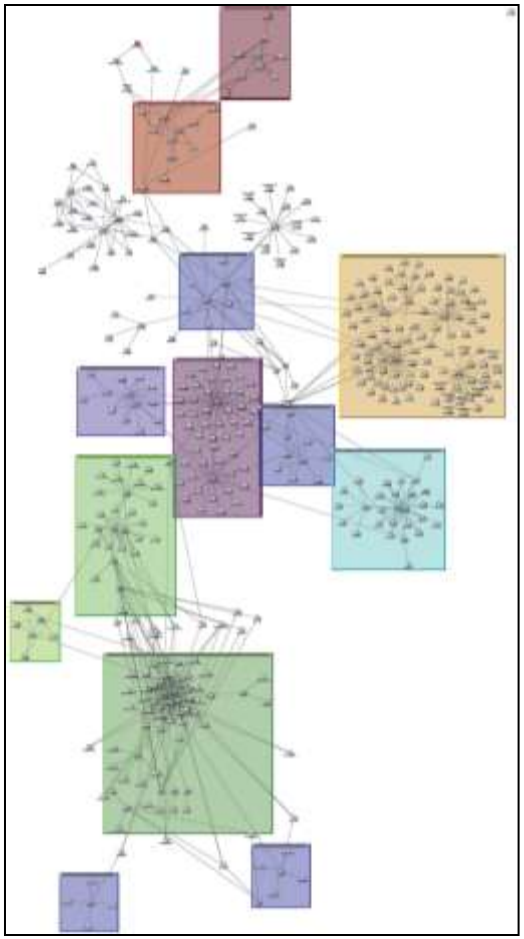
- Unlike rolling stones, changing networks gather moss …

Lesson #4: Networks gather 'cruft'

More sample maps

- 9 PoP maps built out & zoned in one morning
- Export to Visio and PDF

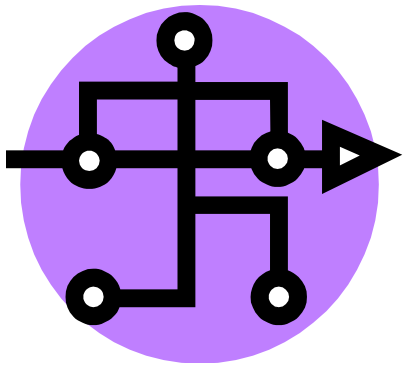




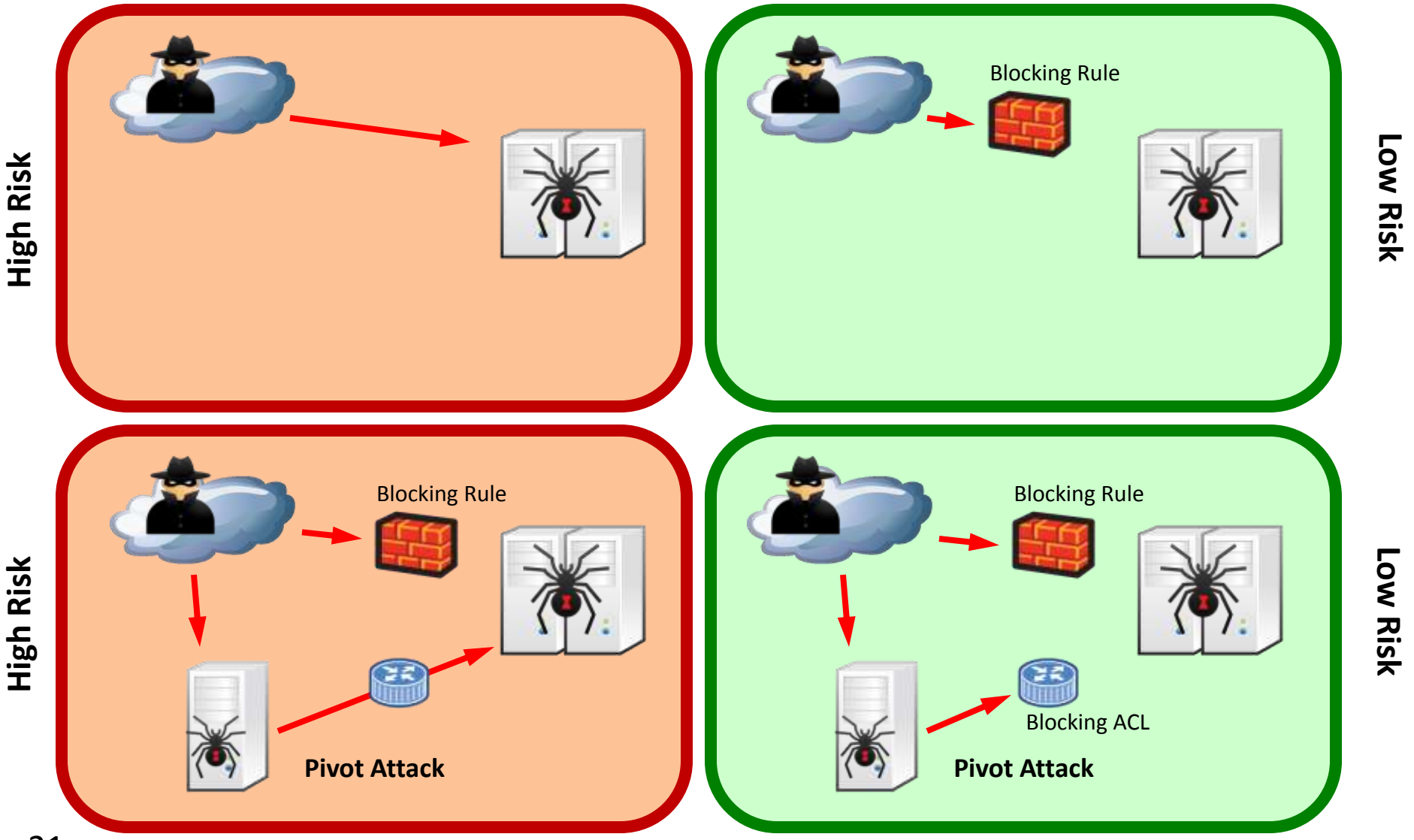
Lesson #5: 'Regular' people can do this.

Offline penetration testing

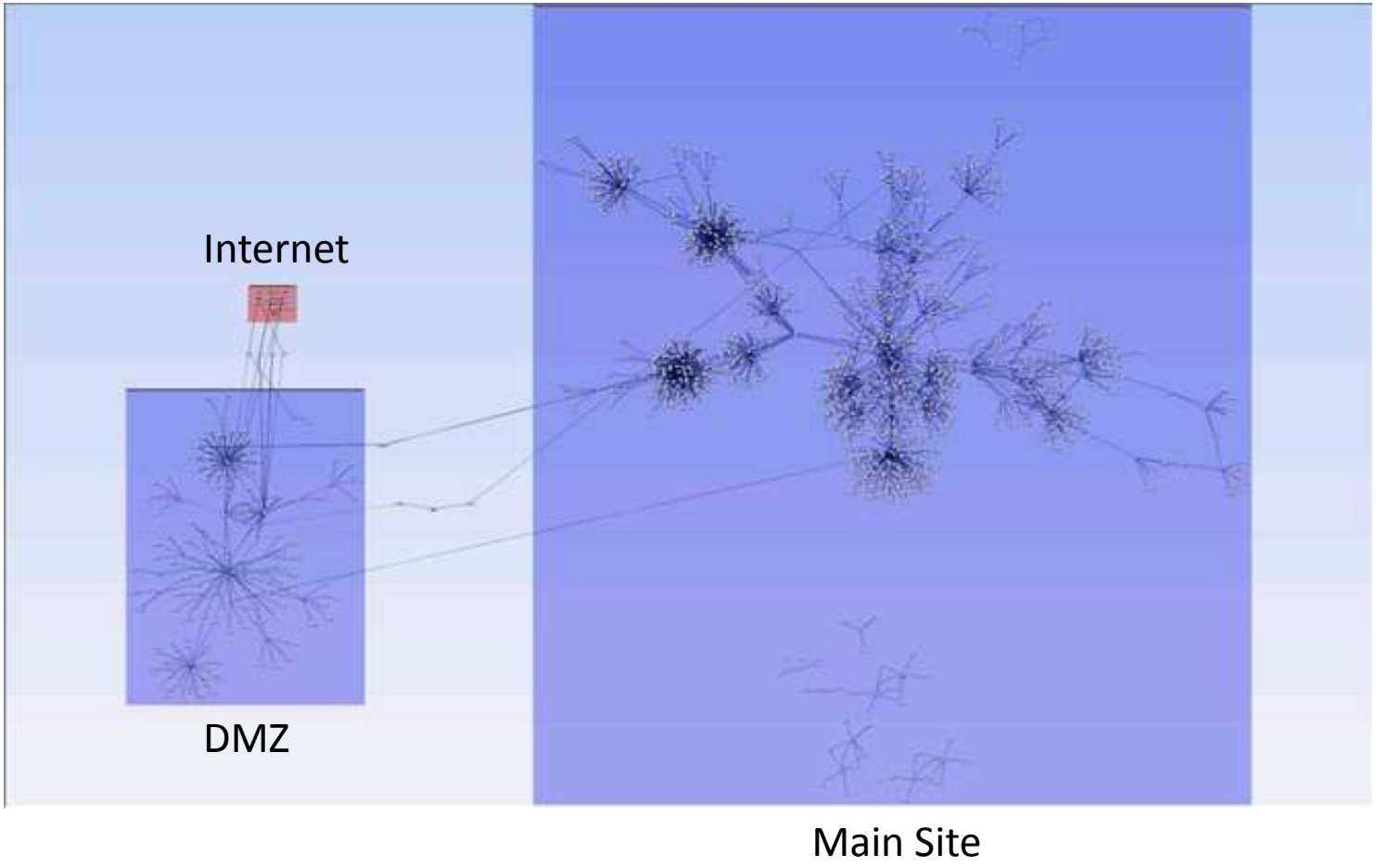
- Next level of analysis is penetration testing
- Combine network map with host scans
- Add access calculation
- Software automatically evaluates attack paths
- Identify high risk defensive weaknesses



