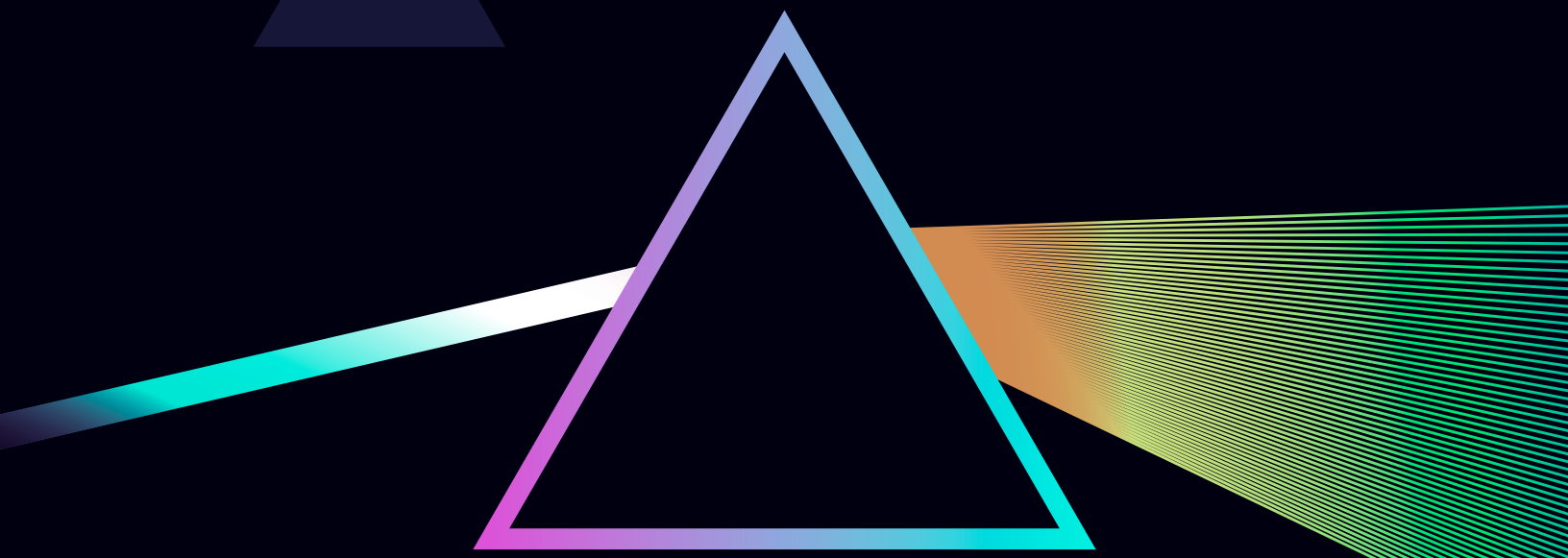
Software-Defined Perimeter

# ARCHITECTURE GUIDE



cloud security alliance®

# ACKNOWLEDGMENTS

CSA thanks everyone who contributed and supported us throughout the development of this guide.

**LEAD AUTHORS**

Jason Garbis
Juanita Koilpillai

**CONTRIBUTORS**

Junaid Islam
Nya Murray
Aaron Palermo
Preeta Raman
Michael Roza

**CSA GLOBAL STAFF**

Shamun Mahmud

# TABLE OF CONTENTS

# INTRODUCTION

The SDP approach combines technical and architectural components that protect networked applications and infrastructure more efficiently and effectively than traditional security tools.

**Today's network security architectures, tools and platforms fall short of meeting the challenges presented by our current security threats.** Whether you're reading the headlines in mainstream media, working day-to-day as a network defender, or are a security vendor, these potential threats may affect you. Ongoing attacks from a variety of sources affect commercial enterprises, governmental organizations, critical infrastructures, and more.

**It's time for us in the information security industry to embrace innovative new tools for network security–specifically via Software-Defined Perimeter (SDP) technologies—and to include all layers of network stacks in our security efforts.** The SDP approach combines technical and architectural components already proven to protect networked applications and infrastructure more efficiently and effectively than traditional security tools. ***SDP Specification: 1.0***, published by the Cloud Security Alliance in April 2014, outlines the basics of SDP technology:

> "The principles behind SDPs are not entirely new. Multiple organizations within the Department of Defense (DoD) and Intelligence Communities (IC) have implemented a similar network architecture based on authentication and authorization prior to network access. Typically used in classified or high-side networks (as defined by the DoD), every server is hidden behind a remote access gateway appliance to which a user must authenticate before visibility of authorized services is available and access is provided. An SDP leverages the logical model used in classified networks and incorporates that model into standard workflow. SDPs require endpoints to authenticate and be authorized first before obtaining network access to protected servers. Then, encrypted connections are created in real time between requesting systems and application infrastructure.[14]"

14 https://downloads.cloudsecurityalliance.org/initiatives/sdp/SDP_Specification_1.0.pdf

# PURPOSE

As a group of security practitioners and solution providers, we are passionate about information security and cyber security. We believe SDP is an important new solution for combating the security threats we all face.

**Since the publication of *SDP Specification: 1.0,*** we as a working group comprised of software vendors, system and security architects, and enterprises, have built and deployed numerous systems adhering to these guidelines. Meanwhile, we have learned a great deal about SDP implementations—especially in areas in which the original specification was lacking.

With this guide, we intend to assist enterprises and practitioners in their acquisition of information about SDP; to show the economic and technical benefits it can provide; and to assist users in implementing SDP in their organizations successfully. We'll consider this document to be a success if the following goals are achieved:

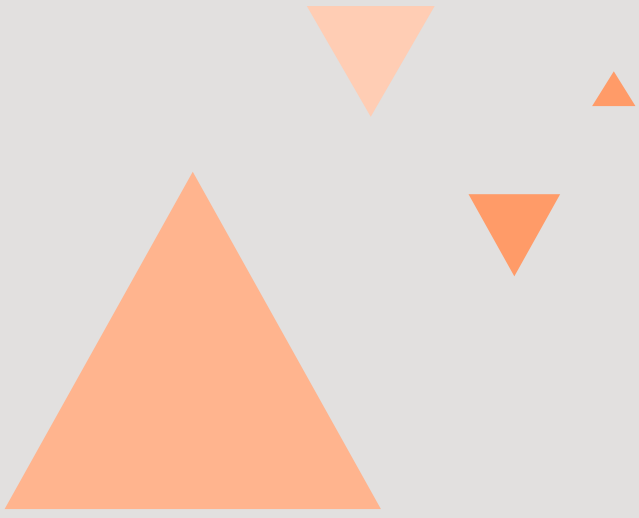
- Increased market awareness, credibility, and enterprise adoption of SDP
- Improved understanding of how SDP can be used in different environments
- Motivation to use SDP to solve enterprise problems
- Use of this document to educate internal stakeholders about SDP
- Enterprises successfully deploy SDP solutions based on the architecture recommendations in this paper

## Target Audience

**The information found in this paper will benefit all security, architecture and technical network teams** considering or currently implementing SDP projects in their organizations.

The primary audience includes **professionals working in information security, enterprise architecture, and security compliance roles**. These individuals are largely responsible for the evaluation, design, deployment, and operation of SDP solutions.

Additionally, those working as **solution providers, service providers and technology vendors** will also benefit from the information provided herein.



# OVERVIEW

## Introduction to the Software-Defined Perimeter (SDP)

**SDP is designed to leverage proven, standards-based components**, such as data encryption; remote attestation (in which a host authenticates remote access); mutual transport layer security (TLS; a method for cryptographically verifying client information); Security Assertion Markup Language (SAML), which relies on cryptography and digital signatures to secure appropriate access; and X.509 certificates, which verify access via public keys. Incorporating these and other standards-based technologies ensures that SDP can be integrated with an organization's existing security systems.

Since the initial publication of the Software-Defined Perimeter (SDP) specification by the Cloud Security Alliance (CSA), CSA has seen tremendous growth in visibility and enterprise adoption of SDP innovation. While traditional network security methods appear to be overwhelming IT and security professionals across all industries, relevant examples of increased use and interest in SDP technologies include the following:

- Five SDP working groups have made significant progress in their areas of focus, including SDP for IaaS, anti-DDoS, and Automotive Secure Communications.[14]
- Multiple commercial SDP offerings are now available from multiple vendors, and in use at multiple enterprises.
- An open-source reference[15] was implemented for the anti-DDoS use case of SDP.
- Four SDP hackathons have resulted in **zero** successful attacks on the SDP test infrastructure.
- Industry analyst reports have begun to include SDP in research and presentations.

14 SDP-for-IaaS: https://cloudsecurityalliance.org/download/sdp-for-iaas/
Anti-DDoS: http://www.waverleylabs.com/open-source-sdp/
Software-Defined Perimeter Working Group Initiatives: https://cloudsecurityalliance.org/group/software-defined-perimeter/#_initiatives
15 http://www.waverleylabs.com/open-source-sdp/demo/

## Security Benefits of SDP

- SDP reduces security risks by minimizing available attack surface.
- SDP protects critical assets and infrastructure by separating access control and data planes to render each of them "black," thereby blocking potential network-based attacks.
- SDP provides an integrated security architecture that is otherwise hard to achieve with existing security point products, such as NAC or anti-malware. SDP integrates the following discrete architectural elements:
  - » User-aware applications
  - » Client-aware devices
  - » Network-aware firewalls/gateways
- SDP provides connection-based security architecture instead of IP-based alternatives, because today's explosion of IPs and loss of perimeter in cloud environments render IP-based securities weak.
- SDP allows control of all connections based on pre-vetting of who can connect; from which devices; and to what services, infrastructure and other parameters.

# Business Benefits of SDP

SDP provides many business benefits, which we outline here for quick reference. We look forward to collaborating with the SDP community to provide in-depth qualitative and quantitative examinations of these benefits in a future publication.

| BUSINESS AREA | BENEFITS OF IMPLEMENTING SDP |
|---|---|
| **Cost and labor savings** | Replacing traditional network security components with SDP reduces licensing and support costs.<br>Implementing and enforcing security policies using SDP reduces operational complexity and reliance on traditional security tools.<br>SDP can also reduce costs by reducing or replacing MPLS or leased line utilization, as organizations can reduce or eliminate the use of private backbone networks.<br>SDP can bring efficiency and simplicity to organizations, which can ultimately help reduce labor needs. |
| **Increased agility of IT operations** | IT processes can act as a drag on business processes. SDP implementations, on the other hand, can be driven automatically by IT or IAM events. These benefits accelerate IT, making it more responsive to business and security demands. |
| **GRC benefits** | SDP delivers reduced risk compared to traditional approaches. SDP suppresses threats and reduces attack surfaces, preventing network-based attacks and the exploitation of application vulnerabilities.<br>SDP can feed into and respond to GRC systems (such as when integrating with SIEM) to streamline compliance activities for systems and applications. |
| **Compliance scope increased and compliance costs reduced** | Compliance data collection, reporting, and auditing processes can be improved by SDP, through the centralized control of connections from users on registered devices to specific applications/services.<br>SDP can provide additional traceability of connectivity for online businesses.<br>The network microsegmentation provided by SDP is frequently used to reduce compliance scope, which can have a significant impact on compliance reporting efforts. |
| **Secure cloud computing adoption** | SDP can help enterprises rapidly, confidently, and securely adopt cloud architectures by reducing the costs and complexity of the required security architecture to support applications in the public-cloud, private-cloud, data-center, and mixed environments.<br>New applications can be deployed faster with equivalent or better security than other options. |
| **Business agility and innovation** | SDP enables businesses to implement their priorities quickly and securely. Examples include:<br>• SDP enabled transition from on-premises call-center agents to home-based agents<br>• SDP enables the outsourcing of non-core business functions to specialized third-parties<br>• SDP enables customer-facing kiosks on remote third-party networks and locations<br>• SDP enables deployment of company assets onto customer sites, creating stronger integration with customers and generating new revenue |

# Primary Functions of SDP

SDP is designed to include five layers of security, at minimum: (1) authenticate and validate devices; (2) authenticate and authorize users; (3) ensure two-way encrypted communications; (4) dynamically provision connections; and (5) control connections to services while keeping them hidden. These and additional components are typically incorporated in SDP implementations.

