

Software Defined Perimeter for Infrastructure as a Service

*Presented by the SDP
Working Group*



The permanent and official location for *Cloud Security Alliance Software Defined Perimeter Working Group* is <https://cloudsecurityalliance.org/group/software-defined-perimeter/>.

© 2016 Cloud Security Alliance – All Rights Reserved All rights reserved.

You may download, store, display on your computer, view, print, and link to International Standardization Council Policies & Procedures Security at <https://cloudsecurityalliance.org/download/international-standardization-council-policies-procedures>, subject to the following: (a) the Report may be used solely for your personal, informational, non-commercial use; (b) the Report may not be modified or altered in any way; (c) the Report may not be redistributed; and (d) the trademark, copyright or other notices may not be removed. You may quote portions of the Report as permitted by the Fair Use provisions of the United States Copyright Act, provided that you attribute the portions to International Standardization Council Policies & Procedures.

ACKNOWLEDGEMENTS

We'd like to offer our sincere thanks to all the people who have collaborated and helped us create this document. Your input, feedback, and different perspectives have made this document better and more meaningful than we could have done on our own.

Thank you!

Lead Authors

Jason Garbis
Puneet Thapliyal

SDP Co-Chairs:

Bob Flores
Junaid Islam

Editor:

John Yeoh

Peer Reviewers:

Brent Bilger
Vince Campitelli
Matthew Carter
Aradhna Chetal
Gerald Greer
Kevin Fletcher
Jeff Huegel
Scott Kennedy
Juanita Koilpillai
Dan Logan
Nya Murray
Elamurian R
Vijay Rangayyan
John Reel
Reza Reza
Colin Robbins
Puneet Thapliyal
Yoshio Turner
Flavio Villanustre
Manish Yadav
Erkki Yli-Juuti
Xing Zhang

Design:

Stephen Lumpe (cover art)
Kendall Scoboria

TABLE OF CONTENTS

GOALS	5
APPROACH AND SCOPE	6
EXECUTIVE SUMMARY.....	7
SDP and the CSA Treacherous 12.....	8
INTRODUCTION: IAAS SECURITY OVERVIEW	10
TECHNICAL FOUNDATION	11
An IaaS Reference Architecture	11
Why is IaaS Security Different?.....	12
Location Is Just Another Attribute	12
The Only Constant is Change.....	12
The IP Address Conundrum.....	12
Security Requirements and Traditional Security Tools	13
Jump Boxes: Look Before You Leap	15
Why SDP and not VPNs?.....	16
Virtual Desktop Infrastructure (VDI)	17
How a Software-Defined Perimeter Solves These Problems.....	17
What is the Software-Defined Perimeter?	18
Policies Based on Users, Not Just IP Addresses	19
SDP Benefits	19
Operational Efficiency.....	19
Streamlined Compliance	19
Reduced Costs.....	19
SDP as a Catalyst for Change.....	19
SDP and Identity and Access Management.....	20
IAAS USE CASES.....	21
Use Case: Secure Access by Developers into IaaS Environment	21
Access Without a Software-Defined Perimeter	21
Access with a Software-Defined Perimeter	21
Comparison	22
Summary	24
Use Case: Secure Business User Access to Internal Corporate Application Services.....	24
Access Without a Software-Defined Perimeter	24
Access With a Software-Defined Perimeter.....	25
Summary	29
Use Case: Secure Admin Access To Public-Facing Services	29
Use Case: Updating User Access When New Server Instances Are Created.....	31
Access Without a Software-Defined Perimeter	31
Access With a Software-Defined Perimeter.....	32
Summary	33
Use Case: Hardware Management Plane Access for Service Provider.....	34
Summary	35
Use Case: Controlling Access Across Multiple Enterprise Accounts	35
Summary	36
RECOMMENDATIONS FOR ENHANCEMENTS TO SDP SPECIFICATION	37

HYBRID AND MULTI-CLOUD ENVIRONMENTS 38
SIDEBAR: ALTERNATIVE COMPUTE MODELS AND SDP 39
SIDEBAR: CONTAINERS AND SDP 40
CONCLUSION AND NEXT STEPS..... 41

Awareness and adoption of the Software-Defined Perimeter architecture (SDP) is rapidly growing,¹ and its effectiveness is being proven across numerous enterprises and use cases. We believe that the time is right to publicly promote the use of SDP to protect Cloud-based resources, given the rate at which organizations are strategically embracing Infrastructure as a Service (IaaS), and the compelling need to secure access to these resources.

This document is intended to explore and explain how a Software-Defined Perimeter (SDP) architecture can improve security, compliance, and operational efficiency when applied to Infrastructure-as-a-Service environments. Readers will obtain a clear sense of the security challenges facing Enterprise users of IaaS (given the Shared Responsibility model), understand the problems that arise from combining native IaaS access controls with traditional network security tools, and learn about how a Software-Defined Perimeter can solve these problems across various use cases. This document is aimed at several audiences – primarily organizations that are customers of IaaS providers, but will also be relevant to service providers themselves that are offering IaaS. These may be commercial service providers, or internal IT teams responsible for private IaaS or shared services environments.

We created this initiative in order to be able to explore the security challenges associated with IaaS in depth, and demonstrate how SDP can solve these problems.

1 Gartner predicts that by end 2017, enterprises using SDP to protect network services will increase from 1% to 10%, and by 2021, 60% of enterprises will have replaced VPNs with SDP solutions. (Gartner, It's Time to Isolate Your Services From the Internet Cesspool, 30 September 2016). Markets And Markets predicts a CAGR of 34% over the next 5 years (<http://www.marketsandmarkets.com/PressReleases/software-defined-perimeter.asp>). Other analysts such as ESG predict growth as well: <http://www.networkworld.com/article/3141930/security/goodbye-nac-hello-software-defined-perimeter-sdp.html>

APPROACH AND SCOPE

- This document uses terminology primarily oriented around the use of public cloud-based IaaS offerings, such as those from Amazon Web Services, Microsoft Azure, Google Compute Engine, and Rackspace Public Cloud. However, our use cases and approaches are equally applicable to on-premises IaaS, for example based on VMware or OpenStack.
- Vendors that have built commercial implementations of the SDP specification have, of course, addressed areas that were not in scope for the SDP v1 specification, in order to build complete offerings. These implementations have a variety of architectures, approaches, and capabilities. In this document we have aimed to be vendor-neutral, and to avoid talking about any vendor-specific capabilities. In those cases where vendor capabilities differ, we'll use terminology such *may*, *typically*, or *often* in attempt to explain these variations without sacrificing readability.
- Because most public IaaS providers only support IPv4 at this time, we are discussing approaches that are to some degree shaped and constrained by the capabilities and limitations of this technology. Adoption of IPv6 will change things, and we plan to address this in a future revision of this document.
- In alignment with the core SDP specification, we're focused on user-to-service access control (North-South) traffic. Server-to-Server (also known as East-West) communications is out of scope for this document, and is something we intend to address in future revisions of both the core SDP spec and this document, and to reflect market adoption trends.² Server-to-Server is a supported model mentioned in the core v1 SDP spec, although it has not yet been as highly adopted as the user-to-service model.
- Discussion of High Availability and Load Balancing considerations are outside the scope of this document
- Discussion of an SDP *Policy Model* is outside the scope of this document.
- The SDP use cases and approaches we discuss here may also be applicable to Platform-as-a-Service systems, depending on how they support and manage network access control.³

While writing this document, we worked hard to resist “scope creep” – we debated many topics that could have been considered, but were ultimately decided to be either better suited for inclusion in the overall v2 spec, or not relevant. See the section [Recommendations for Enhancements to SDP Specification](#) for those topics we think are important and have broader applicability than just IaaS.

Even though we deferred these topics we nonetheless exceeded our target page count, and can only hope that our sin of verbosity is more than compensated for by our sheer entertainment value. All joking aside, we believe we've made the right call in terms of balancing length and scope. But let us know - or better yet, dive in and help us with the next revision!

² Note that in IaaS environments, it's actually in some ways easier to control E-W traffic compared to on-premises environments, since IaaS network security groups default to denying cross-server traffic, which has to be explicitly enabled.

³ For example, if the PaaS system supports source IP address restrictions, it can be configured to only accept traffic from the SDP Gateway, allowing SDP policies to control user access.

IT and security leaders understand that responsibility for cloud Infrastructure-as-a-Service (IaaS) Security is shared between enterprises and cloud providers, and that IaaS has different (and in some ways more challenging) user access and security requirements than traditional on-premises systems. However, these requirements cannot be fully satisfied with traditional security tools and the security constructs provided by the IaaS vendor.

For example, organizations need to achieve the goal of limiting user access to networked resources, but traditional Network Access Control (NAC) and Virtual LAN (VLAN) solutions cannot be used in an IaaS environment, with its multi-tenant, virtualized network infrastructure. Another example: in an IaaS environment, all users require “remote access” to cloud resources, but traditional VPNs are not well-suited to today’s mobile workforce, cross-organization collaboration, or dynamic cloud environments. IaaS security capabilities address some of this, but (ironically) provide a very traditional network firewall perspective on network access control, focused on managing access by IP address and port. Many organizations struggle to make this work in their increasingly identity-centric security and access models.

With a Software-Defined Perimeter (SDP) architecture, organizations can securely provide user access to their IaaS resources without impeding business user or IT productivity. In fact, when properly deployed, an SDP deployment can be a catalyst for changing how network security is accomplished across the entire enterprise – both on-premises and cloud. With SDP, organizations can have a centralized and policy-driven network security platform that covers their entire infrastructure (both on-premises and cloud) and their entire user population. This is a compelling vision – but one that’s realistically achievable with SDP. Numerous organizations worldwide have used SDP to increase their security stance, reduce their attack surface, increase business and IT staff productivity, and reduce their compliance burden – while saving money.

This research document focuses on how SDP can be applied to Infrastructure-as-a-Service environments, with a focus on the following use cases:

- Secure Access by Developers into IaaS Environment
- Secure Business User Access to Internal Corporate Application Services
- Secure Admin Access To Public Facing Services
- Updating User Access When New Server Instances Are Created
- Hardware Management Plane Access for Service Provider
- Controlling Access Across Multiple Enterprise Accounts

In addition, this research note explains how and why traditional approaches to network security are neither effective nor efficient for IaaS environments. We conclude by looking at how SDP can be beneficial in hybrid environments - in fact SDP has been very effectively deployed in traditional on-premises and virtualized environments.

SDP and the CSA Treacherous 12

The Cloud Security Alliance publishes a list of top threats and concerns in order to make educated risk-management decisions regarding cloud adoption strategies. The report reflects the current consensus among security experts in the CSA community about the most significant security issues in the cloud. SDP reduces the attack surface by mitigating and eliminating many of the threats, risks, and vulnerabilities listed in the Top Threats report. This allows organizations to focus resources on other areas.

The following table lists out the top twelve threats ([Treacherous 12](#)) and analyzes the impact by SDP on them, if any.

THREAT	SDP IMPACT
1 Data Breaches	<p>SDP helps by reducing the attack surface of publically exposed hosts, by adding a layer of pre-authentication and pre-authorization. This ensures a “least privileged access” model of security for servers and network and thereby helps in reducing many attack vectors of data breaches.</p> <p>Residual risks: Several other attack vectors for data breaches are not in scope for SDP, including phishing, misconfigurations and end-point protection. Malicious access by authorized users to authorized resources will not be directly prevented by SDP.</p>
2 Weak Identity, Credential and Access Management	<p>In the past, credential theft of enterprise VPN access has led to data loss at many organizations. Since VPNs typically grant users broad access to an entire network, it is one of the weakest points of failure with respect to weak credential management.</p> <p>In contrast, SDP does not allow broad network access and limits the access to only those hosts explicitly allowed. Thus the overall blast radius is limited in case of credential theft. This makes the security architecture much more resilient towards weak identity, credential and access management. SDP can also enforce strong authentication before users can access resources.</p> <p>Residual risks: Organizations must have a solid Joiner-Mover-Leaver IAM process, and ensure that access policies are correctly defined. Overly broad access policies will introduce risk.</p>
3 Insecure Interfaces and APIs	<p>Protecting User Interfaces from unauthorized users is a core SDP capability. With SDP, unauthorized users (i.e. attackers) cannot access the UI, and therefore cannot exploit any vulnerabilities.</p> <p>SDP can also protect APIs if they’re being invoked by processes running on user devices. The primary focus for SDP deployments to date has been protecting user-to-server access.</p> <p>Server-to-server access has not been a focus for SDP to date; however we expect this to be included in SDP scope in the near future.</p> <p>Residual risks: Server-to-Server API calls are not a common use case for SDP at this time, so API service may not be protected by an SDP system.</p>
4 System and Application Vulnerabilities	<p>SDP significantly reduces the attack surface area, hiding system and application vulnerabilities from unauthorized users.</p> <p>Residual risks: Authorized users can access authorized resources, and potentially attack them. Other security systems such as SIEMs or IDSs must be used to monitor access and network activity (see <i>Malicious Insiders</i> below).</p>
5 Account Hijacking	<p>The session cookie based account hijacking is completely mitigated by SDP. The application server simply rejects incoming requests from malicious end-points if they are not pre-authenticated and pre-authorized and carry the appropriate SPA packets. Thus, even if the request carries a hijacked session cookie, it will not be allowed by the SDP gateway.</p> <p>Residual risks: Phishing or password theft is still a risk, but SDP can mitigate this by enforcing strong authentication, and having policies that control access based on attributes such as geolocation.</p>