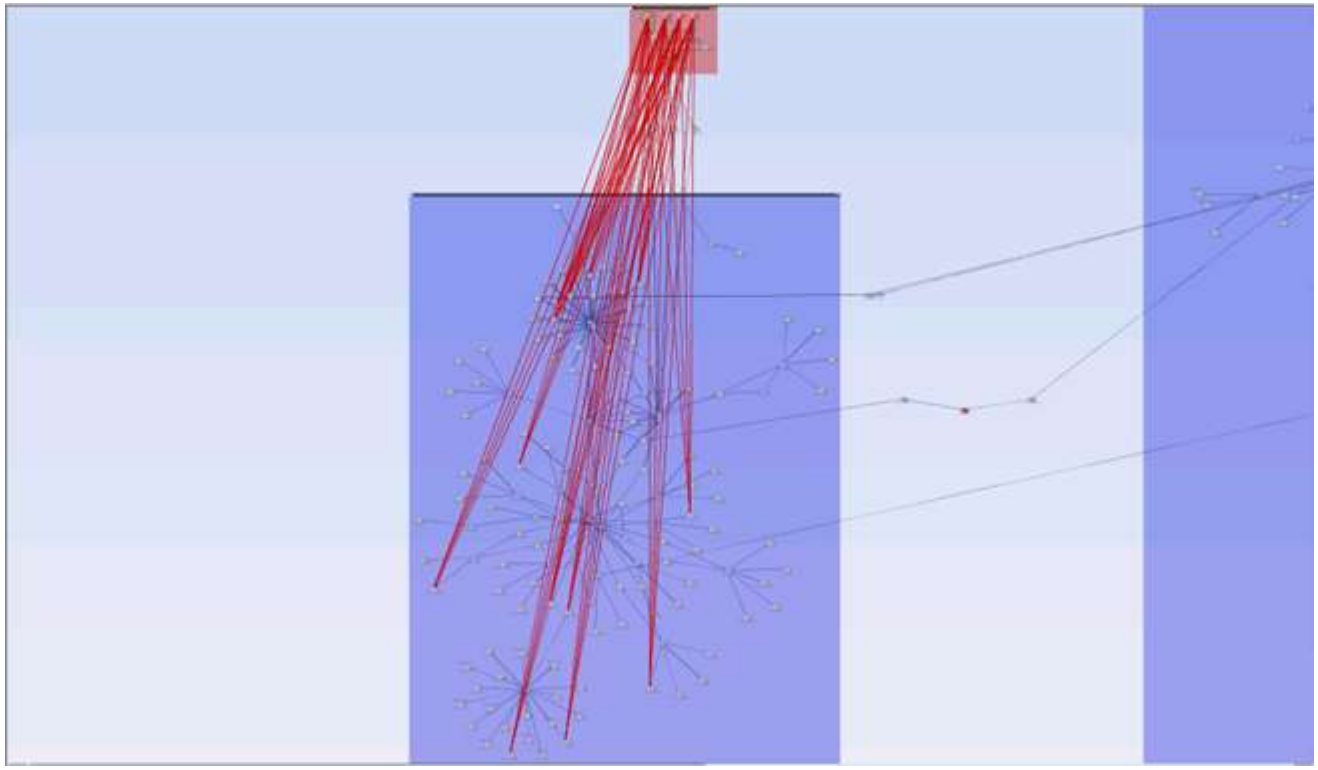
Risk from Network-Based Attacks



Sample attack chain – Before



Step 1 – Vulnerabilities exposed in DMZ

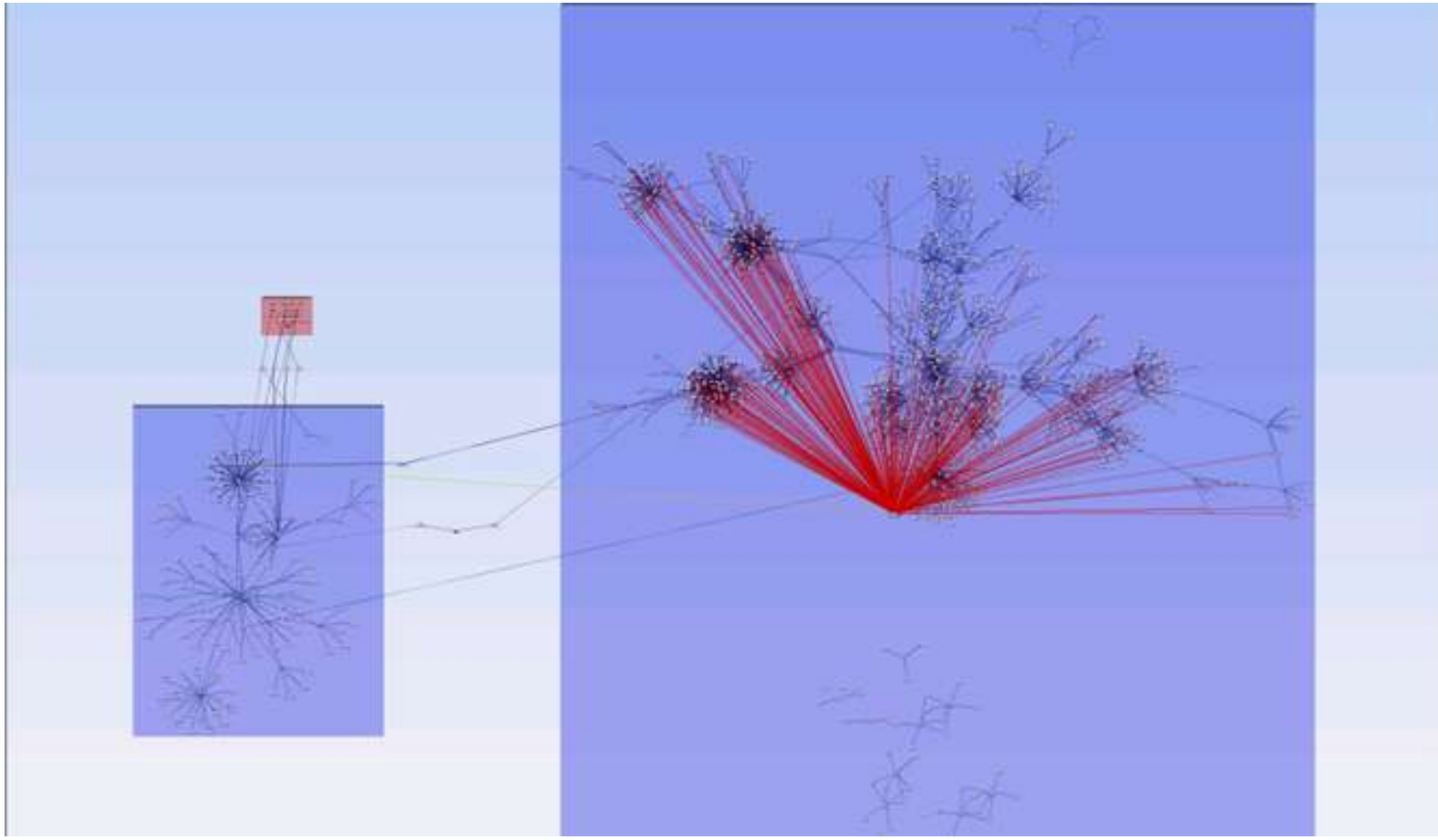


- Attackers can reach these Internet-facing servers

Step 2 – Some attack paths sneak in

- Just a few pivot attacks are possible

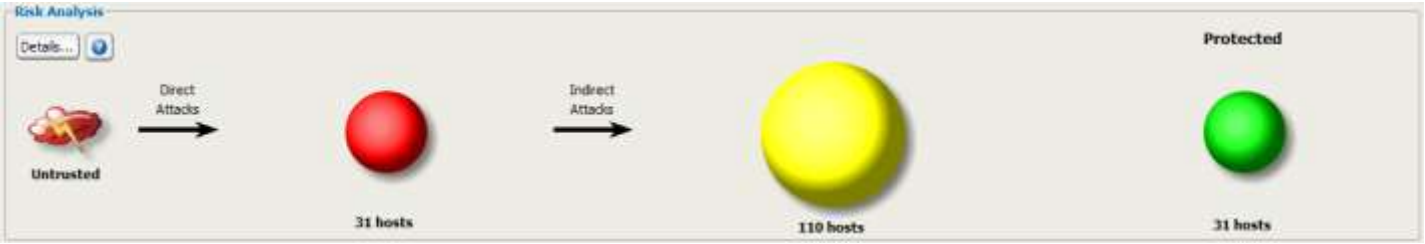
Step 3 – Attack fans out



- An attacker can get in if they find this before you fix it

Penetration test results

- Sample result:



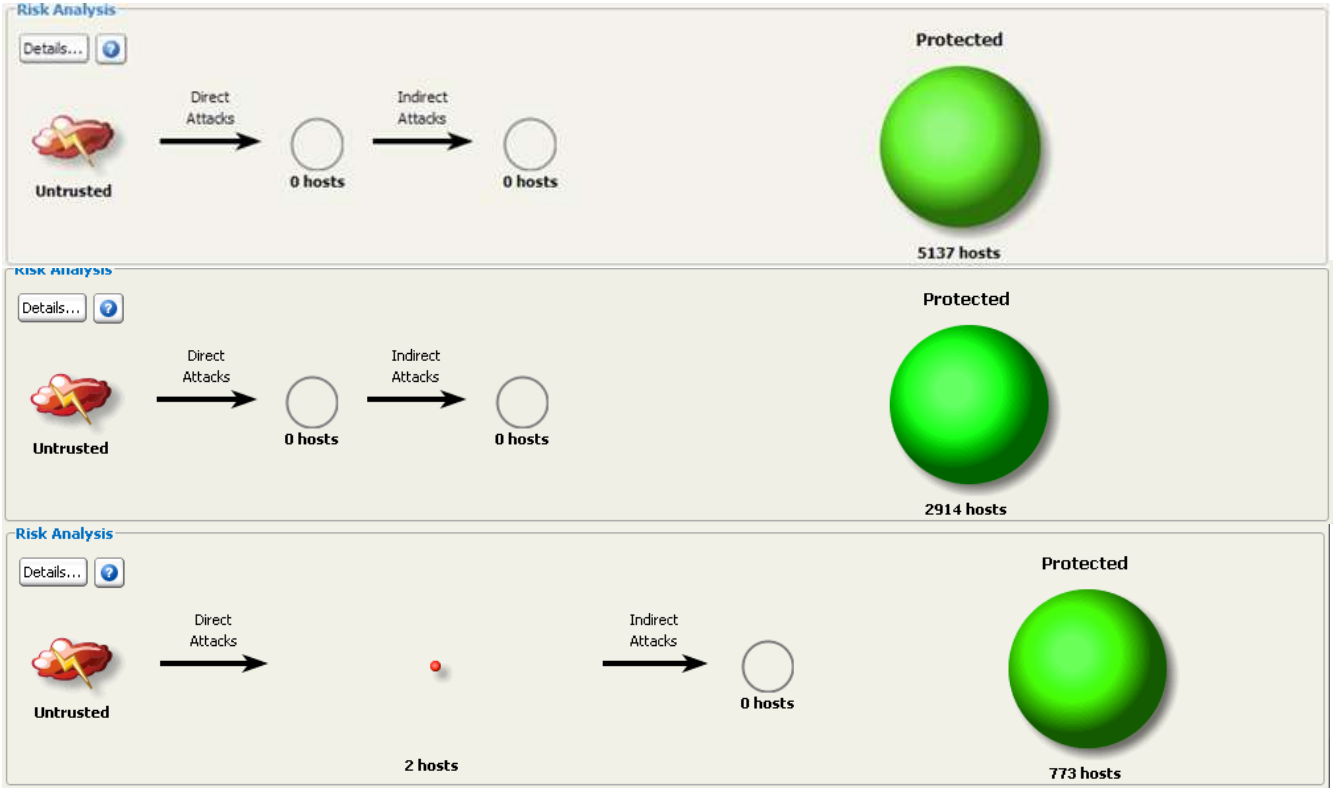
External attackers can reach red hosts

Then pivot to attack yellow hosts

But no attack combination reached green hosts

Results of recent PoP analysis

- Three PoP's out of nine analyzed
- These are very clean – small attack surface



Before vs. After

- Before:
