Migrating to BeyondCorp

Maintaining Productivity While Improving Security

JEFF PECK, BETSY BEYER, COLIN BESKE, AND MAX SALTONSTALL

Jeff Peck is a Technical Program Manager for CorpEng in Google. He previously worked at companies large and small around Silicon Valley, doing software engineering and program management for a variety of projects in the telecom, server, and network application domains. He has a BS in computer, information, and control sciences from the University of Minnesota. jpeck@google.com

Betsy Beyer is a Technical Writer for Google Site Reliability Engineering in NYC. She has previously provided documentation for Google Data Center and Hardware Operations teams. Before moving to New York, Betsy was a lecturer in technical writing at Stanford University. She holds degrees from Stanford and Tulane. bbeyer@google.com

Colin Beske is a Technical Program Manager at Google. Since joining in 2010, he has worked on IT support, printing operations, and internal change management. Prior to Google, he held positions in systems and networking engineering. He has a BA in computer science from Oberlin College. beske@google.com

Max Saltonstall is a Technical Director in the Google Cloud Office of the CTO in New York. Since joining Google in 2011, he has worked on video products, internal change management, IT externalization, and coding puzzles. He has a degree in computer science and psychology from Yale. maxsaltonstall@google.com

If you're familiar with the articles about Google's BeyondCorp network security model published in *;login:* [1-3] over the past two years, you may be thinking, "That all sounds good, but how does my organization move from where we are today to a similar model? What do I need to do? And what's the potential impact on my company and my employees?" This article discusses how we moved from our legacy network to the BeyondCorp model—changing the fundamentals of network access—without reducing the company's productivity.

Among the many challenges that a migration to a BeyondCorp-type model entails, several are particularly notable:

- This process affects the entire company. Getting everyone on board and keeping everyone aligned and informed requires commitment and buy-in from all levels of management. That commitment needs to be reinforced through extensive communications with all parties involved, from the teams that own individual services, to management, to support teams, to users.
- The migration can't be done overnight. The process is multi-layered and incremental, with stages of information gathering, trial deployments, corrections to processes and technology, and exceptions and remediation where and when necessary.
- The process requires changes at many or all layers of the stack: networking, security gateways, client platforms, and backend services. Partitioning the changes in order to make progress independently at different layers makes this multi-pronged undertaking more approachable and manageable.

The following sections discuss how we partitioned the BeyondCorp migration effort, and the tools and technologies we used to bring the various layers into alignment while minimizing negative impact on users.

Prerequisites: Commitment and Communications

Before you can undertake a migration to a BeyondCorp-like model, you need buy-in from top level management and other stakeholders in your organization. Step one here is understanding and communicating the motivation for the migration: you want to reduce the threat of a successful cyberattack while maintaining productivity. You need to document the rationale behind the proposed migrations, the threat model, and the costs of doing "business as usual." Then be prepared to explain to each line-of-business why this process is valuable and essential. As with all security operations, deploying a new model comes with a price: new tools, additional processes, and changes in habits to apply. Top-level management needs to actively support this change and drive the motivation and commitment down to all stakeholders.

Armed with a charter and commitment from management, identify and enlist the support of leaders in crucial areas: security, identity, networking, access control, client and server platform software, business-critical application services, and any third-party partners or outsourced IT functions. The leads should identify and enlist the subject matter experts for each area and commit their time and energy to the process. Our BeyondCorp team was a globally distributed virtual team headed by a director responsible for policy decisions and a technical program manager to drive and coordinate execution. Active membership changed over time, but the stakeholders, team leads, and other contributors were consistently linked through online documentation, group email, and regular face-to-face and video conference meetings to stay informed of current processes and project status.

As the effort progresses, the usual rules of change management apply, because each work group will have its own concerns and priorities. Listen to feedback and adapt to the special circumstances and requirements of each contributor or affected group. Publishing plans and information is necessary but insufficient; interactive communication (ideally done in person, but at minimum conducted over video or audio conferencing) speeds assistance and adoption.