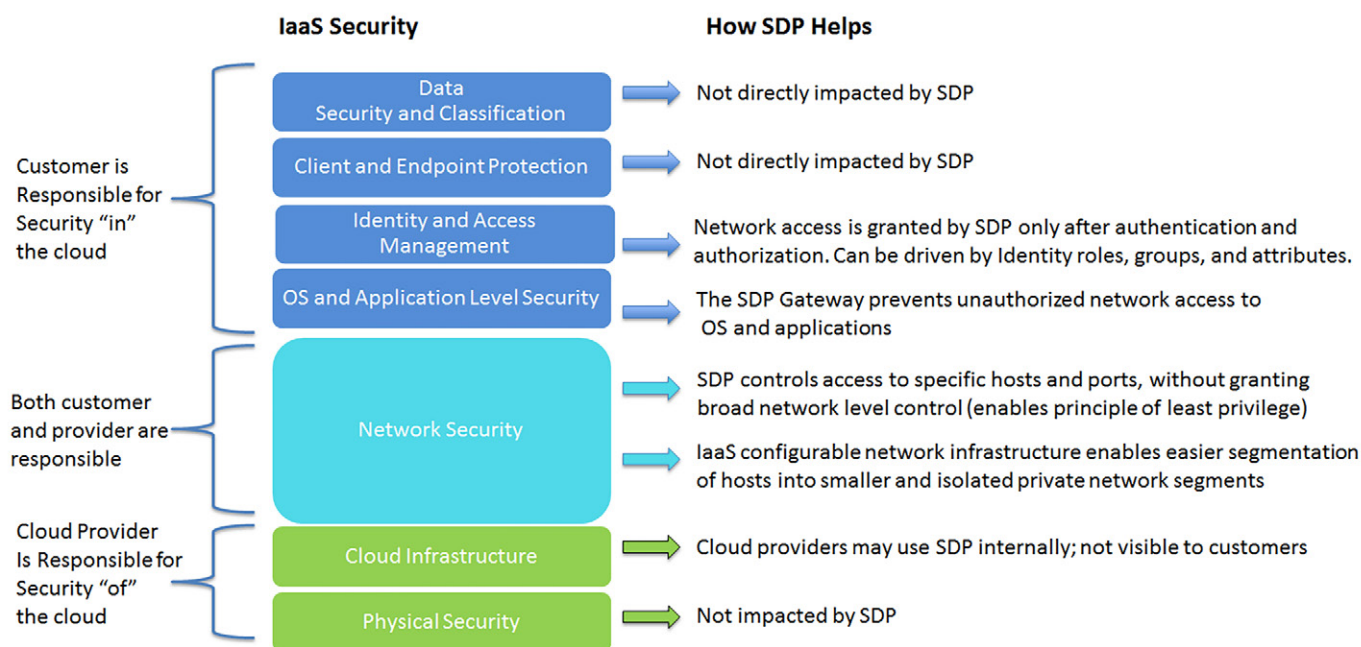
6	Malicious Insiders	<p>SDP will limit the ability of a malicious insider to cause damage. A properly configured SDP system will have access policies that limit users to only those resources required to perform their business function. Therefore, all other resources will be hidden.</p> <p>Residual risks: SDP does not prevent malicious access by authorized users to authorized resources.</p>
7	Advanced Persistent Threats (APTs)	<p>APTs are, by their nature, sophisticated and multi-faceted, and won't be prevented by any single security defense. SDP can certainly reduce the likelihood and spread of an APT by reducing the attack surface, limiting the capability of an infected endpoint to find network targets, and more easily enforcing multifactor authentication throughout an organization.</p> <p>Residual risks: Preventing and detecting APTs requires multiple security systems and processes combined for Defense in Depth.</p>
8	Data Loss	<p>SDP reduces the likelihood of data loss by enforcing the principle of least privilege, and hiding network resources from unauthorized users. SDP should be augmented by appropriate DLP solutions.</p> <p>Residual risks: SDP does not prevent malicious access by authorized users to authorized resources</p>
9	Insufficient Due Diligence	SDP does not apply
10	Abuse and Nefarious Use of Cloud Services	SDP does not directly apply, although SDP vendor products may have capabilities to detect and inform about use of cloud services.
11	Denial of Service	<p>The Single Packet Authorization (SPA) scheme in the SDP architecture makes SDP Controllers and Gateways much more resilient towards thwarting a DoS attack. Processing a SPA takes significantly less resources than a typical TCP handshake, making it possible for the servers to process and drop unsolicited network packets at scale. UDP-based SPA further increases server resiliency over TCP-based SPA.⁴</p> <p>Residual risk: While SPA significantly reduces the computational burden imposed by an invalid SPA packet, it is still non-zero, so a public-facing SDP system could be impacted by a large-scale DDoS attack.</p>
12	Shared Technology Issues	<p>SDP can be used by cloud service providers to secure administrative access to the hardware and virtualization infrastructure, by administrators. See the <i>Hardware Management Plane Access for Service Provider</i> use case below for a discussion.</p> <p>Residual risks: Cloud service providers must use a variety of security systems and processes in addition to SDP.</p>

⁴ The anti-DDoS SDP group has some interesting research on this topic, and some in-progress performance metrics showing server load with traditional TCP connections vs. SPA. Note that UDP-based SPA is even more resilient than TCP-based SPA, as it consumes even fewer server resources, and would better withstand an invalid packet flood attack.

INTRODUCTION: IaaS SECURITY OVERVIEW

There are perhaps many misconceptions in the industry about the security of applications in the cloud. It's well-known that Cloud-based applications and environments **can** be more secure than their on-premises counterparts, if they're deployed properly. The challenge is, of course, that security within Cloud environments follows different models than traditional on-premises security - and these differences can unintentionally result in reduced security.⁵ Migrating workloads to the cloud does not automatically make them secure - doing so takes some thought and some work.

IaaS vendors have generally created and promoted a "Shared Responsibility Model," which declares that the IaaS vendors are responsible for security **of** the cloud, and customers (enterprises) are responsible themselves for the security **in** the cloud. Below is a representative Shared Responsibility model, created by combining the concepts from several leading IaaS providers.⁶



Shared Responsibility for cloud security is something that many organizations are still trying to wrap their arms around, especially given the set of tools that IaaS providers have created. These tools, which we explore below in the IaaS Security Challenges section, tend to be based on relatively static IP addresses, and not on users (or identities). With this approach, customers can't effectively manage network access to Cloud resources on a per-user basis. As a result, they often end up opening their entire cloud network to anyone on the corporate network, and rely on application or instance-level authentication to protect access to these resources.

This is risky from a security perspective – there are far too many vulnerabilities that unauthenticated attackers can explore, requiring only network-level access to resources. It's also a compliance problem, as organizations often have to report on "who has access to what" in sensitive, regulated environments.

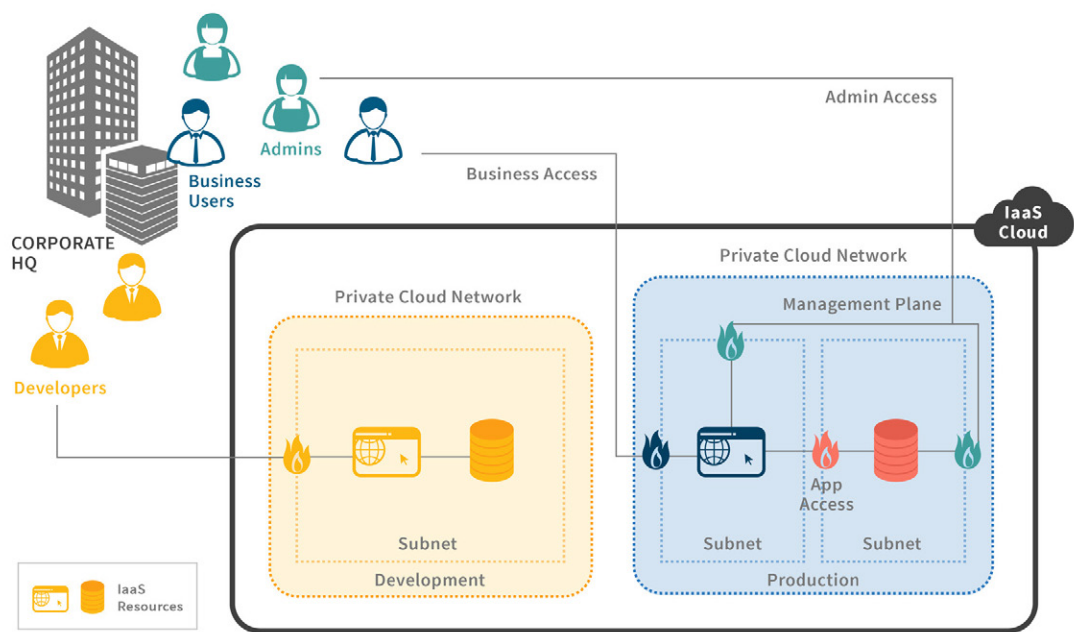
The SDP Architecture does have significant impact on the Shared Responsibility model with the IaaS vendors, as shown in the diagram above. With SDP, cloud customers can apply their own portion of the shared security controls in a much more effective way.

⁵ For example, the January 2017 ransom attacks on unsecured, publicly-exposed NoSQL databases, most of which were running in IaaS environments, is a good example of this.

⁶ In particular, these are drawn from the AWS model <https://aws.amazon.com/compliance/shared-responsibility-model/> and the Microsoft Azure model <https://blogs.msdn.microsoft.com/azuresecurity/2016/04/18/what-does-shared-responsibility-in-the-cloud-mean/>

An IaaS Reference Architecture

Infrastructure-as-a-Service components and architectures are familiar to readers of this document, and do not need introducing here. However, we intend to discuss IaaS concepts in generic terms rather than using provider-specific terms, so it’s important to define them here for clarity. Below is a simplified architecture diagram for an IaaS environment, intended to be applicable for both public cloud and private cloud deployments.



This diagram shows an IaaS cloud environment containing two groups of IaaS resources (virtual machines) separated into two Private Cloud Networks. These Private Cloud Networks could correspond to separate accounts, or separate private areas within a cloud environment (like an AWS Virtual Private Cloud). Most importantly from a network access perspective, these Private Cloud Networks are protected by a Cloud Firewall, which logically controls access into and out of these networks. Network access control into (and between) these Private Cloud Networks can get complicated quickly, and different cloud providers have different sets of tools. For this document we’re deliberately omitting the complexities of constructs such as routing tables, gateways, or NACLs so that we can focus on the challenges of managing user access to IaaS resources. This document is not intended to be a cloud network configuration primer!

This simple model allows us to consistently talk about the security and network aspects of cloud, regardless of the provider. Specifically, we’re going to be using the following terms in this research:

TERM	DESCRIPTION	EXAMPLES
Cloud Firewall	Security construct that controls network traffic into and out of the cloud environment. Managed by assigning server instances to Cloud Firewall groups.	AWS: Security Group Azure: Network Security Group
Private Cloud Network	An isolated network area within the cloud environment, controlled by a single account. This may comprise multiple subnets, and may be accessible by numerous people within a single organization.	AWS: Virtual Private Cloud Azure: Virtual Network

TERM	DESCRIPTION	EXAMPLES
Tag	IaaS systems support assigning name-value pairs to server instances. The tags, which have no semantic meaning within the IaaS system, are very useful as a basis for making access policy decisions by an SDP system.	AWS Tags Azure Tags
Direct Connection	IaaS providers, in partnership with telecommunications providers, offer a dedicated network connection from on-premises networks to the IaaS environment (often using MPLS). This has the advantage of reliable and dedicated bandwidth, which can typically be subdivided into multiple VLANs.	AWS Direct Connect Azure Express Route

Why is IaaS Security Different?

IaaS network access presents a significant security challenge. Network security, as part of the shared cloud responsibility model, rests squarely on the shoulders of the enterprise. Exposing private cloud resources to the public Internet is typically not an acceptable option – relying solely on authentication for protection will clearly fall short of security and compliance requirements – so organizations need to close this security gap at the network layer.

And this represents a typically complex security challenge, for several reasons.

Location Is Just Another Attribute

Different developers – even those sitting next to one another – may require different types of network access to different resources. For example, consider Sally the database administrator, who requires access to port 3306 to all servers running a database, across all projects. Joe, who sits next to Sally, manages the application code for project *Purple*, and needs to SSH into just those servers with the application server, for project *Purple*. And Chris, who works remotely from the rest of the team, is also an application developer on project *Purple*, and requires the same access as Joe despite being hundreds of miles apart.

Location may be one attribute to be considered as part of an access policy, but should no longer be the primary driver of network access levels, as it has been in traditional computing environments.

The Only Constant is Change

While a truism, it's one that's especially true in a cloud environment. First, the compute resources within an IaaS environment are highly dynamic, with server instances being continually created and destroyed.⁷ Manually managing and tracking access to these is near impossible. Second, the developers are also dynamic (although not necessarily from a personality perspective) – at the very least they're likely to work on different projects in different roles, perhaps simultaneously.

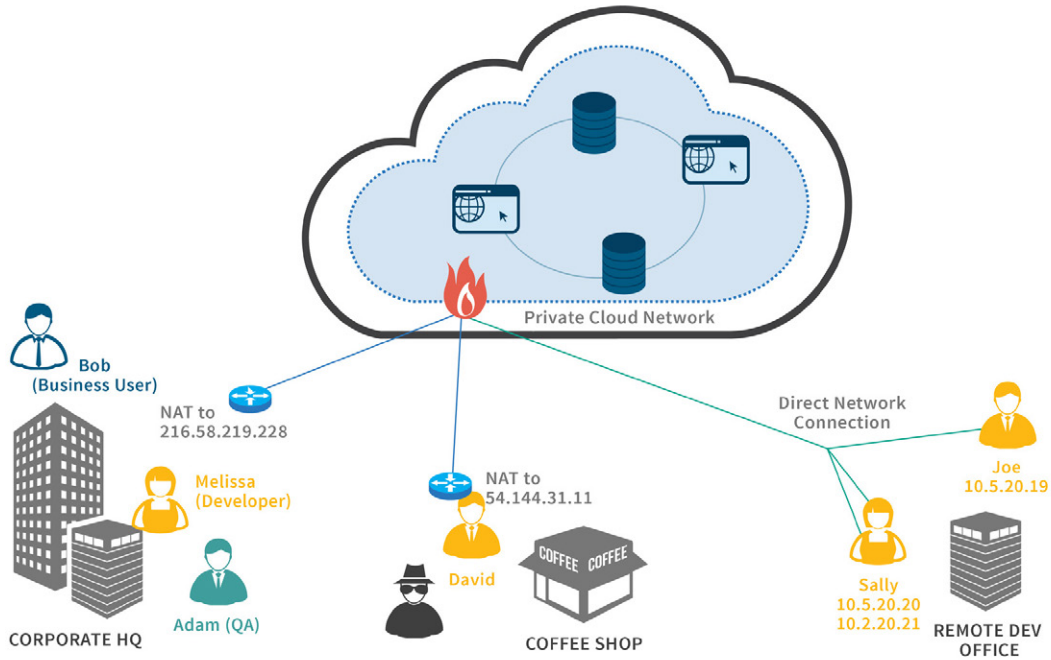
This is magnified in a DevOps environment, where development, QA, release, and operations roles are blended among a team, and access to a “production” resource can rapidly change.