  - Each PoP audit took 90 days
  - Did not consider host vulnerability data
- After:
  - Team recently executed 9 PoP audits in one day
  - Integrated assessment
    - Network configuration analysis
    - Zoned map
    - Host vulnerabilities
    - Attack path analysis
  - Bonus: map and results re-usable on next visit

Lesson #6: Network data + Vuln data + Attack path = GOLD

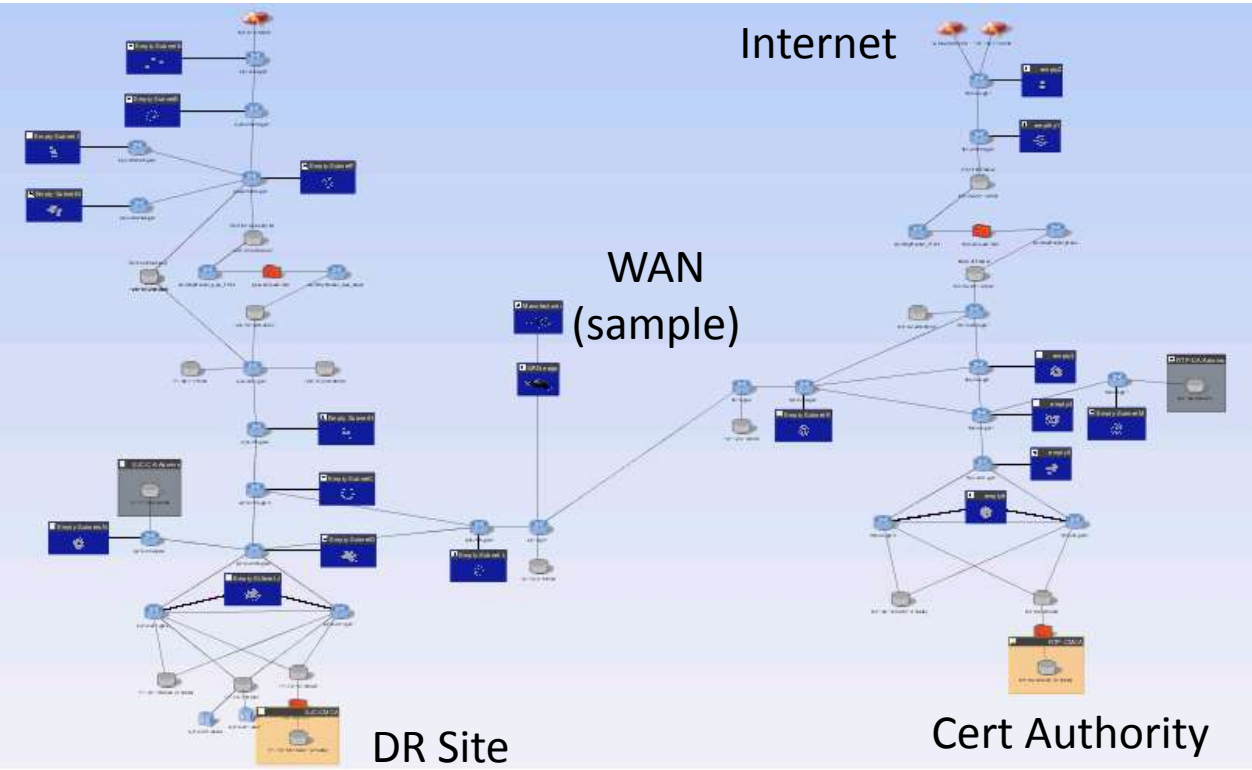
Case Study: Defending critical assets

- PoP audits work outside in
  - Broad scope, hunting major gaps
  - Problem: lots and lots of access to review
  - Can’t quickly capture all rules for all incoming access
  - Some assets deserve focused attention
- For critical assets, work inside out
  - Start from known target
  - Limit scope, increase focus
  - Continuous re-assessment

