

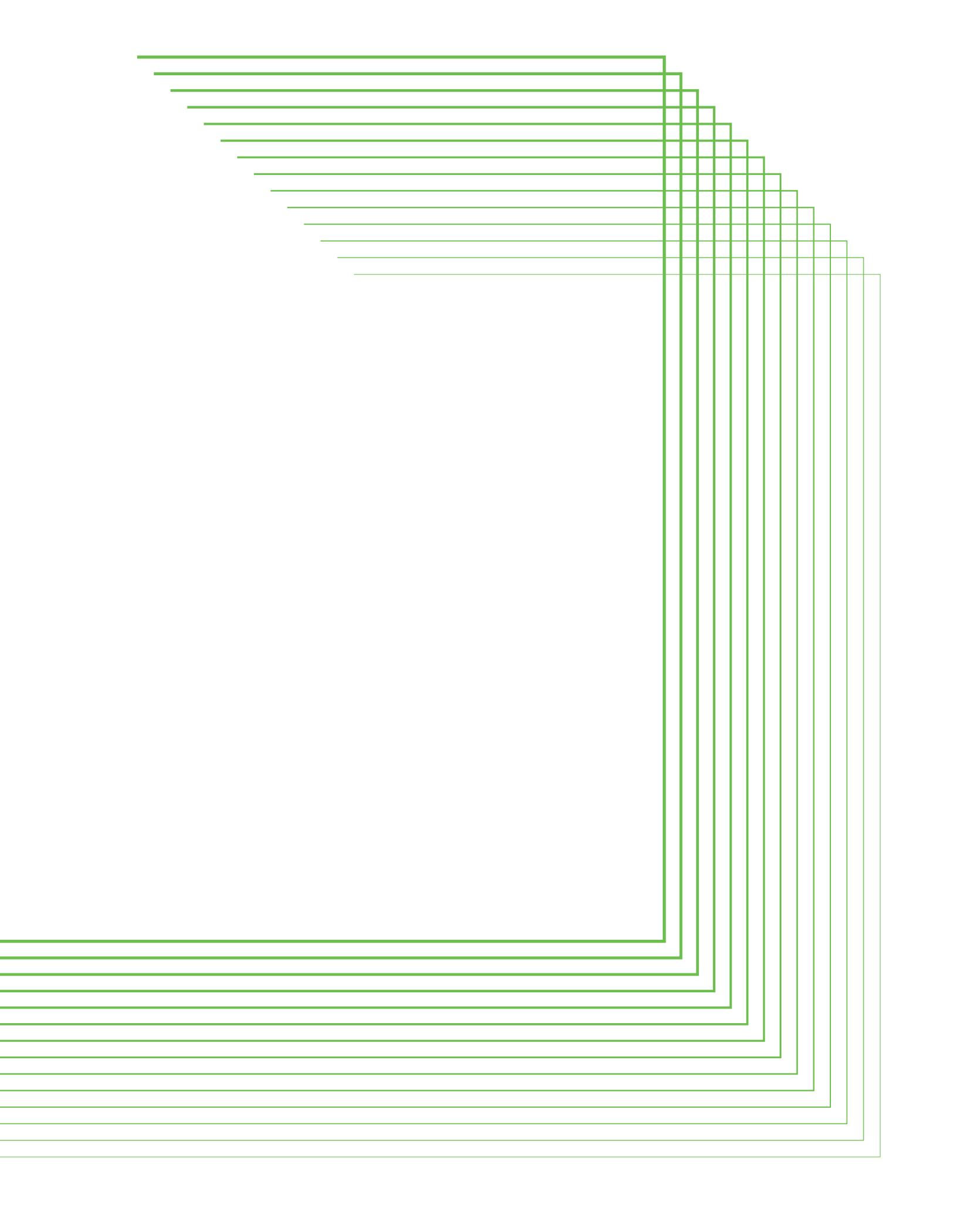
DUO SECURITY PRESENTS

Beyond the Perimeter

Securing the Modern Enterprise With a Zero-Trust Model

Duo Security is
now part of Cisco.  CISCO





Beyond the Perimeter

Securing the Modern Enterprise With a Zero-Trust Model

AUTHOR

Wendy Nather

EDITORS

Thu Pham

Andrew Hickey

DESIGNERS

Chelsea Lewis

Hafsah Mijinyawa

0.0	Securing the New Enterprise Model	1
1.0	A New Enterprise Architecture	3
2.0	The Google BeyondCorp Vision	4
3.0	Getting Started With BeyondCorp	7
4.0	Mapping BeyondCorp Components	9
5.0	Enrolling Users and Their Endpoints	11
6.0	If You Liked It, You Should Have Put a Cert On It	15
7.0	Creating Policies for Big Success	17
8.0	The Maturity Process With BeyondCorp	21
9.0	Summary	23
10.0	How Duo Beyond Can Help	25

0.0

Securing the New Enterprise Model

The new enterprise architecture is redefining the perimeter – enterprise data is now stored outside of corporate walls, and the workforce is increasingly mobile.

This dynamic environment requires a new security model to address insider risk, vulnerable endpoints, policy gaps and more.

The traditional network perimeter draws an invisible line around what belongs in the enterprise, and what doesn't. Historically, it has depended on firewalls and other security measures to protect enterprise assets, but those alone are no longer effective enough to secure the new modern enterprise architecture or a workforce that demands always-available access to cloud applications.

Another dividing line has become razor-thin: when employees access the same third-party SaaS applications for personal use that they do for corporate use (for example, Gmail and storage), the only difference is the login name. Since the perimeter is wherever access control decisions are made and enforced, firewalls won't help in this case; the identity becomes the perimeter.

So the perimeter isn't dead – it's evolved.

The change in the perimeter has been discussed for years; initially by the Jericho Forum, created in 2003 to tackle "de-perimeterisation," and now by Google's BeyondCorp – which began as an internal initiative to enable employees to work from untrusted networks, without the use of a VPN!

The idea is to shift access controls from the network perimeter to individual devices and users.²

But if you've invested a lot already into solidifying perimeter-based security measures, don't worry that they're obsolete – these new security models focus on **adding security on the inside** to ensure that the network perimeter isn't the only line of defense against attackers and security threats.

With more information released about BeyondCorp, a **zero-trust security model** is now within practical reach for many more organizations that want to acknowledge that no traffic within an enterprise's network is any more trustworthy than traffic coming from outside the network.

Enterprise Risks That Live Beyond the Perimeter

The BeyondCorp, or zero-trust security, model addresses several important risks for the enterprise:

- An attack that can bypass the firewall, or that starts on the internal network, can spread out to compromise critical systems and steal sensitive data
- When an application or system is protected with different controls dependent on whether the user is “inside the perimeter” or not, an attacker can compromise the looser set of controls
- External cloud-based applications and mobile users can face attacks that are outside of an enterprise’s traditional perimeter-based protections
- Users make the organization vulnerable by using unmanaged and unpatched devices to connect to critical systems and data

New Security Model: Theory and Implementation

In this guide, you’ll get an overview of the BeyondCorp theory and how to implement it.