| INFORMATION/INFRASTRUCTURE HIDING | | |
|---|---|---|
| **Components of SPD Architecture** | **Security Threats Mitigated or Reduced** | **Additional Benefits** |
| Servers "blackened" | All external network attacks and cross-domain attacks | The SDP components (controller, gateways) will not respond to connections until clients attempting access are authenticated and authorized with security protocols, such as single packet authorization (SPA). |
| Denial of Service (DoS) attacks mitigated | Bandwidth and server DoS attacks (However, note that SDP should be augmented by upstream anti-DoS services provided by an ISP.) | Internet-facing services are typically placed behind a "deny-all" SDP gateway (acting as a network firewall), and are therefore resilient against DoS attacks. SDP gateways are protected against DoS attacks by SPA. |
| Bad packets detected | All external network and cross domain attacks are detected quickly. | The first packet to an accepting host (AH) from any other host is a SPA packet (or similar security construct). If an AH receives any other packet, it reads it as an attack. |

| MUTUALLY ENCRYPTED CONNECTIONS | | |
|---|---|---|
| **Components of SPD Architecture** | **Security Threats Mitigated or Reduced** | **Additional Benefits** |
| Users and devices authenticated | Connections from unauthorized users and devices | Connections between all hosts must use mutual authentication to validate devices and users as authorized members of the SDP. |
| Forged certificates disallowed | Attacks aimed at credential theft | Mutual authentication schemes pin certificates to a known and trusted valid root managed by the SDP. |
| Man-in-the-middle attacks disallowed | Man-in-the middle attacks | Mutual handshake technology protects from man-in-the-middle attacks that exploit online certificate status protocol (OSCP) responses before the server certificate is revoked. |

| "NEED TO KNOW" ACCESS MODEL | | |
|---|---|---|
| **Components of SPD Architecture** | **Security Threats Mitigated or Reduced** | **Additional Benefits** |
| Forensics simplified | Bad packets and bad connections | All bad packets are analyzed and tracked for forensic activities. |
| Fine-grained access control | Data theft from external users from unknown devices | Only authorized users and devices are allowed to make connections to servers. |
| Device validation | Threats from unauthorized devices; credential theft | Keys are proved to be held by proper devices requesting connections. |
| Systems protected from compromised devices | Threats from lateral movement of compromised devices | Users are granted access only to authorized applications. |

| DYNAMIC ACCESS CONTROL | | |
|---|---|---|
| **Components of SPD Architecture** | **Security Threats Mitigated or Reduced** | **Additional Benefits** |
| Dynamic, membership-based enclaves | Network-based attacks | Access to protected resources is enabled by dynamically creating and removing access rules (outbound and inbound). |

| APPLICATION LAYER ACCESS | | |
|---|---|---|
| **Components of SPD Architecture** | **Security Threats Mitigated or Reduced** | **Additional Benefits** |
| Broad network access eliminated | Attack surface minimized; port and vulnerability scanning by malware and malicious users eliminated | Devices can only access specific hosts and services permitted by policy, and cannot access network segments and subnets. |
| Application and service access control | Attack surface minimized; malware and malicious users prevented from connecting to resources | SDP controls which devices and applications are permitted to access specific services, such as application and system services. |

# The Many Potential Uses of SDP

Because SDP is a security architecture, it provides benefits at many different levels, which do not all fit neatly into a narrow scope of classic use cases. This list is not intended to be comprehensive, because there are many other scenarios to which SDP can apply. The chart below provides several likely examples of types of connections that can be secured by an SDP implementation.

| NETWORK SCENARIO | LIMITATIONS OF EXISTING TECHNOLOGY | BENEFITS OF UPGRADING TO SDP |
|---|---|---|
| **Identity-driven network access control** | Traditional network solutions provide only coarse-grained network segmentation and are oriented around IP addresses. Even with newer platforms like software-defined networking (SDN), enterprises have difficulty enforcing user access controls that are timely, identity-focused, and precise. | SDP allows the creation of identity-centric access controls that are relevant to an organization, which are enforced at the network level. For example, SDP can support allowing only finance users web access to a financial management system, and only from corporate-managed devices. SDP can also allow only IT users secure shell (SSH) access to IT systems. |
| **Network microsegmentation** | Increasing network security by microsegmenting services is labor-intensive with traditional network security tools. | SDP enables network microsegmentation, based on user-defined controls. Fine-grained control of network access to specific services is automated by SDP, eliminating manual configuration. |
| **Secure remote access (VPN[4] alternative)** | VPNs provide secure remote access for users but are limited in scope and capability. They do not secure on-premises users, and typically provide only coarse-grained access control (access to entire network segments or subnets). This security and compliance risk often violates the principle of least privilege. | SDP secures both remote and on-premises users. Organizations can use SDP as a holistic solution, and retire VPN point solutions. SDP solutions are also designed for fine-grained access control. All unauthorized resources are inaccessible to users, which adheres to the principle of least privilege.[5] |
| **Third-party user access** | Security teams typically attempt to control third-party access through a combination of VPN, NAC, and VLANs. These solutions are generally siloed, and cannot provide fine-grained or comprehensive access control across hybrid environments. | Securing third party access enables a business to innovate and adapt. For example, users may be transitioned from corporate offices to home offices to reduce costs, or may sometimes work remotely—in which case, certain functions can be outsourced securely to third-party specialists. On-premises access for third-party users can be controlled and secured easily. |

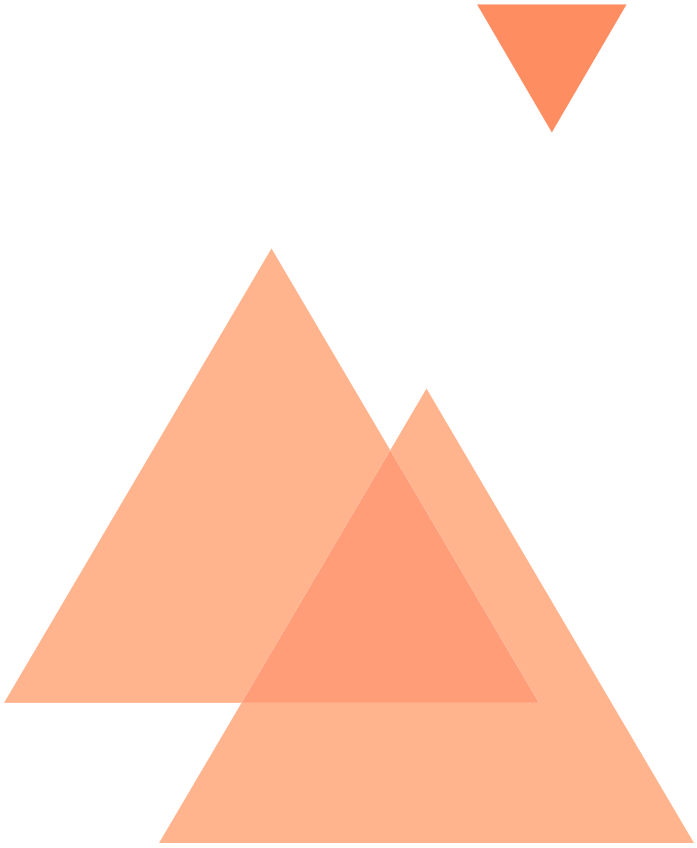
4 Specifically, we're discussing VPNs for remote enterprise user access, not site-to-site VPNs or consumer VPN scenarios.

5 In a paper published by Gartner on September 30, 2016, the authors write: "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."
"It's Time to Isolate Your Services From the Internet Cesspool" https://www.gartner.com/doc/3463617/time-isolate-services-internet-cesspool.

| NETWORK SCENARIO | LIMITATIONS OF EXISTING TECHNOLOGY | BENEFITS OF UPGRADING TO SDP |
|---|---|---|
| **Securing privileged user access** | Privileged user (typically admin) access commonly requires heightened security, monitoring, and compliance oversight. Traditionally, privileged access management (PAM) solutions manage access through credential vaulting, which does not provide network security, remote access, or context-sensitive access. | Access to privileged services can be restricted to authorized users and secured at the network level. Privileged services can be hidden from unauthorized users, thus limiting the attack surface. SDP can ensure access is only allowed when specific conditions are met (e.g. during a defined maintenance window or only from specific devices), and then access can be logged for compliance reporting. |
| **Securing access to high-value applications** | Providing fine-grained authorization to high value applications with sensitive data currently may require complex and time-consuming changes to several layers of functionality. (E.g. application, data external access.) | Access to applications can be restricted by integrating user/identity awareness, network awareness and device awareness; by not exposing a full network; and by relying on applications or application gateways for access control. SDP can also facilitate application upgrades, testing and deployment, and provide the required security framework for DevOps CI/CD. |
| **Securing access to managed servers** | In managed security service provider (MSSP) and larger-scale IT environments, admins need periodic network access to managed servers, potentially on networks with overlapping IP address ranges. This is difficult to achieve with traditional network and security tools, and can lead to onerous compliance reporting requirements. | Access to managed servers can be controlled by a business process (such as an open service desk ticket). The SDP can overlay complex network topologies, simplifying and streamlining access, while logging user activities to meet compliance requirements. |
| **Simplifying network integrations** | Organizations are periodically required to rapidly integrate previously disparate networks—for example, in M&A or disaster recovery scenarios. | With SDP, networks can be quickly and non-disruptively interconnected without requiring wholesale changes. |
| **Enabling secure transition to IaaS cloud environments** | Adoption of Infrastructure-as-a-Service (IaaS) has grown dramatically, yet many organizations using IaaS still cite security as a concern. For example, IaaS access controls may be disconnected from the enterprise, and limited in scope to the cloud provider. | Improved IaaS security. In addition to hiding an application behind a default-deny firewall, all traffic is encrypted, and user access policies can be defined consistently across a heterogeneous enterprise. See ***SDP for IaaS***, published by CSA on February 13, 2017, to learn more.[6] |

6 https://cloudsecurityalliance.org/download/sdp-for-iaas/

| NETWORK SCENARIO | LIMITATIONS OF EXISTING TECHNOLOGY | BENEFITS OF UPGRADING TO SDP |
|---|---|---|
| **Strengthening authentication schemes** | Security and compliance concerns may require the addition of 2FA to existing "legacy" applications. This can be difficult to achieve for non-web apps and for apps that cannot easily be modified. | SDP can require 2FA prior to granting access to specific applications. SDP can use in-place multi-factor authentication (MFA) systems for improved user experience, and can add MFA to enhance the security of legacy applications. |
| **Streamlining enterprise compliance controls and reporting** | Compliance reporting can create tremendously time-consuming and costly workloads for security and IT teams. | SDP reduces compliance scope (via microsegmentation), and automates the compliance reporting task (via identity-centric logging and reporting of access). |
| **Distributed denial of service (DDoS) prevention** | Traditional remote access solutions expose hosts and ports on the Internet, and are subject to DDoS attacks. Not only do all good packets get dropped, but low bandwidth DDoS attacks bypass traditional DDoS security controls. | SDP uses a default-drop firewall, and can be deployed with no visible presence to unauthorized users, allowing only good packets through. |



# SDP ARCHITECTURE

**The primary components of SDP architecture include the client/initiating host (IH), a service/accepting host (AH), and an SDP controller, to which the AH and IH both connect.** SDP hosts can either initiate connections (initiating hosts, or IH), or they can accept connections (accepting hosts, or AH). IH and AH connections are managed by interactions with SDP controllers via a secure control channel. This structure is what enables the control plane to remain separate from the data plane in order to achieve a completely scalable security system. In addition, all of the components can be redundant for scale or uptime purposes. By following the workflow outlined here, connections between these three components can be secured using the techniques outlined in *Figure 1*.

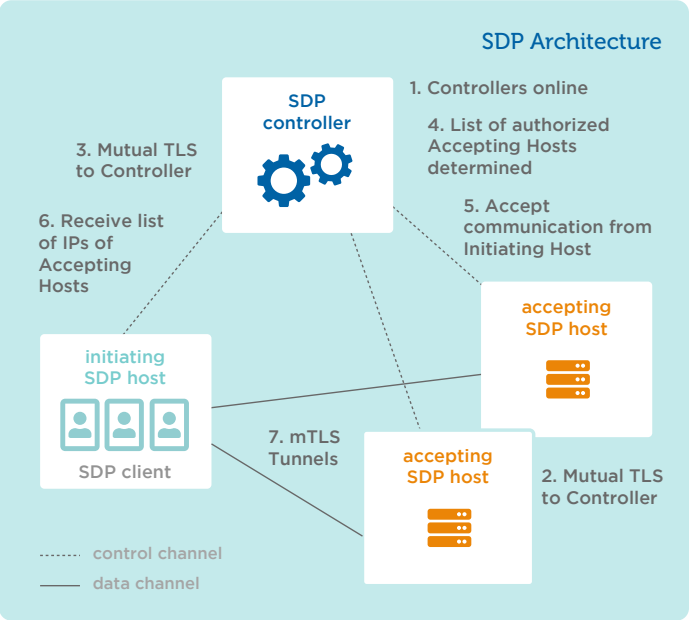



*Figure 1: SDP Architecture, previously published by CSA in **SDP Specification 1.0***

## HOW IT WORKS

The **SDP client** software on the **IH** initiates connections to the SDP.

**IH** devices including laptops, tablets and smartphones are user-facing, meaning the SDP software is run on the device itself. The network can be outside the control of the enterprise operating the SDP.