The IP Address Conundrum

While perhaps not worthy of legendary network administrator Sherlock Holmes (CCIE #1), our IPv4 world represents a real

⁷ One large enterprise that heavily uses an IaaS environment for big data analysis launches over 30,000 instances each day, and 85% of those have a lifespan of fewer than 5 hours!

challenge. Not only do users' IP addresses change regularly, there's not a one-to-one correspondence between users and IP addresses. The diagram below illustrates how complex even a simple environment can be when access rules are solely driven by IP addresses:



LOCATION	NETWORK SETUP	SECURITY IMPLICATIONS
Corporate HQ	All users are mapped to a single IP address. There are many users in this location who require a broad range of network access	The Network Security Group cannot distinguish users, and must grant full access to everyone to all resources. This means that malicious users, attackers, or malware can traverse from on-premises to cloud network unimpeded.
Remote Development Office	Direct network connection preserves each user's IP address	IP addresses are dynamically assigned, and change on a daily basis. Users also access the cloud from multiple devices. Either IT operations teams continually update security group rules (imposing delay on business) or the network is fully opened to the Cloud (reducing security as noted in the row above).
Coffee Shop	One (or very few) users need remote access from various locations, which are probably NATted	Network access from these locations will extend to any malicious users on the same network. It's difficult to manually adjust network access and keep up with users' changing locations and access needs.

Security Requirements and Traditional Security Tools

Fundamentally, there are two problems that need to be solved:

- Secure Remote Access
- Visibility and Control of User Access

Security professionals generally recognize that exposing sensitive services to the public internet is a bad idea, and will want to secure them using one or both of these methods.

SECURE REMOTE ACCESS

First, let's consider the secure, remote access problem. As of today, we haven't invented a way to upload developers into the cloud, so all cloud users are *remote*, meaning that by definition, communications to the cloud occurs over a network connection, either the public internet or perhaps a dedicated Direct Connection.

Organizations typically solve this part of the problem through a VPN, perhaps by establishing a site-to-site VPN (shown as the blue line originating at the Corporate HQ location line in the diagram above), or leveraging a VPN from a user's device directly into the cloud to a VPN concentrator. Alternatively, they might combine solutions, and have users VPN from their device onto the corporate network, and from there have traffic traverse a site-to-site VPN into the cloud.

Using a VPN does technically solve problem A above (secure remote access) – since it provides a secure, encrypted tunnel for network traffic from the user's device to the cloud network. There are some downsides to this, in particular if all user traffic is forced to backhaul to the corporate network, which introduces additional latency, creates a single point of failure, and likely increases bandwidth costs and VPN licensing costs. Connecting over a VPN directly from each user's device to the Cloud helps address some of these issues, but may conflict with the need to simultaneously VPN into the corporate network, for example to access internal development resources.

In general, however, perhaps somewhat surprisingly, using a VPN may well not improve the *confidentiality* and *integrity* aspects of security at all if the application protocols being tunneled over the VPN – such as HTTPS and SSH – already include encryption.

One aspect a VPN can help with is the *availability* aspect of security, since protecting resources behind a VPN ensures that they're not publicly visible, and can thus prevent a DDoS-style attack.

This is a good lead-in to our next section, where we address the need to view and control user access, which VPNs cannot help with.

VISIBILITY AND CONTROL OF USER ACCESS

Regardless of how a user gets to the “front door” of the IaaS environment – whether through a VPN, or not – security teams still need to control (and monitor, and report on) which users can access which resources within the IaaS environment.

IaaS platforms provide built-in tools to manage this – Security Groups in AWS and Network Security Groups in Azure, which we're referring to as Cloud Firewalls in this document – act as simple firewalls, controlling access to servers based on source IP addresses. **This is the fundamental challenge with securing access** – organizations need to solve the problem of *user* access, but are given tools that control access based on *IP addresses*.

Let's take a look at an example of a cloud firewall:

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
HTTP	TCP	80	173.76.247.254/32
HTTP	TCP	80	50.255.155.113/32
HTTP	TCP	80	73.68.25.221/32
HTTP	TCP	80	98.217.113.192/32
HTTP	TCP	80	209.64.11.88/32
HTTP	TCP	80	172.85.50.162/32
HTTP	TCP	80	68.190.210.117/32
RDP	TCP	3389	173.76.247.254/32
RDP	TCP	3389	110.142.238.207/32
RDP	TCP	3389	50.255.155.113/32
RDP	TCP	3389	73.68.25.221/32
RDP	TCP	3389	98.217.113.192/32
RDP	TCP	3389	209.64.11.88/32

This snippet of a firewall configuration illustrates the simple IP address rule approach that IaaS platforms provide. All server instances that are assigned to this cloud firewall group will inherit this set of rules, which allows network access to specific ports. There are several problems with this approach, as any user of IaaS will attest to:

- It provides coarse-grained access to *all* the servers in this cloud firewall
- IP addresses do not correspond to users
- There is no notion of *policy*, and no explanation of *why* a given source IP address is in this list. It's very difficult, and labor-intensive, to attempt to implement any sort of complexity in terms of user access policies
- This list is static, and cannot respond to changes in user locations or permissions
- This approach is unable to take into consideration any notion of trust (such as authentication strength, device profile, or client behavior) and adjust access accordingly
- Any changes requires admin access to the IaaS account
 - » This will either need to be centralized, thus imposing a delay on productivity, or,
 - » Will require giving a broad user population admin level access to the IaaS admin account, which will lead to security, compliance, and operational problems

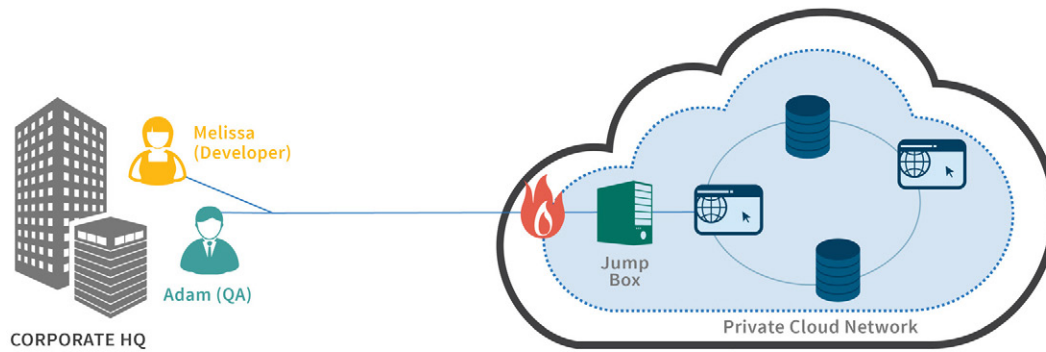
For IaaS environments, secure remote access is no longer a special case. All users are “remote” from the cloud, so security and network teams need to be concerned with how all users are accessing resources, not just a subset of users. That is, *secure remote access* must become a core concern, and part of the overall security strategy for any enterprise adopting IaaS.

Note that in addition to the approach described above - that of adding multiple source IP addresses to a single cloud firewall, on some platforms you could approach this slightly differently, by creating multiple cloud firewalls (for example one for each user's IP addresses), and associating multiple cloud firewalls with a server instance. This has the same logical effect as the approach above, and is arguably slightly better since you could assign meaningful names for the firewall (such as “Sally home and work”), but it comes with a cost of additional overhead, and is still a static solution.

Jump Boxes: Look Before You Leap

Jump Boxes, also known as Jump Servers or Jump Hosts, are servers that are designed to give users in a less-secure zone

access to servers or services running in a more secure zone. For this research document, the scenario is using a Jump Box to broker access to servers inside a cloud environment.



Network access to the Jump Box as shown in the diagram above, can either be open to the Internet, accessible via a Direct Connection, or controlled by a VPN. Access to the Jump Box desktop itself is controlled by user authentication (hopefully multi-factor). Jump Boxes can certainly help control access to cloud resources, by enforcing a single point of access for the managed servers. However, Jump Boxes have a number of limitations that make them ill-suited for broad cloud resource access. They:

- Are typically not multi-user systems, and are intended for single-user access to protected servers
- Are designed for occasional access, such as by system admins, and not for ongoing access
- Provide all-or-nothing network access to all servers on the network behind the Jump Box
- Are a very rich target. Compromising the Jump Box, or a user's device who can access the Jump Box, opens the entire network to an attacker
- Make it difficult to track user access for compliance purposes

It should be clear that Jump Boxes are not a suitable solution for user access to cloud systems.

Why SDP and not VPNs?

VPNs are, of course, a widely deployed and common technology for secure remote user access. Why can't organizations continue to use this proven technology?

VPNs do a good job of providing remote users with secure access to a VLAN or network segment, as if they were physically present on the corporate network. This technology, especially when combined with multi-factor authentication, worked fine for enterprises with a traditional perimeter, and static user and server resources. As Gartner states, "DMZs and legacy VPNs were designed for the networks of the 1990s and have become obsolete because they lack the agility needed to protect digital businesses."⁸

VPNs have two shortcomings that make them ill-suited for today's needs, and ripe for replacement. First, they provide very coarse-grained, all-or-nothing access to the assigned network. Their goal of making remote users behave as if they were present on the local network means that **all** users have full network access to the entire VLAN. It's unrealistic to try to configure VPNs to offer different levels of access for different users. They also cannot easily adjust to changes to the network or set of servers. VPNs simply can't keep up with today's dynamic enterprises.

Second, even if an organization is satisfied with the level of control that a VPN provides, VPNs are a siloed solution that only control remote users – they do nothing to help secure on-premises users. This means that organizations need an entirely different set of technologies and policies to control access by on-premises users. This will more than double the effort required to coordinate and align these two solutions. And as organizations embrace hybrid and cloud-based computing models, it gets even more difficult for VPNs to be effectively used.

⁸ Gartner: G00315586, "It's Time to Isolate Your Services From the Internet Cesspool", 30 September 2016.

Gartner agrees that it's time to replace VPNs with the SDP model: "By 2021, 60% of enterprises will phase out network VPNs for digital business communications in favor of software-defined perimeters, up from less than 1% in 2016."⁹

Virtual Desktop Infrastructure (VDI)

Virtual Desktop Infrastructure (VDI) is a set of technologies that enable organizations to host numerous desktop OS instances on a centralized server farm in a corporate datacenter. These instances may be virtualized instances of desktop OS's, or they may consist of a multi-user version of a desktop OS which many users concurrently login to. Along with VPNs, VDI has been a prominent mechanism used by enterprises for secure remote access into their network and applications.

Overall, there are a few drawbacks of VDI in today's cloud and mobile world. First, the desktop streaming experience tends to be poor on smaller form factor mobile devices. It does not render in a responsive manner and is very hard to use and, therefore, hampers productivity.

Second, many desktop-based client/server applications have been re-written as web applications, thus diminishing the value proposition of VDI. Third, VDI farms can be expensive to procure, especially if they are hardware based. And perhaps most importantly, as more and more workloads move the cloud, organizations have realized that VDI doesn't solve the user access problem for remote applications.

In fact, while VDI does help with part of the remote access problem – typically by encrypting traffic from the client device to the VDI server, it doesn't help solve the core user access problem – that of controlling which network resources a given user can access. In some situations, VDI can make this problem even more difficult by making multiple users appear as a single IP address emanating from a multi-user operating system. In this situation, network access control is effectively impossible with traditional network security solutions.

VDI definitely has certain benefits, but it's not designed to control user access to cloud networks and server resources, and in some ways can make this problem even more difficult.

How a Software-Defined Perimeter Solves These Problems

An SDP can address all the security issue discussed above, providing enterprises with secure remote access to IaaS environments, with fine-grained visibility and control of user access into these environments.

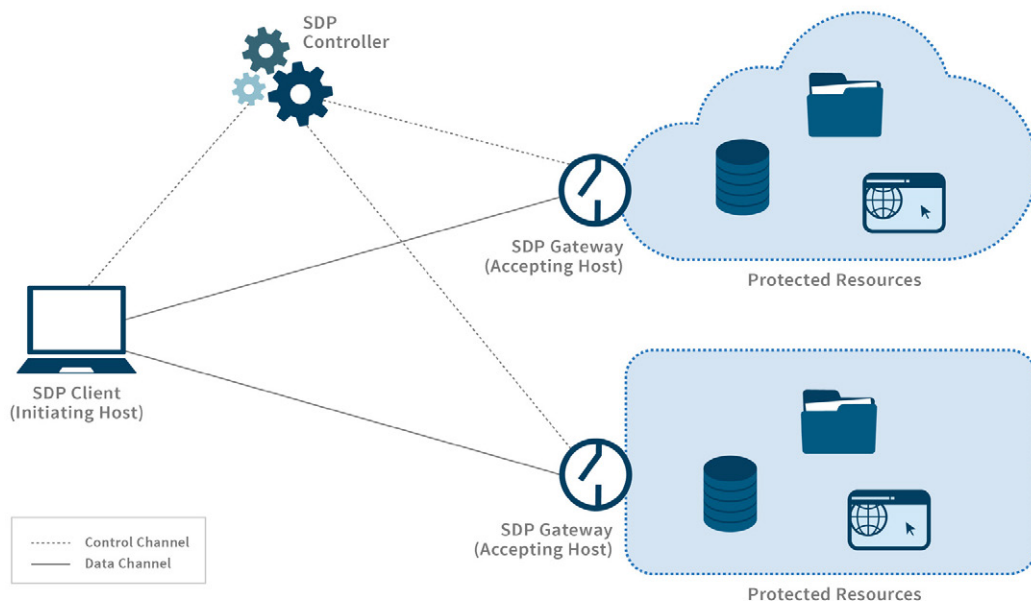
With SDP, organizations can keep cloud resources completely dark to unauthorized users. This completely eliminates many attack vectors including brute-force attacks; network flood attacks, as well as TLS vulnerabilities such as Heartbleed and Poodle. SDP helps organizations successfully manage their share of the responsibility of security IN the cloud, by building a "dark net" around their servers.