For theory, you’ll get:

- An overview of the security theory of Google’s BeyondCorp and the need for a new zero-trust security model to protect enterprise assets
- An in-depth discussion of the various enterprise risks that lie beyond traditional perimeter defenses, and how BeyondCorp can address them

For implementation, we’ll cover the steps your organization can take to start implementing this new framework, such as:

- Enrolling users and their endpoints into inventories
- Using digital certificates to identify endpoints as “trusted” or “managed”
- Classifying resources (such as applications) according to risk levels
- Creating access policies based on the authenticated combination of user and endpoint

Other components include single sign-on, device inspection, the trust inference engine, and the reverse proxy that protects applications and enforces the enterprise access policies. Google describes its own migration process in the fourth white paper in its BeyondCorp series, *Migrating to BeyondCorp*.³

Enterprises often have many of these components already available and can make use of them. Duo’s trusted access product, **Duo Beyond**⁴, has simplified the process to help organizations get up and running as quickly as possible with this new security model – without requiring a multi-million dollar budget or a staff of 500 engineers.

1.0

A New Enterprise Architecture

The kernel explodes in a tiny puff of steam, turning its insides out and expanding far beyond its original size. This describes popcorn, but also describes today's enterprise architecture. With so many external services available, organizations can be partially or even fully "popped," storing their data outside of the traditional firewalled perimeter.

To make things more complicated, a mobile workforce can take its laptops and smartphones to work anywhere, far outside the enterprise's walls and network. And finally, people are using the same software as a service (SaaS) applications for both personal and work purposes. This dynamic environment requires a new security model.

2.0

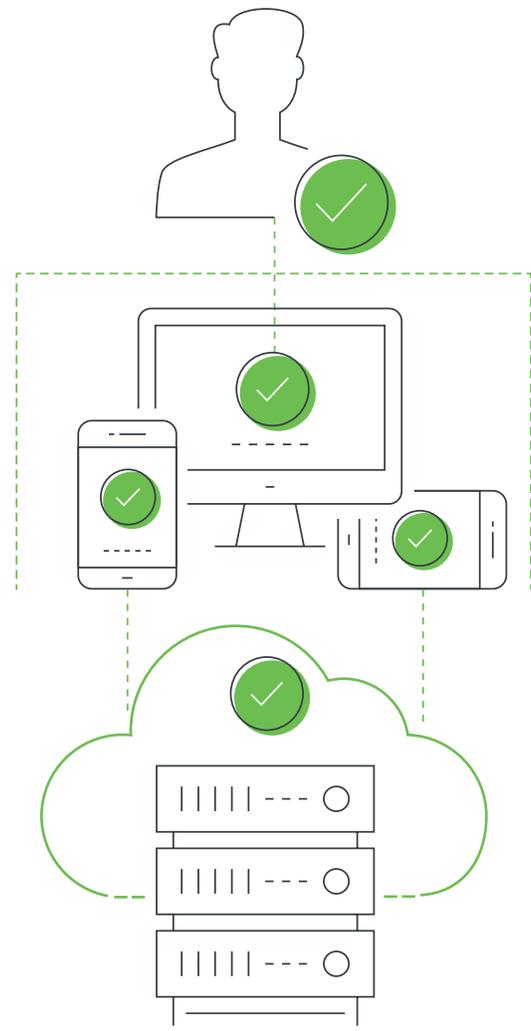
The Google BeyondCorp Vision

Google's vision is similar to John Kindervag's "zero-trust model"⁵ of information security: to assume that no traffic within an enterprise's network is any more trustworthy by default than traffic coming in from the outside. Of course, enterprises can't operate without any kind of trust; the trick is to set the conditions under which they will decide to trust something.*

Google's implementation rests on the combination of validated users using validated endpoint devices. This combination is further locked down with end-to-end encryption between these devices and the resources they access. Finally, users are allowed only the bare minimum access needed for their roles (which is also known as "least privilege").

As long as the user is authenticated with the right combination of factors, and is using an endpoint that has been enrolled and inspected for security vulnerabilities, they can access exactly those resources that they're allowed to by a centralized proxy.

* The word "trust" itself can trigger semantic arguments. Kindervag argues that trust equals vulnerability; since trusting connotes allowing access without verification, his model is called "zero-trust." For others, the word "trust" is used to describe allowing access based on prior verification. In this discussion, we use the latter version to describe users and devices that have been authenticated and that comply with policy, and are therefore trusted to allow them access to resources.



What Risks Does This Approach Address?

The biggest risk, of course, is that an attacker breaks through the perimeter and then has free rein within the trusted internal network. Google specifically referred to the “Aurora” attacks⁶ as an example of what prompted BeyondCorp.

The Insider Risk

The other risk with a fully trusted internal network is that you don't have to start by breaking through the perimeter; if you're an insider planning malfeasance, you're already there. The traditional way to deal with this risk is to segment the network. But creating segmentation after the fact can be a major project, disrupting traffic and application tiers, and in many organizations, it never gets done. **And let's face it – a sufficiently successful outsider looks exactly like an insider.** An external attacker will use the same means to get in that work for the legitimate user, so you have to make sure to limit what everyone can do.

Policy Gaps

Another risk is the attacker exploits the gaps between different policies or enforcement that apply to the same asset. If the same confidential data is available in two different systems using different types of authentication, the attacker will go after the one that's easier to reach – either because it trusts something else you can leverage, or because that one authentication method has a flaw in it.

Attack Scenario: 2FA Workaround

For example, let's say one database requires two-factor authentication (2FA), but the same data is available in another application that doesn't require 2FA – and it has weaker passwords that are shared with a third system. An attacker would try to break into the third system, grab someone's username and password, and use it to get into the non-2FA application. You can prevent this kind of arbitrage by trusting nothing by default and making everyone pass the same tests each time.

Vulnerable Endpoints

A common risk every organization faces is the vulnerable endpoint, where out-of-date software contains security flaws attackers can exploit. At the very least, endpoints should be up to date on the operating system (OS) and plugins they need to use. This isn't always practical due to legacy software that is dependent on older versions of other software, or that is only certified by the vendor for a particular set of infrastructure. But users who simply don't get around to upgrading – especially on their personal devices – are a security headache for the enterprise.

Addressing Risks With Application Policies

With a centralized access proxy, you can have one set of policies for each application, regardless of where the system or user is located. A third-party SaaS could have the same trust requirements for access as an internal web application. This is important because attackers try to come from the “most trusted” location, whether that's a known IP address, an “internal” system, or a favored geographic area. With the BeyondCorp model, it's the combination of validated user and endpoint that earns the trust, not the network.