**AH** devices accept connections from IHs and provide a set of services protected securely by the SDP. AHs typically reside on a network under the enterprise's control (and/or a direct representative).

**An SDP gateway** provides authorized users and devices with access to protected processes and services. The gateway can also enact monitoring, logging, and reporting on these connections.

IH and AH devices connect to an **SDP controller**, which is an appliance or process that secures access to isolated services by ensuring that users are authenticated and authorized, devices are validated, secure communications are established, and user and management traffic on a network remain separate.

The controller and AH are protected by single-packet authorization (SPA), making them invisible and inaccessible to unauthorized users and devices. SPA is referenced throughout this document and in detail on page 21.

Security in an SDP follows this specific step-by-step workflow:

1. One or more SDP controllers are added and activated within the SDP and connected to authentication and authorization services, such as AM, PKI service, device attestation, geolocation, SAML, OpenID, OAuth, LDAP, Kerberos, multi-factor authentication, identity federation, and other similar services.
2. One or more AHs are added and activated within the SDP. They connect to and authenticate the controllers in a secure manner. The AHs do not acknowledge communication from any other host, and will not respond to any non-provisioned requests.
3. Each IH is added and activated within the SDP and connects with and is authenticated by SDP controllers.
4. After authenticating an IH, SDP controllers determine a list of AHs to which the IH is authorized to communicate.
5. The SDP controller instructs the AHs to accept communication from the IH and initiates any optional policies required for encrypted communications.
6. The SDP controller gives the IH the list of AHs, as well as any optional policies required for encrypted communications.
7. The IH initiates a SPA to each authorized AH. The IH then creates two-way encrypted connections (e.g. mutual TLS, or mTLS) to those AHs.
8. The IH communicates with target systems via the mutually encrypted data channel, through the AH. (Note: Step 8 is not depicted in *Figure 1* on the previous page).

## SDP Deployment Models

CSA's ***SDP Specification 1.0*** introduced the following potential ways organizations can deploy an SDP architecture:

- Client-to-Gateway
- Client-to-Server
- Server-to-Server
- Client-to-Server-to-Client
- Client-to-Gateway-to-Client
- Gateway-to-Gateway

### Client-to-Gateway

**When one or more servers must be protected behind a gateway, the connections between client/IH and gateway are secured regardless of underlying network topology.** Gateways may be in the same location or distributed internationally.

In this model, the client/IH is connected to the gateway directly via an mTLS tunnel where the connection terminates. To secure the connection to servers, additional precautions must be taken. The SDP controller may be located in the cloud or near the protected servers so the controller and servers are using the same SDP gateway.

In *Figure 2*, servers (in one or more environments) are protected behind an SDP gateway acting as AH. To secure connections to servers through the gateway,

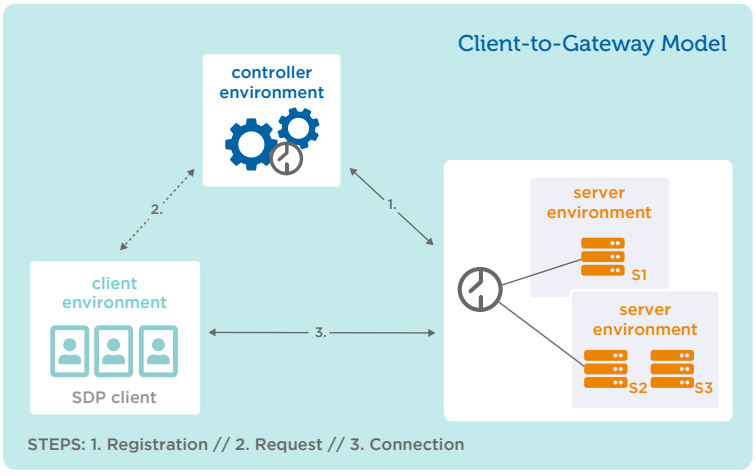



*Figure 2: Client-to-Gateway Model: One or more servers are protected behind the Gateway*

server environments should be controlled by the organization operating the SDP.

The protected servers are unreachable except from a properly onboarded client/IH, and the gateway and controller are secured with SPA with a default-drop firewall, so they are "dark" and inaccessible to unauthorized users and potential attackers.

Protected servers can be included in an SDP without making any changes to the servers. The network on which they reside, however, will need to be configured to permit inbound connections to protected servers from the gateway only, which will prevent unauthorized clients from bypassing the gateway(s).

This model preserves the ability for an organization to use its existing network security components—such as IDS/IP—by deploying them between the SDP gateway and the protected servers. Traffic is also monitored after being extracted from the mTLS tunnel that connects clients to the gateway(s).

The client/IH may be a device or may itself be a server. (See "Server-to-Server" on page 16.)

The Client-to-Gateway model is suited for organizations moving their applications to the cloud. Regardless of where the server environment is located (cloud, on-premises or nearby), organizations must ensure data is secured between the gateway and applications.

This model is also suited to securing on-premises legacy applications, because no changes are required to the IH.

## Client-to-Server

**When an organization is moving applications to an IaaS provider to secure connections end-to-end, this model combines a server and gateway in a single host.** The client/IH may be located in the same location as the server, or distributed. In either case, connections between client/IH and server are secured end-to-end.

This model provides organizations a great deal of flexibility, because server-gateway combinations can be moved between multiple IaaS providers as needed. This model is also appropriate for securing on-premises legacy applications that cannot be upgraded.

In this model, the client/IH is connected directly to a secure server via an mTLS tunnel, at which point the connection terminates. The SDP controller may be located on the server (so the controller and server are using the same gateway), or

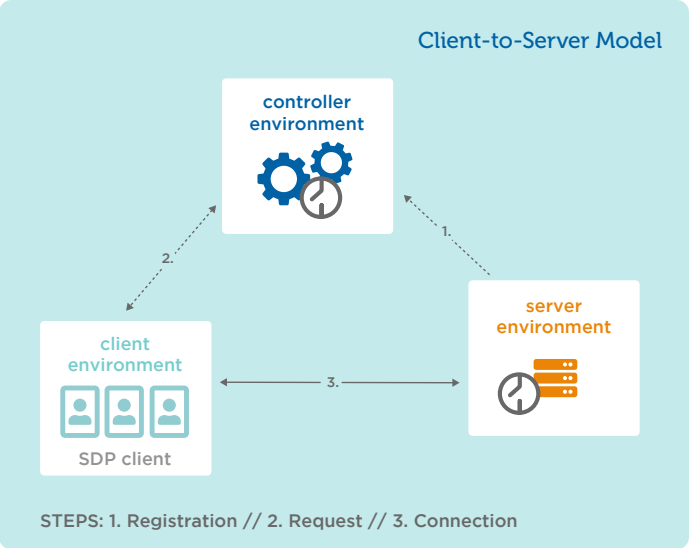



*Figure 3: Client-to-Server Model: Server runs the Gateway software locally*

in the cloud.

The server is protected behind an SDP gateway (acting as AH). The secure connection to the server (in the server environments) going through the gateway, may be under the control of the owner of the application/services on the server, giving the owner full control of these connections.

The protected servers are unreachable except from an appropriately onboarded client/IH, and the gateway and controller are secured with SPA with a default-drop firewall. This means the servers are "dark" and inaccessible to internal and external attackers and unauthorized users, which provides superior protection from an inside threat.

With this model, the protected servers will need to be outfitted with the gateways. The network on which the servers reside will not need to be configured to restrict inbound connections to the protected servers. The gateways (the enforcement points) on these servers use SPA to prevent unauthorized connections.

This model makes it easier to use existing network security components, such as IDS/IPS or SIEMs. Traffic can be monitored by analyzing these dropped packets from the SDP gateway/protected servers, thereby preserving the mTLS connections between the client/IH and the servers. (Also note that the client/IH, although depicted as a user device, may itself be a server. In this instance, refer to the Server-to-Server model below.)

The Client-to-Server model is well-suited for organizations moving their applications to the cloud. Regardless of where

the server environment is located (cloud or on-premises), organizations have full control over connections to their applications in the cloud.

## Server-to-Server

**This model is best suited for Internet of Things (IoT) and Virtual Machine (VM) environments, and ensures that all connections between servers are encrypted regardless of the underlying network or IP infrastructure.** The Server-to-Server model also ensures communications are explicitly permitted by an organization's SDP whitelist policy. Communications between servers across untrusted networks are secured, and servers remain hidden from all unauthorized connections using a lightweight SPA protocol.

This model is similar to the Client-to-Server model on the previous page, except that the IH is itself a server, and can also act as an SDP AH. Like the Client-to-Server model, the Server-to-Server model requires that the SDP gateway or similar lightweight technology be installed on each server and renders all server-to-server traffic "dark" to other elements of the security ecosystem. Network-based IDS/IPS will need to be configured to get dropped packets from the SDP gateway instead of externally. In addition, organizations may rely on host-based IDS/IP systems.

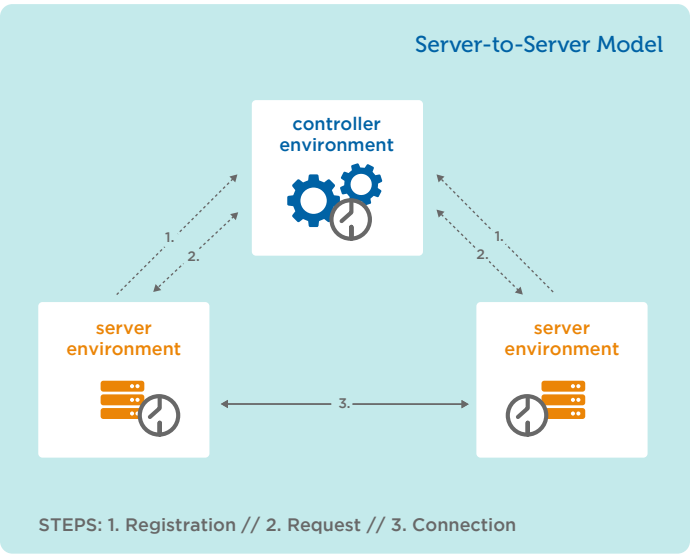



*Figure 4: Server-to-Server Model: Any communication including API calls and system services.*

The SDP controller may be located on the servers so that the controller and the servers are using the same SDP gateway. The SDP controllers may also reside in the cloud.

Servers are protected behind an SDP gateway acting as AH. The secure connections to the servers (in the server environments) going through the gateway are under the control of the owner of the application/services on the server by default, giving the owner full control of these connections.

The protected servers are unreachable except from other whitelisted servers, and the gateway and controller are secured with SPA with a default-drop firewall, so the servers are "dark" and inaccessible to attackers and unauthorized users (both internal and external), providing for extra protection from the insider threat.

With this model, the protected servers will need to be outfitted with the gateways or lightweight SPA protocol mechanisms. The network on which the servers reside will not need to be configured to restrict inbound connections to the protected servers. The gateways (the enforcement point) on these servers utilize SPA to prevent both internal and external unauthorized connections.

This model makes it easier to use network security components such as IDS/IPS and SIEMs. Traffic can be monitored by analyzing all the dropped packets from the SDP gateway/protected servers, thereby preserving the mTLS connections between the protected servers.

This model is well suited for all environments in which organizations are moving their IoT and VM environments to the cloud. Regardless of where the server environment is located (cloud or on-premises), organizations can have full control over the connections to their environments in the cloud.

## Client-to-Server-to-Client

**In some instances, peer-to-peer traffic passes through an intermediary server, such as in IP phone, chat and video conferencing services.** In these cases, the SDP obfuscates the IP addresses of connecting clients, encrypts network connections between the components, and protects the server/AH from unauthorized network connections via SPA.

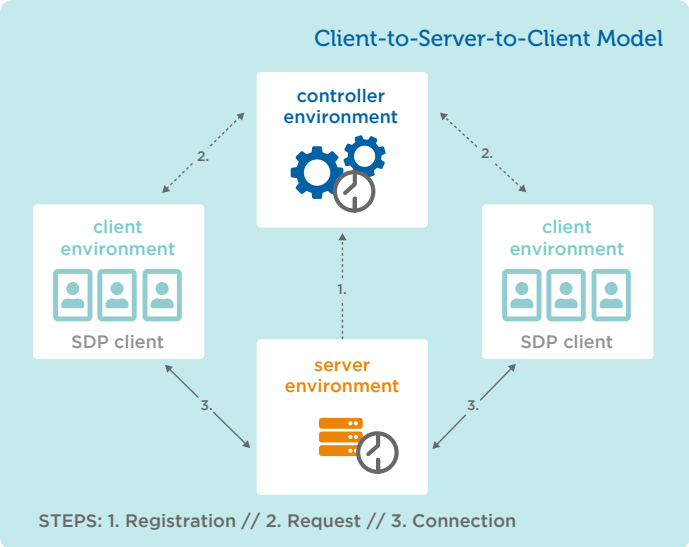



*Figure 5: Client-to-Server-to-Client Model: Used for peer-to-peer-style services such as IP phone or chat.*

The SDP controller may be located on the servers (so controller and servers are using the same SDP gateway) or in the cloud. As depicted above, the server is protected behind an SDP gateway acting as AH. The secure connections to the server going through the gateway are controlled by the owner of the applications/services on the server by default.