Distributed public key infrastructure

- Main site, plus disaster recovery site

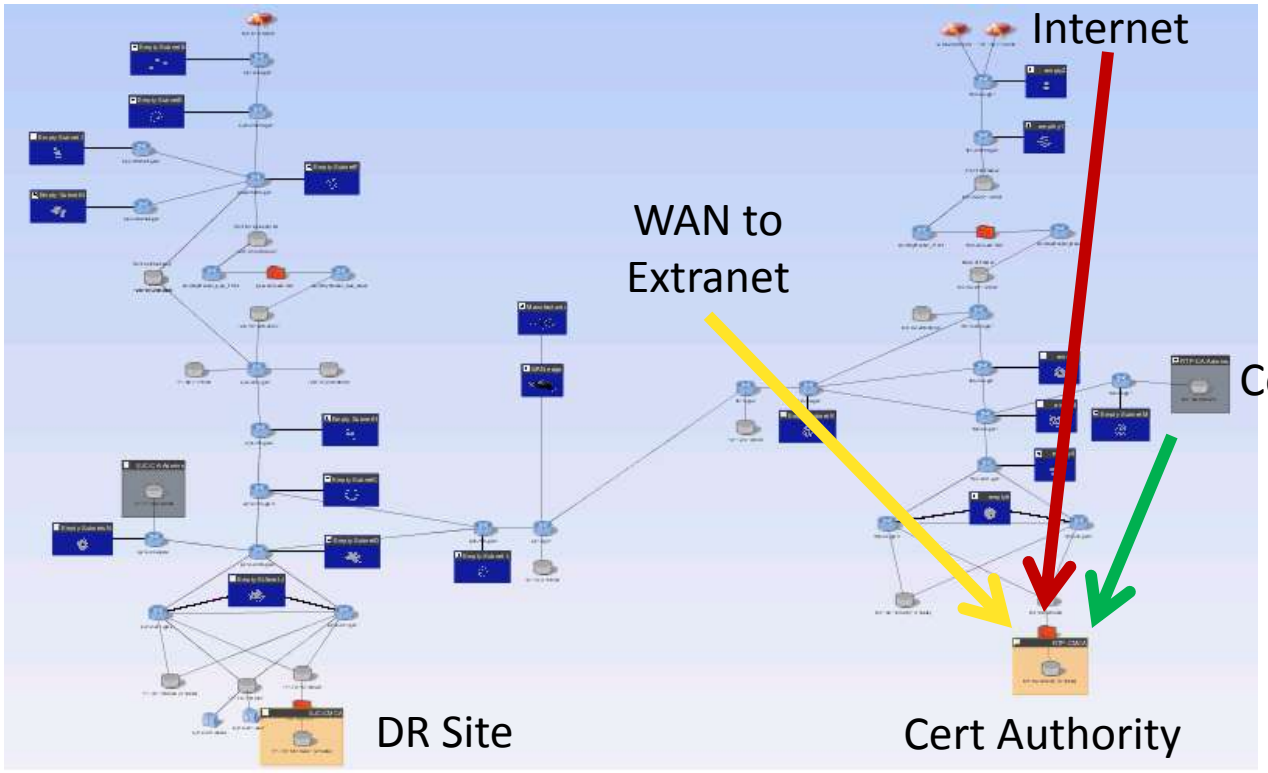
  Building the “crossbar” was easy – we sampled from Atlas



Lesson #7: A reference atlas is your friend

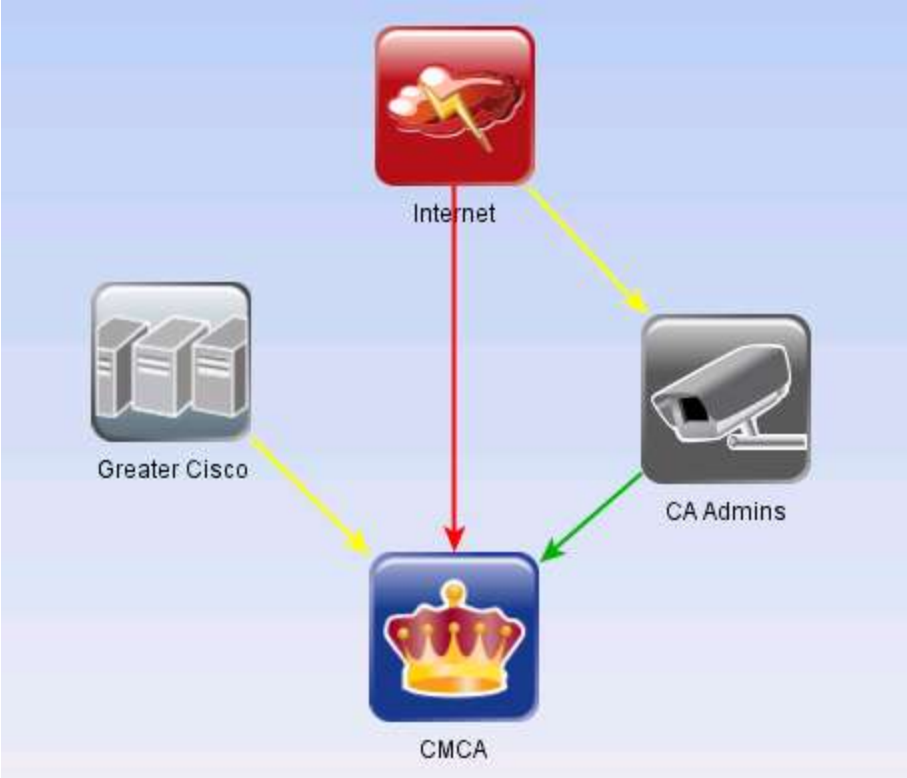
Distributed public key infrastructure

- Access strictly controlled
  - Untrusted 3rd party manufacturers need to request certs
  - Only cert admins should have general access



Capture high level rules

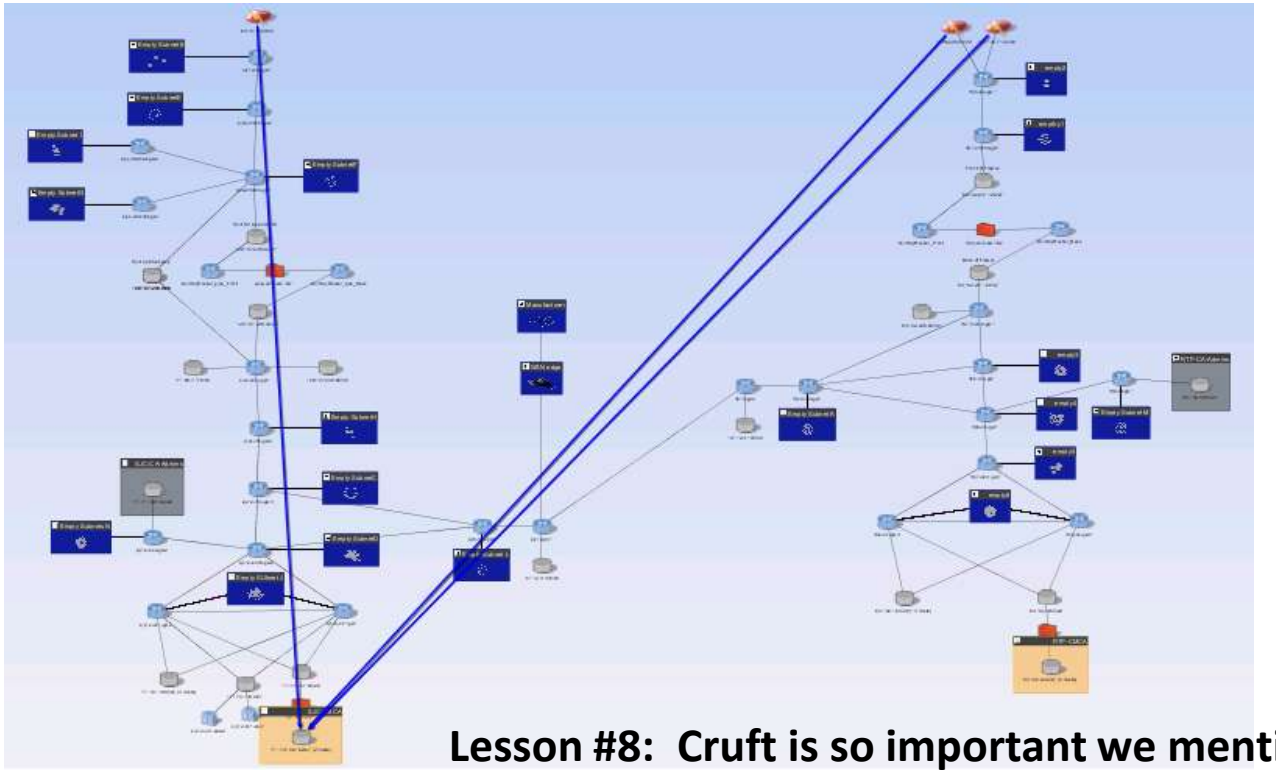
- Capture relationships of major zones
- Arrows show there is some unwanted access