Note: You can have different requirements based on whether it's an internal or external app, but once you start making that distinction, you're back on the road to destroying that security model you just tried to implement. Make sure your policies are based on business criticality and confidentiality, not on “inside” versus “outside.”



3.0

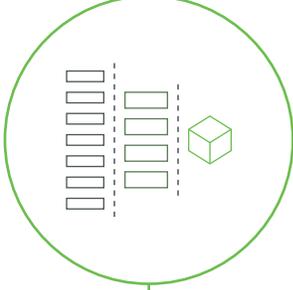
Getting Started With BeyondCorp

If you're already in a hybrid environment – with some of your infrastructure located on-premises and some hosted in the cloud – it's time to think about how you could potentially use the BeyondCorp model to rebalance your security policies to extend to cover assets that aren't within your perimeter. If you have a large network and haven't been able to segment it as much as you'd like, or for tighter control, the BeyondCorp model offers a chance to focus on combining user and endpoint verification with encryption. Think of it conceptually as a triad: authenticated users, verified devices,

and applications protected through discrete policies that are appropriate for the types of data they contain.

The good news is that you don't have to do everything all at once. While Google's description of a comprehensive migration sounds daunting, moving to this different concept of security also works when you do it incrementally. **Remember, you're not actually getting rid of the perimeter controls; you're raising the level of security on the inside so that it looks more like the outside.** Any progress is a significant improvement.

Here are some of the high-level steps to plan for:



1. **Enroll your users and their endpoints.** This may require a discovery process, since users might not always be using the corporate assets you assigned them. By routing those users to a popular application through an authentication gateway such as the ones Duo provides, you can get an inventory on the fly, and discover which devices are actually connecting to your corporate systems.⁷
2. **Deploy certificates** to the user endpoints you want to identify as “managed” or “trusted.” The level of trust is up to you, but for some organizations it means these endpoints are officially supported and maintained by the enterprise; for others who embrace Bring Your Own Device (BYOD), it means that you’ve done the initial hygiene check during enrollment and validated that the device belongs to an authorized user.
3. **Classify applications according to risk levels** so you can enforce different access policies for each. Some resources require global access and contain less sensitive data, such as internal announcement pages, employee directories, and cafeteria menus. Other resources, such as financial systems, HR, customer or patient data, or intellectual property, would have more restricted access.
4. **Create access policies** based on the requirements for each application or system you want to protect. These policies can include how often you want users to re-authenticate; whether they can use personal devices; and which level of hygiene you want to enforce. These policies can be adjusted dynamically based on security events. For example, if a new vulnerability is being actively exploited in a particular endpoint OS or plugin, you can block affected users until they update it.⁸ This drives users to update on their own rather than waiting for IT to organize a scheduled maintenance window (no more Terror Tuesday, when patches are released!).

Remember, you’re not actually getting rid of the perimeter controls; you’re raising the level of security on the inside so that it looks more like the outside.

As a result, you will get **better visibility and a tighter set of controls** over what your users and endpoints are accessing, regardless of where they are. By adapting to the new reality – that applications, users and devices can change locations at the drop of a hat – you’ll be able to maintain a more consistent level of security and user experience.

Mapping BeyondCorp Components

If you were building your own BeyondCorp, what components would it entail?

User/Group Database

To keep the information and attributes about your users, and to group them where necessary according to organizational, geographical or other aspects.

Device Inventory Database

An up-to-date repository for information on all devices you allow to access the network, including type, purpose, network addresses, asset tags, components, configuration, and responsible user or maintainer.

Managed Devices

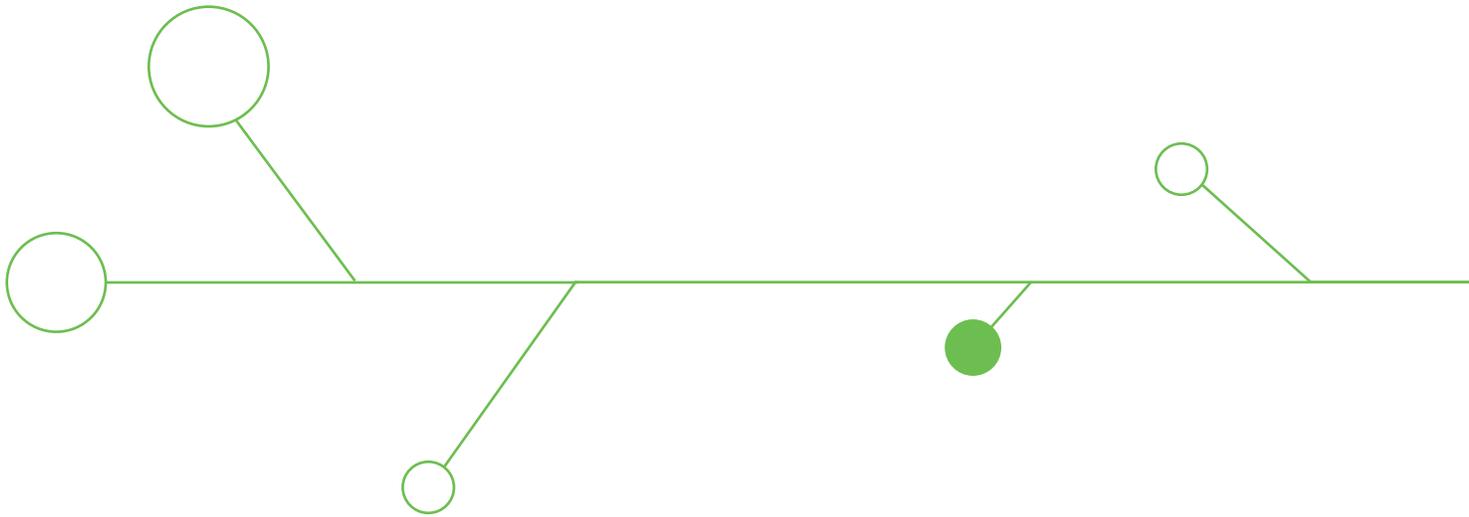
If you do not allow BYOD, this will be the whitelist of corporate-owned devices you allow to access your resources. If you are using an enterprise asset manager such as LANDESK, Jamf, or Active Directory, you probably have this list already.

Certificate Issuer

This is used to mark your managed or otherwise approved devices with a client-side certificate. Depending on which types of certificates you plan to use, the public key infrastructure (PKI) for this may already be part of another security product.

Trust Inference

Deciding what conditions will cause you to place or lose trust in a given device (such as hardware changes). The trust inferrer will rely on a steady input of data from the sources you choose. Google's description includes checking to see whether the device is encrypted; whether it has all management agents working; whether its software is up to date; and whether all of the information about that device is current.⁹



Access Control Engine

The repository of all your access policies, such as “only this group of users, together with their up-to-date, assigned and managed devices, may use this sensitive application.”

