High Performance Network Security at SLAC

ESnet Site Coordinators Committee Meeting,
17-18 January 2013, Hawaii (US)

Antonio Ceseracciu

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

Introduction

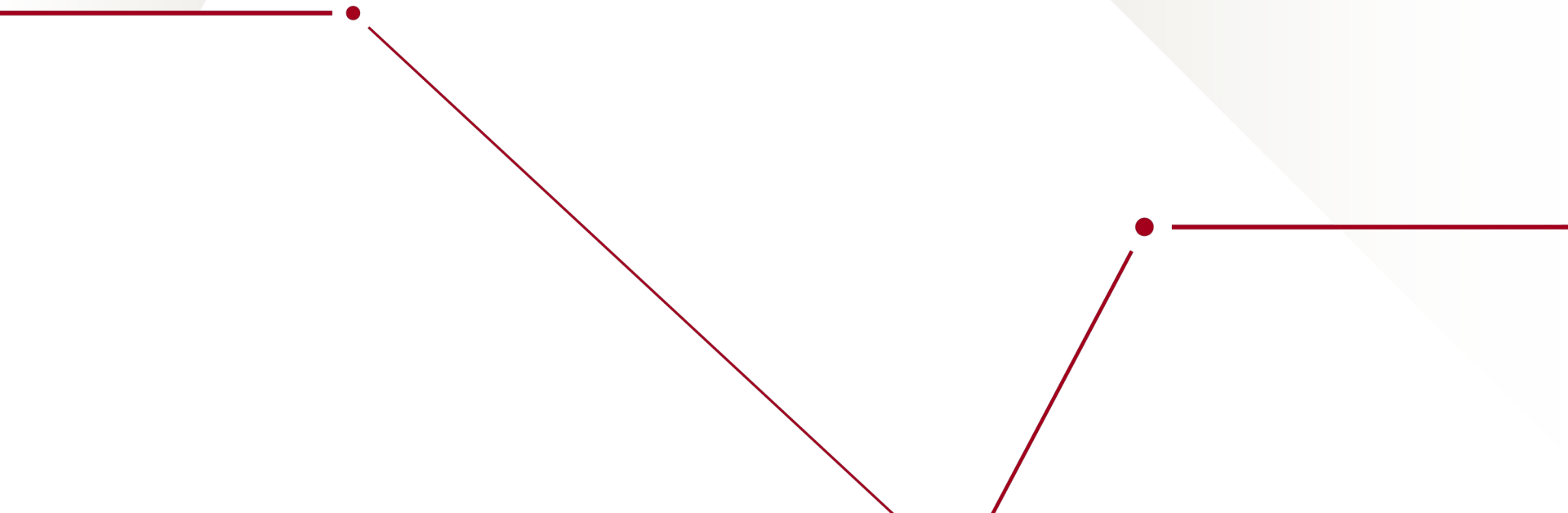
SLAC

- Assume familiarity with Science DMZ model.
- Look at alternative approaches to implement Science DMZ concepts within a site network.
  - Specifically, discuss operational experience at SLAC.

Discussion in two parts:

- Network Policy and Architecture.
- Distributed ACL management.

