PO SECURITY

Redefining PAM for the modern enterprise

Authz Control Plane for every user, NHI and AI agent

Neha Duggal
Chief Product Officer, PO Security

FORTUNE CYBER 60

Latio CLOUD SECURITY SEGMENT LEADER 2025

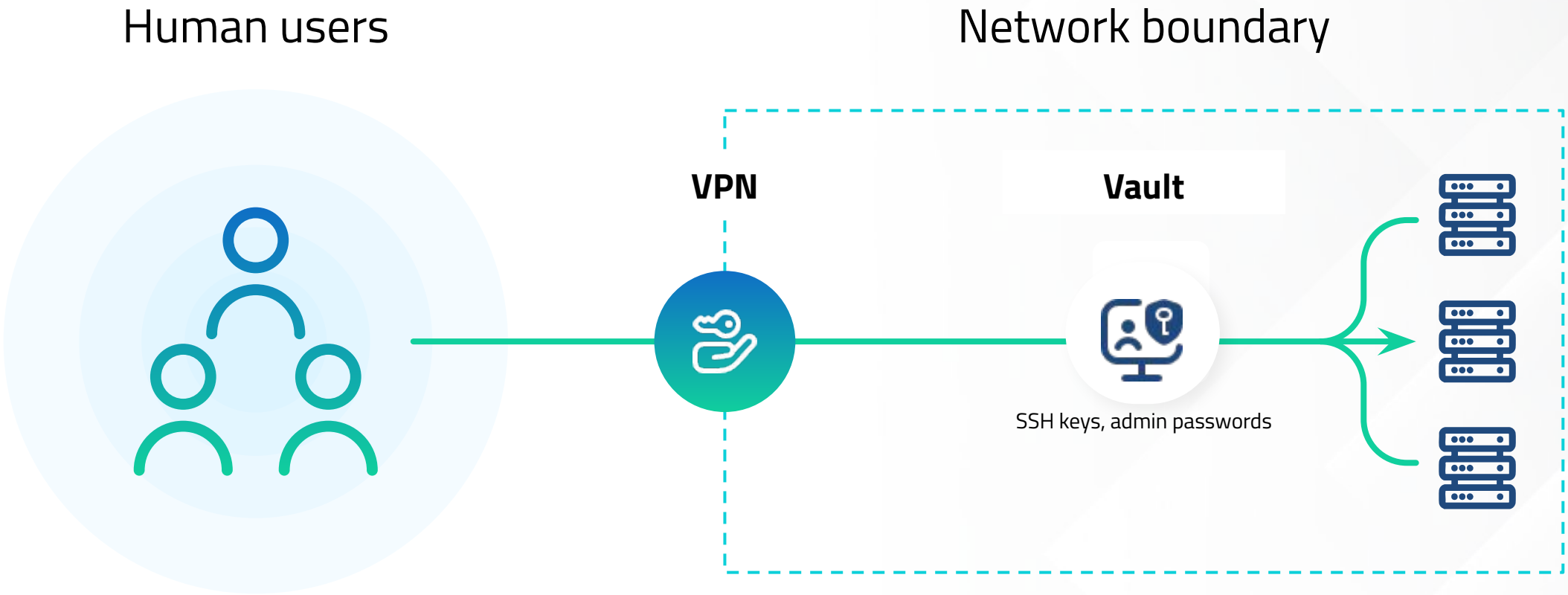
RSAC Innovation Sandbox 2024 FINALIST

PO SECURITY

	20 years ago ...	Now ...
Connectivity	**VPN**	**SDWAN/SASE**
Authentication	**Vaults** (using static keys)	**IdP** (Okta, Entra ID)
Authorization	**–** (network segmentation)	**UNSOLVED** (Authz Control Plane)

Managing privileged access *was* straightforward

Back when infrastructure was static and access was linear



Vault-led PAM was built for static infrastructure

Environment:
On-prem data centers with static servers, databases and IPs

Approach:
Vaults with stored/rotated static credentials

Authentication:
Manual authentication via SSH keys or root passwords stored in vaults

Authorization:
Per-system, coarse-grained access, manually scoped and managed

Auditability:
Focused on credential activity and session recordings for compliance reporting

Limitations:
Secret sprawl, operational bottlenecks, poor scalability in modern environments