The protected server is unreachable except from other appropriately onboarded clients, and the gateway and controller are secured with SPA with a default-drop firewall, so the server is "dark" and inaccessible to attackers and unauthorized users (both internal and external) providing for extra protection from the insider threat.

With this model, the protected server will need to be outfitted with the gateways or lightweight SPA protocol mechanisms. The network on which the servers reside will not need to be configured to restrict inbound connections to the protected server. The gateway (the enforcement point) on the server use SPA to prevent both internal and external unauthorized connections.

This model makes it easier to use network security components, such as IDS/IPS and SIEMs. Traffic can be monitored by analyzing all the dropped packets from the SDP gateway/protected servers thereby preserving the mTLS connections between the clients and the protected server.

This model is well suited for all environments in which organizations are moving their peer-to-peer applications to the cloud. Regardless of where the server environment is located (cloud or on-premises), organizations can have full control over the connections to the clients.

## Client-to-Gateway-to-Client

**This model is a variation of Client-to-Server-to-Client, above.** This model supports peer-to-peer network protocols requiring clients to connect directly to one another while enforcing SDP access policies.

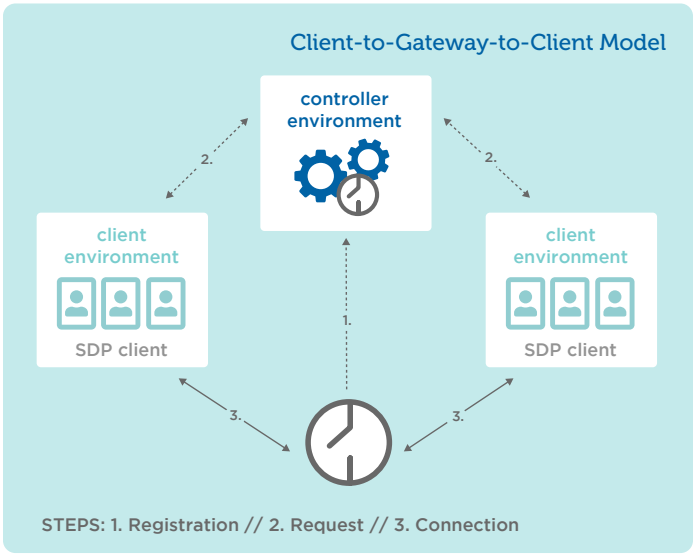



*Figure 6: Client-to-Gateway-to-Client Model: Used to secure client-to-client communications.*

This results in a logical connection between the clients (each acting as either IH, AH, or both depending on the application protocol). Note that the application protocol will determine how the clients connect to one another; the SDP gateway acts as a firewall between them.

More information will be added for this model in a future publication.

## Gateway-to-Gateway

**The Gateway-to-Gateway model was not included in the initial publication of *SDP Specification 1.0*.** This model is well-suited for certain Internet of Things (IoT) environments. In this scenario, one or more servers sit behind the AH such that the AH acts as the gateway between clients and servers. At the same time, one or more clients sit behind an IH such that the IH also acts as a gateway.

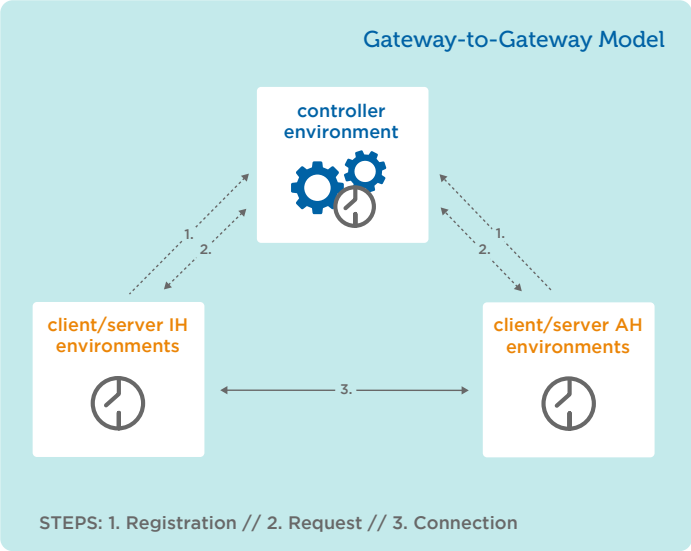



*Figure 7: Gateway-to-Gateway Model: One or more servers or clients are protected behind the Gateway*

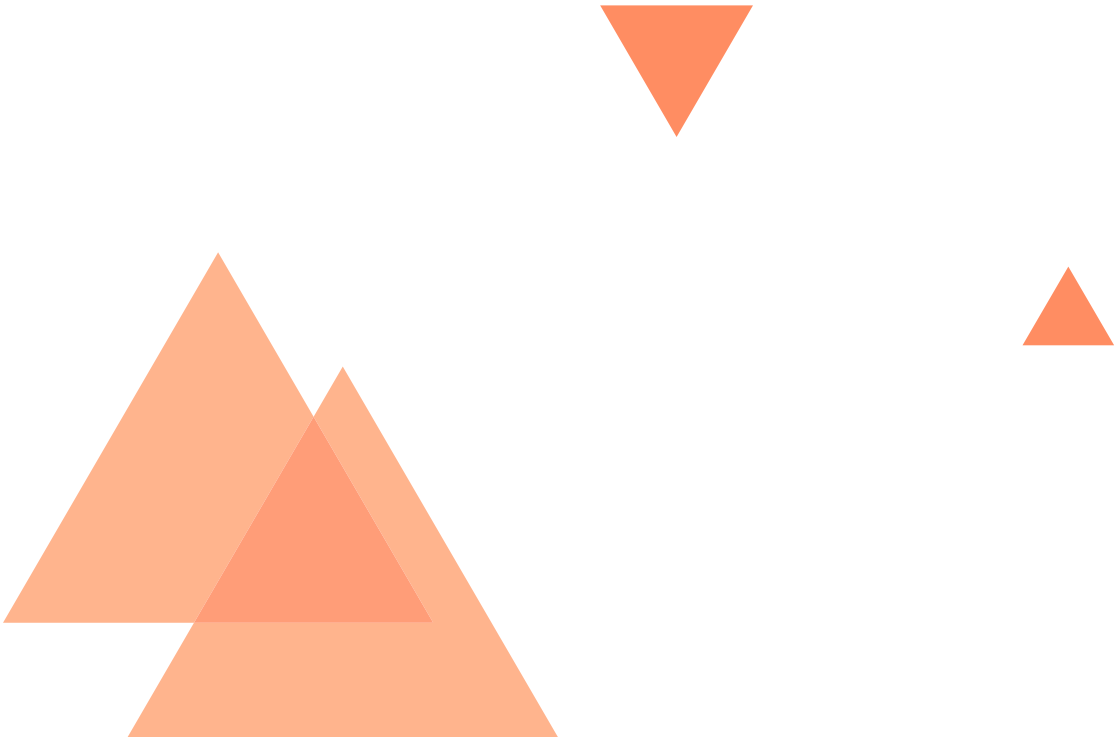
In this model, client devices do not run the SDP software. The devices may include those for which it is not desirable or possible to install an SDP client, such as printers, scanners, sensors, and IoT devices. In this model, the gateways operate as firewalls, and also potentially as a router or proxy, depending on the implementation.

## SDP Deployment Models and Corresponding Scenarios

The table below shows which deployment models can take advantage of which SDP scenarios. Each type of deployment requires different connections to be secured.

| NETWORK SCENARIO | CLIENT-TO-GATEWAY | CLIENT-TO-SERVER | SERVER-TO-SERVER | CLIENT-TO-SERVER-TO-CLIENT | CLIENT-TO-GATEWAY-TO-CLIENT | GATEWAY-TO-GATEWAY |
|---|---|---|---|---|---|---|
| **Identity-driven network access control** | Y | Y* | Y | Y | Y | Y** |
| All SDP models support identity-driven network access control.<br>* This model provides secure connections to the network and the service.<br>** The degree to which SDP can identify devices for this model depends on the way the specific SDP implementation performs device identification and validation. For example, MAC addresses provide weaker identity validation than 802.1x. | | | | | | |
| **Network microsegmentation** | Y* | Y ** | Y *** | Y | Y | Y |
| All SDP models provide network microsegmentation by securing individual connections.<br>* This model provides microsegmentation by securing connections between clients and gateways, but does not micro-segment connections to servers behind gateways.<br>** This model provides network microsegmentation by securing all connections to servers. In addition, servers hosting gateways are hidden.<br>*** This model provides network microsegmentation by securing all connections to the server. In addition, servers hosting the gateway are hidden. | | | | | | |
| **Securing remote access (VPN alternative)** | Y | Y | Y | Y | Y | Y |
| SDP is a replacement for traditional VPNs. In all cases, the controller and gateway/AH must be accessible to remote devices so they can use SPA to initiate connections. | | | | | | |
| **Third-party user access** | Y | Y | Y* | Y | Y | Y |
| SDP supports third-party access for all scenarios depending on the connections required to be secured. Third parties may be remote or on-premises, and may also have a separate identity provider to which they authenticate.<br>* SDP secures connections from all third-party applications accessing internal applications for this model, with the third-party application acting as the client. | | | | | | |
| **Securing privileged user access** | Y | Y | N | Y | Y | N |
| SDP secures privileged user access for connections from the client. Typically, privileged user access refers to clients accessing servers, but can apply to all the models, depending on the application in question. | | | | | | |
| **Securing access to high-value applications** | Y | Y | Y | Y | Y | N |
| All models except Gateway-to-Gateway secure all connections related to high-value applications secured by that particular model. | | | | | | |
| **Securing access to managed servers** | Y | Y* | Y | Y | N | Y |
| This scenario is for service provider access to managed servers. The servers can be hidden by gateways entirely, or in the case of a managed services environment, only the management plane is hidden by gateways.<br>* In this model, the SDP gateway software is deployed on the server. The servers are hidden, and the MSSP/managed service detects and controls connections to services. | | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Simplifying network integrations** | Y | Y* | Y* | Y | Y | Y |
| All SDP deployment models support this scenario, with different types of secure connections required.<br>* For these models, an additional advantage is that the services on the server can be hidden with a gateway. | | | | | | |
| **Secure Transition to IaaS Cloud Environments** | Y | Y | Y | Y | Y | Y |
| This scenario involves moving services from on-premises to the cloud. | | | | | | |
| **Robust Authentication Schemes** | Y | Y | Y* | Y | Y | Y |
| All SDP models provide the ability to strengthen authentication, often via multi-factor/step-up authentication.<br>* This model, with no user, cannot prompt for a one-time password. It can support multi-factor authentication, however, such as using a PKI or a server-based HSM. Identity management systems can (and should) also be used for systems or devices, and not just users. | | | | | | |
| **Streamlined Compliance Controls and Reporting** | Y | Y | Y | Y | Y | Y |
| All SDP models help enterprises streamline compliance by integrating the controls. | | | | | | |
| **DDoS Attack Prevention** | Y | Y | Y | Y | Y | Y |
| Because all SDP models use SPA within the gateway, they increase the organization's resilience to DDoS attacks. In cases where a deny-all gateway is not used, Internet-facing services are more frequently subject to DDoS compared to internally-hosted services. | | | | | | |



# SDP Connection Security

**As an architecture, SDP provides the protocol to secure connections at all layers of a network stack.** *Figure 8* depicts the connections secured by each SDP deployment model. By deploying gateways and controllers at key locations, implementers can focus on securing connections most critical to their organizations and can secure these connections from network-based and cross-domain attacks.

## Single-Packet Authorization (SPA)

**One of the most critical elements of SDP technology is that it requires and enforces an "authenticate before connect" model, which compensates for the open and insecure nature of TCP/IP.** SDP accomplishes this through single-packet authorization (SPA), a lightweight security protocol that validates a device or user's identity before permitting network access to the relevant system component (controller or gateway).

The information for a connection request, including the requester's IP address, is encrypted and authenticated in a single network message. The purpose of SPA is to allow a service to be darkened via a default-drop firewall. The system should drop all TCP and UDP packets and not respond to those attempts, providing no information to potential attackers that the port is being monitored. After authentication and authorization, the user is granted access to the service. SPA is integral to SDP, where it initiates communication in the connections made between clients and controllers; gateways and controllers; and clients and gateways.