Network Policy and Architecture

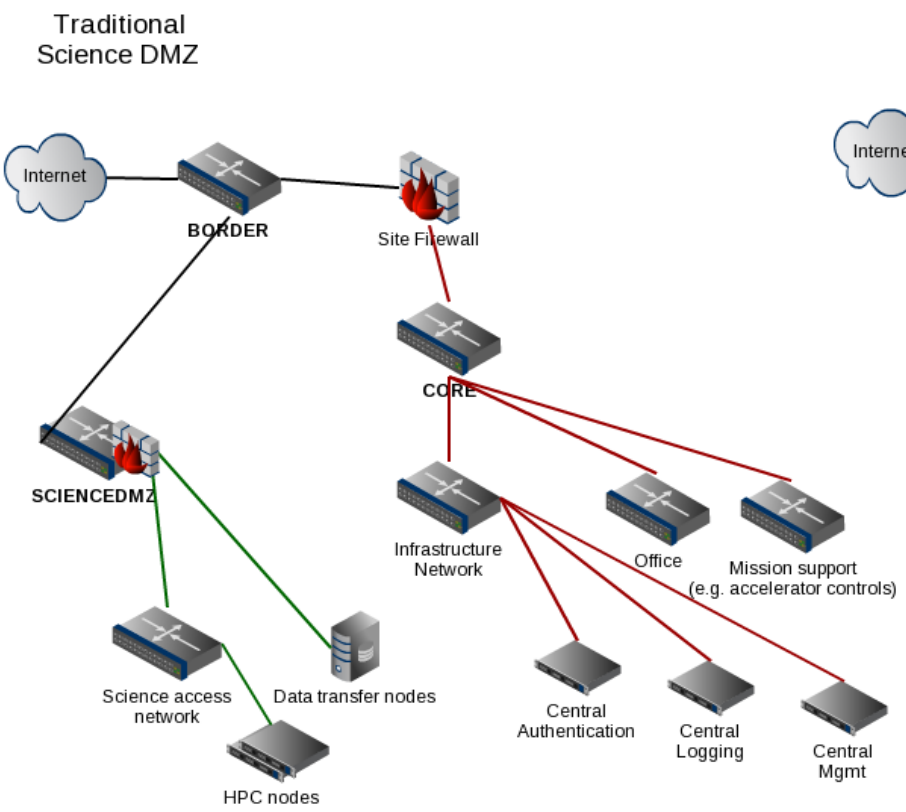


Typical Science DMZ vs. Distributed Science DMZ

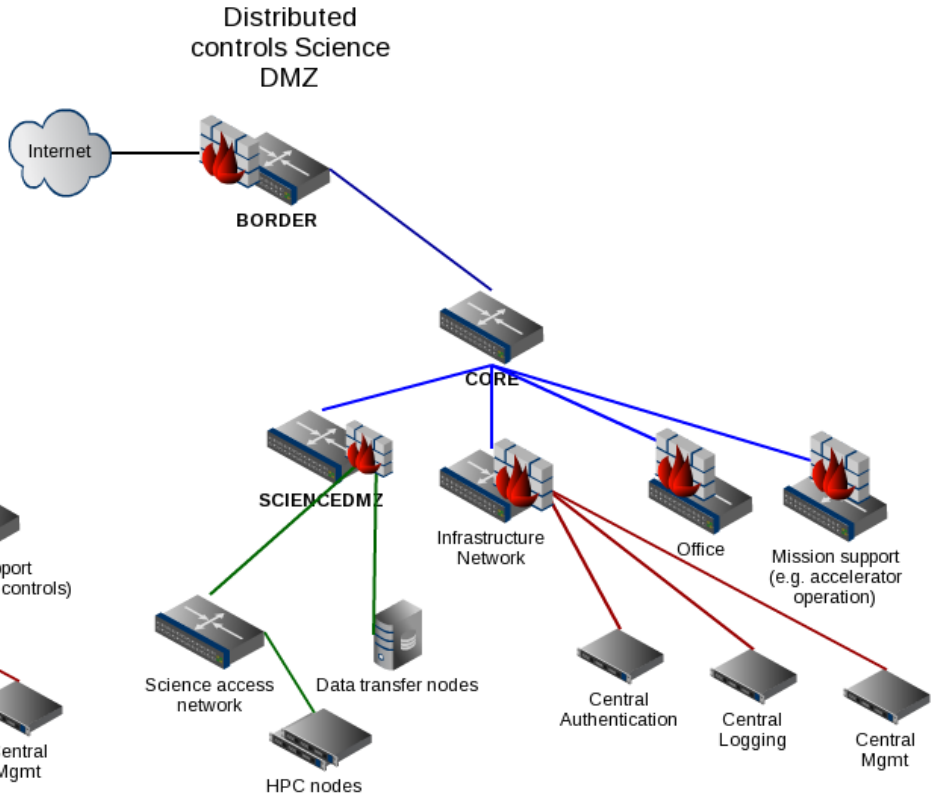
SLAC

- Split firewall: stateless at border, stateful inside.
  - The idea is to identify parts of your firewall policy which can be effectively implemented on stateless devices, and express those as router ACLs.
    - Examples: bad ports; non-internet-accessible IP space, BCP38, ...
- Duality between Site Firewall + science DMZ, split firewall with distributed internal controls.
  - i.e. any given network policy can be implemented in either scenario.
  - Distributed internal firewalls are required anyway if you have multiple Security zones in your internal network. Your backbone network may already be semi-trusted!

Traditional and Distributed Science DMZ Diagrams



Science DMZ (green network) sits outside of main firewall. Therefore it cannot access central services, which are on the internal network (red) and must be managed as a separate site (accounts etc); or have "holes" in the firewall, or backdoor networks, which potentially compromise site network policy.



Here, we split the functions of the Site firewall, such that the parts that can be implemented statelessly are done at Border using ACLs. This way, the core network becomes a middle security zone (blue).
More extensive security controls are pushed closer to the respective target systems.
Depending on the requirements of each enclave, the inner controls can be on stateful firewalls or router ACLs.

Stateless Performance Advantages

SLAC

- The distributed firewalls approach requires high performance data transfers to traverse core, as it pushes Science DMZ inside the network - so network core needs to be provisioned appropriately.
- Stateless ACLs work correctly with asymmetric flows.
- Stateless ACLs are processed at line rate on any modern router, provided you don't exceed the limits of the TCAMs.
- Lack of connection state tracking eliminates a possible attack vector.