Partitioning for Progress

The overall objective of the BeyondCorp program is to transition from a network that allows clients to directly access servers to a new network design, one that removes the privilege of direct access to backend servers. For more details, see "BeyondCorp: A New Approach to Enterprise Security" [1], the first article in this series. To this end, we considered removing privileged access from the legacy VLAN by blocking each application or server in sequence. This strategy was less than ideal for two reasons: it would be difficult to deploy and coordinate at the network layer, and it posed increased risks to productivity at the application layer. Instead, we decided to deploy a new VLAN in its final BeyondCorp configuration. This VLAN only permits access to the server network through access control gateways, ensuring that all traffic flows are authenticated, authorized, and encrypted. Rather than incrementally restricting the privileges of the legacy VLAN, we incrementally moved devices to this new end-state VLAN.

The VLAN migration project achieved the complex but critical goal of removing user devices from the legacy "privileged" network and assigning them to the new Managed Non-Privileged Client (MNP) VLAN. This move had a key constraint: any legacy application that expected or required direct access to the server network would fail when run from a workstation on the new VLAN. Therefore, achieving this migration without breaking business-critical operations was an immediate subgoal. We used a three-pronged strategy to meet this subgoal:

1. Extensively analyzing network traffic logs
2. Identifying and remediating noncompliant applications
3. Migrating devices **after** determining they would be successful on the new network

This approach allowed the network layer to roll out the new configuration and achieve stability independently from other parts of the BeyondCorp program. The BeyondCorp design includes the use of 802.1x for network admission and VLAN assignment, which we utilized to isolate the network layer from the details of the migration policies. Higher level software and data analysis determined each device's VLAN assignment, which the RADIUS servers then communicated to the network layer.

Realizing these goals was a vast undertaking that required changes at almost every layer of the stack. Rather than attempting to introduce change to all of these layers in a single transition (undoubtedly a recipe for disaster), we pursued a partitioned approach that entailed:

- Decoupling network layer projects: new VLANs, 802.1x, RADIUS policy server
- Decoupling client platform upgrades: certificate generation and installation, user authentication tools
- Migrating devices incrementally as we remediated services and workflows
- Continuously refining our processes and procedures

First Steps: An 802.1x Network

In the first phase of BeyondCorp, we installed certificates on each user device and transitioned to 802.1x for all network access grants. This seemingly simple step implied several new developments: a certificate authority, tools to install certificates on company-managed devices (for each OS type), enabling 802.1x on the network switches, and integrating with a policy-driven RADIUS service. We undertook all of these developments in parallel.

The security team designed a new Certificate Authority with APIs to enable the various per-OS platform management teams to obtain and install certificates on their platforms. Each platform team independently deployed the software, tools, and telemetry to enforce and monitor certificate rollout to each device. We created the processes for mass distribution and maintenance of certificates while we were still working on integration with the access switches.

Likewise, re-provisioning the access switches to include the new VLAN definitions proceeded in parallel—we enabled and later required 802.1x and RADIUS-provided VLAN assignments. Automated scripts audited the switch upgrades to identify switches not yet provisioned with the new VLAN. As a result, the RADIUS server would not request a VLAN assignment that wasn't available on a particular switch.

We used 802.1x so we could move control of VLAN assignments from the network layer to a VLAN policy server. Because we wanted to reduce failures caused by the new RADIUS server, the initial policy simply matched the existing assignments (which included complex blacklists and whitelists). We first deployed the policy server in an auditing mode that compared the new assignments with the legacy assignments. When the differences were sufficiently few, we enabled the new policy. From that point on, we could manage device assignment to VLANs in near-real time using high-level software and data-driven policies. Using this simple initial policy allowed us to enable dynamic VLAN assignments in the network while the end-state (and transition) policies were still being developed.

Success-Oriented Migration

It took years to fully deploy the 802.1x layer, and several more years before the inventory-based tiered access VLAN assignments were available as input to the RADIUS policy server [2]. While those developments were underway, we wanted to identify our two main groups of users and application services: those that were ready for BeyondCorp versus those that needed to upgrade their network and security capabilities to become BeyondCorp compliant. Our first step was to capture and analyze traffic from the network routers. By logging and analyzing a fractional sample of all traffic through the corporate routers, we discovered patterns of noncompliant usage. As a second-order benefit, this analysis also helped us discover unusual, unexpected, and unauthorized traffic on the network. Identifying these applications meant we could start the reengineering earlier and avoid disrupting the users of these systems.