Investigate unexpected access

- Note: no flow into primary
- Only DR site had unexpected Internet access

  Even that was for limited sources, but still unexpected



Lesson #8: Cruft is so important we mention it twice

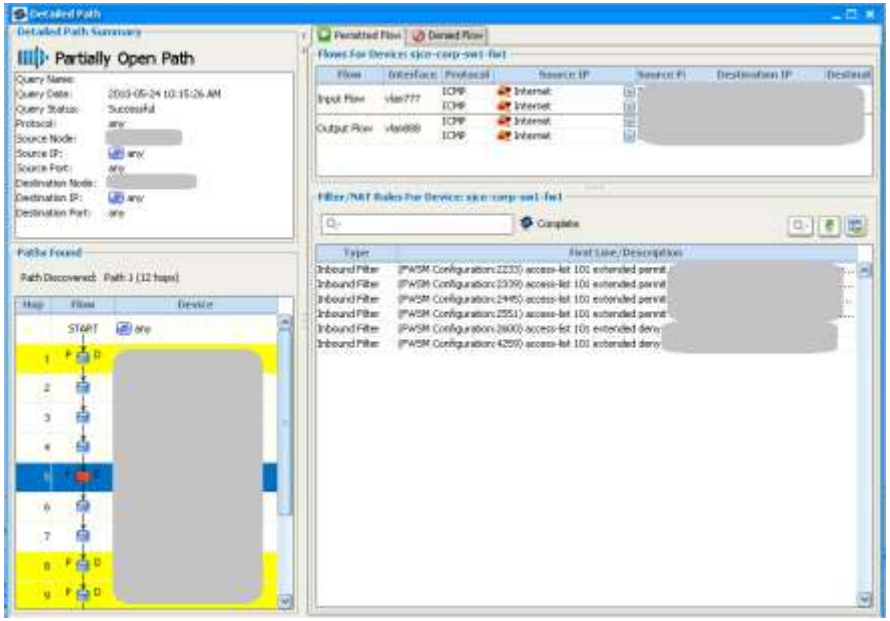
Remove unwanted access

- Drill down to detailed path for unexpected access
- Identify exact cause

  In this case, an out of date group definition on firewall

Access Found

"Subway Map" showing path



Flow through one hop

Specific rules

Before vs. After

- Before:

  Important details buried in large, complex network

- After:

  Focused rule-set to test defenses

  Built out over 2 days

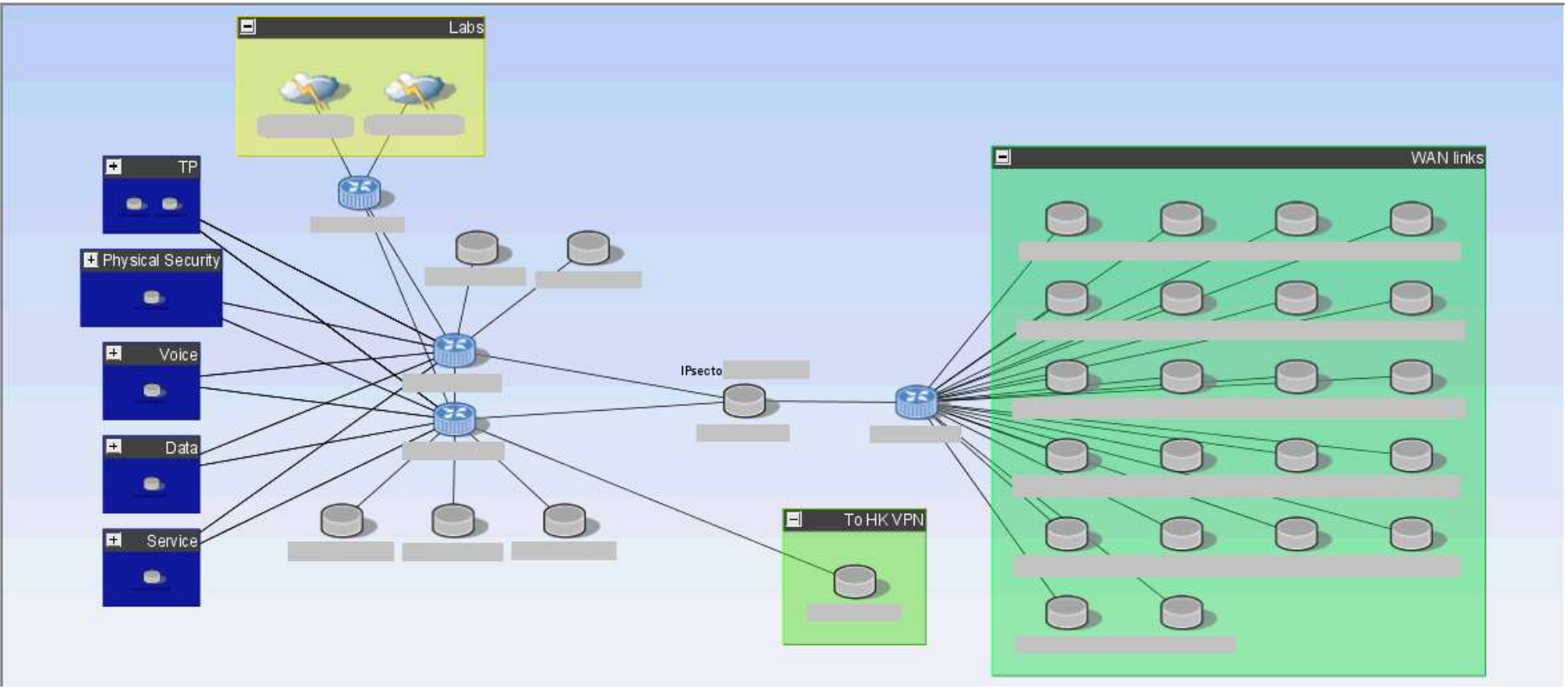
  Daily re-evaluation as network changes come and go