Security *Gap*

SLAC

- “Real” firewalls offer connection state, deep packet inspection, integration with Identity Management.
- For many threat models, the gap is small, but details matter...
  - “*The devil is in the details*”
- Many of the early threats blocked by firewalls such as TCP replay and SYN flooding are not a problem for modern OS's.
- A growing fraction of "business" traffic is encrypted, making deep packet inspection less helpful.
- The logging capabilities of a firewall are superior, but netflow on a router can address the logging requirement. There are netflow-based IDS tools.

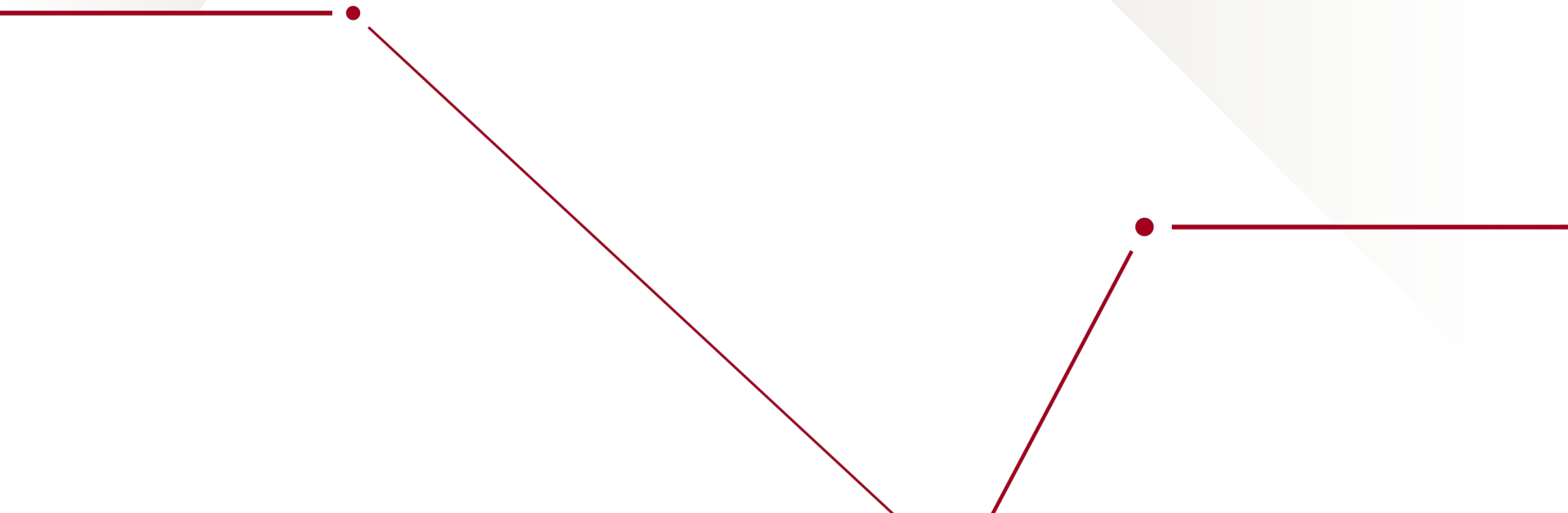
Site Integration

SLAC

For multi-purpose sites it can be difficult to build shared science DMZ's or multiple isolated science DMZ's, because individual experiments are too small or unorganized but, they still benefit from high speed data transfers!

A clean high performance network path into various experimental or user networks is a scalable way to address the requirement.

Distributed ACL Management



Distributed ACL Challenge

SLAC

- As discussed before, centralized firewall architectures require a choke point in the network design, which can result in suboptimal traffic routing, and reduce the capacity of the network.
- Alternative design is distributed control near the access layer, e.g. SVI.
  - Each zone can choose appropriate implementation for controls, e.g. stateless or stateful.
- Also consider: use host based fw, with central policy management.
- Distributed ACLs can represent a management challenge
  - the same groups of hosts and networks are referenced in multiple ACLs on multiple devices.
  - IPv4/IPv6 duplication of policy compounds the problem.
- Some form of automation is fundamental.

SLAC Approach

A solution being pursued at SLAC is an ACL management system based on **Capirca.**

- Capirca is a meta-language to express firewall rules / ACL's, and compilers for multiple platforms.
- It was created and used internally at Google, and released under open source license.
- Idea: take advantage of automation to distribute enforcement across the network, exploit the integration with IPAM to [semi-]automatically update firewall rules when hosts or subnets change IP address or disappear.