While implementations of SPA may differ slightly, they should share the following tenets:

1. Packet must be encrypted and authenticated
2. Packet must self-contain all necessary information; packet headers alone are not trusted
3. Packet must not depend on admin or root-level access in order to generate and send; no raw packet manipulation allowed
4. Server must receive and process packets as silently as possible; no response or verification will be sent

### BENEFITS OF SPA

**SPA plays a huge role in SDP.** One of the goals of SDP is to overcome the fundamentally open/insecure nature of TCP/IP, which follows a "connect, then authenticate" model. Given today's cyber security threat landscape, it is unacceptable to permit malicious actors to scan and connect to our enterprise systems. SPA in combination with SDP addresses this vulnerability in two ways. Applications using the SDP architecture are hidden behind an SDP gateway/AH so they are accessible only to authorized users. In addition, the SDP components themselves—the controller and gateway—are protected by SPA. This allows them to be deployed securely

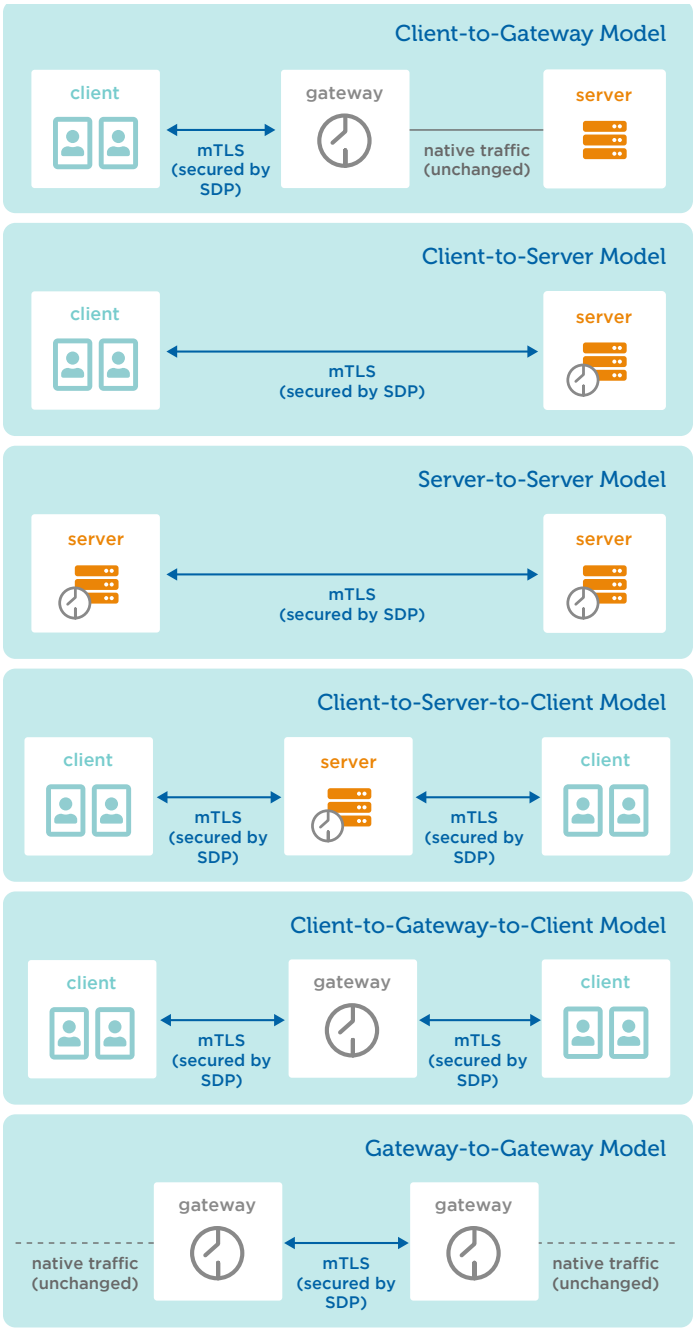



*Figure 8: Connections secured by each deployment model*

in Internet-facing situations, ensuring legitimate users have productive and reliable access, while remaining invisible to unauthorized users. SPA offers the critical benefit of **service darkening**. A default-drop firewall stance mitigates port scanning and related reconnaissance techniques. This type of firewall renders SPA components invisible to unauthorized users, significantly reducing the attack surface of an entire SDP. SPA offers more security than VPNs, with open ports and other known vulnerabilities in many implementations.

Another advantage of SPA over similar technologies is **zero-day protection**. When a vulnerability is newly discovered, it is inherently less damaging when only authenticated users can access the affected service(s).

SPA also protects against Distributed Denial of Service (DDoS) attacks. A relatively small amount of traffic has the potential to take an HTTPS service offline, if that service is exposed to the public internet for attack. SPA only makes a service visible to authenticated users, so any DDoS attack is handled by a default-drop firewall instead of by the protected service itself.

### LIMITATIONS OF SPA

**SPA is only one part of SDP's layers of security, and is not complete on its own.** While SPA implementations should be designed to be resilient to replay attacks, SPA can be subject to Man-In-The-Middle (MITM) attacks. Specifically, if a MITM adversary is able to capture/alter a SPA packet, the adversary may be able to establish the TCP connection to the controller/AH rather than to the authorized client. However, this adversary will be unable to complete the mTLS connection without the client's certificate. The controller/AH should therefore reject this connection attempt and close the TCP connection. Despite the MITM scenario, SPA is far more secure than standard TCP.

SPA implementations may differ slightly from different vendors. An open-source SPA reference implementation is available in the Fwknop project, or "FireWall KNock OPerator."[14] For more detailed technical information about SPA, refer to "Appendix 2" on page 38. Another excellent reference is the chapter "Trusting the Traffic" in Zero Trust Networks by Evan Gilman and Doug Barth (O'Reilly Media, Inc., 2017).

## SDP and Access Control

**The value of SDP as an emerging architecture is that it strengthens access control management and sets standards for implementing user access management, network access control, and system authentication control.** SDP has the ability to enforce access control by preventing any network-level access from unauthorized users and/or using unvalidated devices. Because SDP deploys a deny-all gateway, it blocks or allows or prevents network packets from flowing between the IH and AH. At a minimum, SDP enables organizations to define and control their own access policies determining which identities should access which network services and from which validated devices.

SDP does not attempt to displace existing identity and access management solutions, but enhances user authentication-only access control. SDP significantly decreases potential attack surfaces by integrating user authentication and authorization with other security components (see "Primary Functions of SDP" on page 8). As an example, user Jane might not have credentials to sign in to a company's production financial management server, but if it is simply visible to her device on the network, it presents a risk. If Jane's company implements an SDP architecture, the financial management server is hidden from Jane's device. So even if an attacker has a foothold on Jane's machine, SDP will prevent connections from her device to the financial management server that she does not have credentials for. Even if Jane did have credentials allowing access to the financial management server, having an SDP client installed on her device provides additional protection. The attacker would still be thwarted by multi-factor user authentication coupled with robust device validation.

14 https://www.cipherdyne.org/fwknop/

# COMPLEMENTARY ARCHITECTURES

## Zero Trust and BeyondCorp

**In addition to the Software-Defined Perimeter, there are two other new approaches in today's security landscape:** the "Zero Trust" concept initially driven by industry analyst firm Forrester, and Google's internal "BeyondCorp" initiative.

### Forrester's Zero Trust Model

Forrester's Zero Trust model[15], which has increased in scope over the past few years, is built on three principles:

- Ensuring all resources are accessed securely, regardless of the location of user or resource
- Logging and inspecting all traffic
- Enforcing the principle of least privilege

These foundations are aligned with what SDP provides. An SDP architecture may be the best way to implement the principles of Zero Trust. Because SDP strongly authenticates users and devices, and also encrypts network connections, it ensures that all resources are accessed securely, regardless of user location. Because SDP is typically deployed as an overlay across existing networks, SDP also ensures that all resources are accessed securely regardless of the location of the resource—on-premises, in the cloud, or elsewhere. In an SDP implementation, network connections also are controlled, providing a centralized place to log which identities (human or machine) are accessing which resources. SDP can be integrated easily with a network traffic inspection system if deep packet inspection is desired. Finally, and perhaps most importantly, SDP inherently and strongly enforces the principle of least privilege. Because SDP begins with a default, deny-all gateway, and access strictly follows a whitelist access model, identities can only access networked applications (and the SDP itself) when explicitly allowed by an SDP control. This is the essence of the principle of least privilege.

### Google's BeyondCorp Model

**BeyondCorp is Google's internal network and access security platform, designed to enable their employees access to internal resources.** BeyondCorp heavily emphasizes device validation to manage corporate-issued Chromebooks. The system has been well-researched and documented,[16] and it has been a successful internal project at Google over the past five years.

BeyondCorp differs from SDP in a number of ways. BeyondCorp is a web proxy-based solution that supports HTTP, HTTPS, and SSH protocols. SDP implementations generally support more (and in some implementations, all) IP protocols. SDP also supports more fine-grained access controls than BeyondCorp. In Google's system, applications are assigned to a small number of "trust tiers." SDP supports finer-grained and individualized access controls, driven by user and device context.

While Google's BeyondCorp is not commercially available, Google has included some elements of the platform in their open-source Istio platform for securing microservices.[17]

15 https://www.forrester.com/report/The+Forrester+Wave+Zero+Trust+eXtended+ZTX+Ecosystem+Providers+Q4+2018/-/E-RES141666 and https://go.forrester.com/blogs/next-generation-access-and-zero-trust/

16 https://cloud.google.com/beyondcorp/#researchPapers

17 https://cloud.google.com/istio/

Google has also launched a free component, called the Identity-Aware Proxy (IAP),[18] for controlling access into resources in the Google Cloud Platform (GCP). The IAP is not an SDP, nor does it have the full capability of BeyondCorp. According to Google, "Cloud IAP is a building block toward BeyondCorp."

If your enterprise is considering creating a Zero Trust environment, or if your team likes the BeyondCorp approach, you may also wish to evaluate SDP as it offers similar benefits with multiple commercially available versions.

To summarize, SDP as an architecture can ensure a successful implementation of Zero Trust principles. The BeyondCorp implementation should provide readers with the impetus to incorporate the SDP architecture into their BYOD strategy.

18 https://cloud.google.com/iap/

# SDP AND YOUR ENTERPRISE

**Enterprise Information Security Architecture can be complex, considering that numerous stakeholders across an organization all have security risks and concerns.** SDP ensures secure connections irrespective of the underlying IP infrastructure. SDP can be the basis for enterprise security architectures because it comprises the following key concepts:

1. authorize users and validate devices before allowing connections
2. bi-directional encrypted communication
3. dynamic rules on deny-all firewalls and hiding servers
4. integrate application context and fine-grained access control

In this section, we present a number of questions architects should consider as they plan to deploy SDP in their enterprise. The questions will help architects consider various aspects of security, including user populations, networks, server environments, and security and compliance requirements.

## HOW DOES AN SDP DEPLOYMENT FIT INTO EXISTING NETWORK TECHNOLOGIES?

**Architects must decide which SDP deployment model to use, and they must understand that for some models gateways may represent an additional in-line network component.** This may have implications on their organization's network, such as requiring some firewall or routing changes to ensure protected servers are invisible and accessible only through the SDP gateway.

## HOW DOES SDP AFFECT MONITORING AND LOGGING SYSTEMS?

**Because SDP uses mTLS between IH and AH, network traffic becomes opaque to intermediary services that may be in place for security, performance, or reliability monitoring purposes.** Architects must understand what systems are in operation, and how SDP-related changes to the network traffic may affect these systems. Because SDPs typically provide richer, identity-centric logging of user access, they can also be used to augment and enhance existing monitoring systems. In addition, all dropped packets from SDP gateways and controllers can be logged in an SIEM for further analysis. The "who, when, what, where" information for every connection becomes easier to collect.

## HOW DOES SDP AFFECT APPLICATION RELEASE/DEVOPS PROCESSES AND TOOLSETS, INCLUDING API INTEGRATIONS?

**Many organizations have adopted high-velocity application release processes, such as DevOps or CI/CD.**[19] These processes, and their supporting automation framework, require thoughtful integration with security systems, and SDP is no exception. SDPs can effectively secure connections by authorized users to the development environment during DevOps. SDP can also be used during operations to protect connections even from legitimate users to specially protected servers and applications.

Security architects must understand their SDP deployment model and how their organization's DevOps mechanisms will integrate with it. Security teams should look at the set of APIs supported by their SDP implementation, as API integration is often a requirement for DevOps toolset integration.

## HOW DOES SDP IMPACT USERS, ESPECIALLY BUSINESS USERS?

**Security teams often strive to make their solutions as transparent as possible for users.** SDP supports this approach. If the principle of least privilege is achieved, users will have full access to everything they need, and will not notice that unnecessary access has been denied. Depending on the SDP deployment model, users will run the SDP client software on their devices. Security architects should collaborate with IT to plan the user experience, client software distribution, and device onboarding processes.