SDP relies on pre-authentication and pre-authorization as its two foundational pillars. By authenticating and authorizing the person and the device before even allowing a single packet to reach the target server, SDP enforces the Principle of Least Privilege at the network layer, significantly reducing the attack surface.

⁹ Ibid.

What is the Software-Defined Perimeter?

A Software-Defined Perimeter¹⁰ (SDP) architecture is made up of three main components, shown below:



CLIENT (Initiating Host)	A Client, which runs on each user's device
CONTROLLER	The component to which users authenticate (optionally with integration to an Identity Management system), and which evaluates policies before granting each user their individualized network entitlements
GATEWAY (Accepting Host)	<p>Gateways broker access to protected resources. Traffic from the client is sent through an encrypted tunnel to each Gateway, where it is decrypted and sent to the appropriate application (protected resource).</p> <p>As shown in the diagram above, the SDP architecture supports multiple distributed Gateways, each protecting a set of application or system resources.</p>

Using SDP terminology, the client (user device) is the *Initiating Host*, and the Gateway is the *Accepting Host*. After being authenticated by the controller the client establishes encrypted tunnels to each of the Gateways (the diagram above shows two distributed Gateways, each protecting a distinct set of resources, managed by a single Controller).

One important element of the SDP specification is Single-Packet Authorization (SPA). With this technique, clients create an HMAC-based One Time Password based on a shared secret (seed), and submit this to the SDP Controller and Gateway as the first network packet sent during connection setup. (It's also used for Gateway-Controller connection setup.)

Because the SDP Controller and Gateway reject invalid packets (presumably from an unauthorized user), they can prevent the establishment of TCP connections from unauthorized users or devices. Because invalid clients can be distinguished by analyzing a single packet, the computational load incurred by the SDP controller and gateway are minimal.¹¹ This reduces the effectiveness of DDoS attacks, and enables the safer and more reliable deployment of SDP services in public-facing networks.

¹⁰ The SDP version 1.0 specification is available here: <https://cloudsecurityalliance.org/download/sdp-specification-v1-0/>

¹¹ The anti-DDoS SDP group has some interesting research on this topic, and some in-progress performance metrics showing server load with traditional TCP connections vs. SPA. Note that UDP-based SPA is even more resilient than TCP-based SPA, as it consumes even fewer server resources, and would better withstand an invalid packet flood attack.

Policies Based on Users, Not Just IP Addresses

Because SDP systems are user-centric (i.e. they validate the user and the device before permitting any access), they permit organizations to create access policies based on user attributes. By leveraging aspects such as directory group membership, IAM-assigned attributes, and Roles, companies can define and control access to cloud resources in a way that's meaningful to business, security, and compliance teams. This compares favorably to traditional network security systems that are solely based on IP addresses, and don't consider users at all.

SDP Benefits

Organizations deploying SDP benefitted in several dimensions beyond that of directly improving security, which we have hopefully expressed clearly throughout the rest of this document. Other benefits of SDP include operational efficiency, streamlined compliance, and reduced costs. Let's briefly explore each of these.

Operational Efficiency

The automated policy enforcement performed by SDP provides significant operational benefits, compared to the manual work typically required to achieve a given level of security. Looking at this from another perspective, the level of security that an organization can easily obtain with SDP is effectively impossible to achieve with traditional security tools.

Streamlined Compliance

SDP implementations typically provide detailed visibility of per-user access entitlements and activities, due to the Gateways that log and control all client network traffic. SDP can therefore provide automated compliance reporting based on these details.

And, because SDP enables fine-grained control of user access, enterprises obtain the ability to reduce compliance scope by segmenting their network into smaller and well-isolated parts.

Reduced Costs

SDP can help organizations reduce costs in several ways. First, automated enforcement of access policies reduces the need to manually update and test firewall rules in response to user or server changes. In larger organizations, this is typically part of the daily workload for IT, and therefore an presents an opportunity to reduce both workload and labor costs (especially in an outsourced model). It will also accelerate business and technical user productivity, which is while worthwhile in its own right can also reduce hard costs (particularly for hourly or outsourced workers).

Second, streamlined compliance will reduce the time and effort required to prepare for and perform audits. Both of these activities often require third-party consultants, and every hour of time saved is a direct cost savings.

Finally, SDP can also help organizations save money as an alternative to other technologies. For example, we've seen enterprises that have chosen SDP over a traditional NAC after comparing the network switch upgrade requirements imposed by NAC – saving them hundreds of thousands of dollars in capital expenditures alone.

SDP as a Catalyst for Change

Of particular interest to us is the notion that SDP can be a catalyst for change. We believe that SDP represents a security architecture breakthrough, and will soon become a widely adopted approach for protecting network services.

More and more frequently, we're seeing organizations overtly embrace SDP as the new way that they're approaching security, and using this as an opportunity to replace traditional security technologies, such as VPNs, NAC, or DMZs, with a

more effective, more dynamic, and more secure alternative.

SDP and Identity and Access Management

SDP and Identity and Access Management (IAM) are naturally complementary in several ways. First, SDP implementations are typically designed to make use of already-deployed IAM systems for authentication, accelerating the rollout of SDP. This authentication may occur via connections to on-premises LDAP or AD servers, or use a standard such as SAML.

Second, SDP implementations typically use IAM attributes for users - such as directory group memberships, directory attributes, or Roles - as elements of SDP policies. For example, an SDP policy might state, “All users in directory group *Sales* may access server *SalesPortal* on port 443.” This is an excellent example of how an SDP system can add value to (and extend the power of) an existing IAM deployment.¹²

Finally, SDP systems can also be included in the identity lifecycle managed by IAM systems. Often referred to as “Joiner, Mover, Leaver,” IAM systems manage the business and technical processes associated with changes to user accounts and access. Enterprises deploying SDP should include SDP-managed network entitlements into their IAM provisioning system. For example, when an IAM system creates a new account in application X for Sally Smith, the SDP system should create a corresponding network entitlement at the same time.

Taken together, this integration works very well to support third-party users accessing an SDP system. The SDP controller trusts the third-party IAM system for both authentication and for ownership of the user’s identity lifecycle. So when a third-party user is deactivated in their IAM system – which is core to their user deprovisioning business process – the user will automatically be unable to access SDP-protected resources, since they can no longer authenticate via federation. This federation nicely solves a common problem with third-party access.

There’s a lot remaining to be written about how SDP and IAM can work together, but (as much as we love both of these) such analysis is not in scope for this document. It is, however, under consideration for inclusion in the SDP v2 spec.

¹² This example is a realistic policy, but in some larger-scale environments may impose challenges. Vendors should consider policies that support parameterization, for example a policy that leverages identity and system attributes to effectively state “Only users in a {Department} may access their {Department portal} on port 443”.

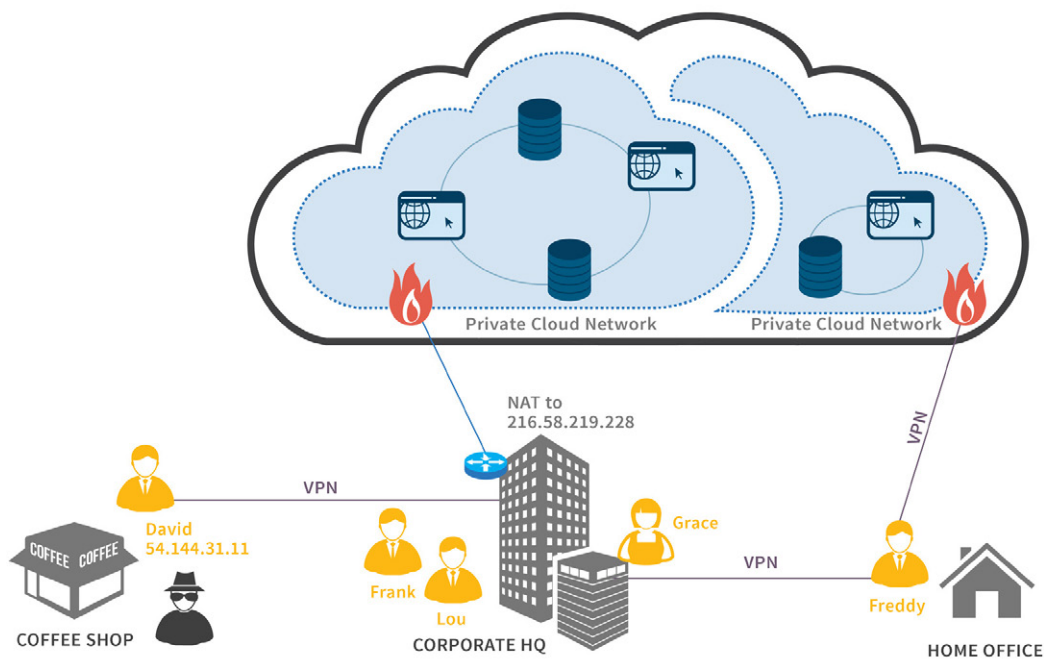
This section contains the six use cases that represent the core requirements for IaaS access, which SDP can help with.

Use Case: Secure Access by Developers into IaaS Environment

Developers need administrative access to IaaS resources for development, testing, and deployment. These users require access to a wide range of ports and protocols, and access to a constantly changing set of IaaS resources.


Developers may be working with sensitive data, and in DevOps environments will be working with production systems. As a result, organizations have security and compliance needs that require visibility and control of access to systems.

Access Without a Software-Defined Perimeter

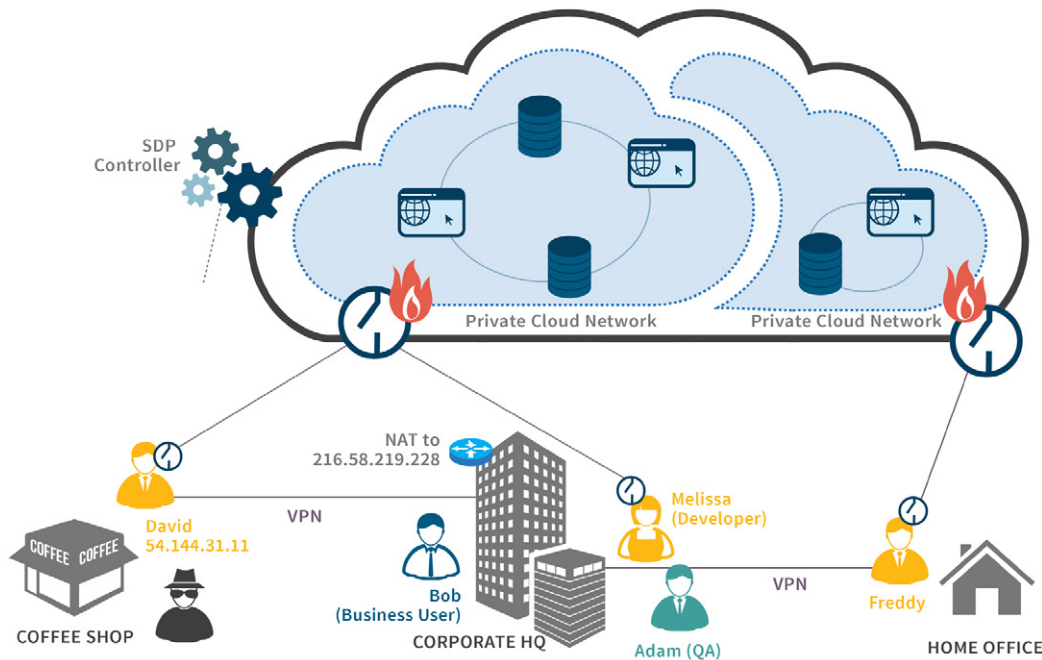


As shown in the diagram above, various populations of developers require access into two Private Cloud Network environments. These developers have different access requirements, and are in many different locations. The Cloud Firewall is the only control point for network traffic, and is essentially a simple table of allowed connections, mapping source IP addresses to target servers and ports.

Access with a Software-Defined Perimeter

The SDP is deployed as follows. A Controller (depicted as  in the diagram on the following page), is running in a location where it's accessible to all users (connections not shown in the diagram for clarity). This may be running in a publicly accessible location at the edge of the Cloud as shown, or perhaps running in a DMZ at corporate HQ. Access to the Controller is protected by Single-Packet Authorization, so exposing it to the Internet does not materially increase risk.

After users are properly authenticated by the controller, they access resources on the Private Cloud Network through the Gateways. The Gateway is also protected by SPA, and all user traffic is transmitted through an encrypted tunnel across the Internet. The Gateway enforces access policies on a per-user basis, achieving the principle of least privilege. The Gateways are situated at the entry point of each Private Cloud Network, and control all inbound traffic.



Comparison

REQUIREMENT: Grace, Lou, and Frank work at Corporate HQ and need to collaborate on an application, and access ports 22 (SSH), 443 (HTTPS), 3306 (MySQL), and 3389 (RDP) on multiple server instances.

CHALLENGE: All systems at HQ are NATted to a single IP address, 216.58.219.228

WITHOUT AN SDP	WITH AN SDP
<p>Approach: Cloud Firewall must be configured to allow traffic from 216.58.219.228 to ALL ports on ALL servers in the Private Cloud Network. These servers must have publicly accessible IP addresses assigned.</p>	<p>Approach: Each user establishes a mutually authenticated, tunneled connection from their device (Initiating Host) to the Gateway, and then to the target resources in the Cloud.</p> <p>The Cloud Firewall configuration becomes much simpler:</p> <ul style="list-style-type: none"> • The Gateway is open to all traffic from the entire internet. Because it's accessible only through SPA, it's more resilient to DOS and other attacks.¹³ • The protected resources are situated behind the Gateway on private IP addresses, and are not accessible from the Internet. Their Cloud Firewall is configured to only accept connections from the Gateway's IP address
<p>Implications: All users and systems on the corporate network have full access to the Private Cloud Network, violating the principle of least privilege and increasing the attack surface. This cloud network can be scanned, and attackers can exploit vulnerabilities.</p> <p>Server access is protected only by authentication, not at a network level. Key management can be a burden on developers.</p> <p>Compliance is more difficult, since all users have network access to all systems.</p>	<p>Implications: Because each user's connection to the Gateway is individually established and strongly authenticated, the fact that source IP addresses are NATted becomes irrelevant. The Gateway can enforce and control access to Cloud resources on a per-user basis, in a fine-grained way. The organization can define policies that are tied to users, devices, and roles.</p>

¹³ For more information, see the SDP research from the anti-DDoS group, as well as the annual SDP Hackathons sponsored by the CSA.