Access Proxy

The part that carries out the connections and policy enforcement. Google’s description of its own access proxy can be found in their white paper, *Beyond Corp: The Access Proxy*.¹⁰ It is much more complex and handles traffic load balancing, Transport Layer Security (TLS), authentication, access control list (ACL) evaluation, authorization, and self-service for users.

Single Sign-On (SSO)

Make it much easier on your users by providing one portal for access to all of their applications and systems.¹¹

Other Components You Will Need

Google’s BeyondCorp architecture doesn’t explicitly mention **two-factor authentication (2FA)**¹² or **multi-factor authentication (MFA)**, since in Google’s case, it’s integrated with its own **identity provider (IdP)** service, but it’s vital to the strategy of making it harder to

compromise an account. In addition to MFA, if you don’t have a centralized system for identity management, this will likely make the BeyondCorp implementation more complex.

5.0

Enrolling Users and Their Endpoints

The first step to take toward the implementation of this new enterprise security model is enrolling your users and endpoints. Enrollment usually involves a combination of inventory, inspection and verification. You create a list of entities to be entered into the system you'll use to authenticate them and grant them access (in this case, a list of authorized users and a list of endpoints they're using).

You can use bulk enrollment – that is, you can use the list to create entries for each one without requiring your users to do anything – or you can use self-enrollment, where the users make contact and supply shared data so you can recognize them.

Inventory: What corporate-owned or managed endpoints do you have, and who are the authorized users? What other endpoints are you going to allow?

Inspection: Does it conform with your security requirements? (Note: enrollment isn't the only time you'll inspect the endpoint; it should happen automatically with every access decision.)

Verification: Is this the known user who is presenting the endpoint for enrollment?
Is this the same endpoint you have in your inventory?

Ultimately, BeyondCorp is a new way of thinking about security and trust. Applying the “zero-trust” attitude to every enterprise design and process is the real peak maturity on this curve.



Inventory

Start with what you know you have. Regardless of whether you pre-enroll devices you're aware of or if you let users enroll them individually, the process needs to have controls in place so you have visibility over which assets you expect to see. Most enterprises have some sort of IT asset and configuration management in place, whether it's Active Directory, LANDesk, Jamf, or other products.

Starting with a basic list of hardware tags (or phone numbers) and assigned users will let you recognize corporate systems as opposed to personal ones. For best coverage, plan to start with bulk enrollment, and then fill in the gaps with self-enrollment, because you'll need to plan for ...



Discovery

An important issue within the inventory process is discovery:

- Are you sure you know all your users and all their endpoints?
- How will you handle new or forgotten users?
- How will you deal with changes in endpoints?

One way to handle this is to put discovery into the enrollment workflow, and place it where the users have to go to access something important. **Make sure they will access it early and often.** Everyone accesses an HR system sooner or later when they need to download their tax forms, but that will only be once a year. It's better to place discovery in front of something they use all the time, such as email, reference wikis or directories.

Don't neglect discovery. Many organizations have had policies against using personal devices on the corporate network, but they found out through discovery that literally hundreds of users were doing it anyway.



Ensuring Trust With User-Device Pairs

What do you do about enrolling shared devices? Remember that it's the combination of user and endpoint that you decide to trust, so you can't just decide to trust all devices independently of the users; an attacker could take control of a given endpoint and leverage any other known username and password to get access. To avoid this, you need to enforce user-device pairs by adding **multi-factor authentication**.

To break in, the attacker would need to have the username, password, access to the second factor (such as a software token on a phone), and the endpoint — making it more difficult to get unauthorized access with every piece you add to the puzzle.

So make sure you have an entry only for those combinations of user and device you expect to see. Sharing may not happen that often, but when it's needed, the enrollment process should accommodate it.

To break in, an attacker would need to have the username, password, access to the second factor (such as a software token on a phone), and the endpoint — making it more difficult to get unauthorized access with every piece you add to the puzzle.



Verification

As discussed, it's the combination that earns the trust, so you need to make sure to authenticate the user during self-enrollment. From that time forward, the user will be re-authenticating (with more than one factor!) to the access proxy, along with that user's assigned endpoints.

How do you uniquely identify an endpoint? It's harder than it sounds, particularly when hardware components and their IDs get replaced. Google described how it used a combination of observed and prescribed data to do this. Organizations will probably end up using whatever data they can most easily obtain and match; whatever you do, aim for consistency. Google decided it would be the certificate that was the arbiter of endpoint identity: if the certificate didn't match what was enrolled, it didn't matter whether any of the system components matched.





Inspection

It would be great if the user's endpoint was in a known clean state when it was enrolled, but this isn't always possible. At the very least, you can decide on what hygiene and configuration settings you want to see:

- No known dangerous apps installed
- Encryption and lock screen turned on
- Updated operating systems and plugins

If you already have an agent installed on the endpoint, you can get whatever data it provides. If you don't, or if this is the first time you're seeing the device, you'll need something that can perform this inspection without an agent.¹³

When you're building a device inventory and collecting data on the state of those devices at scale, you'll need to build and manage the data pipelines separately. Google's BeyondCorp paper described how its multiple device inventories required collecting and normalizing everything into a meta-inventory to feed its downstream components.¹⁴ Configuration data, event log data, information from security infrastructure such as endpoint monitoring, anti-malware and SIEM can all potentially have a role to play in how you infer the current security state of the device at the point of an access request.

If You Liked It, You Should Have Put a Cert On It

What Does “Trusted” Mean?

Trust policies and their requirements will be determined by each organization. It used to be that if a user provided the correct login name and password, it proved that the right person was at the keyboard – and we all know how well that worked out.

We ran into the same problem with devices: because it was on the corporate network, we assumed it was supposed to be there, and it got access to anything it asked for. Both of these “tests” failed for a number of reasons:

- Stolen passwords
- Spoofed network addresses
- Compromised endpoints
- The ability to spread out laterally to other vulnerable systems

Now, the path to trust needs more checkpoints, such as authentication factors and conditions placed on the device. One of these conditions can be whether it's a managed, corporate-owned endpoint.

Why “Managed?”

A managed endpoint is presumably owned by the enterprise, or at least known: it may be tracked as part of an inventory, enrolled in a configuration and patch management program, and monitored for security events. For this reason, you may choose to trust it more than you would trust an unmanaged, personal device.

Many organizations have the policy that only the endpoints they own and assign to staff can be used to access business data. However, this policy can be difficult to enforce, especially if there's no way to check. There are different ways to try:

• **Virtual Private Network (VPN) Software**

If the endpoint has the VPN client installed, it's assumed to be an approved and managed asset, so whoever is using it will be allowed to access the internal network from the outside (say, at home, or from a hotel or coffee shop). SSL VPN software doesn't require an installed client, so it provides more convenience for the user, but it also removes that implicit enforcement.