Effective for on-prem, but fragile and slow-moving

Bastion-led PAM struggles with ephemeral cloud

Environment:
Early cloud adoption and on prem environments, network-based

Approach:
Bastions and proxies that route sessions through jump hosts with shared accounts

Authentication:
Static SSH keys, early SSO integrations via jump hosts

Authorization:
Persistent access, coarse-grained roles, limited visibility into actual permissions

Trade-offs:
Increases friction, reused credentials, fragile workflows

Limitations:
Does not enforce fine-grained, just-in-time access or govern cloud-native entitlements

Helped centralize hybrid access but lacks fine-grained cloud controls

P0 SECURITY

Resources

Cloud infrastructure

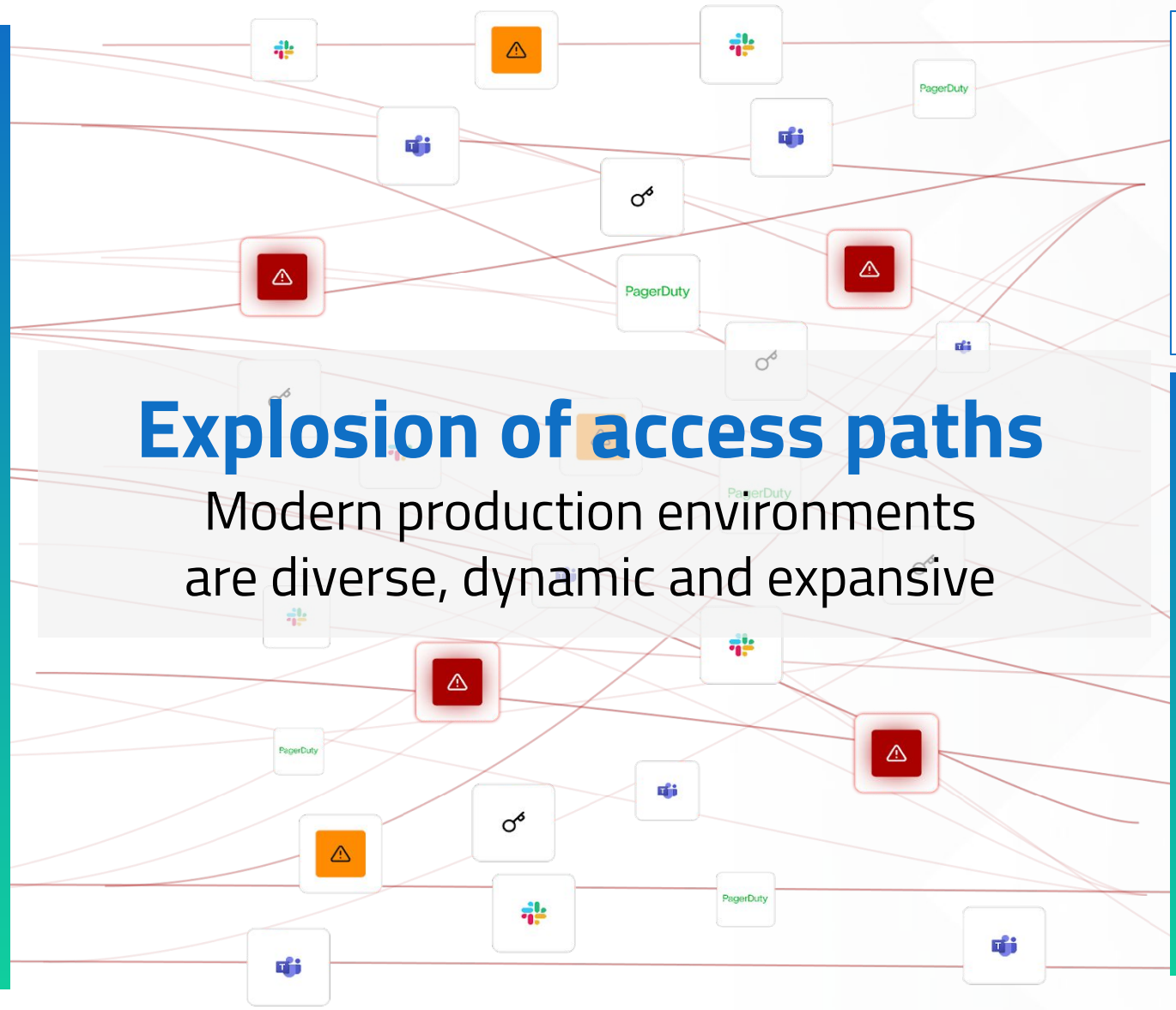
Code repositories

Servers and databases

CI/CD

Dev tools

Legacy/on-prem infrastructure



Explosion of access paths

Modern production environments
are diverse, dynamic and expansive

Workforce

Developers | Engineers

IT Security and Compliance teams

Non-humans

Machines

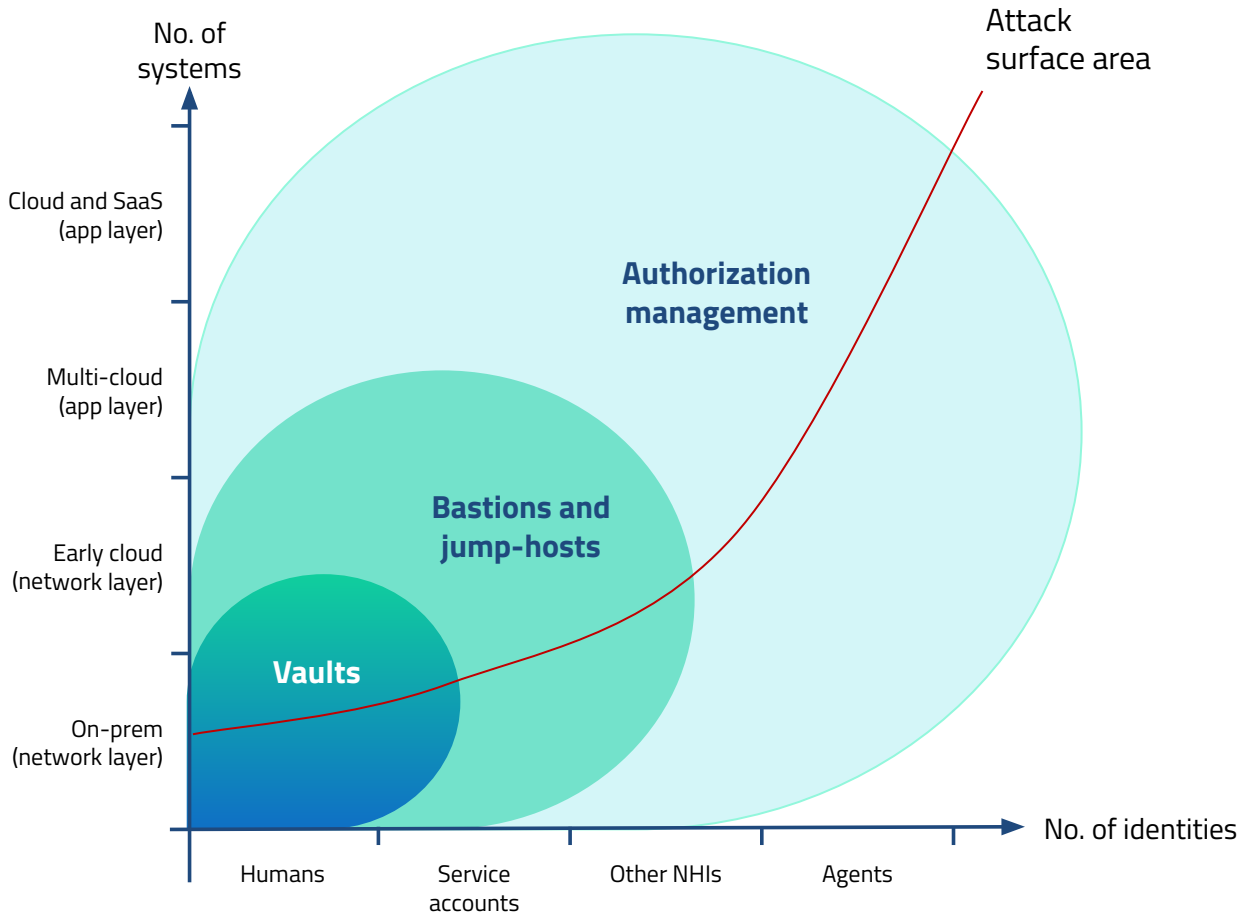
Service accounts, service principals and workload IAM