  Automatic mail summarizing status

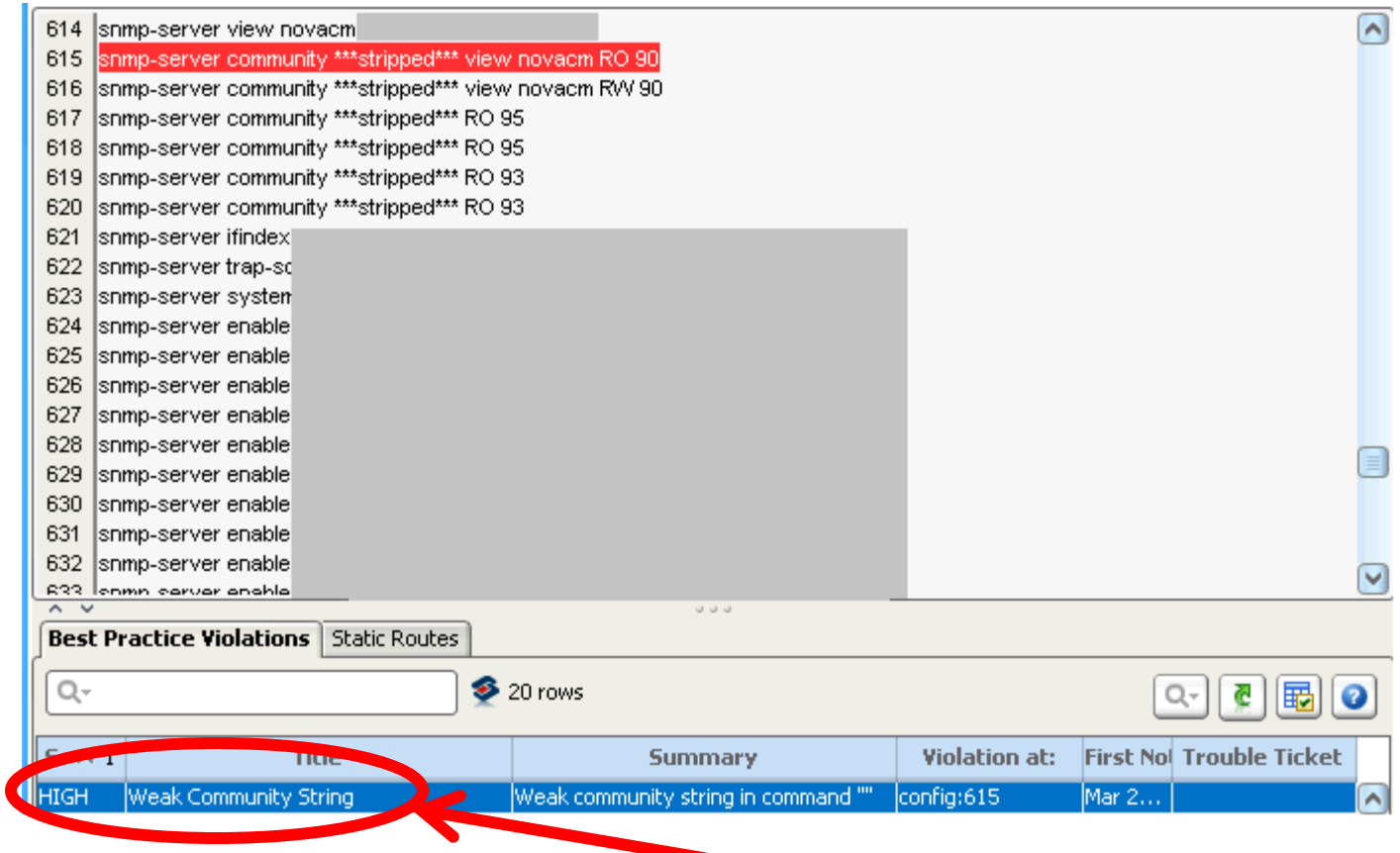
Case Study: “Surprise!”

- Ad hoc network support
- Sudden addition of complete network to secure
- M&A, or in this case, short-lived Expo network
- Requires very rapid assessment
- Continuous tracking during high visibility phase
  - Until end of expo, or for M&A, integration into normal ops

China Expo Center Topology



Best Practice?



- Weak Community String

Best Practice Checks

Examples of Best Practice Violations

RedSeal

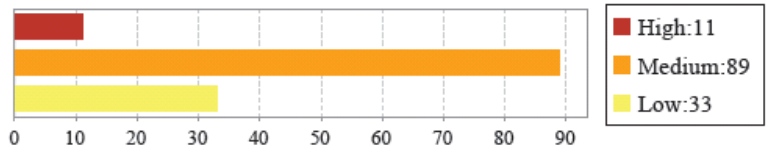
Description: Identify which Best Practice rules are violated, and where.

Best Practice checks, sorted by name, are shown with itemizations of instances of violations of the check, sorted by frequency of violation.

User Name: uiadmin

Parameters: View = Primary Capability, Minimum selected severity = low, Max violations per folder to show = 10, Sort violation by = name, Sort checks by = frequency,

Severity Summary



Summary Data

	Devices in this report
Total Network Devices	6
Avg. Violations / Device	22.2
Unique Violations	13
Total Violation Instances	133

Non-contiguous Wildcard — Severity: low — Check ID: RS-21

Description: A wildcard in the configuration references a set of non-contiguous IP addresses. This is frequently done by mistake—0.0.0.240, which addresses 16 non-contiguous hosts, might easily get set instead of the intended 0.0.0.15 wildcard. (If the ***redundant-security-rule*** test has also failed for the same block of addresses, fix the non-contiguous problem first. It may be producing a false-positive ***redundant-rule*** warning.)

Remediation: If not intentional, the wildcard should be replaced with a contiguous wildcard.

Primary Capability > Router — 1 of 5 network devices have at least 1 issue

Device	Summary	Violation ID	First Noticed	Last Noticed
[redacted]	Non-contiguous wildcard found Line 2673 permit tcp any [redacted] 0.0.0.32 eq www	119	Mar 26 2010	Mar 26 2010
	Non-contiguous wildcard found Line 2790 permit ip any [redacted] 0.0.0.128	124	Mar 26 2010	Mar 26 2010
	Non-contiguous wildcard found Line 2827 permit ip any [redacted] 0.0.0.128	126	Mar 26 2010	Mar 26 2010

Inverted Mask in Access List — Severity: medium — Check ID: RS-92

Description: An inverted subnet mask was found in an access list rule. An inverted mask can inflate a range of 255 addresses to as many as 16.7 million, causing severe performance degradation of the RedSeal analysis engine. RedSeal ignores rules containing inverted masks, since they are almost certainly configuration errors.

A common mistake when configuring access lists is to specify the mask using ***do care*** bits when the platform expects ***don't care*** bits. That is, for example, to match hosts of the form 172.16.1.*, the correct form for IOS and Foundry is 172.16.1.0 0.0.0.255. An operator may sometimes enter 172.16.1.0 255.255.255.0 by mistake. Since the mask uses ***don't care*** bits, this actually matches hosts of the form *.*.*.0. Also note that the router can remove any values covered by ***don't care*** bits, so the incorrect entry will show up as 0.0.0.0 255.255.255.0 instead of what the operator typed originally. Permitting every address that ends in zero is almost certainly not the intended filter, since *.*.*.0 specifies 16.7 million distinct permissible addresses.

Remediation: Verify the original intent of this line and replace with the correct host and mask.

Mar 29, 2010, 11:34 PM (EDT)

Page 1 of 7

Lesson #9: Computers are better at reading phone books than you are. Get over it.

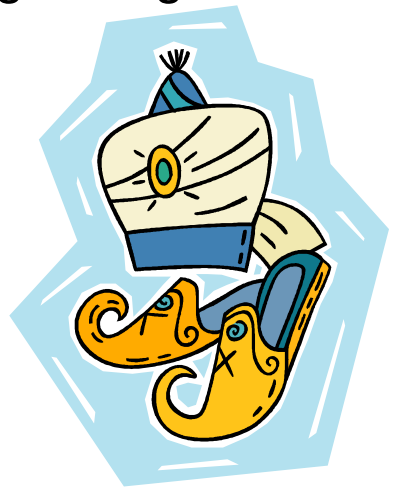
Before vs. After

- Before:
  - Very hard to keep up with new projects
  - Availability wins – move fast, bring it up, move on
  - Security gaps don't cause phone calls, availability gaps do
- After:
  - Assessments at the speed of business
  - Automation is key
  - Use rules with expiry dates to stop accumulation of cruft