Example

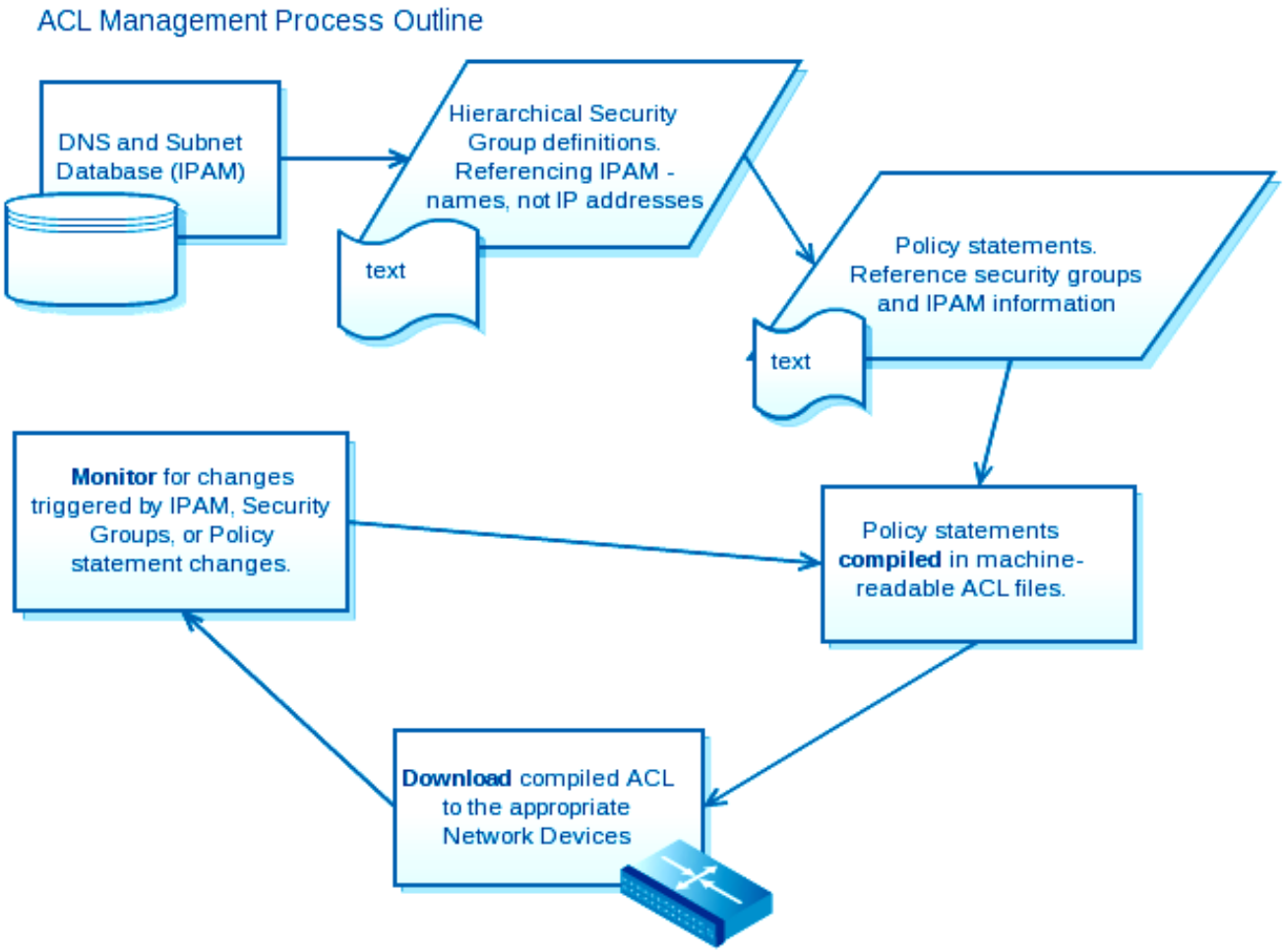
- The ACL name and target device are declared in the header. networks and host names are referenced by name, not IP address.
- There are tools to test the ACL offline, before applying it to the

```
header {
     comment:: "policy for ..."
     comment:: "This ACL is generated from a Capirca source file.
           Do not edit manually”
     comment:: "targetDevice: rtr-farmltda"
     target:: cisco toBBR-LTDA-VM
}
term allow-ssh-fm-ltda-login {
     source-address:: LTDA-LOGIN my-workstation
     protocol:: tcp
     destination-port:: SSH
     action:: accept
}
term default-deny {
     protocol:: ip
     action:: deny
     logging:: true
}
```

https://confluence.slac.stanford.edu/display/NetMan/SLAC%27s+modifications+to+the+Capirca+ACL+management+package

ACL Lifecycle Workflow

- The **download** step is initiated manually, to ensure that a human looks at the diffs before committing a policy.
- The **monitor** loop is effective particularly in the common case where a server referenced in some ACL changes IP address. When the new IP address is registered in the IPAM system (DNS), the generated ACL automatically changes, and the nightly verification procedure alerts on that.



Thank you.

Questions?