• **Network Access Control (NAC) Software**

With common port-based NAC, if the endpoint has an 802.1x certificate installed, it's assumed to be an approved and managed asset, so whoever is using it will be allowed to connect to the internal network from inside the building.

• **Mobile Device Management (MDM) Software**

Enrolling mobile devices into this system allows you to enforce configuration policies by installing an agent.

In each of these cases, you've marked the endpoint as trusted by installing something on it (or given it a second factor, "something it has"). What else could this marking mean for a "trusted endpoint?" It could be used for endpoints that don't belong to the organization, but that have been vetted (for example, a consultant's laptop that has been scanned). The important point is that **you've seen the device before and expect to grant it access**, as opposed to endpoints that are trying to access your applications that you've never seen before and may be used by attackers. Either way, it can be used to control which devices can access your business data.

Unfortunately, if you can't manage an endpoint, it's generally more difficult to convince that endpoint owner to let you install something on it. A certificate or other method of fingerprinting is lightweight, and may be more acceptable than installing and running software. Still, the key requirement is to make that marking unforgeable and prevent it from being copied to another device.

Since you will be making trust decisions based on the marking's presence or absence, it functions as yet another authentication factor, and it needs protection in the same way you must protect the primary user credentials (username and password) and the second factor (such as a one-time password, U2F device¹⁵, or push-based authentication).

All the Single Endpoints

In Google's BeyondCorp framework, certificates offer a way to identify the device as managed. You can take it a step further by including device and user data in the certificate, tying them together so **neither ones credentials can be leveraged alone**. You can set policies so users must use known and approved endpoints to access the most critical data and applications (for example, privileged users must use a corporate-owned device).

Likewise, even if a user loses credentials to an attacker, the attacker still needs to use a valid endpoint belonging to that user to get into an application – it's not enough to have the username and password with a different corporate device. **Trusting the devices only if they're with the right user** is a new step towards tighter security the BeyondCorp model makes possible.

Creating Policies for Big Success

Your access proxy takes on the role of enforcing access to corporate resources, regardless of whether they're outside or inside your traditional perimeter. Enforcement strategy is one way we express risk tolerance; rightsizing those policies depends on factors such as sensitivity, threat, user community, regulatory requirements, and any number of other things. And enforcing policies consistently for both sides of the firewall is a key tenet of the BeyondCorp model.

Tiers of Trust

A major drawback to the classic network perimeter security model was organizations tended to have one level of trust everywhere on the inside. Building in different tiers required network segmentation that was often too complex to implement. With BeyondCorp encouraging a new look at separating out levels of trust at the application layer, it's important to determine where your most critical and sensitive data, applications and control functions are so you can protect them with higher trust requirements.

Some examples of the most critical accesses might be:

- Control systems, which are used to grant or change access (such as administrative consoles, configuration management systems, identity stores, certificate authorities, and authentication servers)
- Systems that manage availability (load balancers, backups, HVAC systems)
- Financial and human resources applications (including payment systems)
- Research and engineering systems holding intellectual property
- Applications and storage for customer, patient, student or citizen data

To access these, users and their devices may need a higher level of trust, which means they need to pass more tests and comply with stricter requirements. Start with a baseline level of trust for all users and all devices regardless of what they're accessing, and then add more to reach the level of risk management you need for access to the most sensitive tiers.

Wielding Access Policies

Your access policies are much more flexible than a stop-or-go approach. Like a multi-use tool, you can use them to bludgeon, nudge, slice or tap. Here are some of the types of access policies to consider.

- Warning** Strongly recommending or requiring action at some point in the future.
- Blocking** The heaviest of the policies, preventing access entirely.
- Logging** Taking note of a condition or event.
- Mitigating** Loosening or reversing the effects of another policy based on certain risk scenarios.
- Responding** Taking short-term actions to react to a particular situation.



Warning

You can use warning policies to drive behavior. **A warning is a reminder with a little weight behind it: if you don't do what the reminder says, sooner or later, you will suffer a consequence.** For example, most organizations put a grace period in their policies to give users time to update their software before they're either forcibly upgraded, or they're blocked until they catch up. So, if a new version of a particular browser comes out, your users have one month to upgrade to it, or be blocked after that grace period has expired.

If your warning policy has no consequence attached to it – that is, the user may override or ignore the warning every time – then it's little more than an irritating flag that pops up in the middle of that user's workflow. And if the warning is about something the user can't take action on, it's even more frustrating.

If a system can't be updated because of some other dependency, then the warning serves no purpose and merely trains the user to ignore the irritant. When it comes to access policies, make sure you ask for a concrete action that's within the recipient's capability, and be prepared to take an enforcement action within a reasonable time period based on your risk estimates.



Blocking

A policy for blocking is best suited to situations where you don't have wiggle room. For example, many organizations want to block access to critical applications from non-managed personal devices. Either the device is corporate-owned and "blessed," or it isn't.

Many organizations are interested in blocking based on geolocation.¹⁶ If you are quite sure you never need to allow access from certain regions, a general block will work, but that's not always an option if you do business with them or you have users who travel there.

Bear in mind that blocking based on IP address or a derived geolocation won't necessarily protect you from a determined attacker who can spoof those things, but in general, it can work as a filtering mechanism for large segments of the population who should not even be trying to authenticate to your applications.



Mitigating

There are some policies used to mitigate the effects of other policies. Multi-factor authentication is an important security control, but some users don't like having to use it every time they need to use a resource. An organization may decide that after the initial authentication to a system, the risk is low enough to delay re-authenticating for a certain period of time.

One example of this is “remembering” a user or device, or both. Most services that offer MFA allow each user to “remember this device for 30 days,” for example.¹⁷ Setting that time period involves making a risk calculation on your side as to how likely a user's device could be lost or stolen; it's a tradeoff against convenience. The same principle applies to application session length – how often you need the user to re-authenticate if, say, they don't lock their device when it's not being used.

Another possible mitigating policy is to skip the second authentication factor for devices on particular trusted network segments. However, once you begin trusting something more when it's on the “inside” of your network perimeter, you're in danger of undermining what BeyondCorp is all about: the idea that you shouldn't trust the inside any more than the outside. So use these “loosening” policies with caution.



Responding

Organizations can also put temporary policies in place to respond to a particular event. If a critical vulnerability is announced for a plugin, for example, and you know your users are at risk because the vulnerability is already being exploited, then you may want to block users until they get the patched version installed.¹⁹ In other words, you would shrink the time window or grace period of a regular policy for just this one situation.

Other response-type policies could include placing geolocation or network restrictions on a device someone can't find – until they either find it again, or determine it was really lost or stolen. If they find it where they expected, they can use it again right away, but if someone else tries to use it from a different location, they won't be able to access corporate data with it.

The same idea applies to an employee who is leaving; while they work out the notice period, their access policies might be tightened so they can't access applications that contain large stores of sensitive data.