19 https://en.wikipedia.org/wiki/DevOps and https://en.wikipedia.org/wiki/CI/CD

# Enterprise Information Security Architecture Elements

*Figure 9* below examines the primary elements of an enterprise security architecture. The diagram is a simplified view of a hybrid enterprise, depicting the primary elements of a prototypical security infrastructure and the relationships between these elements.

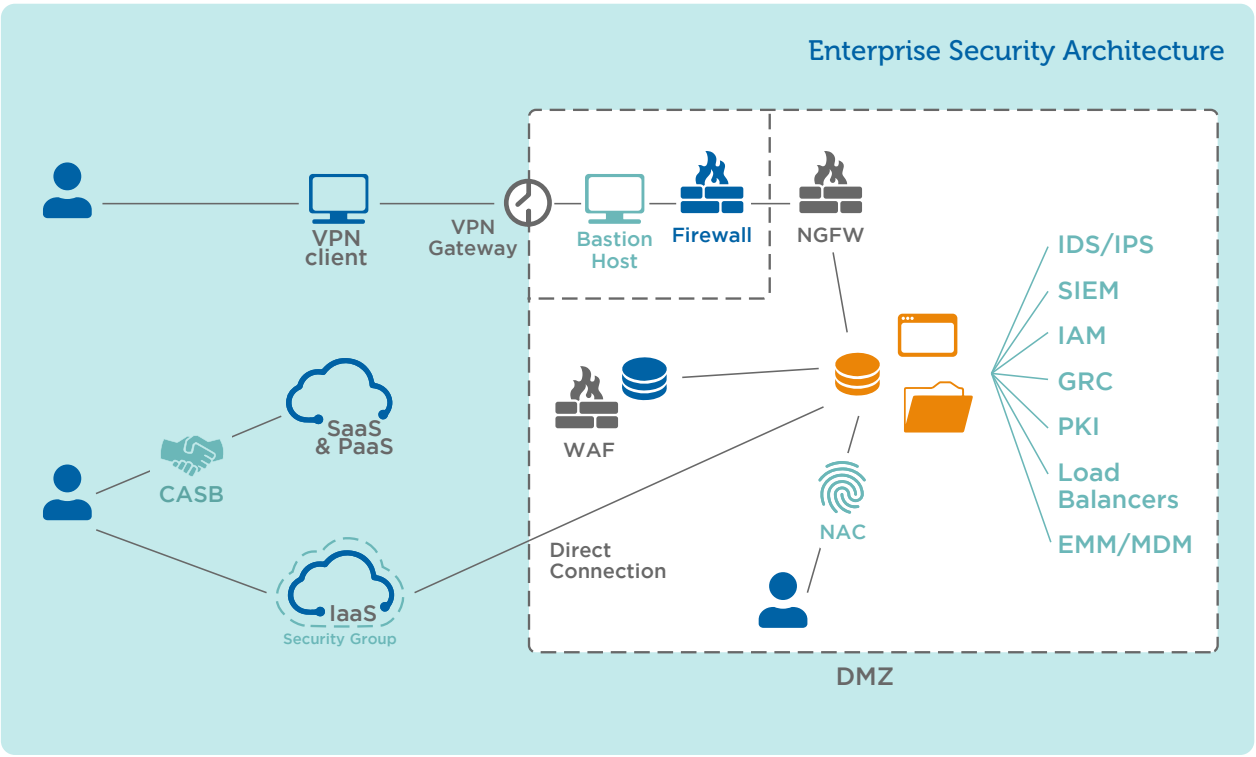



*Figure 9: Primary Elements of an Enterprise Security Architecture*

This example of enterprise security architecture is comprised of on-premises and cloud-based resources (IaaS and SaaS/PaaS), with a standard accompanying set of security, IT, and compliance components. How these standard components integrate with SDP is explored in more detail in the following pages.

## Security Information and Event Management (SIEM)

**SIEM systems[14] provide analysis of log information and security alerts generated by applications and network components.** SIEMs centralize the storage and interpretation of logs and allow near-real-time analysis, which enables security personnel to take defensive actions quickly. SIEM systems also provide the automated centralized reporting usually required for regulatory compliance.

SIEM systems, whether hosted on-premises or in the cloud, are a well-established and mainstream part of IT and security management systems. While commercial SDP solutions typically provide an internal logging capability, their value is magnified when SDP logs are directed to an enterprise SIEM system that aggregates information from multiple sources. An enterprise system might receive feeds directly from distributed SDP components, or it may deploy multiple collection agents in a hierarchical manner. The SIEM performs inspections and flags anomalies by forwarding predefined and customized events to a centralized management console or by sending alerts to designated individuals via email.

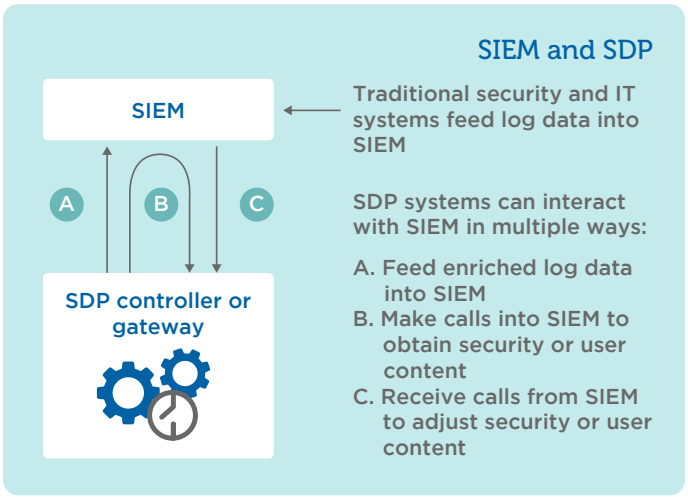



*Figure 10: SIEM with an SDP*

Because they control access in a manner that vets identities and devices, SDPs provide SIEMs with richer information than typical network and application monitoring tools. SDPs provide, in real time, the "who, what, where" information about every connection made, thereby increasing the value of SIEM systems. Compare this to how SIEMs are currently being used for logging: security analysts have to piece together information from multiple logs to even identify unauthorized users (the "who"). Identifying unauthorized connections made from "what" to "where" is even more challenging. If an SDP client is on a user's device, however, specific information from the device can also be collected. All dropped packets from the SDP gateway can be stored for further analysis into potential hack attempts or for evaluating attrition.

This level of recording is superior to the lists of IP addresses and ports generated by traditional firewalls. SDPs also enhance the SIEM's ability to associate user activity occurring across multiple devices. Associating user activity in this way is often difficult to achieve without SDP, especially with the advent of BYOD and mobile devices.

Integrating SIEMs with SDP deployments helps accomplish the goal of moving security operations from a reactive to a proactive endeavor. In order to control risk, an existing SIEM should also be considered an important source of information, in addition to being a sink for SDP log information.The SIEM can help control risk by disconnecting users, disallowing connections from unvalidated devices or from certain hosts, and dropping suspicious connections. For example, if an SIEM indicates a higher-than-normal risk level indicating unauthorized user activity, the SDP will then drop all connections from the user until further analysis can be performed. SDP complements SIEM by addressing and controlling connections in seconds.

Like all systems that generate log information, SDP logs represent a potential data privacy concern for an enterprise. Because network connections (and their metadata) may be associated with specific users in the logs, organizations need to take precautions during deployment of SDPs in order to address this concern.

SDP augments and improves the ability of SIEM systems to prevent, detect, and respond to different types of attacks. The following page shows some examples of the types of attacks that can be mitigated. (A more exhaustive list of attacks that may be prevented by integrating SDP with SIEMs is slated for a future CSA publication.)

14 https://en.wikipedia.org/wiki/Security_information_and_event_management

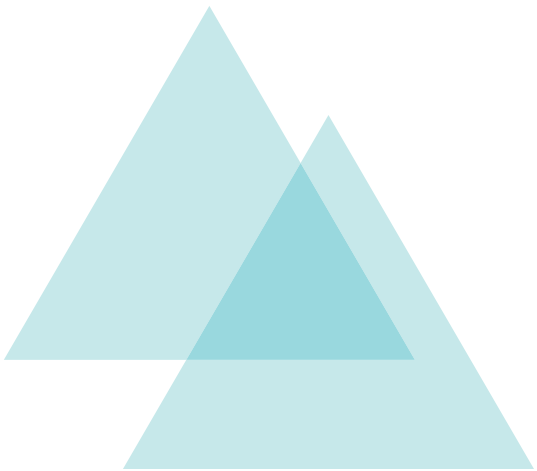
| TYPE OF SECURITY ATTACK | MITIGATION | HOW SDP INTEGRATES WITH SIEM |
|---|---|---|
| **Port scan / Network reconnaissance** | Block and Notify | SDP blocks all unauthorized network activity, and can log all connection requests for use within the SIEM. |
| **DoS attack** | Block and Notify | Because the SDP is protected by SPA, DoS attacks are largely ineffective. SPA drops bad packets, which can be logged to the SIEM. |
| **Malicious use of authorized resources** | Detect and Address | Authorized user access to authorized resources is permitted by SDP, but the SIEM can analyze user activity for anomalous behavior, and the SDP can then disallow access by the authorized user until further analysis can be performed. |
| **Use of stolen credentials** | Block and Notify | SDP requires multiple factors prior to connection, rendering stolen passwords insufficient for attackers to obtain access. |

## Traditional Firewalls

**Traditional firewalls monitor and control network traffic according to a set of rules based on the seven-layered Open Systems Interconnection (OSI) model**, in which layers 2, 3 and 4 of OSI are applied: data-link layer (2), network layer (3), and transport layer (4). They adhere to the 5-tuple[14] method, which filters network packet data based on source and destination IPs and ports, and definitions of the network protocol flowing over the connection. Firewalls may also support other functions, including network address translation (NAT) and port address translation (PAT).

Firewalls have been a mainstay of enterprise network security for decades. They do, however, have many limitations since they are only one piece of the security infrastructure and operate only within the limited world of the 5-tuple. Typically, traditional firewalls are only capable of expressing static rule sets and cannot express or enforce rules based on identity information.

SDPs, which use firewalls or implement the equivalent network traffic enforcement capabilities, significantly improve the way enterprises currently use firewalls. SDP will take on much of the network access control enforcement that organizations are attempting to control with firewalls. Enterprises can reduce their firewall rulesets considerably with SDP. Rather than attempting to model identity-centric access controls within the constraints of the 5-tuple, SDP allows more accurate representation and enforcement via SDP access controls. In addition to reducing the effort required to write, test, debug, and deploy firewall rules in complex environments, SDP also enables richer and more precise access control mechanisms.



14 https://www.techopedia.com/definition/28190/5-tuple

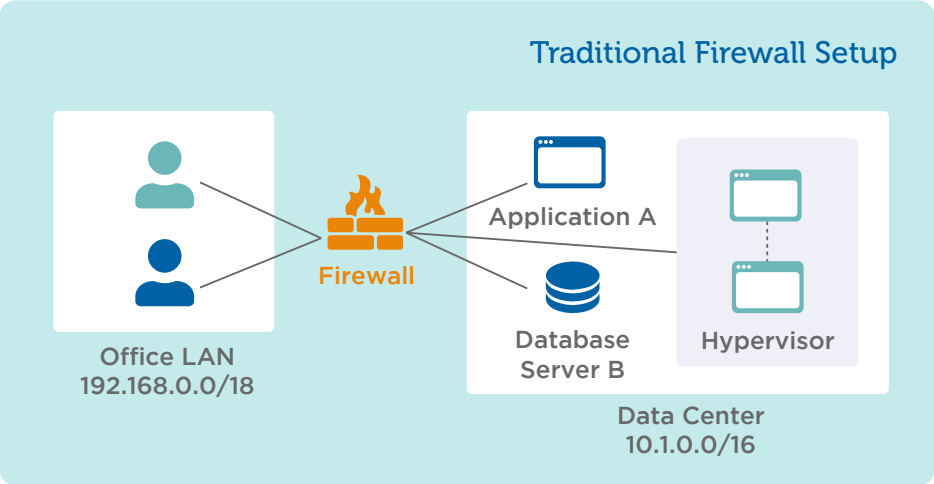



*Figure 11: Traditional Firewall Setup*

*Figure 11*, above, depicts the difficulties of attempting to secure access in a traditional office LAN environment, in which a single firewall controls access from the user subnet (192.168.100.0/18) to a local data center subnet.

The firewall cannot distinguish between users on the office network because they appear only as IP addresses. In addition, because many users are on laptops that regularly connect and disconnect, users' IP addresses frequently change.