Some networking use cases, such as workstations using an NFS/CIFS file server, were obviously noncompliant. Although a NFS/CIFS file server is a simple way for users to maintain a single, common copy of their files, the underlying protocol didn't support our desired security properties (strong encryption and authentication). To eliminate this dependency, we initiated a major project early on to accomplish two goals: moving NFS home directories to local disk with automatic backup to secure cloud storage, and replacing other NFS usage with Google Drive or other secure file-sharing technologies. Even so, some applications, like CAD (computer-aided design) editors, are deeply dependent on NFS and required special solutions before we could move their users and workstations to the restricted MNP VLAN. We discuss the details of our framework for handling these special requirements in the "Remediating Difficult Use Cases" section below.

Other noncompliant workflows were not so obvious but would nevertheless fail when subjected to the restrictions of the MNP network ACL. This failure was by design, as we couldn't assume that NFS, RDP, SQL, etc. had adequate authentication, authorization, and encryption. Detecting these workflows and re-enabling productivity by changing the device's network assignment is difficult and time-consuming when remediation must happen at the network layer. To avoid large impacts on productivity (not to mention user morale), we needed an analysis-driven strategy to detect failing workflows and correct them before assigning users to the MNP VLAN.

To facilitate easy analysis and user workflow testing on the non-privileged network, we created a client-based network ACL simulator that identified network packets that would be blocked by the MNP ACL. The underlying technology used Capirca (see [4] for the source code) to create local iptables or Packet Filter rules from the actual MNP network ACL. During the analysis and migration phase, user devices continued to operate on the privileged network, while the MNP-simulator monitored network traffic and logged the source and destination of all non-MNP-compatible traffic to a central repository. The IP source address identified the failing user, and the IP destination address identified the failing service. By analyzing the logs over time (with appropriate privacy constraints in place), we could identify devices with MNP-compliant traffic and assign them to the MNP VLAN. Likewise, we could identify devices, users, and services that relied on noncompliant traffic and initiate projects to move those services to alternative solutions. Over time, more devices became compliant and were automatically assigned to the MNP VLAN.

In a second mode, the MNP-simulator can actually block/drop the non-MNP traffic, thereby enforcing the MNP ACL without relying on network level deployment of the MNP VLAN and the 802.1x pipeline. Although we ultimately enforce the ACL in the network equipment, where it is isolated from user (or hacker) abuse, enabling and disabling this "enforcement" mode in the client workstation is much easier and faster during the trial and transition period. Client-side enforcement served as both an important step in the migration process and a self-service tool for testing. Without this feature, we wouldn't have gained the confidence we needed to move devices to MNP at nearly the speed (or with the high level of success) that we did.

Figure 1 shows the pipeline for moving Google computers to the Managed Non-Privileged (MNP) network.

Handling Easy Use Cases with the Access Proxy

Google's basic security policy requires that all traffic that flows from workstations to servers is:

- Authenticated (to identify the device and user making the request)
- Authorized (to verify that the user and device are allowed to access the backend resource)
- Encrypted (to prevent eavesdropping)
- Independently logged (to aid in forensic analysis)

The Access Proxy [3] achieves all these requirements for HTTP/S traffic and for our HTTP-encapsulated SSH traffic.

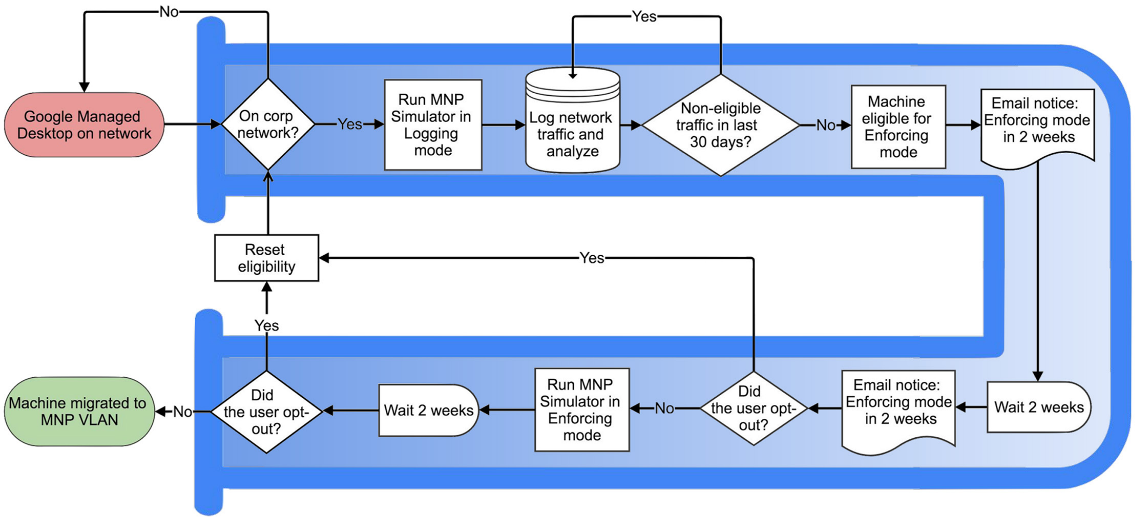
Google Managed Desktop on network
On corp network?
No
Yes
Run MNP Simulator in Logging mode
Log network traffic and analyze
Non-eligible traffic in last 30 days?
Yes
No
Machine eligible for Enforcing mode
Email notice: Enforcing mode in 2 weeks
Reset eligibility
Yes
Yes
Wait 2 weeks
Email notice: Enforcing mode in 2 weeks
Did the user opt-out?
No
Run MNP Simulator in Enforcing mode
Wait 2 weeks
Did the user opt-out?
No
Machine migrated to MNP VLAN

