Poster: Geographically Distributed Management of Enterprise Network Security Policy

Iffat Anjum
North Carolina State University
ianjum@ncsu.edu

Jessica Sokal
Northeastern University
sokal.j@northeastern.edu

Hafiza Ramzah Rehman
North Carolina State University
hrehman@ncsu.edu

Ben Weintraub
Northeastern University
weintraub.b@norteastern.edu

Ethan Leba
Northeastern University
leba.e@northeastern.edu

William Enck
North Carolina State University
whenck@ncsu.edu

Cristina NitaRotaru
Northeastern University
c.nitarotaru@northeastern.edu

Bradley Reaves
North Carolina State University
bgreaves@ncsu.edu

***Abstract*—Commercially-available software-defined networking (SDN) technologies will play an important role in protecting the on-premises resources that remain as enterprises transition to zero-trust architectures. However, existing solutions assume the entire network resides in a single geographic location, requiring organizations with multiple sites to manually ensure consistency of security policy across all sites. In this poster, we present MSNetViews[1], which extends a single, globally-defined and managed, enterprise network security policy to many geographically distributed sites. Each site operates independently and enforces a least-information *policy slice* that is dynamically parameterized with user location as employees roam between sites. We build a prototype of MSNetViews and show that for an enterprise with globally distributed sites, the average time for policy state to settle after a user roams to a new site is well below two seconds. As such, we demonstrate the utility of SDN towards achieving zero trust for on-premises network resources, even for organizations with many geographically distributed sites.**

I. Extended Abstract

NetViews [1] is the most recent work in a series of research proposals that use reactive software-defined networking (SDN) to enforce least-privilege, per-request connections between hosts within a single enterprise location. We present MultiSite NetViews (MSNetViews), which extends a single, globally-defined, enterprise network security policy to many geographically distributed sites. MSNetViews extends prior work by modeling both role- and location-based access control policies, allowing independently operating reactive-SDN networks to dynamically react to employee movement between sites. A key contribution of our work is the idea of "policy slicing," which not only allows a global manager to selectively inform individual sites of policy and user location updates, but also limits the security policy available at each site on a "need-to-know" basis. Our performance evaluation shows that for an enterprise with sites globally distributed, the average time for policy state to settle after a user roams to a new site is well below two seconds, which is negligible with respect to the overall experience of traveling often hours and authenticating to a new network.

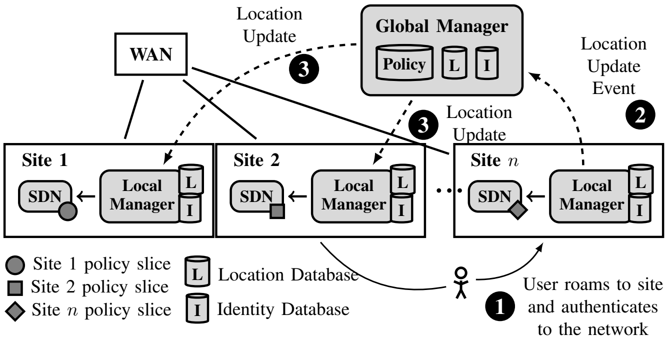


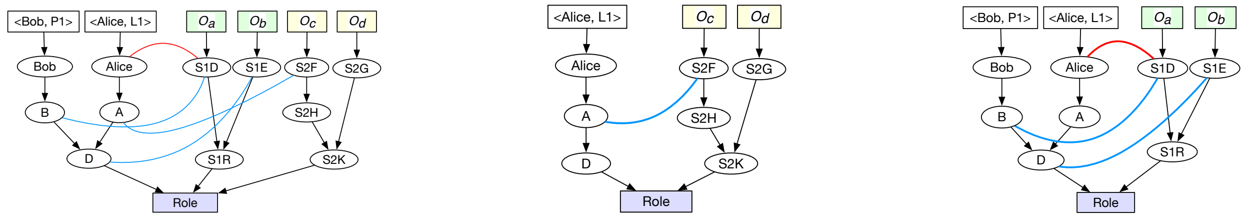
Fig. 1: Overview of our approach. Each site runs its own SDN network with a unique subset of the global policy. Users roaming between sites cause location update events that are propagated to other sites as appropriate. Each SDN controller has a different access control policy, as indicated by the different shapes (circle, square, diamond).

Enterprise networks require least-privilege policies that restrict network communication between on-premises hosts. Extending prior solutions to consider multisite environments requires overcoming the following research challenges.

- *Users commonly move between sites, requiring differentiated access based on their location.* The policy and enforcement must dynamically update based on a user's location, and ensure that state is consistent across sites.
- *Compromise of a single site should not leak the global policy.* Security policies are often confidential. An exposed policy at one site should not leak policy details of unrelated sites.
- *Site administrators should only modify policies for their local resources.* Updates to the global policy should be controlled and maintain policy semantics.