Managing Exceptions to Policies

For every policy, there is an equal and opposite exception. There may be good reasons why a set of endpoints can't be fully updated:

- They don't have regular access to enough network bandwidth
- They're dependent on one application that requires a certified stack to operate
- It's too politically sensitive to block your CEO even if she rooted her own phone
- You never allow traffic through an anonymized proxy, except that one time when an employee is traveling abroad and can't access some home resources any other way

Strictly speaking, a firewall is an exception in itself: you know it's risky to connect to the internet, but you do it anyway because there are strong business reasons to do so. The firewall embodies and manages those exceptions ("Okay, but only for web applications ..."). For your users, have a workflow process ready to receive exception requests, and for yourself, be ready to record and approve them with reminders to follow up if the policy exceptions are only temporary.

Another reason to add policy exceptions is to introduce change over time. You may have stricter policies in place for a smaller user group to try them out before deploying them to the rest of the population. Exceptions can also help troubleshoot all sorts of problems if you suspect they're being caused by an access policy: for the one user, you create an exception for each policy you know is being applied to them, until the guilty one surfaces (or all of them are ruled out).

From the Big to the Small

Access policies can be used not only at the network and application levels, but also at the device and behavior levels. You can start by blocking access to whole categories of outliers (such as banning any use of an insecure browser), and then work your way toward requiring better endpoint hygiene, such as screen locks. In some cases, you can require your users to validate their 2FA confirmation with a fingerprint, so even if an attacker has access to the unlocked device, they still can't finish logging into the application.

The most important thing is to carve away at the devices, software, sources and behaviors you know you don't want to allow, thereby reducing your exposure to attacks. Changing the security lifestyle of an organization takes dedicated work, but once you have the controls fit more closely to where they belong – the users, their devices and the applications – you'll be addressing the gaps in today's traditional security paradigm and moving *Beyond* it.

The Maturity Process With BeyondCorp

Rome wasn't built in a day, nor was BeyondCorp. Google describes in detail what it learned from the deployment process (ACLs are complicated), and building the architecture from scratch offers many such learning opportunities. As mentioned before, organizations don't have to consider it a cutover-style project, but rather an evolution as new controls are added to fill the gaps in the old ones.

Some proposed stages of maturity are listed below.

EARLY MATURITY:

Building Inventories

This is where you should start if you don't already have centralized identity management. User, application and local system accounts need to be tracked in one place, even if they're not managed from there. You don't have to collect them all at once – strictly speaking, you could just collect the first set of users from the first application you plan to protect in the BeyondCorp fashion – but if you plan to implement BeyondCorp throughout your enterprise, you'll eventually end up with all users in the same repository.

The same goes for devices: know what you have, sort out the ones you manage, and track changes to them. The output from this inventory will help you decide what device policies you want to enforce (can you require encryption for all of them?).

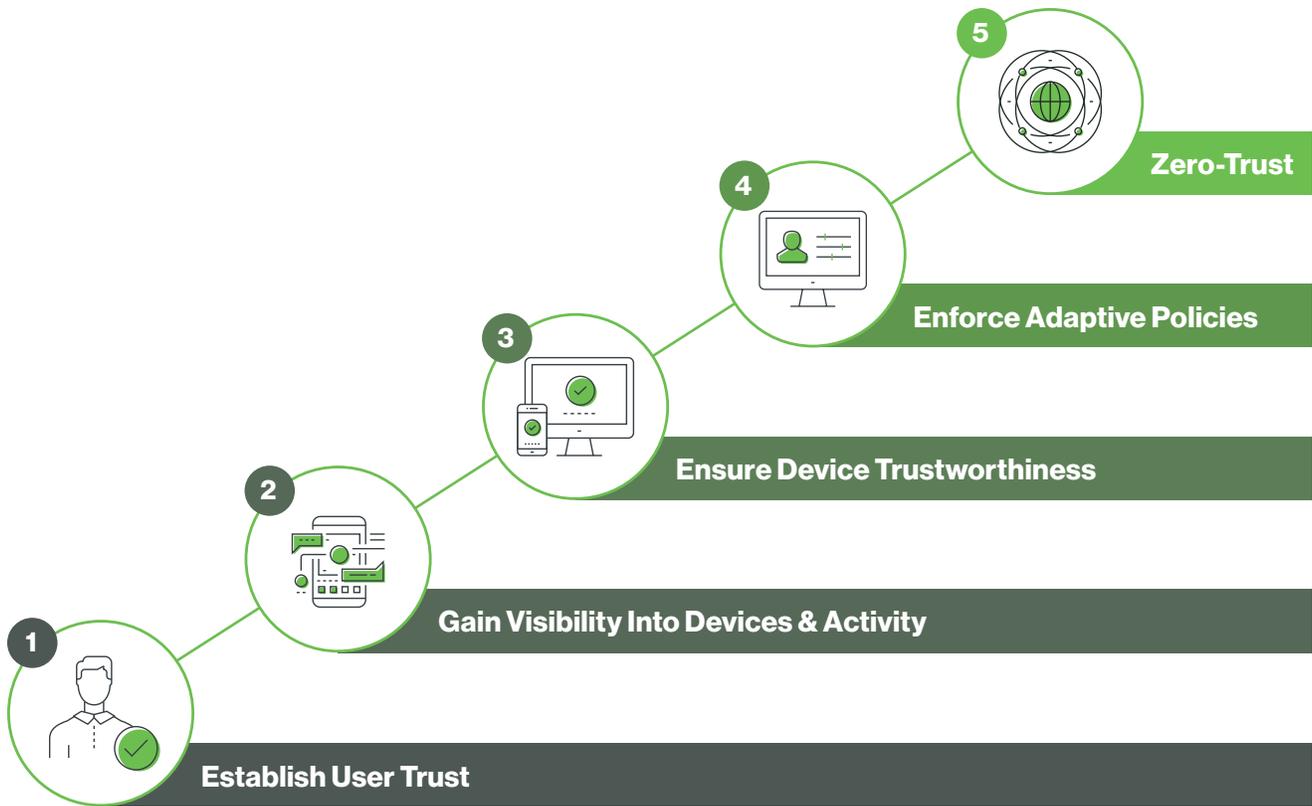
Who can reach this stage? Organizations of any size, although for larger ones it will take longer, usually due to decentralized user and asset management.

MID-STAGE MATURITY:

Core Deployment

As you start increasing the level of control you have over access to your most important resources, you'll grow the groups of user-device pairs you manage. This generally happens on an application-by-application basis, since the authentication for each one will need to move to the access proxy. An organization with a core deployment using the BeyondCorp framework might have its system administration and infrastructure tools (such as Active Directory) migrated, along with financial systems, human resources, and applications using intellectual property or regulated data.