REQUIREMENT: David is a developer who works remotely, and periodically must access multiple servers in the cloud system from insecure networks such as a coffee shop. He also needs to access development resources on the HQ network. These services use multiple protocols and ports (22, 443, 3389).

CHALLENGE: The coffee shop network is NATted to a single IP address, 54.144.131.11.

WITHOUT AN SDP	WITH AN SDP
<p>Approach: Opening the Cloud Firewall to the entire Internet is not acceptable, and even allowing all traffic from 54.144.131.11 is too much of a security risk, so David first VPNs to the office network, and then access the cloud network as if he were on the corporate LAN</p>	<p>Approach: David's device authenticates to the Controller, and is then granted access to resource protected by the Gateway. David is no longer required to VPN into the office network, improving network performance and reducing network bandwidth usage and costs.</p>
<p>Implications: David requires a VPN connection to the HQ network (which he already requires to access on-premises resources)</p> <p>All traffic must be backhauled into and out of corporate network, adding latency and bandwidth cost</p> <p>This solution reduces to the requirement above, where all users and devices on corporate network have full access to the cloud network.</p>	<p>Implications: Because traffic is encrypted from David's device to the Gateway, there is no risk for him to be using a public wireless network or the public Internet. The Cloud Firewall configuration doesn't have to change – it's open to the Internet (but protected by SPA) – so David can be productive no matter where he is, and the security infrastructure continues to work consistently regardless of his location.</p>

REQUIREMENT: Freddy is a developer who works from his home office, and needs access to a Private Cloud Network that's separate from the rest of the team. This environment contains sensitive, regulated information so he's set up a VPN in order to access it. He also needs access to development resources on the HQ network.

CHALLENGE: Freddy's location doesn't change, but he needs ongoing access to both Cloud and HQ resources. Secure network connections are required for security purposes. but he cannot run two VPNs concurrently on the same machine.

WITHOUT AN SDP	WITH AN SDP
<p>Approach: Freddy accesses these resources through different environments on his development machine – he VPNs into the cloud through a VM, and accesses the HQ network through a VPN running in his host OS</p>	<p>Approach: Freddy establishes a secure connection to the Gateway for access to the protected Cloud resources.</p>
<p>Implications: This approach causes productivity issues for Freddy, since some of his tools and development tasks require access to both environments from the same system.</p> <p>Because Freddy is the only one accessing this environment right now, compliance and audit reporting is not a problem. But he knows that in a few weeks, as other team members join this project, he is going to have problems tracking and reporting on access, as well as managing this access. He doesn't yet know how he will enable this access. Should he open up the Cloud Firewall to everyone in the office? What about remote developers? Will he have to manage everyone's VPN access?</p>	<p>Implications: He can use his VPN connection to the office network concurrently, with no conflicts or issues, since the SDP connection looks like a regular network connection from the perspective of his desktop OS, not a VPN. So Freddy becomes more productive.</p> <p>Freddy can control and report on access to these resources easily, through a set of policies that he designs. Provisioning access to new users is a simple matter of editing his policies or editing user attributes, and gives him the ability to control access in a fine-grained way.</p>

Summary

For this use case, SDP provides compelling benefits to the organization:

- Secure developer access regardless of location
- Precise control of services each developer can access – by service and port
- Simpler compliance reporting
- Simpler security policy configuration
- Increased developer productivity

Use Case: Secure Business User Access to Internal Corporate Application Services

Business users need secure access to enterprise applications running in an IaaS environment. These business users require access to a wide range of applications such as HRMS, Financials, Procurement, Expenses, Supply Chain, etc. These applications could be provided as packaged applications from vendors or they could be home-grown applications developed by the internal IT developers.

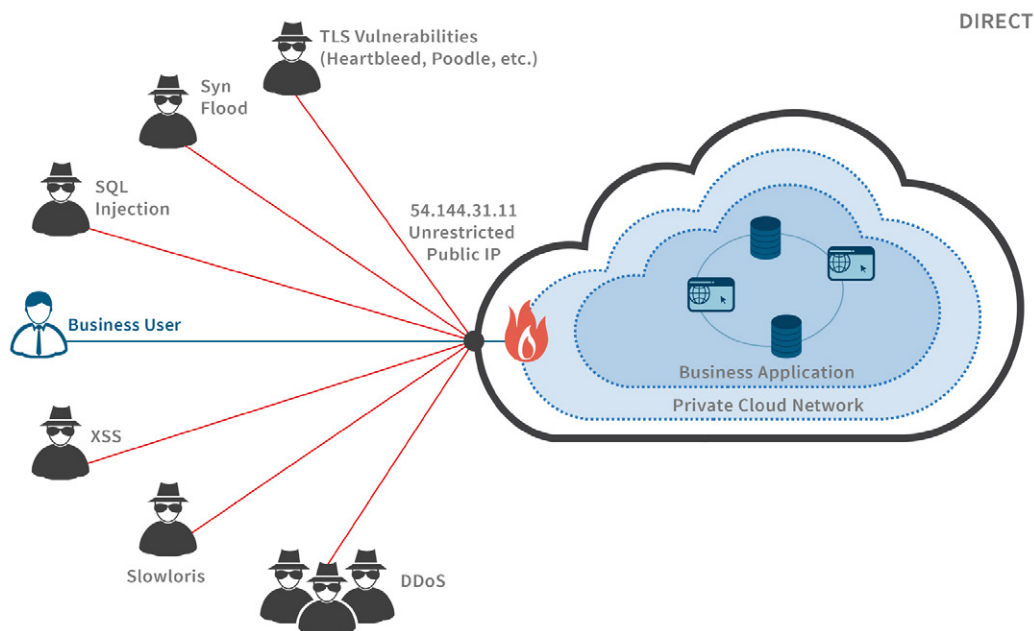
In these cases, Business Users typically do not require low level access to the network or machines (e.g. SSH or RDP).

The applications may be either in Production or Test/QA.

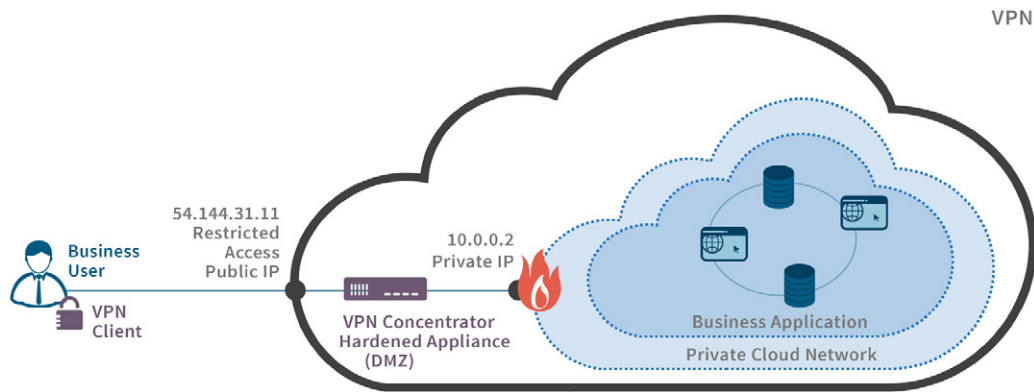
Access Without a Software-Defined Perimeter

There are three common ways of providing the secure remote access of Applications to the Business Users – 1) Direct, 2) VPN, and 3) VDI.

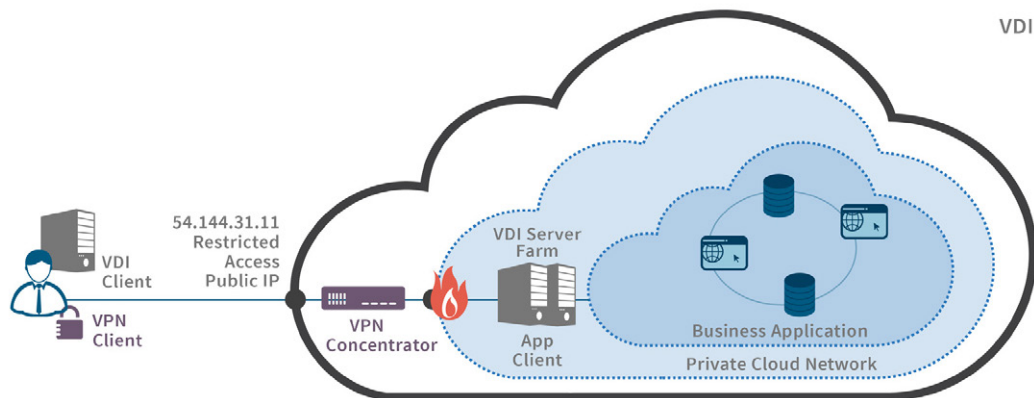
Direct: In case of Direct access, the application is typically a Web Application that is provisioned out to the public internet, without any consideration of access restrictions. In these cases, the application is exposed to the elements and is prone to all sorts of attack vectors including Brute Force, DDoS, XSS and any TLS vulnerabilities such as Heartbleed or Poodle.



VPN: With VPN, the private network and all the resources are extended to the Business User's device.

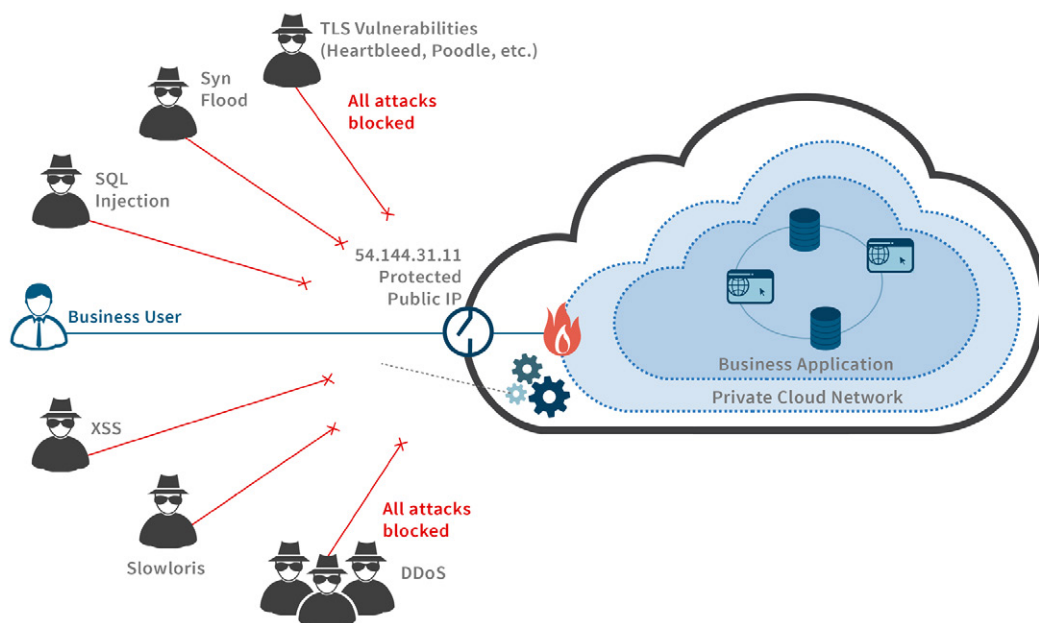


VDI: And with VDI, a virtual computer (usually Windows OS) is made available to the Business Users, that may in turn be used as a launch pad to the enterprise applications. The business applications are typically client/server applications that require a thick Windows client.