AI agents

Amazon Bedrock

Copilot

Legacy PAM does not extend to today's reality



Inherent risk
Any human or non-human with credentials has standing access to production

Audit blind spots
Shared accounts make it impossible to log who actually did what

Operational drag
Heavy infrastructure requires significant engineering effort and adds user friction

Centralizing control for every user, NHI and agent

Sensitive resources

- Cloud infrastructure
- On-prem infrastructure
- Servers (SSH/Sudo)
- Containers and repositories
- Databases
- CI/CD
- Developer tools
- Agentic platforms
- Custom applications

APIs

PO SECURITY

AUTHZ CONTROL PLANE

Just-Enough-Privilege and Just-in-Time Access

Human users | Workloads | AI agents

ACCESS GRAPH

Access paths, patterns, relationships

LDAP

BRING YOUR OWN IDENTITY

Users, groups, credentials, roles, permissions, entitlements

Use cases

- Replace standing privilege with JIT production access
- Govern NHIs with short-lived credentials and auto rotation
- Modernize PAM for hybrid infrastructure
- Simplify audit prep with session-level replay
- Control first-party agents

Getting to Zero Standing Privilege (ZSP)

DISCOVERY

Continuously surface and map access paths

POSTURE

Prioritize access sensitivity and privilege risk

POLICY

Establish least-privilege programs and guardrails

ACCESS

Enforce just-enough and JIT access at runtime

AUDIT

Automate session recordings and monitor for drift

The simplest way to secure access at scale.

Stop managing credentials and start managing privilege

P0 SECURITY

AUTHZ CONTROL PLANE

Persistent identity, ephemeral access

Shrink your attack surface

Enforce just-enough and Just-in-Time access, replacing static credentials and standing privilege

Remove governance overhead

Provision access to a user's IdP-native identity, removing shared accounts and manual reconciliation

Simplify operations

Streamline workflows with API-led orchestration, no added infrastructure to deploy or manage

Case study

splunk> a CISCO company

PO SECURITY

Use Case: JIT SSH/Sudo Access

- SSH to EC2, k8s (EKS) for 2000+ developers / CS engineers
- Fine-grained, just-in-time access controls
- Frictionless developer experience

Challenges

- Okta ASA (bastion-led PAM) limitations:
  - **Hard to maintain and scale** across hybrid environments
  - **Led to standing access** with no support for JiT escalations for fine-grained cloud access
  - **No identity-native provisioning** or support for NHIs leading to governance overhead

PO solution

Rip-and-replace of Okta ASA (bastion-led PAM):

- Short-lived, JIT SSH access
- Identity-native provisioning for users and NHIs
- Eliminated standing privileges
- Consistent developer experience with high adoption
- Future proof infrastructure, simple to maintain and deploy

Case study

CNA

PO SECURITY

Use Case: NHI Governance

- 1000+ projects in GCP; 20k+ service accounts and static keys
- Visibility into over-privileged accounts (NHIs) and stale keys
- Automate governance and risk remediation at scale