Case Study: Managing daily change

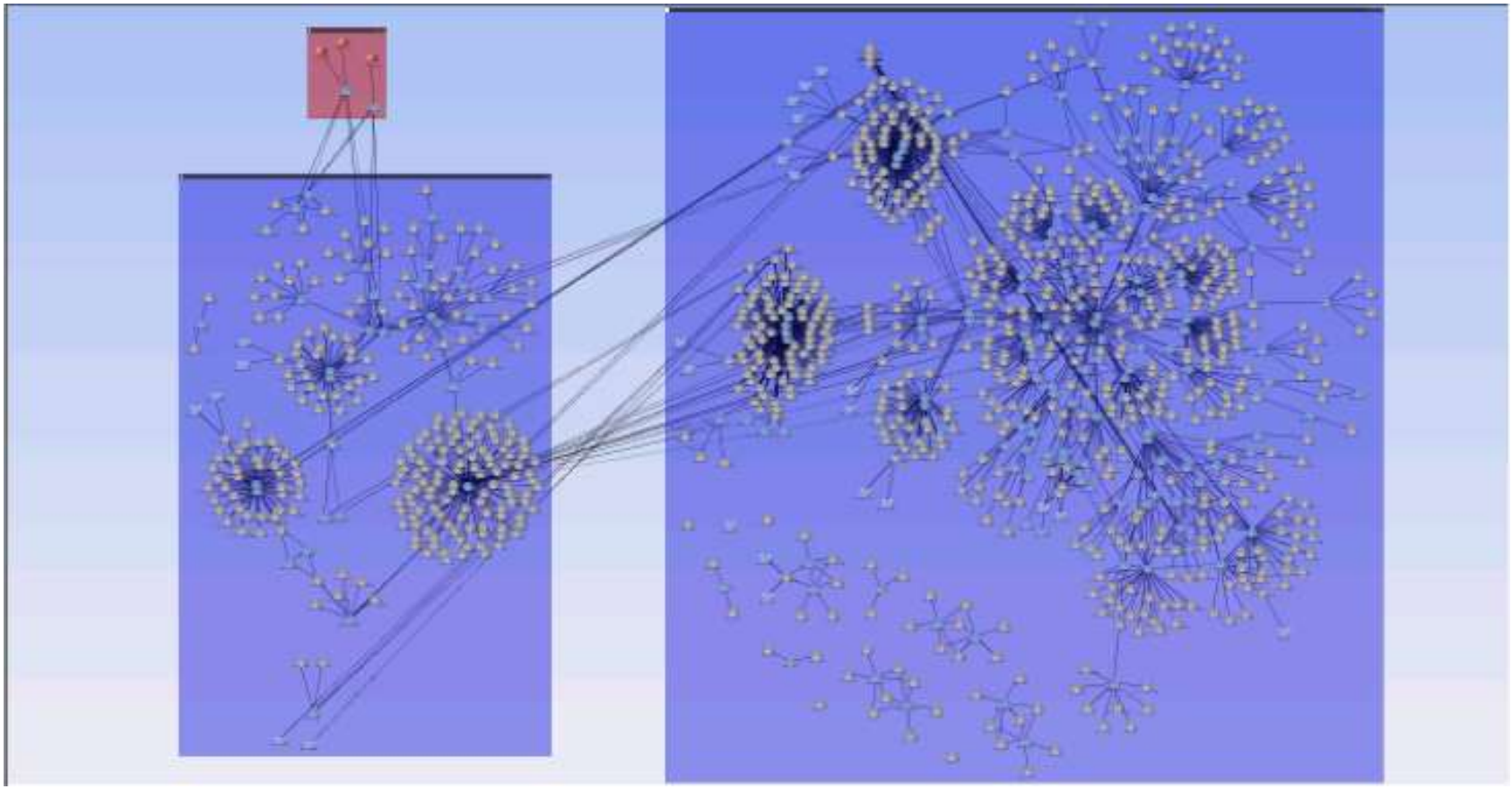
- Business change requests come thick & fast
- Security teams are asked to approve
- No standard basis to approve
- Can't position security team as "Dr No"

  Need clear, unequivocal reasons when rejecting changes
- Causes "the Carnac moment"