Access With a Software-Defined Perimeter

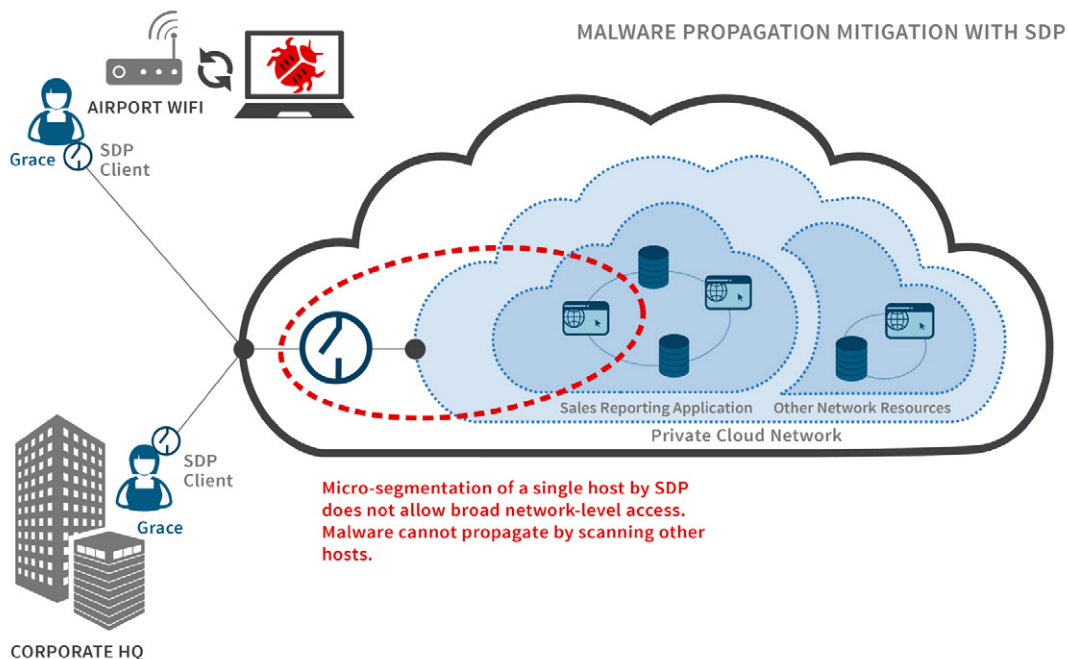
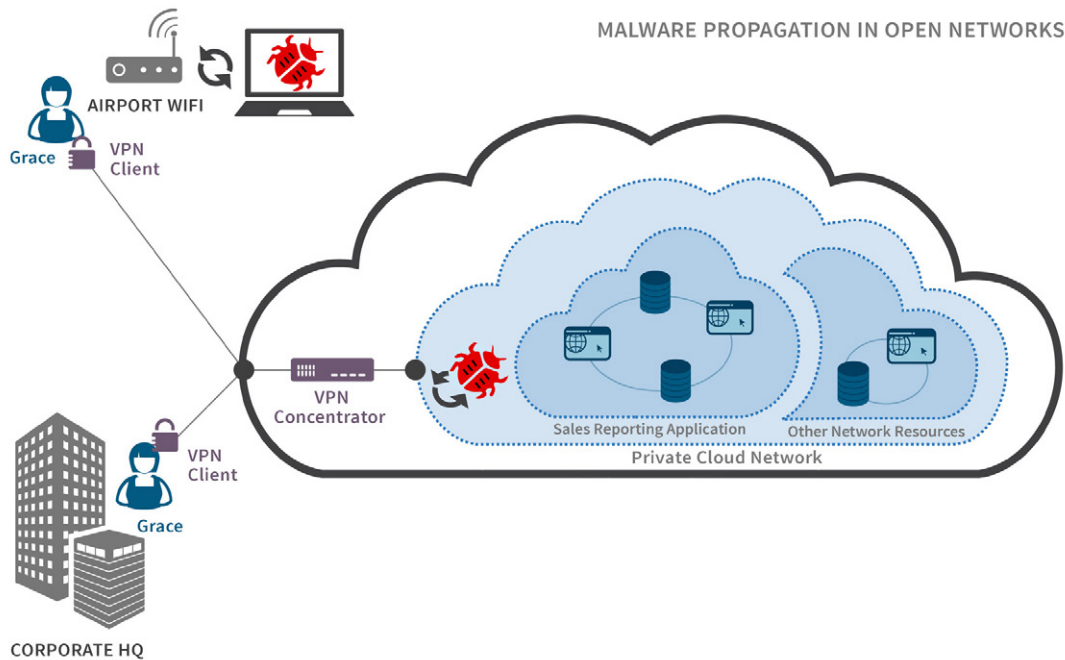
With an SDP solution, only authorized users can access the business application services. In fact, unauthorized users cannot even access the SDP Gateway – it's protected by Single-Packet Authorization, and is effectively “dark” to the attackers.



Let's examine the different users, their access requirements, and how this access can be managed.

REQUIREMENT: Grace works in the Sales department at the Corporate HQ. She needs to access the Top Accounts Report via the new Sales Reporting application built by the IT team. She travels to multiple customer locations and would be required to run the report often from remote locations. The application is hosted on Amazon AWS.

CHALLENGE: Grace works hard during her travels and accesses several free networks at the airports and coffee shops. In the past, IT Security has observed malware originating from her laptop when she return from her travels. They worry that malware may propagate into the AWS infrastructure when she accesses the new Top Accounts report via VPN or when she returns to the Corporate HQ ("Malware Mondays").



WITHOUT AN SDP

WITH AN SDP

Approach: No malware should be able to spread within the network. So, the IT Security folks create a network segmentation for isolating the application server on AWS.

Implications: Creating micro-segmentation with traditional network tools is complicated. The number of ACL records tends to grow exponentially¹⁵ as the number of applications grow.

Soon, the Network Admin are overburdened with supporting thousands of ACLs. Each request for opening a new ACL takes many days to analyze and process. Productivity suffers and IT loses agility, slowing down deployment of new apps.

Approach: SDP gives the benefit of isolating the network to individual servers (rather than a whole network). SDP provides safe and secure remote access to Grace, while preventing malware propagation via open network access (e.g. via VPN).¹⁴

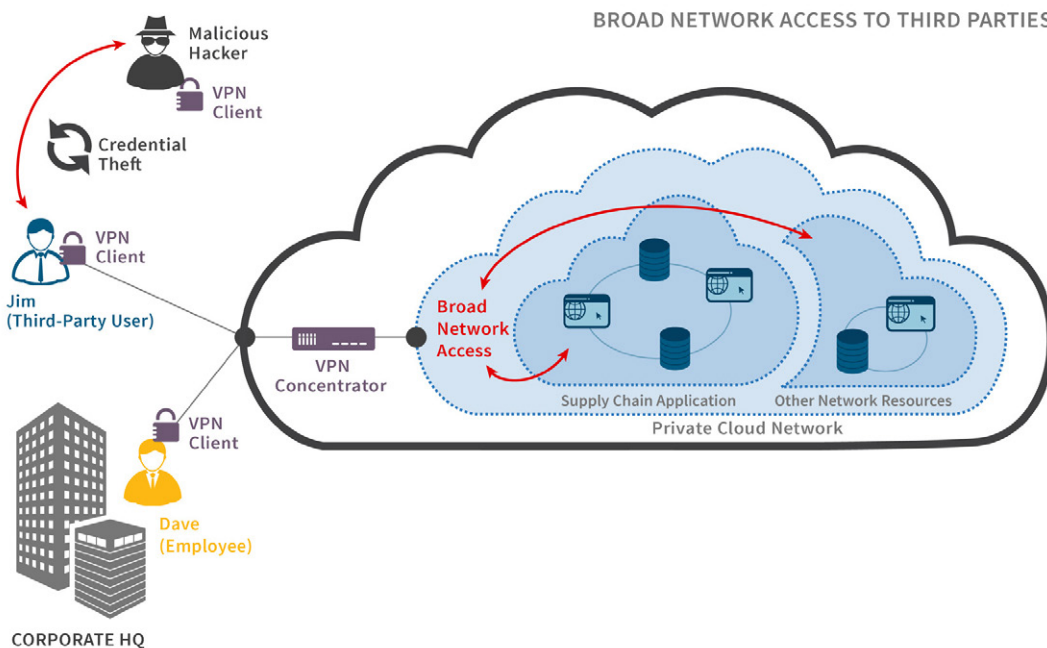
Implications: Business users now have full access to the Corporate Applications hosted on the Public Cloud Network. Limited network access is the default behaviour via SDP. So, it does not violate the principle of least privilege access, unlike traditional VPNs.

Network admins are not overburdened with the growing complications in the corporate network. Malware propagation via open networks is now prevented.

REQUIREMENT: Dave looks after the Supply Chain applications in the IT department. A new business process requires their supplier contact, Jim, to enter shipment details in their Supply Chain application, as soon as the goods are shipped.

CHALLENGE: This requires granting application access to a set of contacts at the third-party vendor. These business users (e.g. Jim) are not the employees of the company, so Dave has limited control over their security policies, training, etc.

Dave does not want to grant them a broad network level access (VPN) into the company's network. He is worried that in case of credential theft at the Vendor's end, his entire company network would get compromised. How can he limit the "blast radius"?



14 Controlling East-West traffic with SDP will be substantively addressed in the v2 SDP spec. As noted elsewhere in this document, this is an easier problem to solve in IaaS environments versus on-premises.

15 Technically, $O(n^2)$ in order to model application-to-application connections.

WITHOUT AN SDP	WITH AN SDP
<p>Approach: Dave creates a VLAN for this one application server, so it is isolated from the rest of the network. However, the Supply Chain application is very complicated and has integration into several other systems in the network. There exist cross-VPC connections and Firewall rules and network level ACLs. So, maintaining the complex network configuration is troublesome, as one IP address change in any system might bring down the whole Supply Chain application.</p>	<p>Approach: With SDP, they get the benefit of isolating the network to individual servers (rather than a whole network). The remote connection does not expose any other network resources to the Vendor. There is no way for a malicious hacker to sniff and explore for other weak resources in the network.</p> <p>Dave choses SDP to provide safe and secure remote access to Vendor contacts.</p>
<p>Implications: Dave pushes back on providing application access to third parties, as it is too complex and risky from a network security perspective.</p> <p>The Supply Chain users wish they could plan better for incoming supplies from their vendors. The impact of this was that the LOB no longer had visibility into incoming shipments of data center components. The business suffers from incorrect orders and late shipments.</p>	<p>Implications: Even though Dave does not have control over the third-party business users to enforce security best practices and training, with SDP he is able to limit the blast radius in case of credential theft. It becomes much safer to grant access to 3rd parties. Thereby, improving the efficiency of the Supply Chain and keeping the business growing.</p>

REQUIREMENT: Jim runs a Business Analytics report every week, even when he is on the road. The report is a Client/Server applications (not Web App) that is written only for the Windows OS. So, IT has deployed a VDI solution for Jim and his team. Jim first logs onto the remote desktop via VDI and then launches the reporting app.

CHALLENGE: The Client/Server reporting application is a packaged application purchased from a large Vendor. The suggested deployment model is only “on premises,” i.e. the vendor suggests not to provision the Server side of the application via public Internet, as it is not rugged/secure. That is to say, the server must be on same network as client.

Therefore, for remote users, the IT Security team has decided to use VDI where both the Client and the Server always remain in the same network. However, the cost of building and maintaining the VDI server farm is very high and is increasing as the number of mobile/traveling employees is increasing.

The challenge is to securely expose the Server part of the Client/Server app, so the Business Users could run the Client directly on their own Windows laptops.

WITHOUT AN SDP	WITH AN SDP
<p>Approach: Jim continues to add more infrastructure to support the VDI farm.</p>	<p>Approach: With SDP, Jim can securely open the Server/Port to only the devices belonging to selected Business Users, after they have been authenticated/authorized. For the rest of the world, the Server remains “dark.”</p> <p>Jim choses SDP to provide safe and secure remote access to the Client/Server apps required by the Business Analytics users.</p>
<p>Implications: Setting up a large VDI farms increases both CapEx for the hardware as well as OpEx for the maintenance and upkeep of the VDI servers.</p> <p>So, IT has to cut down the budget of other essential projects and VDI Farm is eating up all of the dollars.</p>	<p>Implications: The reasons for maintaining the VDI infrastructure decrease with the SDP approach. The Business Users become more productive as they avoid a daily hop into Remote Desktop before they could run the application.</p> <p>Soon, the VDI farm can be retired thereby freeing up IT budget for other initiatives.</p>

Summary

For this use case, SDP provides compelling benefits to the organization:

- Secure access to corporate applications by remote business users.
- Precise control of applications that a user may access.
- Increased third-party business integration.
- Simpler compliance reporting
- Reduced cost on VDI related infrastructure
- Simpler security policy configuration
- Increased business process productivity.

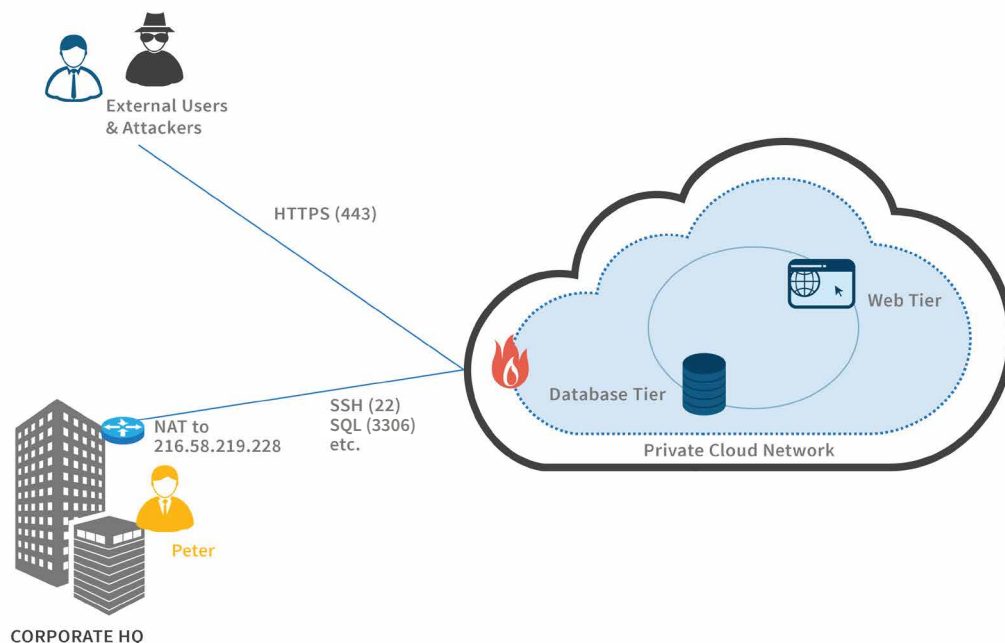
Use Case: Secure Admin Access To Public-Facing Services

When an application is provisioned in the cloud, many of its backend services need to be accessed remotely by the sysadmins, devops and other power users. These services may include:

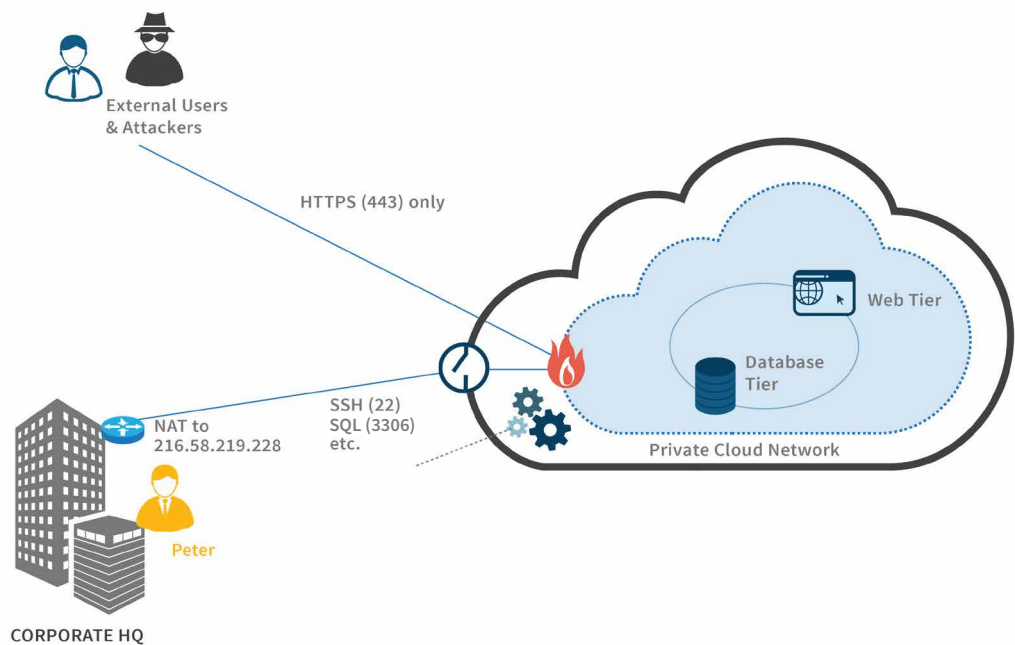
- Database tier via SQL interface (e.g. SQL Navigator for Oracle, PgAdmin for Postgres, etc.)
- SSH into servers
- Admin access to applications (e.g. admin for Wordpress blog)
- Database tools over HTTPS (e.g. PhpMyAdmin for MySQL)

In these cases, it is not prudent to allow these services to be available unrestricted on the public internet. Exposing them to the elements increases the chance of brute force, mis-configurations, zero-day exploits, etc.