Figure 1: The pipeline for moving Google computers to the Managed Non-Privileged (MNP) network

Happily, most of our high-usage applications are browser-based Web applications. This condition is both "happy" and by design: Google is somewhat unique in the industry in its core philosophy of using browser-based applications when possible. We provided tools and documentation to each Web application provider so each could configure their application to run behind the Access Proxy.

When an application is behind the Access Proxy, corporate and public DNS contains a CNAME that resolves to the Access Proxy, so the URLs for such applications work from both corporate and public networks with equivalent ease and security. The ability to access corporate applications from public networks meant that authenticated remote users could access the corporate Web applications without diverting to initiate a VPN connection. As a result, the overhead for using and supporting VPN connections for remote work immediately and dramatically decreased. According to our rough estimates, the resultant productivity gains easily outweigh the implementation costs of BeyondCorp.

Once browser-based applications were secured behind the Access Proxy, we could make dramatic progress. We activated an automatic process for analyzing, verifying, and migrating devices to the non-privileged network; within a year this process moved over 50% of the fleet to non-privileged network access.

Remediating Difficult Use Cases

While we could handle the vast majority of applications via the Access Proxy, other applications weren't so easy. Our plans and schedules also had to address the reality of the long tail of non-Web cases that required additional time and resources to migrate. Evolving these use cases to become compliant required new tools, technology, and workflow modifications.

In particular, some of our workgroups use third-party desktop or "thick client" applications that are not HTTP-based, which entail a special set of problems. For example:

- Some tools are intrinsically designed to rely on network mounted file shares.
- Java applications may use RMI (Remote Method Invocation) or other direct socket connections.
- Many tools may be linked to license servers using non-HTTP sockets and protocols.

Even applications that use HTTP may be problematic due to obscure, unexpected failure modes. For example, some applications aren't designed to present a client certificate or proper user credentials, while some have logic built into the load balancing layer that doesn't mesh well with the Access Proxy. For some of these cases, we tweaked the Access Proxy to allow traffic coming from the MNP VLAN to pass without a certificate. We felt comfortable with this temporary strategy because the device had to present a certificate in order to access MNP. Each problematic case required a diagnosis and remediation project.

To address the class of hard cases, we developed a solution using a multi-port encrypted tunnel to carry application traffic between the client and server:

- When initiating a connection from client to server, the Access Proxy applies the usual user and device authentication and authorization.
- Routing tables on the client direct packets to a TUN device that captures and encrypts traffic to specific backend servers.
- The encrypted packets flow directly between the client and encryption server using a UDP-based encapsulation protocol.
- The encryption servers only allow traffic to the services and ports for which the application needs access.

Use Case	Solution
Browser-based HTTP/S	Access proxy
Naive HTTP cmd-line applications: We provide a client-side proxy server that supplies the platform certificate to achieve an authenticated and encrypted connection to the Access Proxy. We then direct the naive application to that localhost proxy.	Local authenticating proxy
Single TCP connection: For applications that need a TCP socket to a server, we can often arrange to establish an SSH connection to a backend bastion, and tunnel the port for the naive TCP application.	SSH tunnel and port forwarding
Many ports or unpredictable port numbers	Encrypted service tunnel
Latency-sensitive, real-time, UDP flow	Encrypted service tunnel

Table 1: Approaches to solving problematic workflows

This approach allows legacy third-party applications to more securely connect to their servers from any network and still assert the BeyondCorp invariants of authentication, authorization, and encryption.