A typical data center network houses a variety of workloads, including both testing and production systems. While some applications are long-lived and have static IP addresses, others are built on virtual machines, which are regularly created and destroyed (and therefore have unpredictable IPs). While no individual user requires access to all servers (and all ports on those servers) in the data center, this environment effectively forces the firewall to have a rule set that permits all IPs in the office LAN to access all IPs in the data center network. Compare the traditional firewall setup with *Figure 12*, below, which depicts a simplified version of the SDP Client-to-Gateway model (omitting the controller for clarity). Note that other SDP deployment models similarly apply.

In this example, the network firewall has been replaced by the SDP gateway, which performs the same functions.[15] Because the SDP controls access based on specific identities and the devices they use, it enables the organization to enforce fine-grained access control into the data center. The open, flat network—which represents a large attack surface—has been minimized. Note that even more fine-grained access to specific services can be gained by adding more SDP gateways closer to or on the servers in the data center. (See "Client-to-Server" on page 15.)

15 In a real-world deployment, the firewall may remain in place, but with a minimal rule set, such as permitting office LAN traffic to reach only the SDP gateway, which then enforces user connectivity from specific devices to specific services.

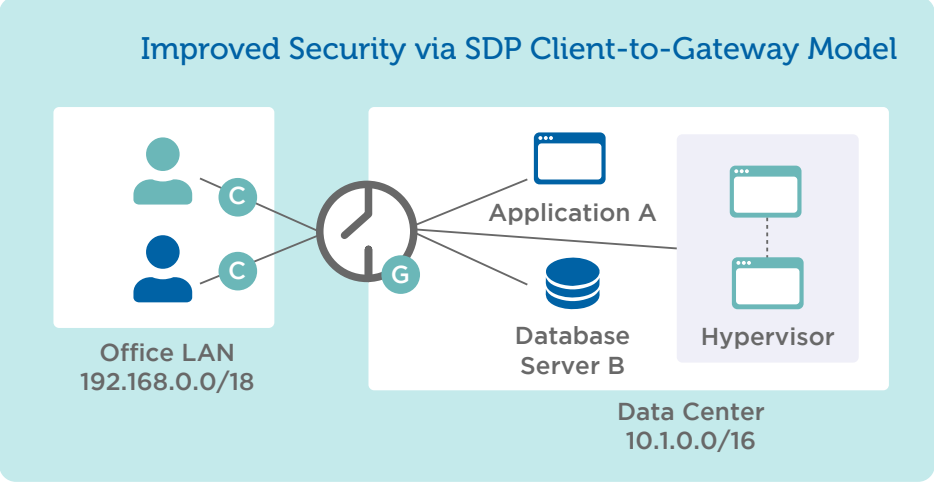



*Figure 12: Improved Security via an SDP Client-to-Gateway Model*

## Intrusion Detection/Prevention Systems (IDS/IPS)

**Intrusion Detection and Intrusion Prevention Systems (IDS/IPS)[16]—treated synonymously here—are security components that monitor a network or system for malicious activity or policy violations.** They may be network-based (inspecting traffic) or host-based (inspecting activity and potentially network traffic). SDP can bolster the deployment of IDS/IPS systems, although it may require changes to network-based IDSs. In smaller operations (e.g. single network remote offices), SDP may even eliminate the need for IDS/IPS, which can reduce costs.

Additionally, because SDP uses mutual TLS encryption between the client and the gateway, network traffic becomes opaque to network-based IDSs. IDSs typically proxy the TLS traffic by importing certificates, which has a side effect of increasing attack surface.[17] It is generally not possible within SDP to increase the attack surface because it is based on mutual TLS (often with ephemeral certificates) and is therefore resilient to the MITM role played by the IDS.

The mTLS segment of these logical connections (depicted in blue) are opaque to any outside system, because they are encrypted using the SDP's credentialing and are, by design, inaccessible to systems attempting to analyze this traffic. This change will impact intermediary security and network monitoring systems in a manner similar to the shift from TLS 1.2 to 1.3, in which certain use cases are no longer possible.[18] (For a graphical representation for each type of SDP deployment, see "SDP Connection Security" on page 21.)

In addition, SDPs support the ability to push unencrypted traffic (e.g. dropped packets) to a remote IDS. Alternatively, a host-based IDS may enhance security operations more than a network-based IDS/IPS can. SDP is not the only trend impacting network-based IDSs. The move to cloud-based applications has also increased the effectiveness and adoption of host-based IDSs.

While deploying SDP may require some changes to IDS systems, it will bring the benefit of reducing noise in the system by blocking all unauthorized network traffic. This shift allows the IDS, and the team operating it, to focus on network traffic to authorized applications, and to shift resources to detecting insider threats.

SDP also can simplify and enhance the creation and effectiveness of a "honeypot" system. Because protected systems are invisible to attackers, SDP increases the likelihood that a malicious actor will find and attack the honeypot, because it represents a larger proportion of visible systems to all users. An SDP-honeypot system can lead to faster detection of malicious activity on the network.

## Virtual Private Networks (VPNs)

**VPNs establish secure private network connections across untrusted networks.** VPNs are commonly used for secure remote access (e.g. off-site employee to corporate site), secure inter-site communications, and even between different companies (a site-to-site extranet). VPNs commonly use TLS or IPSec.[19]

While VPNs provide encapsulation and encryption of network traffic, they also suffer from limitations that are better addressed by SDP. While licensing costs of VPNs may be low, anecdotally they can require significant operational effort. VPNs generally provide broad, overly permissive network access. VPNs typically also provide only basic access control limits (e.g. based on subnet ranges). These limitations represent security and compliance risks in many organizations. In distributed environments, VPNs may require the unnecessary backhauling of user traffic through a corporate data center, adding latency and bandwidth costs. The VPN server itself also is exposed as a service on the Internet, rendering them potentially vulnerable to attackers.

In addition, VPNs impose a considerable burden on users, delivering an often-poor user experience. Users are required to remember which applications require use of the VPN (and which ones do not), and they are also required to manually connect and disconnect. For users who need to access multiple remote locations, VPNs often prevent them from connecting to both locations simultaneously, requiring them to switch back and forth between environments. Whenever cloud migration is involved, VPN management balloons in complexity, forcing IT administrators to configure and sync VPN and firewall policies across multiple locations. This complexity makes it even more difficult to eliminate unwarranted access.

VPNs are a prime target for replacement by SDP. Similar to

16 https://en.wikipedia.org/wiki/Intrusion_detection_system
17 https://www.sans.org/reading-room/whitepapers/vpns/snort-ssl-tls-inspection-37735
18 https://tools.ietf.org/id/draft-camwinget-tls-use-cases-00.html
19 https://en.wikipedia.org/wiki/Virtual_private_network

VPNs, SDP requires a client to be installed on the user's device. By using SDP instead of VPN, organizations can have a single access control platform consistent for remote, on-premises and mobile device users. And because SDPs, especially those exposed to the Internet, provide zero visibility via SPA and dynamic firewalls, they are considerably more resilient to attacks than typical VPN servers.

## Next-Generation Firewalls (NGFW)

**In general, NGFWs[20] have the attributes of traditional firewalls, plus additional capabilities—making them "next generation."** They monitor access and examine network packets according to predefined rules and filter data using the information in OSI layers 2 through 4. NGFWs also use the information in layers 5 through 7 (session layer, presentation layer, and application layer) to perform additional functions.

NGFWs provide the following capabilities, depending on vendor:

- **Application Awareness:** Recognizes applications to determine what types of attacks to seek
- **Intrusion Detection System (IDS):** Monitors the security status of the network
- **Intrusion Prevention System (IPS):** Denies traffic in order to prevent security breaches
- **Identity Awareness (User and Group Control):** Manages which resources users can access
- **Virtual Private Network (VPN):** NGFWs may provide this capability for remote user access from untrusted networks

While NGFWs represent significant improvements over traditional firewalls, they also have limitations when compared with SDP:

- **Latency:** Like any IDS/IPS, they impose additional latency on network traffic, especially when performing file inspection.
- **Scalability:** They may require substantial hardware in order to scale up.
- **Rule complexity:** Some NGFW vendors include capabilities related to identity, such as user and group attributes, but these can be complex to implement.

SDP is a natural complement to NGFWs already deployed. Enterprises can use SDP to ensure user access policies, while also using their NGFWs for core firewall protection and IDS/IPS for traffic inspection. The benefit of integrating SDP with your NGFW is to enforce zero visibility as well as making the NGFW more dynamic (described in more detail later in this section). User access policies can also be enforced by integrating NGFWs with IAMs or AD, but SDP offers truly secure connections that can be controlled.

In some ways, NGFW architectures compete and overlap with SDP. In the past decade, NGFW vendors have been successful and have innovated to solve some of the same problems that SDP addresses. By combining NGFW and VPN capabilities with user and application awareness, enterprises can, to some degree, accomplish many of the goals of SDP. However, there are architectural differences between doing this and implementing SDP. NGFW systems are IP-based, whereas SDP is connection-based. NGFWs offer limited identity and application-centric capabilities. Their access policy models are typically coarse-grained, providing users broader network access than what is strictly necessary. NGFWs also tend to be less dynamic than SDPs, which can include external systems in access decisions. For example, an SDP may only permit developer access to staging servers during an approved change management window. SDPs also are capable of enforcing step-up authentication, which is typically not supported by NGFWs.

NGFWs are still firewalls, and they often mandate traditional perimeter-centric network architectures with site-to-site connections. SDP deployments usually support more distributed and flexible networks, thereby enabling a flexible network segmentation capability. SDP is fundamentally based on a "need to know" (whitelist) security model, which hides unauthorized services from unauthorized users and unauthorized devices. SDP leverages SPA and dynamic firewalls to protect and hide authorized connections. NGFWs typically operate in more highly visible environments.

20 https://en.wikipedia.org/wiki/Next-generation_firewall

## Identity and Access Management (IAM)

**IAM systems provide mechanisms for users and devices to validate their identities (via authentication), and to store managed attributes and group memberships about those identities.** The SDP architecture is designed to integrate with existing enterprise IAM providers, such as LDAP, Active Directory, and SAML.

SDPs control access are typically based on factors including IAM attributes and group memberships, along with attributes of the devices being used to make connections. The combination of user and device authorization criteria help create granular access rules that grant or restrict access, ensuring only authorized access from authorized users on authorized devices to authorized applications.

SDP integration with IAM is not only used for initial user authentication, but also for step-up authentication, such as prompting for an OTP in order to access sensitive systems, or under certain circumstances (such as remote access versus on-premises). IAM systems can also communicate with SDP via API calls in response to identity lifecycle actions, such as disabling an account, changing group membership, dropping connections from users, or changing user roles.

IAM is used within SDP to authenticate users, to provide information that SDP uses to make authorization decisions, and to enable enriched audit logs for all access granted to users from registered devices. Tying application access (not network access) to users (not IP addresses) yields useful connection information for logging and significantly reduces IT overhead when it is necessary to audit historical access for security or compliance reasons.

IAM tools typically focus on the business processes that maintain the identity lifecycle and standardize how identity information is used to control access to resources. For example, the mechanism for "provisioning" users for access is usually a combination of manual and automated processes. SDP supports these IAM processes because it relies on the identity attributes and group memberships managed by IAM tools. As user attributes or group memberships change, SDP automatically detects these changes and alters user access without altering the IAM processes.

SDP integrates with SAML.[21] Within an SDP deployment, a SAML provider may act as an identity provider for user attributes and/or as an authentication provider (e.g. for MFA). In addition to SAML, there are many open authentication protocols, such as OAuth,[22] OpenID Connect,[23] W3C Web Authentication (WebAuthn),[24] and the FIDO Alliance Client-to-Authenticator Protocol (CTAP).[25] (These protocols will be explored in future SDP-related research.)

## Network Access Control (NAC) Solutions

**NAC solutions[26] typically control which devices can connect to a network and which network subjects can be accessed.** These solutions most commonly use standards-based hardware (802.1X) and software to validate devices prior to granting them access to the network, and they operate at Layer 2 of the OSI model.

At the time the device first appears on the network, NACs perform device validation and then assign devices to a network segment (VLAN). In practice, NACs are used to assign devices coarsely to a small number of networks. Most organizations have only a few networks, such as "Guest," "Employee," and

21 https://en.wikipedia.org/wiki/Security_Assertion_Markup_Language
22 https://en.wikipedia.org/wiki/OAuth and https://oauth.net/
23 https://en.wikipedia.org/wiki/OpenID_Connect
24 https://www.w3.org/TR/webauthn/
25 https://fidoalliance.org/specs/fido-v2.0-id-20180227/fido-client-to-authenticator-protocol-v2.0-id-20180227.html and https://fidoalliance.org/fido2/
26 https://en.wikipedia.org/wiki/Network_Access_Control

"Production." Because NACs operate at Layer 2 of the network, they often require specific network gear, do not operate in cloud environments, and cannot be used remotely.