RTP Campus Network Map

Internet



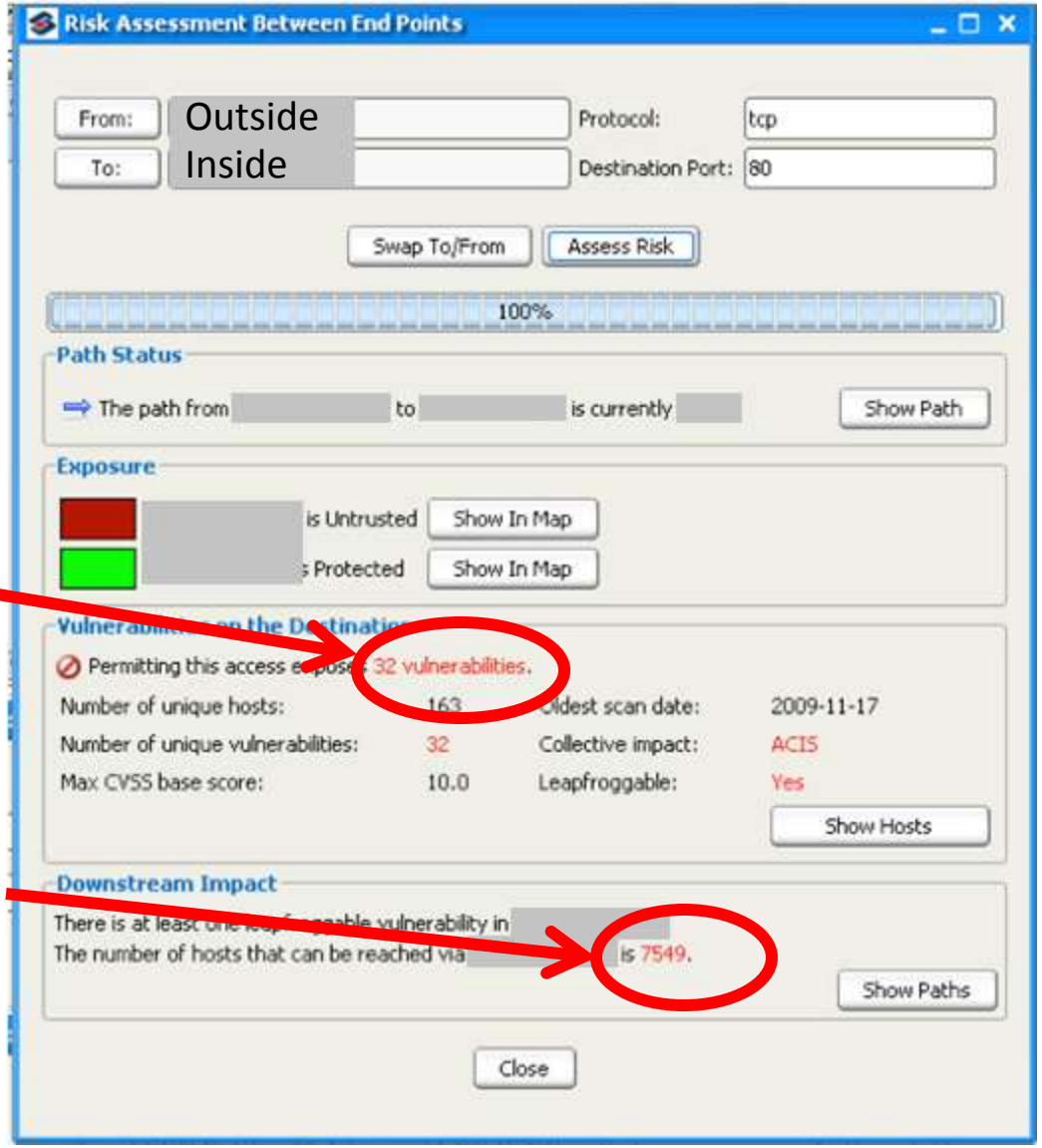
DMZ **Cisco Campus**

Client Connection Request

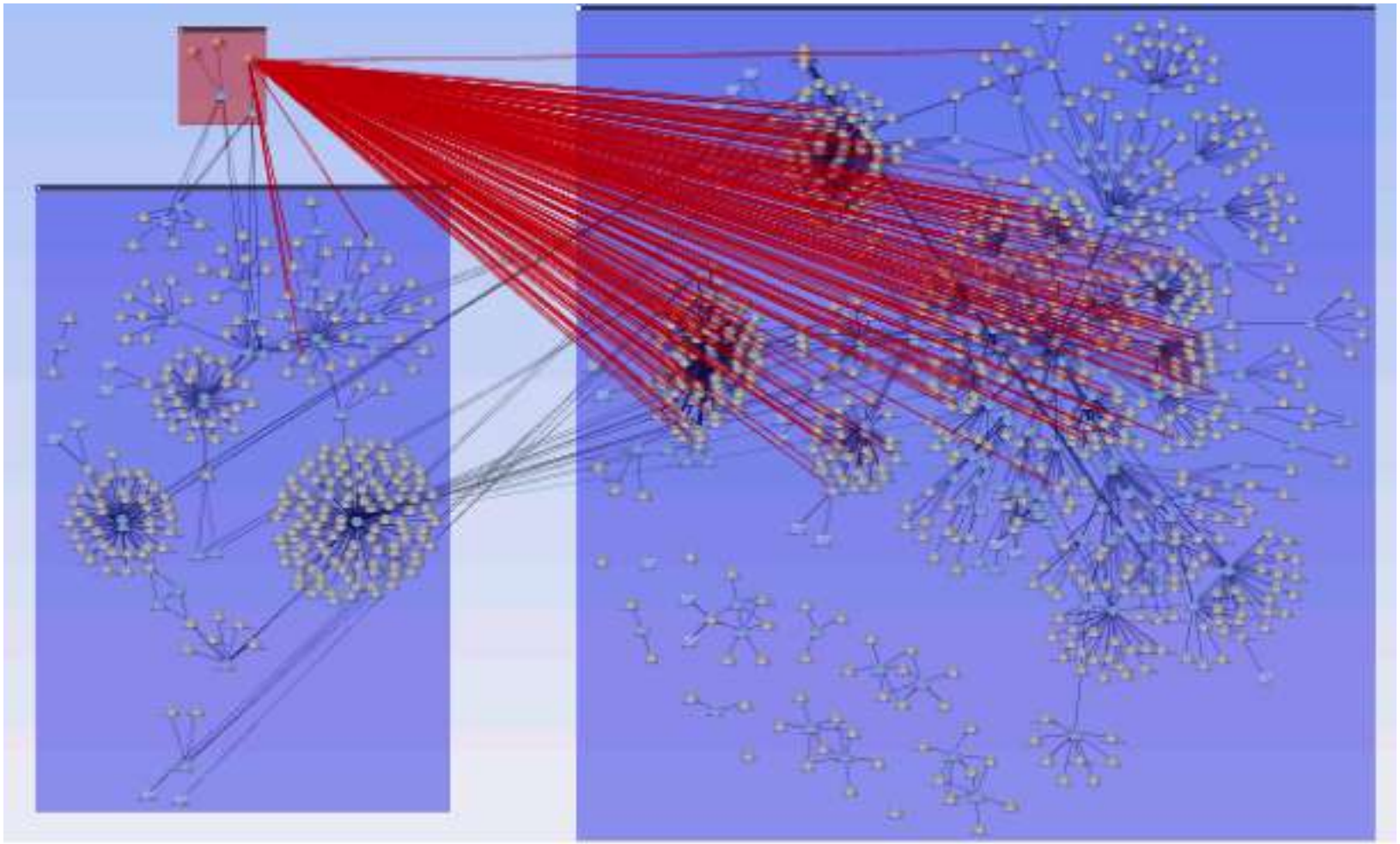
- Create Network Model
- Input Vulnerability Data
- Business need: Open one Class C network :80

  - Connection exposes 32 vulnerabilities

Downstream Effect?
Exposes 7,549 Vulnerabilities

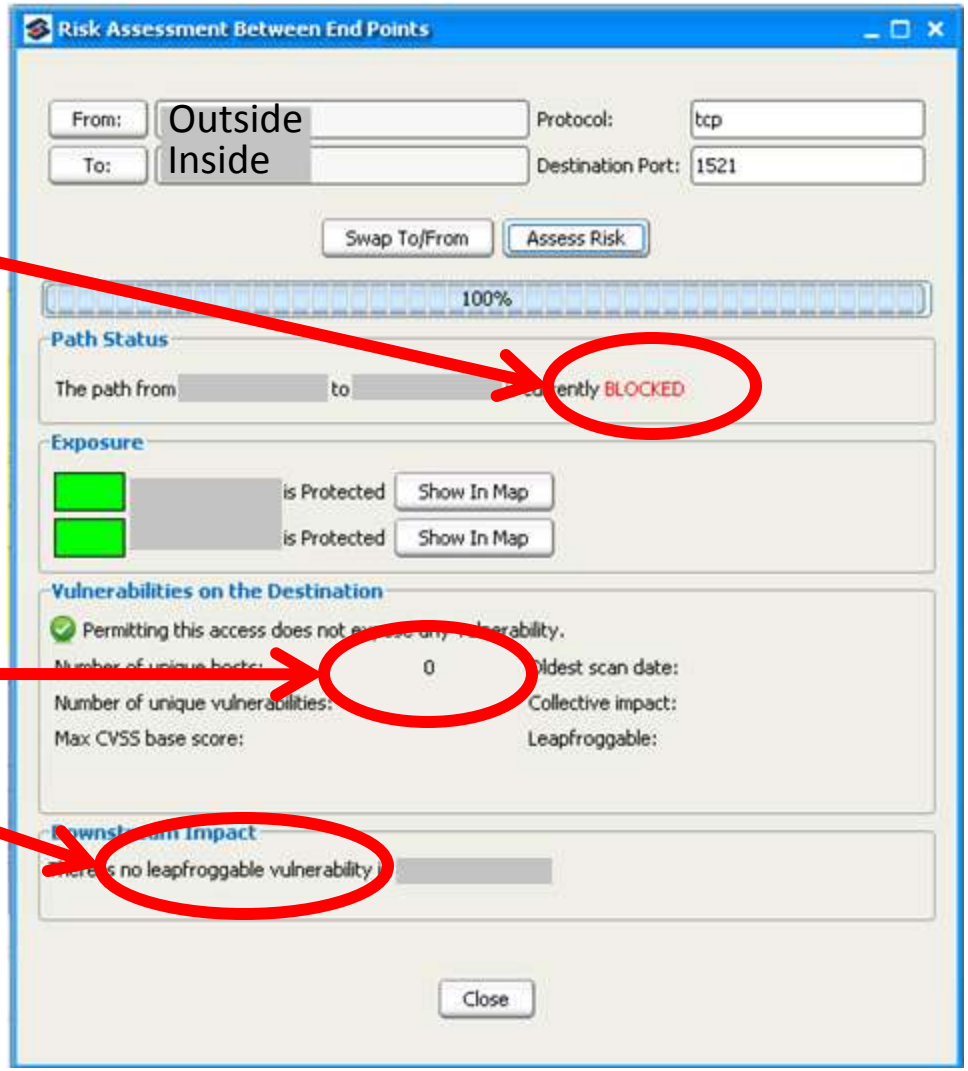


Client Connection Exposure



Acceptable Risk Assessment

- Access is BLOCKED
- No hosts vulnerable; nothing Leapfroggable



Before vs. After

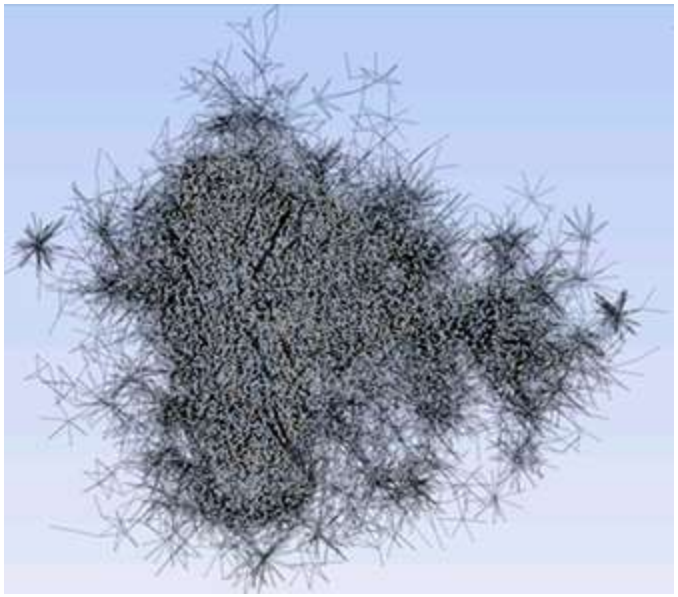
- Before
  - The Carnac moment
  - Could only enforce general best practices (“spell checking”)
  - Exceptions granted based on need, no real risk evaluation

- After
  - Push-button assessment of impact
  - Visuals to demonstrate nature of exposure
  - Automatic pin-pointing of rules needing to change

Lesson #10: We can finally have a coherent discussion with the business

Automating network audit

Before:



After:

Lesson Summary

- Lesson 1 – You need a config repository.
- Lesson 2 – Naming conventions are your friend.
- Lesson 3 – Pictures easily explain difficult concepts.
- Lesson 4 – Networks gather ‘cruft’.
- Lesson 5 – ‘Regular’ people can do this.
- Lesson 6 – Network data + Vuln data + Attack path = GOLD.
- Lesson 7 – A reference atlas is your friend.
- Lesson 8 – Cruft is so important we mention it twice.
- Lesson 9 – Computers are better at reading phone books than you are. Get over it.
- Lesson 10 – We can finally have a coherent discussion with the business.

Thank you

- Questions?
- Contact:

  ddexter@cisco.com