As shown in Figure 1, MSNetViews operates at two scales: global and local. The global manager coordinates policy management between sites. We envision the global manager residing in a cloud service, though it could be hosted within a site. Site administrators specify their network access control

[1] The paper is currently under submission on the same topic at SACMAT 2023



(a) Global policy containing sites S_1 and S_2

(b) Policy slice for site S_2

(c) Policy slice for site S_1

Fig. 2: Example demonstrating the use of policy slicing to create site-specific least-information policies. Here, we excluded the location policy class for simplicity.

TABLE I: MSNetViews Policy Invariant Rules

Rule	Name	Purpose
1	Dangling PE	Each policy element must lead to at least one policy class.
2	Exclusive UA	Each user attribute must lead to only one policy class.
3	Exclusive OA	Each object attribute must lead to only one policy class.
4	Exclusive Associations	The source and target attributes of an association relation must lead to same policy class.
5	Exclusive Prohibitions	The source and target attributes of a prohibition relation must lead to same policy class.

policy directly in the global manager through an administrator console. When the global policy is modified, local policies are generated, sent, and reloaded at each site. A local manager operates at each site, managing the local policy state, users, and interactions with the global manager. Each site runs its own independent reactive SDN network with its own SDN controller. This global-local design never requires policies to be merged, eliminating the potential for policy conflicts. It also (1) significantly decreases first-packet latency and (2) allows internal traffic to continue if the WAN connection fails.

In presenting MSNetViews, we demonstrate that SDN can not only provide an invaluable primitive for achieving zero trust within a single enterprise location but also across many geographically distributed locations.

II. Research Challenges

In this section, we discuss the methodology for the research challenges.

A. Supporting Roaming Users

Shown in Figure 1, a user's physical movement between sites initiates a location and identity update. The local manager at the new site updates its policy state using the NGAC obligations. The local manager then informs the global manager, which in turn informs the local managers in the other sites (on a "need-to-know" basis, as discussed in Section II-B). Those local managers update their policy state using their NGAC obligations. This design ensures that only the global manager is allowed to initiate updates to the policy state, identity, and location repositories. As such, MSNetViews avoids potentially malicious or unwanted local alterations. It also simplifies maintaining state consistency as the global manager serves as a permanent, central leader.

B. Policy Slicing

Each site only needs to know how to control access to *local resources*. As such, MSNetViews uses policy slicing to compute and distribute a *least-information policy* to each site. The policy slicing algorithm takes as input (1) a global policy, (2) a site S_x, and (3) the set of objects for S_x. It outputs a least-information policy. The algorithm traverses the global policy to identify all policy elements, assignments, associations, and prohibitions to determine which ones are *relevant* for site S_x (algorithm omitted for brevity).

Figure 2 provides an example (excluding location policy class) that depicts the high-level intuition behind policy slicing. Intuitively, the policy slices for site S_1 (Fig. 2b) and site S_2 (Fig. 2c) are created by identifying the objects specific to each site and then finding all of the object attributes, user attributes, and users that refer to those objects by inspecting the assignments, associations, and prohibitions.

As shown in Figure 2b, site S_2 only needs the subset of the policy relevant to resource O_c and O_d. This site-specific policy includes object attribute $S2G$, because there is an assignment path $O_d \rightsquigarrow S2G$. Furthermore, the policy includes user attribute A, because there is an association rule connecting A to $S2F$. Next, the policy includes user $\langle Alice, L1\rangle$, because $\langle Alice, L1\rangle \rightsquigarrow A$. The policy slice for site S_1 in Figure 2c is constructed similarly, also considering prohibition relations.

C. Policy Checker

Policy checks are needed for the following scenarios: (a) adding or removing a policy element; (b) adding or removing an association, or prohibition; and (c) changing the policy elements referenced by an assignment, association, or prohibition. Table I defines five policy invariant rules we added to supplement those in the policy-machine-core.

References

[1] Iffat Anjum, Daniel Kostecki, Ethan Leba, Jessica Sokal, Rajit Bharambe, William Enck, Cristina Nita-Rotaru, and Bradley Reaves. Removing the reliance on perimeters for security using network views. In *Proceedings of the ACM Symposium on Access Control Models and Technologies*, SACMAT, 2022.

We developed a framework for geographically distributed management of enterprise network security policy. Each site operates independently and enforces a least-information policy slice that is dynamically parameterized with user location as employees roam between sites. We demonstrate the utility of SDN towards achieving zero trust for on-premises network resources for organizations with many sites.