SDP is a modernized solution to NAC that integrates user and device access. However, there are some environments in which it makes sense to use NAC—for example, hardware devices like printers, copiers, landline phones, and security cameras. These devices often have 802.1X support built-in and cannot typically support the installation of an SDP client. The SDP Gateway-to-Gateway model to protect these devices and control user access to them is a better option and will be a topic for future SDP research.

## Endpoint Management (EMM/MDM/UEM)

**Many enterprises utilize endpoint management systems, often categorized as Enterprise Mobility Management (EMM), Mobile Device Management (MDM), or Unified Endpoint Management (UEM).** These are important elements of enterprise IT and security, and their value and importance is augmented by an SDP deployment.

Endpoint management systems can be used to automate the distribution and installation of SDP clients across user devices. Because these systems often use the same IAM systems as SDP, the rollouts can be closely coordinated to simplify the user experience. These systems often also provide functionally rich capabilities around device introspection and profile evaluation. An SDP can make API calls into a device management platform to obtain information about a specific device, and it can then make dynamic access decisions based on the information.

Alternatively, organizations that do not have an endpoint management system, can take advantage of the management and control of devices provided by SDP.

## Web Application Firewalls (WAF)

**Web Application Firewalls (WAFs) are designed to filter, monitor, and block HTTP(S) web traffic to and from web applications.** WAFs introspect application protocol traffic to block attacks from application security flaws, such as SQL injection, cross-site scripting (XSS), and file inclusion.[27] WAFs are not a network access control or network security solution, despite typically operating inline in the network between users and applications in a manner similar to IDS/IPS. WAFs primarily examine HTTP(S) protocol traffic to detect and block malicious content.

WAFs are complementary to SDP. For example, in the Client-to-Gateway model, WAFs are deployed behind an SDP gateway, operating on the native web application traffic after it has been extracted from the SDP mTLS tunnel. In the Client-to-Server and Server-to-Server models, WAFs integrate with the SDP gateway on the servers for more distributed control of HTTP(S) traffic inspection.

## Load Balancers

**Load balancers are part of many network and application architectures.** Load balancers include DNS and network-based solutions, and architects need to understand how their organization uses them when planning SDP deployment.

For example, network-based load balancers are typically deployed inline on the network, between clients and servers, and similar to the WAF discussion above, may not be able to inspect the mTLS connection between SDP components. The specifics of the SDP deployment and load balancing approach need careful analysis to ensure that SDP can be deployed to maximum benefit.

## Cloud Access Security Brokers (CASB)

**CASBs sit between cloud service users and cloud applications and monitor all activity related to the enforcement of security policies.** They offer a variety of services, including monitoring user activity, warning administrators about potentially hazardous actions, enforcing security policy compliance, and automatically preventing malware.[28] CASBs may sit in-line between users and cloud services, or they may use SaaS APIs to sit "inside" the SaaS system, depending on the vendor and the level of API support within the SaaS platform.

CASB capabilities typically do not overlap with those of SDP because they generally operate at Layer 7 (application layer), examining application traffic. They typically do not provide network security or access control. However, their operations can be simplified by also using SDP for data protection and user behavior analysis.

27 https://en.wikipedia.org/wiki/Web_application_firewall

28 https://en.wikipedia.org/wiki/Cloud_access_security_broker

## Infrastructure as a Service (IaaS)

**Security for IaaS platforms is built around an industry-standard "shared responsibility" model**,[29] in which cloud providers have certain responsibilities (security of the cloud), while customers are responsible for securing their applications (security within the cloud). IaaS customers use cloud network security groups[30] to control access to their cloud resources. These network security groups are configured and operate as simple firewalls. These security measures can be integrated with SDP to create a much more robust security environment.[31]

## Software as a Service (SaaS)

**SaaS applications such as Salesforce.com and Office 365 are multi-tenant and are publicly accessible on the Internet.** Preventing network-level access by unauthorized users is currently not a goal for these systems. Organizations may wish to strengthen security when using SaaS applications for the following reasons:

- Ensure only authorized users on authorized devices can access the SaaS tenant for that particular organization
- Ensure managed corporate IAM credentials are used to authenticate to SaaS applications
- Enforce multi-factor authentication for access to the SaaS application
- Ensure all user visits to the SaaS application are identified and logged

A growing number of SaaS providers recognize that their enterprise customers want these benefits and have begun including "source IP address and device restriction" capabilities. These features work equally well with SDP and traditional VPNs, and enable a SaaS customer to restrict user access (logins and usage) for their domain (tenant) via specific IP addresses. For SDP, that source IP is an element of the system (the gateway) through which user traffic is routed and authorized or denied.

29 https://aws.amazon.com/compliance/shared-responsibility-model/ and https://blogs.msdn.microsoft.com/azuresecurity/2016/04/18/what-does-shared-responsibility-in-the-cloud-mean/
30 https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html and https://docs.microsoft.com/en-us/azure/virtual-network/security-overview
31 https://downloads.cloudsecurityalliance.org/assets/research/sdp/sdp_for_iaas.pdf

## Platform as a Service (PaaS)

**Platform-as-a-Service[32] offerings allow enterprises to build and deploy custom applications with less overhead than standard hardware or IaaS-based systems.** Unlike IaaS and SaaS, network access control into PaaS systems (and thus the degree to which SDP is relevant) is dependent on the function provided and the ways in which the PaaS provider has enabled external access controls.

However, the major PaaS providers support the same network security models for their PaaS platforms as for IaaS. For example, the Microsoft Azure PaaS security model [33]supports source IP address restrictions, via Azure network security groups. So do the Google Cloud Platform App Engine[34] and AWS Elastic Beanstalk.[35] The various SDP models can be deployed depending on what the PaaS applications are and which connections need to be secured.

## Governance, Risk Management, and Compliance (GRC)

**The discipline of governance, risk management, and compliance,[36] which is often part of an enterprise's overall enterprise security framework, helps ensure organizations achieve security objectives and act with integrity.** GRC systems, often implemented through purchased GRC software,[37] define and enforce controls around many organizational systems, including IT, typically via standards and guidelines (e.g. SOX, PCI, etc.).

SDPs can interact with and support GRC systems by enforcing and documenting access controls required by GRC. For example, a GRC system may require that production systems be isolated from non-production systems, and that all user access to production systems be logged. An SDP can enforce this network segmentation, and it can provide the GRC system with an audit log for validation.

32 https://en.wikipedia.org/wiki/Platform_as_a_service
33 https://docs.microsoft.com/en-us/azure/security/security-paas-deployments
34 https://cloud.google.com/vpc/docs/firewalls
35 https://aws.amazon.com/premiumsupport/knowledge-center/security-group-elastic-beanstalk/ and https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features.managing.ec2.html
36 https://en.wikipedia.org/wiki/Governance,_risk_management,_and_compliance
37 https://searchcio.techtarget.com/definition/GRC-governance-risk-management-and-compliance-software

## Public Key Infrastructure (PKI)

**Public Key Infrastructure (PKI)[38] is "a set of roles, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates and manage private and public keys used for encryption, decryption, hashing and signing."** SDPs may use PKI to generate TLS certificates and secure connections. Even if no PKI infrastructure exists, SDPs can provide TLS certificates to secure connections.[39] Existing PKIs are a natural integration point for SDPs because they can be used by the SDP certificate generation as well as optionally for user authentication.

## Software-Defined Networking (SDN)

**Software-Defined Networking is an API-driven orchestration of the IT network infrastructure used to enable orchestration of network routing within an IT network.** SDN enables efficient network configuration in order to improve performance and monitoring.[40] The focus of SDN is traffic efficiency—not security and authorization. A well-run SDN system delivers reliable, efficient, and adaptive network bandwidth to an enterprise.

SDP provides orchestration of connections between objects on the network regardless of the underlying network infrastructure. SDP can be integrated to benefit from the deployment of an SDN, but does not require one. For example, SDP and SDN controllers can be integrated. An SDN can also provide network Quality of Service (QoS) for the opaque, encrypted mTLS connection.

## Serverless Computing Models

**As computing models evolve, security tools and architectures must evolve along with them.** One example is the growth in "serverless" computing models,[41] in which a cloud provider offers the capacity to run either custom code (in a "function as a service" model) or a pre-built set of code (such as in a "serverless database").

A "function as a service" model may expose a universal public endpoint to the entire Internet and use an API key to control authentication and authorization. In this case, the SDP models would not apply because these interactions are designed to be public. However, other services (or other cloud providers) may choose to follow a different security model, in which each customer has their own dedicated access point for the "as a service" function. In this model, SDP can be applied to secure the private access point by an SDP gateway.

## Architectural Considerations

**While broad enough to cover a substantial set of network access scenarios, SDP is not intended to solve all security issues.** Some exclusions include the following areas:

- Securing or controlling access to public network services (such as a website that does not require authentication)—SDP is better suited to membership-based services
- Endpoint protection
- Certain computing models, such as serverless computing
- Certain network connection topologies, such as peer-to-peer, depending on the SDP deployment model (see "SDP Deployment Models and Corresponding Scenarios" on page 19)

38 https://en.wikipedia.org/wiki/Public_key_infrastructure
39 https://cloudsecurityalliance.org/download/software-defined-perimeter-glossary/
40 https://en.wikipedia.org/wiki/Software-defined_networking
41 https://en.wikipedia.org/wiki/Serverless_computing

# CONCLUSION

**Information security presents such critical challenges to today's enterprises and governmental institutions that we must all adopt more effective approaches to securing our organizations' data assets.** The Software-Defined Perimeter (SDP) approach can give security professionals in organizations the tools they are seeking to provide a strong, adaptable, and manageable foundation for robust development, operations and security. We hope this document provides security professionals with a better understanding of how SDP architecture works and how it can be deployed into their unique situations.

However, our job is not done. We have more topics to cover in future research, including papers on each of the discrete SDP deployment models and integrations presented above, plus elaboration of the benefits of SDP for various businesses, compliance controls mapping tools based on SDP deployments, and many other publication goals.

Most importantly, we recognize that we do not have all the answers! We invite you to join our SDP Working Group to engage in the conversation and offer your contributions. As security professionals who support a secure, open, and available Internet, and as ethically motivated human beings, we are working hard to secure a better future. We hope you'll join us on this journey. Learn more about participating at https://cloudsecurityalliance.org/working-groups/software-defined-perimeter/.

Appendix 1

# ADDITIONAL RESOURCES

"Software-Defined Perimeter Working Group: SDP Specification 1.0," by Brent Bilger, Alan Boehme, Bob Flores, Zvi Guterman, Mark Hoover, Michaela Iorga, Junaid Islam, Marc Kolenko, Juanita Koilpilla, Gabor Lengyel, Gram Ludlow, Ted Schroeder, and Jeff Schweitzer (CSA, April 2014).
SDP v1.0 Spec

"Software-Defined Perimeter for Infrastructure as a Service," by Jason Garbis and Puneet Thapliyal (CSA, 2016).
SDP for IaaS

"Software-Defined Perimeter Working Group Glossary," Cloud Security Alliance (CSA, 2018).
SDP Glossary

"Zero Trust Networks: Building Secure Systems in Untrusted Networks," by Evan Gilman and Doug Barth (June 2017).
http://shop.oreilly.com/product/0636920052265.do

"fwknop: Single Packet Authorization > Port Knocking," by Michael Rash (CipherDyne).
http://www.cipherdyne.org/fwknop/

"Open Source Software-Defined Perimeter," Waverley Labs.
http://www.waverleylabs.com/open-source-sdp/

# Appendix 2
# SPA DETAILS

Below is the format of the SPA packet as defined in the SDP 1.0 specification.

| | | |
|---|---|---|
| **CIPHERTEXT** | Nonce | Prevents servicing outdated SPA packets |
| | Timestamp | Most common: Service Access Request |
| | Message Type | Possibly deprecated: access request, NAT access request, gateway command message |
| | Message String | Source IP address to allow, service ID(s) to open |
| | Optional Fields | NOTE: Gateway knows which port to open as well as whether and where to forward the connection. |
| | Digest | NOTE: Might be used to request tunneling of service traffic. |
| **CLEARTEXT** | HMAC | Before encryption, this SHA256 hash is calculated over the ciphertext portion of the message and then used by the server to verify message integrity after a successful message decrypt. |

A suggested improvement to this definition is to add a cleartext client ID, which would allow for more efficient handling of incoming packets. An effort to design a binary SPA format is under way, as well as the creation of an RFC document to describe the format.

SPA is most effective when sent as a single UDP packet. In some use cases, this is not practical as network environments may block some or all outgoing UDP packets. In such cases, SPA packets can be sent over TCP connections. This technically violates the "Single Packet" nature of SPA, but is sometimes necessary as a practical consideration.

A SPA packet can be sent by the connecting machine or another device. An example of this use is when a mobile device is used to send a SPA packet on behalf of a desktop computer. In some scenarios, this also is a reasonable workaround for an environment that blocks UDP packets.