Table 1 shows our general approach to resolving difficult workflows. For more detailed information, see "BeyondCorp Part III: The Access Proxy" [3]. In some cases, the solution shown in the table also required users to modify a workflow by running a script or providing the necessary authentication before accessing the backend resources.

Some essential framework services were noncompliant. Rather than block all migration, we temporarily opened access from MNP to the specific ports or servers for these critical services. To prevent these temporary exceptions from becoming commonplace and subverting the basic goals of BeyondCorp, we only allowed such exceptions when a service had a concrete plan for implementing and deploying a compliant solution.

As we remediated each application or use case, the automated process for analysis, verification, and migration moved more users and devices to the non-privileged VLAN. As we progressed, the network logging and analysis provided ready metrics about the number of users and devices that were successful on MNP.

Incrementally Rolling Out and Continually Refining Our Approach

The MNP simulator, analysis pipeline, and the subsequent automatic assignment of devices to the MNP VLAN was a significant software development and process creation project. As such, we developed and deployed it incrementally: we tested each phase on small groups, continuously fixed the software, adjusted user messaging when appropriate, trained the tech support team, and then gradually expanded to full-scale deployment.

The simulation and pre-analysis approach helped us avoid negative impact on users while we identified users of noncompliant workflows. However, because this approach assigned all newly provisioned, unanalyzed devices to the privileged network and didn't prevent unmigrated users from using or creating new noncompliant applications, it wasn't an acceptable long-term strategy. After reducing the number of exceptions by remediating the high volume use cases, we changed our approach to a policy of "MNP by default." Proceeding site by site, we assigned all devices to MNP, granting exceptions to devices belonging to users in job functions that use unremediated applications. This policy-based assignment marked the evolution from "*Prove* the user will be successful before migrating their devices" to "*Assume* the user will be successful and migrate their devices."

Scaling Support to Minimize Impact on Employees

Using the tools and processes discussed above, we were able to automatically identify, contact, and migrate entire groups of users. However, we also needed ways to assist people and communicate with users, both in advance of change and when something went wrong. A combination of specialized training for tech support and strategies to scale user communications and interactions was critical in shifting workflows to the new model.

Empowering Tech Support

We trained a select group of technicians in our support organization to become champions of the new BeyondCorp model and primary local points of contact. From the early stages of rollout, these techs helped affected users return to work quickly without compromising migration strategies, and also efficiently escalated appropriate issues to implementation and policy experts.

Initially, these specially trained technicians were granted more advanced access to remediation systems than their fellow technicians. As the first observers of the BeyondCorp rollout, they could anticipate what access, tools, and processes the rest of tech support would need. Additionally, they trained the rest of the support organization through global tech talks, discussion lists, brown bag lunches, and office hours. As knowledge was disseminated, we expanded system access to all of support.

Establishing local subject matter experts enabled us to engage directly with teams that had incompatible workflows. By working with one knowledgeable point of contact, teams had direct lines of communication to project experts and could collaboratively find solutions. Simultaneously, technicians were empowered and encouraged to add new temporary workarounds or fixes to internal documentation as soon as they identified problems. Distributing the power to solve problems to as wide a network as possible enabled us to efficiently share knowledge and scale support.

Self-Service Help

To avoid a flood of queries and concerns, we needed a way to minimize confusion and answer common questions without personal intervention by support personnel. When a user was selected for migration, we automatically sent them an email containing a clear timeline, an idea of how the migration would impact their work, and links to project information, FAQs, self-help, and escalation points.

We also provided a self-service Web portal that allowed users subject to business-critical time constraints to delay their migration. To answer questions and further disseminate knowledge at scale, we created an internal discussion list where people could crowdsource answers. Using analysis of common questions, we were able to quickly iterate the initial email communication and project documentation.

Throughout the rollout we also iterated and improved error messaging with a dedicated Web application. This application clearly identified common problems (for example, explaining why a user was denied access to a certain resource), provided steps for resolution, and linked to knowledge-base articles. Users could fix common issues such as group membership and certificate problems themselves, further reducing tech support requests. The Web application also helped technicians by coalescing information from the many different layers and systems into a single series of actions to solve an error.