MSNetViews: Geographically Distributed Management of Enterprise Network Security Policy

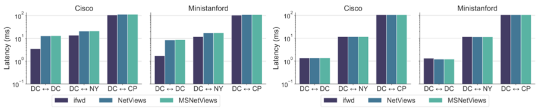
Iffat Anjum, Jessica Sokal, Hafiza Ramzah Rehman, Ben Weintraub, Ethan Leba, William Enck, Cristina NitaRotaru, Brad Reaves

Introduction

- Users move between sites, requiring differentiated access based on their location. The policy and enforcement must dynamically update based on a user's location and ensure that state is consistent across sites. We propose policy model for supporting *user roaming*.
- Compromise of a single site should not leak the global policy. Security policies are often confidential. An exposed policy at one site should not leak policy details of unrelated sites. We propose *policy slicing* to minimize policy information loss when a site is compromised.
- Site administrators should only modify policies for their local resources. Updates to the global policy should be controlled and maintained by policy semantics. We propose a *policy checker* to maintain policy correctness.

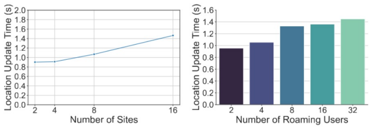
Methods

- Policy Support for Roaming: MSNetViews achieves dynamic policy updates using NGAC's concept of obligations, which are rules that accept events as input. Based on that input, the policy engine executes a set of pre-defined actions that change the enforceable rules.
- The policy slicing algorithm takes as input a global policy, a site, and the set of objects for that site. The algorithm traverses the global policy to identify all policy elements, assignments, associations, and prohibitions to determine which ones are *relevant* for the site.
- MSNetViews policy checker is built upon the NGAC policy engine implementation. We define five policy invariant rules to supplement those in the policy engine.



(a) Average Initial Packet Latency

(b) Average n^{th} Packet Latency



(a) Location update time of one roaming user as a function of number of *relevant* sites

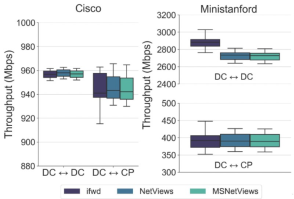
(b) Avg. location update time per user as a function of number of users roaming between two sites

Results

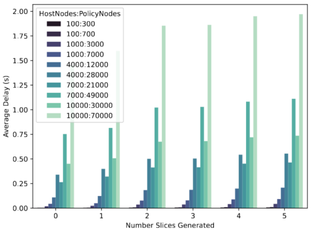
- Impact one system latency and throughput: The performance of MSNetViews is comparable to NetViews and ifwd with negligible impact on latency and throughput.
- Time MSNetViews takes to stabilize after a user roams between sites: For a single user roaming between 2 sites this delay is under 2 seconds depending on how many relevant sites need to be informed ranging from 2 to 16 relevant sites.
- How expensive are policy checking and update operations, and do they scale: The policy checker and slicer takes under 3 seconds for as much as 10k hosts and 70k policy nodes.

Takeaways

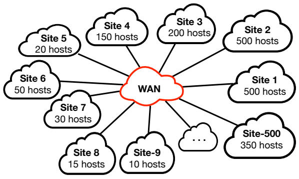
- In presenting MSNetViews, we demonstrate that SDN can not only provide an invaluable primitive for achieving zero trust within a single enterprise location, but also across many geographically distributed locations.



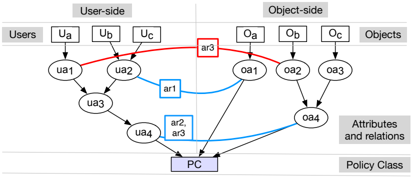
Aggregate throughput for MSNetViews, NetViews, and ifwd under two WAN latencies between sites



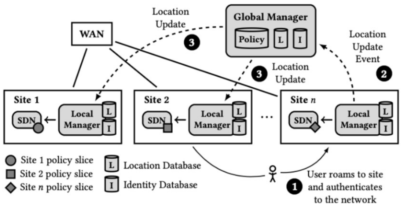
Effect of #Slices Needed to be Generated for Policy Updates



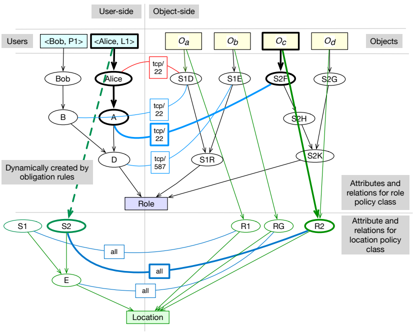
Example Enterprise Network Scenario



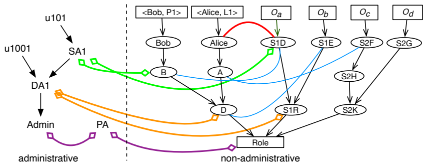
Example of NGAC Policy Structure



Overview of MSNetViews



MSNetViews Policy Example



Administrative Policy Example

NC STATE UNIVERSITY

WSPR