Without an SDP, these back-end services will be exposed to the public internet, or restricted to certain source IP addresses which must be manually maintained in the cloud firewall, and which may still expose these services to many users.



With an SDP solution in place, only those services that are intended for public consumption (e.g. HTTPS) are exposed to the internet. All other services are hidden by the SDP Gateway, and access is controlled by policy. No need for an additional or isolated VPN.



REQUIREMENT: Peter is an database administrator for the Financial applications. His company has moved their infrastructure to a leading IaaS provider. Peter has a task of tuning a database SQL query to improve the application performance. So, he opens the DB port through the firewall and connects the SQL Navigator tool to the DB for the tuning project.

CHALLENGE: The DB port is now open to the world. Malicious auto-bots quickly discover the open port and attempt brute-forcing of the admin password. It is possible that they will work through the dictionary in few days and gain access to the critical financial data of the company, discover default passwords, or exploit a known (but unpatched) vulnerability in the database platform.

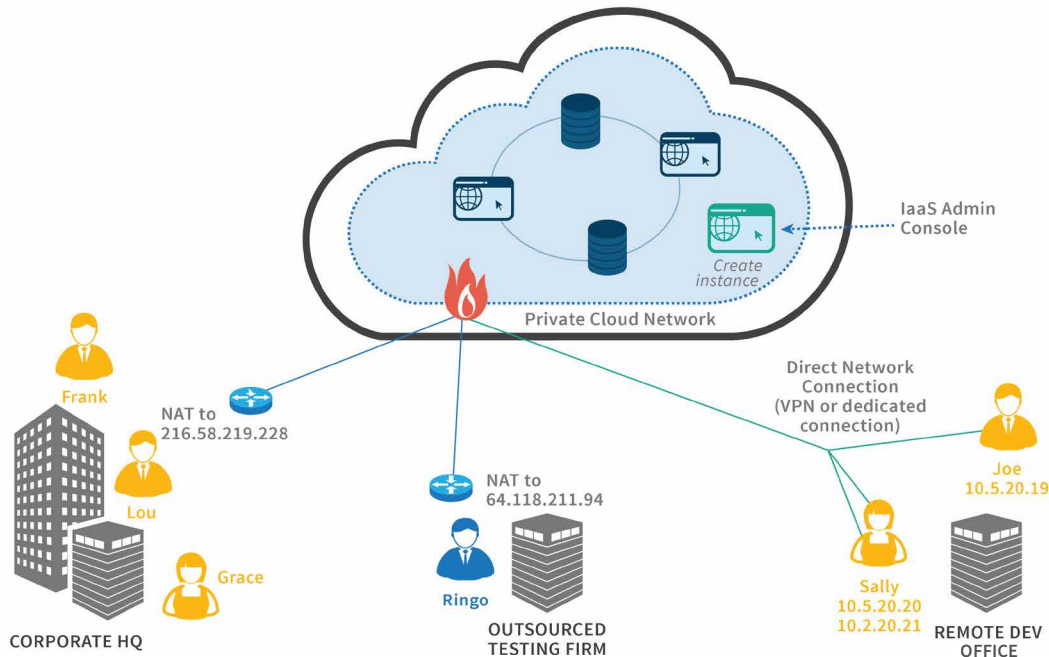
WITHOUT AN SDP	WITH AN SDP
<p>Approach: Peter sets up VPN for the cloud network.</p>	<p>Approach: With SDP, Peter keeps the port closed to the rest of the world. Malicious hackers don't realize that the DB service is running.</p> <p>The access to the DB port is limited only to Peter's device, after proper authentication and authorization</p>
<p>Implications: Peter has to set up a VPN to a network that is not their original datacenter's private network. He has to configure two different VPN endpoints on his machine and has to choose which one to connect into, every time he has to access network resources. Peter is a SQL developer, and not an IT admin, and doesn't fully understand the VPN configuration. He dislike using the VPN as it slows him down.</p>	<p>Implications: Peter is able to get secure private access to the DB without jumping through multiple configurations, and without having to configure network security.</p>

Use Case: Updating User Access When New Server Instances Are Created

Cloud environments are by their very nature, dynamic, and it's fair to say that most organizations leverage this aspect of IaaS to increase their development velocity and agility. In particular, in IaaS environments it's quick and simple to create and destroy server instances so organizations do this on a frequent (if not continual) basis.¹⁶

Access Without a Software-Defined Perimeter

As depicted in the diagram below, a person using the IaaS admin console (or a system making an API call) creates a new server instance. Required network changes depends on their location, cloud connection type, and needs, and is discussed in the tables on the following pages.

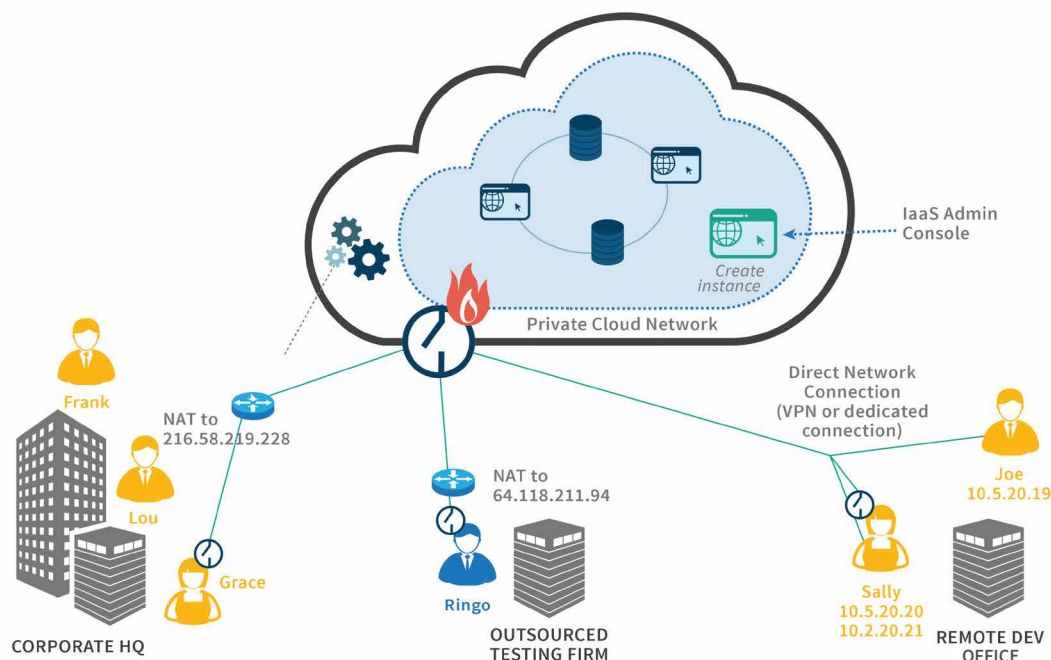


¹⁶ Note 1: For this use case, we're examining server instances that are manually launched by the IaaS admin console, or launched via IaaS API from a development or deployment script. Automated scaling of server instances (for example based on a CPU load threshold) by the IaaS infrastructure is not discussed here. Access to these instances is covered by existing use cases, since they're effectively clones of already-running resources that exist solely for load balancing and scale purposes. Access to them will be governed by the same access policies that control the core set of servers in the pool, which are covered in other use cases in this document.

Note 2: Because network access is granted by assigning an instance to a Cloud Firewall group, no action is required to remove access when an instance is terminated. Cloud Firewalls don't grant access to specific server IP addresses, and therefore there's no risk of access being mistakenly inherited through IP address reuse.

Access with a Software-Defined Perimeter

Cloud servers are all protected by the SDP system, with a Gateway acting as the sole network entry point into the Private Cloud Network. New server instances have metadata (tags).



REQUIREMENT: Grace launches an instance and needs SSH access only (port 22)

CHALLENGE: Grace is on a network that accesses the Cloud from a NATted IP address.

WITHOUT AN SDP	WITH AN SDP
<p>Approach: This instance must be assigned a public IP address, and the Cloud Firewall must grant full access to this instance from the NATted IP address 216.58.219.228, or to the entire Internet (0.0.0.0).</p> <p>Implications: While there is no immediate change required to the Cloud Firewall, access to this server is unrestricted for any user or device on the corporate network. This represents a significant security risk.</p> <p>Because IP addresses are NATted, it's impossible to restrict network access to just a single user or a single port. Instead, the security team requires that SSH access to these instances be controlled by a separate key. Management and tracking of these key files is a headache and a security risk for developers.</p>	<p>Approach: Access to the new server instance must go via the SDP Gateway, which is publicly accessible and itself protected by SPA. The SDP system detects this new server instance, and based on its metadata (tags), automatically grants Grace access to port 22.</p> <p>Implications: Grace is automatically granted the minimum level of access necessary to be productive, without any manual reconfiguration or IT operations involvement.</p>

REQUIREMENT: Sally launches an instance and needs access on ports for HTTPS, RDP and MySQL, from both her devices.

CHALLENGE: Sally accesses the cloud over the direct network connection from her office into the cloud – these resources look like they're on the local network.

WITHOUT AN SDP	WITH AN SDP
<p>Approach: If the goal is to limit access to just Sally, the Cloud Firewall must be updated to allow Sally's current IP addresses to access this specific new server instance. If the goal is to give Sally access without imposing any delay, the instance must be configured to allow all devices on the local network to access this new server instance on all ports.</p>	<p>Approach: The SDP system detects this new server instance, and based on its metadata (tags), automatically grants Sally access to the appropriate ports.</p>
<p>Implications: Requiring ongoing changes to the Cloud Firewall is an operational burden that most organizations are not willing to take on, as it adds time and cost. Most organizations grant open network access and rely solely on authentication.</p>	<p>Implications: Sally is automatically granted the minimum level of access necessary to be productive, without any manual reconfiguration or IT operations involvement.</p>

REQUIREMENT: Ringo works at an outsourced QA team, and needs Web access (443) in order to test this new instance.

CHALLENGE: Ringo's office is NATted to a single public IP address, which doesn't change. Because Ringo is in another time zone, he sometimes must work from home in order to collaborate with the team in real-time.

WITHOUT AN SDP	WITH AN SDP
<p>Approach: The Cloud Firewall must allow access from Ringo's public IP address. This can be limited to specific server instances by assigning only certain servers to this Cloud Firewall group. The Firewall must also allow access from Ringo's (periodically changing) home IP address.</p>	<p>Approach: The SDP system detects this new server instance, and based on its metadata (tags), automatically grants Ringo access to port 443.</p>
<p>Implications: The user launching this new server instance must assign it to the security group that grants access from Ringo's NATted IP address. All users on the local network with Ringo also have network access to this instance on port 443.</p> <p>Every time Ringo's home IP address changes, he must request Cloud Firewall updates from IT. This can take up to 24 hours, and impacts the overall team's productivity.</p>	<p>Implications: Ringo is automatically granted the minimum level of access necessary to be productive, without any manual reconfiguration or IT operations involvement.</p> <p>Because access is granted for Ringo as a user, it's not tied to his IP address. This means that Ringo is immediately productive, and has the same level of secure access regardless of where he is working from.</p>

Summary

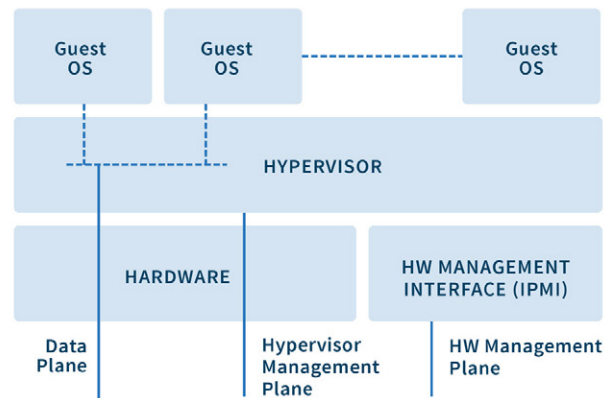
For this use case, SDP provides compelling benefits to the organization:

- Automatic detection of new server instances, and automated assignment of user access based on instance metadata
- Secure developer access regardless of location
- Full user productivity – no delays waiting for network access changes
- Access driven by policies, and not based on Cloud Firewall configuration
- Reduction in IT operations effort and cost

Use Case: Hardware Management Plane Access for Service Provider

The cloud, despite our abstracted notion of an infinitely-scalable and completely virtualized platform, actually *does* run on network and compute hardware at some level (that is, it's not "virtualized turtles all the way down"). And this hardware must, of course, be managed by a service provider, whether they are an on-premises IT department offering a private cloud platform, a managed hosting or co-location provider, or a commercial IaaS vendor.