Internal Publicity Campaign

To raise awareness of BeyondCorp, we ran an internal publicity campaign with laptop stickers, common logos and wording, and visible articles posted throughout our offices. These materials pointed to self-service help and office hours open to anyone with any question. By focusing on informing, educating, and helping, we directly built trust, goodwill, and buy-in with our users. Corporate communications and tech writer involvement were critical throughout the process—especially in the early phases, when we needed to provide a clear picture of the program's intent and impact.

Phased Rollout

BeyondCorp began as a small-scale pilot, geographically close to the project team. We increased the rollout over time by progressively targeting locations with local technical experts, eventually expanding to increasingly risky workflows and sites further from the project team. We didn't migrate critical business workflows until we had a history of success, strong buy-in from users, and confidence in our strategy. During this process, tech support load decreased as rollout size and affected workflows increased. Phasing our approach was a key element of its success.

End Result

By continually analyzing and improving all of the methods described above, we built a system that ensured the BeyondCorp rollout could scale globally without negatively affecting business, support, or user experience. Rather than simply "throwing more people at the issue," we scaled our efforts by building systems and processes to efficiently handle questions, escalations, and training. Additionally, we were able to trust our users to help us enable change by relying on information, openness, and agreement on a shared goal.

We carefully tracked support incidents caused by the BeyondCorp rollout as we moved more and more of the company onto this model. In recent months, BeyondCorp is responsible for only 0.3% of issues handled by our tech support organization. From an initial rate of 0.8%, escalations have steadily decreased with the help of improved documentation, training, messaging, and rollout methodology. Compared to similar wide-scale internal IT changes at Google, BeyondCorp has caused 30% fewer support issues.

Conclusion

There is always tension between the urgency to improve security and resistance to changing the habits of end users. When infrastructure and workflow changes threaten to impact productivity, this tension only escalates. Achieving a balance between progress and stability is more art than science. BeyondCorp's keys to success and acceptance were analysis, adaptive planning, and proactive communications.

By partitioning BeyondCorp changes into independent units, we could make progress in parallel, and user impact at each stage was minimal. Although it took years to deploy BeyondCorp across its many layers, each milestone came with benefits.

Cumulatively, we made remote access significantly easier and faster, simplified network management, and strengthened our security posture.

Creating the technology to implement the BeyondCorp security model is a challenge. Planning the rollout and managing the migration of users to that technology is just as challenging. It's essential to ensure that each transition has minimal impact on users and does not break ongoing productivity. Each successful transition brings fresh awareness of the value of the program and provides continued enthusiasm and acceptance of the program goals by both users and management. We succeeded by empowering a cross-functional team with representatives from each of the technology and implementation teams, security and policy stakeholders, and specialists in end-user support and communications.

At Google, we've been able to apply what we learned during the BeyondCorp effort to other programs and services—most notably, the new services we've recently added to Google's Cloud Platform (such as the Identity-Aware Proxy). One of the biggest lessons of BeyondCorp was the importance of phasing a project and continuing to refine and develop our strategies as we encountered additional use cases. While this article focuses on Google's specific experience, the lessons it shares can be adopted at any organization, regardless of size, so long as the effort has solid backing from relevant stakeholders.

Acknowledgments

Thanks to the following people for their help in making this article possible: Heather Adkins, Jeff Baird, Darren Bilby, John Brady, Victor Escobedo, Cynthia Horiguchi, Michael Janosko, Rob Peasegood, Dan Polsby, Val Stiris, and Rory Ward.

References

[1] R. Ward and B. Beyer, "BeyondCorp, A New Approach to Enterprise Security," *;login:,* vol. 39, no. 6 (December 2014), pp. 6–11: https://www.usenix.org/system/files/login/articles/login_dec14_02_ward.pdf.

[2] B. Osborn, J. McWilliams, B. Beyer, and M. Saltonstall, "BeyondCorp: Design to Deployment at Google," *;login:,* vol. 41, no. 1 (Spring 2016), pp. 28–35: https://www.usenix.org/publications/login/spring2016/osborn.

[3] L. Cittadini, B. Spear, B. Beyer, and M. Saltonstall, "BeyondCorp Part III: The Access Proxy," *;login:,* vol. 41, no. 4 (Winter 2016), pp. 28–33: https://www.usenix.org/publications/login/winter2016/cittadini.

[4] Capirca is a tool designed to utilize common definitions of networks, services, and high-level policy files to facilitate the development and manipulation of network access control lists: github.com/google/capirca.