Who can reach this stage? Any organization that is able to create policies, issue endpoint certificates, set up the access proxy, implement MFA, use an identity provider for primary authentication, and change the domain name service (DNS) entries for the relevant applications. This requires a certain level of technical expertise as well as control over the environment, so smaller organizations that outsource all their support may run into obstacles here.



PEAK MATURITY:

All the Users, All the Devices, and All the Apps

Can you ever make the network irrelevant? That's the end state of BeyondCorp, and although that theoretically means enterprises can ditch their traditional firewalls, practically speaking, it's not likely to happen.

Who can reach this stage? Any enterprise that is still hosting any connected infrastructure will be responsible for protecting it against many other sorts of network-based attacks (such as denial-of-service), not just authentication-level ones.

9.0

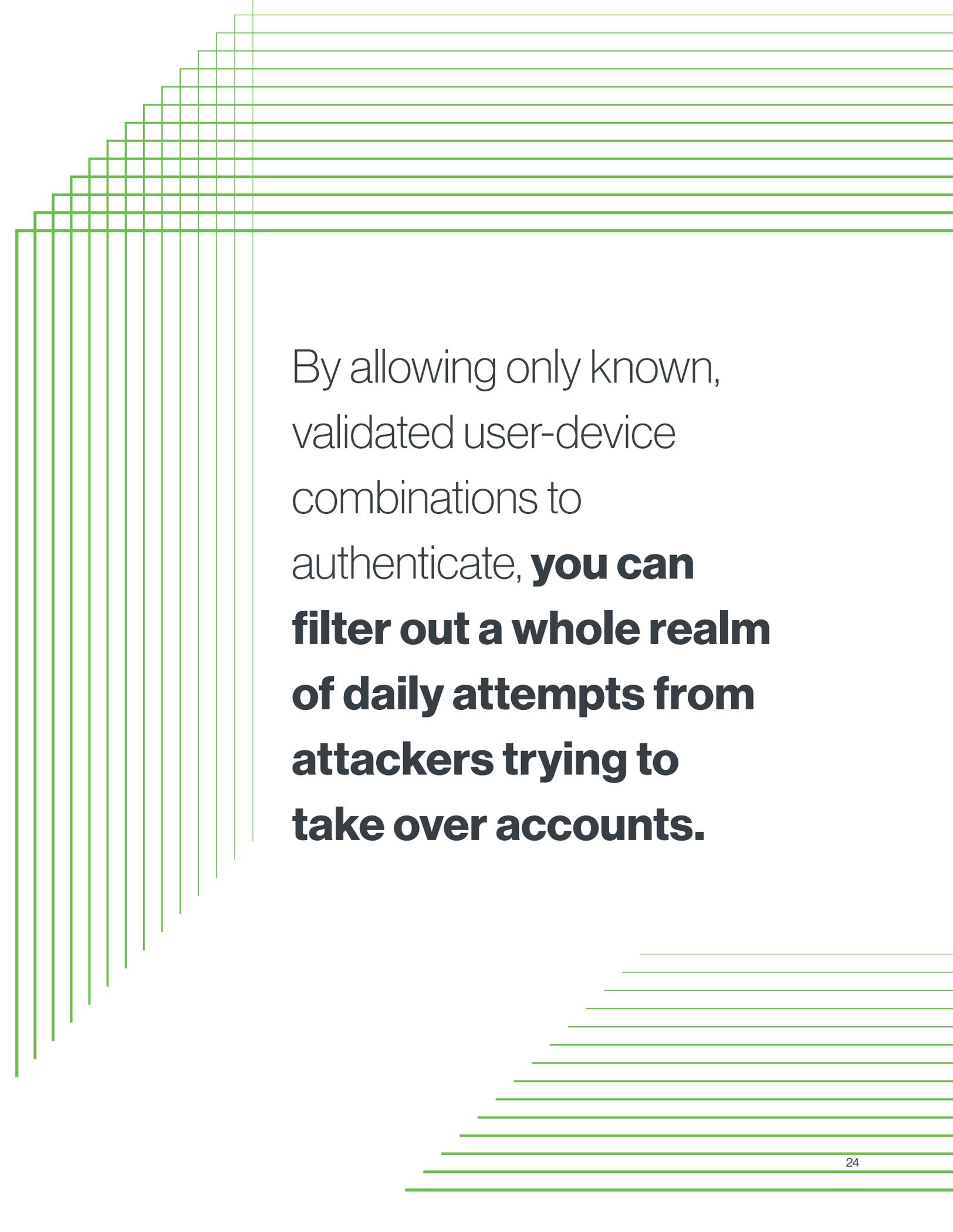
Summary

BeyondCorp is not a silver bullet that will take care of all risks; it's a way of increasing the security level of what used to be viewed as a "safe" environment. Until you remove the complication of legacy systems, software and protocols, or move all the hosting off-premises, you'll need your traditional perimeter to continue standing its watch.

By contrast, cloud-first organizations can use the BeyondCorp model to increase the control they have today over access to third-party SaaS applications. **By allowing only known, validated user-device combinations to authenticate, you can filter out a whole realm of daily attempts from attackers trying to take over accounts.** Anyone can try to log into a public SaaS

application today with a stolen set of credentials - but it'll be harder if they have to use that user's fully-patched endpoint and their 2FA on yet another device.

Ultimately, BeyondCorp is a new way of thinking about security and trust. Applying the "zero-trust" attitude to every enterprise design and process is the real peak maturity on this curve.

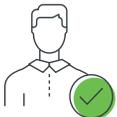


By allowing only known, validated user-device combinations to authenticate, **you can filter out a whole realm of daily attempts from attackers trying to take over accounts.**

How Duo Beyond Can Help

Building an entirely new infrastructure to accommodate a modern way of thinking takes a long time, and for many enterprises simply isn't practical. At Duo, we've shortened and simplified the path by building a platform called **Duo Beyond** that allows you to base application access on the trustworthiness of the user and their devices, instead of the networks from where access originates.

Duo Beyond resolves several security challenges through a single security platform so your organization can quickly and securely adopt a zero-trust model:



1. Establish Trust in User Identities

Verify the identity of all users with Duo's easy-to-use, strong two-factor authentication before granting access to corporate applications and resources.



2. Enhance Visibility Into Users' Devices and Activity

Gain visibility into every device used to access corporate applications, whether or not the device is corporate-managed, without onerous device management agents.



3. Ensure Trustworthiness of User Devices

Inspect all devices used to access corporate applications and resources at the time of access to determine their security posture and trustworthiness. Devices that do not meet the minimum security and trust requirements set by your organization are denied access to protected applications.



4. Enforce Risk-Based and Adaptive Access Policies

Protect every application by defining policies that limit access only to users and devices that meet your organization's risk tolerance levels. Define, with fine granularity, which users and which devices can access what applications under which circumstances.