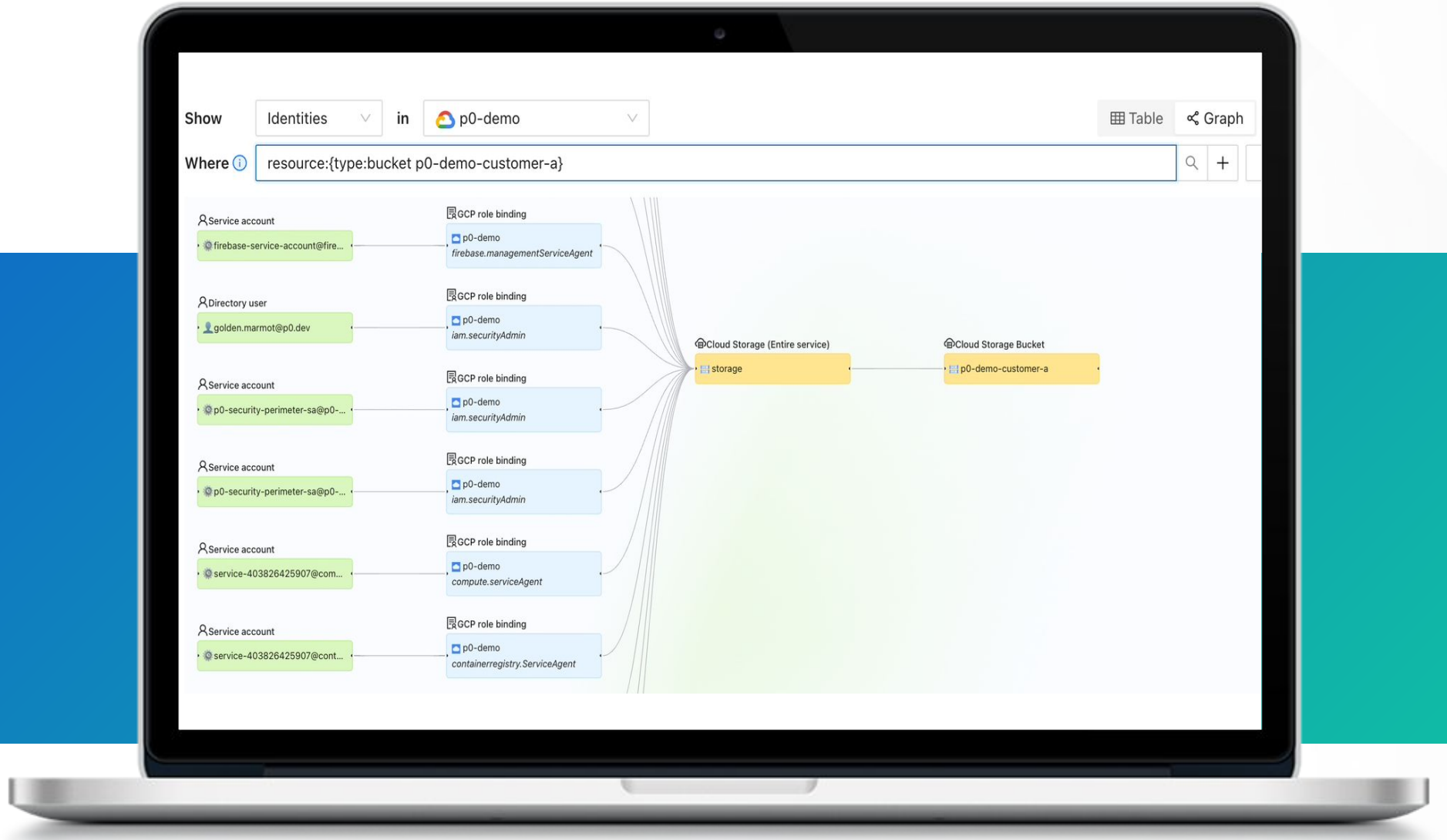
Challenges

- Native GCP SCC Premium and CNAPP limitations:
  - **Invisible NHI sprawl,** 20k service account and static keys, Policy Analyzer was paywalled, no visibility into GCP access
  - **Lack of NHI governance** with developers creating service accounts without oversight
  - **Manual overhead** of homegrown scripts, JIRA tickets, and emails for remediation and secret rotation didn't scale

PO solution

1000s of hours saved in governance overhead:

- Comprehensive visibility and governance of NHIs in GCP
- Eliminated static credentials, automated secrets rotation
- Replaced GCP's SCC Premium SKU entirely


Show
Identities
in
p0-demo
Table
Graph
Where
resource:{type:bucket p0-demo-customer-a}
Service account
firebase-service-account@fire...
GCP role binding
p0-demo
firebase.managementServiceAgent
Directory user
golden.marmot@p0.dev
GCP role binding
p0-demo
iam.securityAdmin
Service account
p0-security-perimeter-sa@p0-...
GCP role binding
p0-demo
iam.securityAdmin
Service account
p0-security-perimeter-sa@p0-...
GCP role binding
p0-demo
iam.securityAdmin
Service account
service-403826425907@com...
GCP role binding
p0-demo
compute.serviceAgent
Service account
service-403826425907@cont...
GCP role binding
p0-demo
containerregistry.ServiceAgent
Cloud Storage (Entire service)
storage
Cloud Storage Bucket
p0-demo-customer-a

A demo

Q&A

P0 SECURITY

Contact us

sales@p0.dev