The logical diagram below illustrates the network access paths to be managed:



The *Data Plane* is the standard network used for access to and from the Guest OS instances. Our discussions throughout this document have all been centered on this network.

The *Hardware Management Plane* is a network that's built-in to many hardware platforms, typically built on an Intel specification known as IPMI (Intelligent Platform Management Interface). This may be generically known as the Baseboard Management Controller, or informally referred to as the "lights out network." Most server manufacturers support this via an embedded hardware card with its own physical network connection, known as DRAC for Dell servers, and ILO for HP Servers.¹⁷

These IPMI services are unfortunately well known for many vulnerabilities, including default credentials that often are not changed, and susceptibility to simple attacks.¹⁸

The *Hypervisor Management Plane* is how administrators access the hypervisor admin functions, either via GUI console or API. While hypervisors in general have better access controls than IPMI systems, they should still be configured for access only through a separate network card, on a physically separate network or a VLAN, and protected by SDP. The discussions below, while focused on IPMI, are also applicable to the hypervisor management plane.

REQUIREMENT: Secure administrator access to the IPMI network interface on various ports. This access must have strong authentication, and be logged for security and compliance reporting purposes. Ideally, admins will have 24/7 access to this network, but only on an as-needed basis. However, this as-needed access is often time-sensitive, as IT may be responding to a server outage. There should be business process - such as request and approval - to control access, with a closed-loop mechanism for ensuring access is removed once it's no longer needed.

CHALLENGE: IPMI has many known weaknesses, ranging from exploitable vulnerabilities to limited management capabilities. IPMI requires a separate network, either physically separate or through a VLAN.

¹⁷ See https://en.wikipedia.org/wiki/Intelligent_Platform_Management_Interface for additional information.

¹⁸ See <https://community.rapid7.com/community/metasploit/blog/2013/07/02/a-penetration-testers-guide-to-ipmi>

WITHOUT AN SDP	WITH AN SDP
<p>Approach: The default approach, which is strongly not recommended, is to rely on default credentials within the IPMI system, and limit access to the IPMI network to only authorized users.</p> <p>A slightly better approach, although it incurs significant overhead, is to manage each server's access credentials individually.</p> <p>An even better approach is to tie the IPMI authentication to an enterprise's LDAP/RADIUS system.</p>	<p>Approach: With SDP, the IPMI servers can simply be placed on a network segment protected by an SDP Gateway. That is, no network traffic would be able to reach any IPMI interface unless permitted by an SDP Policy.</p> <p>The SDP system can leverage various user and system attributes—such as group membership, device profile, location, or time of day.</p>
<p>Implications: None of these solutions are strong – leaving IPMI systems with default credentials is simply asking for trouble – it's too easy for a malicious actor to obtain access to the network, for example via misconfiguration.</p> <p>Configuring user access credentials on a per-server basis provides better security, but is unworkable in environments of any size – there is simply too much manual work and credential tracking to make this practical.</p> <p>Leveraging an organization's LDAP/RADIUS system for authentication is far better, but still requires that any user that might need IPMI access at some point have full access to the IPMI network at all times.</p> <p>Controlling network access through firewall rules is technically possible but introduces too much process overhead, and will delay admin access to a server.</p>	<p>Implications: User access to IPMI interfaces can be driven by policies, and can be easily and dynamically based on an immediate “need to know.” For example, the SDP system could validate the existence of an open Service Desk ticket that lists a specific user and a specific server, before permitting access. This easily supports a business process for requesting and approving access to this sensitive entitlement.</p> <p>And, the SDP system can enforce access rules based on location, such as allowing access only from the on-premises corporate network, and blocking any access from a remote location.</p> <p>SDP can also integrate with an organization's IAM system to enforce strong authentication.</p>

Summary

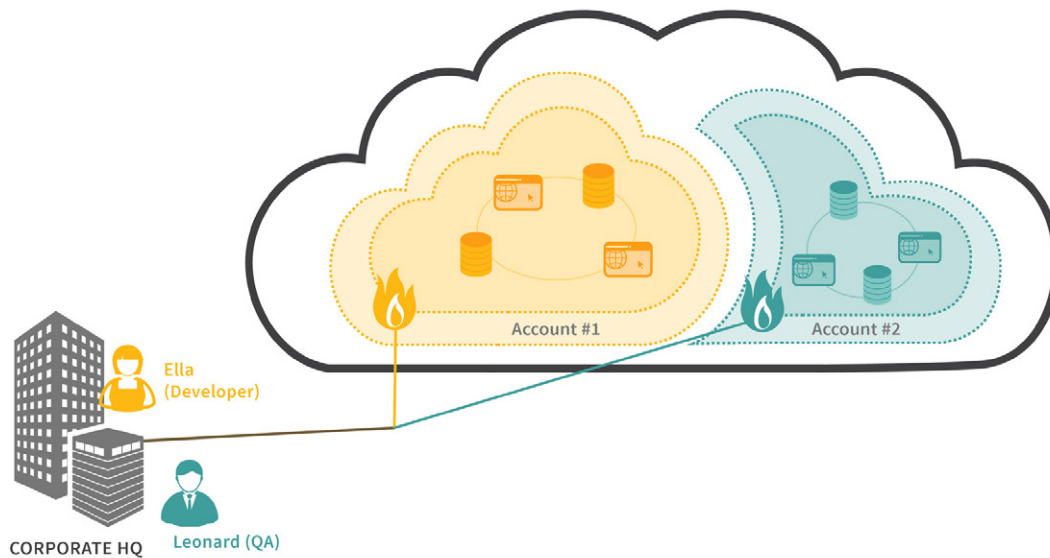
For this use case, SDP provides the following benefits:

- Secure and controlled access to the high-risk and vulnerable IPMI network
- Fine-grained control of which users can access these systems, and when
- Control via simple, user-centric policies
- Enforcement of strong authentication via integration with IAM
- Achieving rapid access for emergency / server outage situations without sacrificing security or compliance
- Comprehensive and detailed logs of who access which systems (and when) for compliance purposes
- A solution that scales with the growth and dynamics of the data center

Use Case: Controlling Access Across Multiple Enterprise Accounts

In this scenario, an organization has multiple distinct accounts within the IaaS provider. This may be deliberate, for security or compliance reasons, or it may be accidental due to different groups having set up accounts independently. The diagram below shows two accounts, but organizations typically have many more, especially if they have adopted a one-account-per-application strategy, for example.

The connection from the enterprise to the cloud, shown in teal in the diagram below, may be routed over a shared cloud Direct Connection (effectively a dedicated site-to-cloud VPN), or over the internet. In either case, the same challenges apply, described in the following diagram.



REQUIREMENT: Control which users can access which services consistently and efficiently across these accounts.

CHALLENGE: While separate accounts may have consolidated billing or shared IAM access to the Cloud admin console, they do not share Cloud Firewalls.

WITHOUT AN SDP	WITH AN SDP
<p>Approach: If organizations attempt to tightly control user access to resources, the problems noted in the use cases above still exist, and are in fact multiplied by the number of accounts.</p> <p>They may attempt to automate some of this by using cloud APIs to update many security groups, but ultimately are still facing the problem of trying to enforce user-centric security controls with the IP address-centric Cloud Firewall model</p>	<p>Approach: By placing an SDP Gateway in front of each Cloud Firewall, organizations can immediately simplify and improve their security. Each account's Cloud Firewall can be reduced to essentially a simple rule – allow traffic to the protected services only from the SDP Gateway.</p>
<p>Implications: As a result, organizations will realistically just open up all cloud resources to all users on the network, and rely on authentication to protect them. As we've covered above, this is not a strong security or compliance stance.</p>	<p>Implications: SDP policies can be applied consistently, regardless of the number or type of Cloud accounts in use. These policies provide full compliance tracking, and are consistently integrated with the enterprise's IAM systems.</p>

Summary

For this use case, SDP provides the following benefits:

- Enables the use of multiple Cloud provider accounts without sacrificing security
- Eliminates manual effort associated with management of Cloud Firewall rules
- Consistently enforces access policies across all accounts, for all user populations

Throughout the process of doing this research, we deliberately took a practitioner's perspective on SDP, looking at real-world use cases that are generally supported by available SDP implementations and architectures. By necessity, these use cases and requirements extended beyond the current (v1) SDP spec.

During our debates, conversations, and writing, we took note of the areas that we recognize are needed for a complete SDP implementation (as evidenced by the problems that SDP vendor products have solved).

Many of these topics go beyond just IaaS – they are needed as part of the overall SDP specification, and we decided that they were out of scope for this document – but are perfect for potential inclusion in the currently-in-progress v2 of the SDP specification. We look forward to participating in its creation, and the discussions on the following topics:

- Policy model for network entitlements
- IAM integration
 - » Directory attributes and SAML assertions as part of a policy model
 - » Step-up authentication triggers
- High Availability and Load Balancing approaches
- Support for additional network protocols beyond HTTP (e.g. SSH, UDP)
- Deep packet inspection and/or session proxying
- Incremental deployment considerations into existing enterprise environments
- Instance metadata and auto detection for cloud and on-premises environments
- SDP Cost Savings / ROI Model
- Discussion of Microsegmentation of Server-to-Server traffic (East-West traffic)
- Residual risks and potential vulnerabilities of SDP systems (for example, Man-in-the-middle attacks, or compromise of the Controller or Gateway)

Most organizations will have a complex and heterogeneous IT environment for the foreseeable future. Rather than looking at this as a problem to be eliminated, security teams need to embrace this richness – and its associated complexity – as a part of doing business. Different areas of the business have different requirements, and I think we can safely predict that there will never be a “one size fits all” IT infrastructure for all parts of today’s enterprises.

What this means is that security teams must look for the right types of tools and technologies that provide consistent security across these environments. While there will always be some platform-specific tools - such as systems management, automation, or endpoint management, we believe that from a security perspective, it’s critical for organizations to be able to establish policies and processes that are user-centric, and that work consistently across their platforms.

For example, organizations clearly want a single platform from which they can define and enforce policies and processes around which identities can access which systems. This platform must operate consistently across their on-premises physical, co-located, virtualized, private cloud and public cloud resources. Not doing so is setting up the organization for increased complexity, risk, and operational costs.

We believe that SDP, because it’s user-centric, independent of the underlying compute platform, and because it has the ability to strongly enforce access controls at the network level, is the right way for enterprises to meet their security goals in today’s complex environments.

We've seen a steady increase in the availability and adoption of so-called "serverless" compute models, whereby Cloud providers add new types of Platform-as-a-Service offerings to their inventory. These can range from an "as-a-service" twist on more traditional capabilities such as relational databases or message queuing, to the more novel "function as a service" (such as AWS Lambda, Azure Functions, and Google's Cloud Functions), and new IoT-centered offerings, among many others.

What all of these have in common is that they distinctly do not expose a traditional OS to the customer, which means that the network access control problems to be solved may be different from those associated with IaaS platforms.

In some cases, these services fall perfectly in line with the IaaS scenarios we've discussed here. For example, relational databases as a service are exactly the type of service that benefits from being protected by SDP. In fact many IaaS providers use the same network access models to control access to their relational instances as to their IaaS instances, so the SDP approaches we've described here are completely relevant.

In other cases, some of these services uses other security models. For example, the "function as a service" offerings tend to be either accessible through a publicly exposed URL, or via an API gateway of some sort. These offering may not be compatible today with an SDP approach, since the client-gateway-service model won't make sense. We believe that these models will evolve, as will SDP, and that this will be an interesting area for future work.

In any case, if your organization is using (or considering) some of these alternative compute models, make sure you and your security team engages with the developers to get a mutual understanding of the tool's security model, and how it fits in with the rest of your security architecture.

Containers are another rapidly growing trend, with many organizations adopting them as an underlying technology that enables a high-velocity DevOps methodology/lifestyle. Containers bring with them some interesting new security and access challenges. There are (of course) different network access models for the different container and clustering technologies, but to oversimplify they map to one of the following:

- Each pod (group of containers within a single OS process) gets a public IP shared by its containers (Kubernetes model)
- Each container gets a private IP, which is NATted to a public IP for the pod (Docker model)

In both cases, SDP can be applied effectively. The pods and their containers can certainly be placed behind an SDP gateway, and SDP policies defined to control user to service access. Protected services correspond to IP addresses or metadata for dynamic resolution within the Containers, just like with IaaS environments. And any pod-specific mappings from ports to containers can work just fine behind the SPD Gateway, there's no impact by the addition of SDP.

There are, of course, alternative approaches to networking within containers, so look carefully at the tools your teams are using. But in general, the mainstream approaches listed above are compatible with, and in fact work well with SDP. This is another area that will benefit from future research and validation.

CONCLUSION AND NEXT STEPS

We hope that this research has proven useful for you, whether you're an enterprise, a service provider, or an independent practitioner. Ideally, this document has helped improve your understanding of specific network access challenges associated with IaaS environments, and educated you on the ways in which a Software-Defined Perimeter can help solve these problems.

We'd like to make sure you're able to think of IaaS resources as something other than just an extension of your on-premises network. Embracing the cloud has many benefits but often requires many changes to take full advantage of – we hope that this research note helps you think about the cloud differently, and that you see this is an opportunity to transform the way users access these resources to be more secure, more flexible, and more efficient.

We believe that SDP represents a significant step forward in security – for the first time enabling dynamic, identity-centric security to be applied at the network layer – and are enthusiastic about seeing it more widely embraced by enterprises to meet today's security and business needs. As Gartner states,

“Connectivity complexity has made the old security compromises unsustainable, driving the need for a newer approach that meets the digital business needs for complexity, traffic volume and flexibility while avoiding the inherent vulnerabilities of the old models.”¹⁹

Of course, SDP doesn't solve every security problem – there are many parts of infosec that are simply not in scope for SDP, as well as residual risks that may be associated with a particular product, or driven by the particulars of an enterprise's implementation.

But overall, we believe that the Software-Defined Perimeter is a novel and compelling approach for improving security with particular applicability to Infrastructure-as-a-Service environments, and hope that this document has convinced you as well.

¹⁹ Gartner: It's Time to Isolate Your Services From the Internet Cesspool, 30 September 2016.