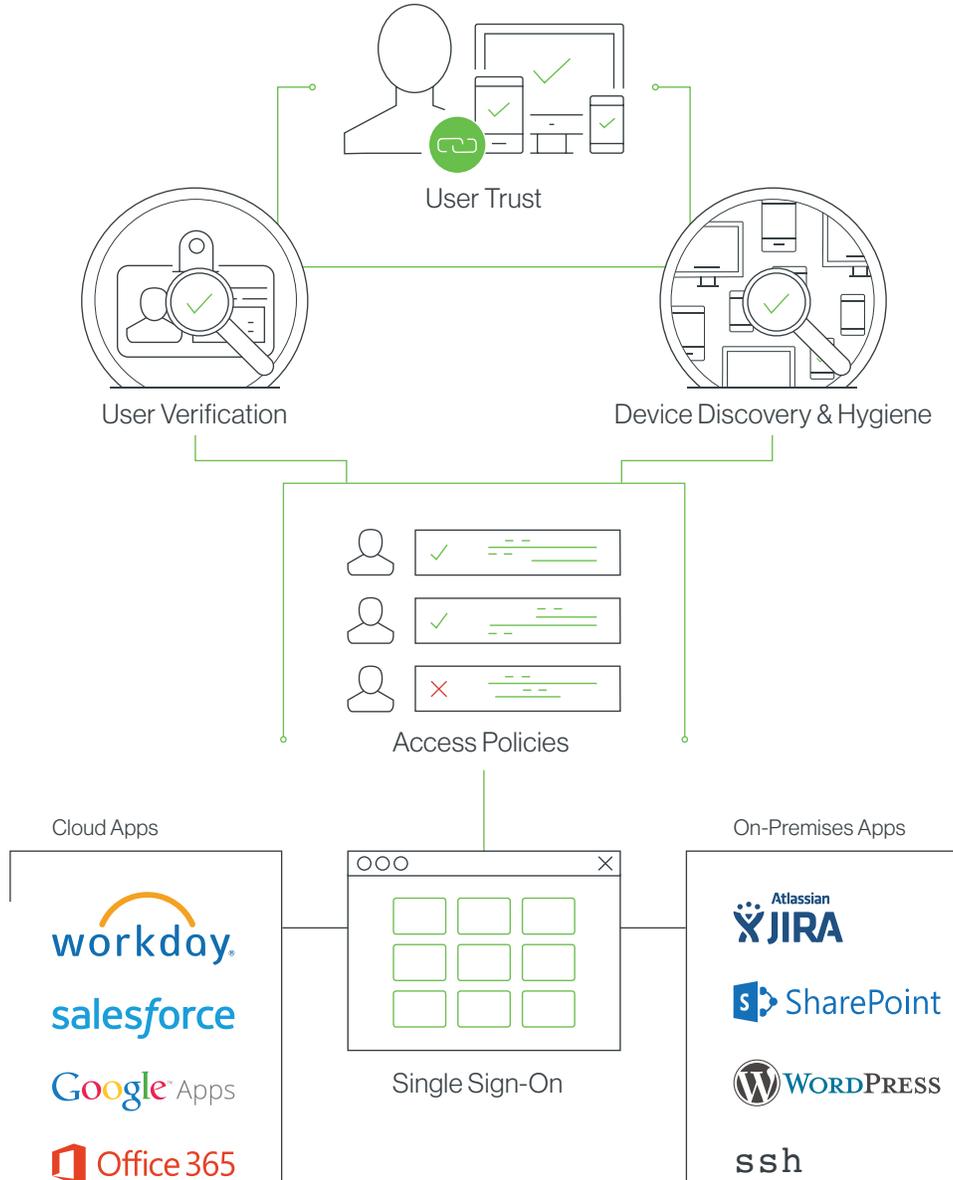
5. Enable Secure Connections to All Applications

Grant users secure access to all protected applications through a frictionless single sign-on interface accessible from anywhere, without a VPN. Protect all applications – legacy, on-premises, and cloud-based – in this fashion.



Duo Beyond

Trusted Users. Trusted Devices. Every Application.



Duo Beyond has made the BeyondCorp journey possible for companies such as KAYAK[®], allowing them to tighten their security controls both inside and outside the perimeter, and saving them months or years of effort piecing together their own solutions.



Learn more about Duo Beyond and start your free 30-day trial at duo.com/beyond.

References

- ¹ *BeyondCorp: A New Approach to Enterprise Security*; <https://research.google.com/pubs/pub43231.html>; Dec. 2014
- ² *Google's BeyondCorp: A New Approach to Enterprise Security*; <https://cloud.google.com/beyondcorp/>; 2/27/2018
- ³ *Migrating to BeyondCorp: Maintain Productivity While Improving Security*; <https://research.google.com/pubs/pub46134.html>; 2017
- ⁴ *Announcing Our New Edition, Duo Beyond!*; <https://duo.com/blog/announcing-our-new-edition-duo-beyond/>; 2/8/2018
- ⁵ *NIST: Developing a Framework to Improve Critical Infrastructure Cybersecurity*; https://www.nist.gov/sites/default/files/documents/2017/06/05/040813_forrester_research.pdf; 4/8/2013
- ⁶ *CNET: Behind the China Attacks on Google (FAQ)*; <https://www.cnet.com/news/behind-the-china-attacks-on-google-faq/>; 1/13/2010
- ⁷ *Duo Network Gateway*; <https://duo.com/product/every-application/application-access-policies/duo-network-gateway>
- ⁸ *Duo's Self-Remediation*; <https://duo.com/product/trusted-devices/self-remediation>
- ⁹ *BeyondCorp: Design to Deployment at Google*; <https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/44860.pdf>; Spring 2016
- ¹⁰ *Beyond Corp: The Access Proxy*; <https://research.google.com/pubs/pub45728.html>; 2016
- ¹¹ *Duo's Secure Single Sign-On*; <https://duo.com/product/every-application/single-sign-on>
- ¹² *Duo's Two-Factor Authentication*; <https://duo.com/product/trusted-users/two-factor-authentication>
- ¹³ *Duo's Device Insight*; <https://duo.com/product/trusted-devices/device-insight>
- ¹⁴ *BeyondCorp: A New Approach to Enterprise Security*; <https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43231.pdf>; Dec. 2014
- ¹⁵ *Duo's U2F: Universal 2nd Factor*; <https://duo.com/product/trusted-users/two-factor-authentication/authentication-methods/u2f>
- ¹⁶ *Duo's User Access Policies*; <https://duo.com/product/trusted-users/user-access-policies>
- ¹⁷ *Duo's Trusted Devices and Networks*; <https://duo.com/product/trusted-devices/device-access-policies/trusted-devices-and-networks>
- ¹⁸ *Duo's Endpoint Remediation*; <https://duo.com/product/trusted-devices/endpoint-remediation>
- ¹⁹ *Duo for KAYAK*; <https://duo.com/use-cases/case-studies/kayak>

