



cloud
CSA security
allianceSM



The diagram features a large white cloud at the bottom. Two white lines extend upwards from the cloud. The left line connects to a white cloud containing the CSA logo. The right line connects to a smaller white cloud. Both lines have USB connector icons at their ends: a blue USB-A connector on the left and a white USB-B connector on the right.

Software Defined Perimeter Working Group

Software Defined Perimeter

December 2013

© 2013 Cloud Security Alliance – All Rights Reserved.

All rights reserved. You may download, store, display on your computer, view, print, and link to the Cloud Security Alliance Software Defined Perimeter at <http://www.cloudsecurityalliance.org>, subject to the following: (a) the Document may be used solely for your personal, informational, non-commercial use; (b) the Document may not be modified or altered in any way; (c) the Document may not be redistributed; and (d) the trademark, copyright or other notices may not be removed. You may quote portions of the Document as permitted by the Fair Use provisions of the United States Copyright Act, provided that you attribute the portions to the Cloud Security Alliance Software Defined Perimeter (2013).

Acknowledgments

Editors

Brent Bilger, VP Solutions Architecture, Vidder

Contributors

Alan Boehme, Director CSA; Chief of Enterprise Architecture and Emerging Technologies, The Coca-Cola Company

Bob Flores, Former CTO, Central Intelligence Agency

Jeff Schweitzer, Chief Innovation Architect, Verizon

Junaid Islam, CTO, Vidder

Contents

1.0 Introduction.....	5
1.1 The Changing Perimeter	5
1.2 SDP Concept	6
2.0 SDP Architecture.....	6
2.1 SDP Controller	7
2.2 Initiating SDP Hosts	7
2.3 Accepting SDP Hosts.....	7
2.4 SDP Workflow	8
3.0 SDP Implementations	9
3.1 Client-to-Gateway	9
3.2 Client-to-Server	9
3.3 Server-to-Server	9
3.4 Client-to-Server-to-Client	9
4.0 SDP Applications.....	10
4.1 Enterprise Application Isolation	10
4.2 Private Cloud and Hybrid Cloud	10
4.3 Software as a Service.....	10
4.4 Infrastructure as a Service.....	10
4.5 Platform as a Service	10
4.6 Cloud-Based VDI	11
4.7 Internet-of-Things.....	11
5.0 SDP's Relationship to IKE/IPsec and TLS.....	11
6.0 Working Group Purpose	12
7.0 Core Values.....	12
8.0 Scope	12
9.0 Deliverables	12
10.0 Membership & Structure of the Working Group	13
11.0 Advisory Roles	13
12.0 Duration.....	13

1.0 Introduction

This document explains the software defined perimeter (SDP) security framework and how it can be deployed to protect application infrastructure from network-based attacks. The SDP incorporates security standards from organizations such as the National Institute of Standards and Technology (NIST) as well as security concepts from organizations such as the U.S. Department of Defense (DoD) into an integrated framework. The Cloud Security Alliance (CSA) intends to create a public standard that is freely available for use without license fees or restrictions.

The premise of the traditional enterprise network architecture is to create an internal network separated from the outside world by a fixed perimeter that consists of a series of firewall functions that block external users from coming in, but allow internal users to get out. Traditional fixed perimeters allowed internal services to remain secure from external threats for a number of years due to the powerful but simple characteristics of blocking visibility and accessibility from outside the perimeter to internal applications and infrastructure. But the traditional fixed perimeter model is rapidly becoming obsolete because of BYOD and phishing attacks providing untrusted access inside the perimeter and SaaS and IaaS changing the location of the perimeter.

Software defined perimeters address these issues by giving application owners the ability to deploy perimeters that retain the traditional model's value of invisibility and inaccessibility to "outsiders," but can be deployed anywhere – on the internet, in the cloud, at a hosting center, on the private corporate network, or across some or all of these locations.

The SDP brings together standard security tools including PKI, TLS, IPsec, SAML, and standards, as well as concepts such as federation, device attestation, and geo-location to enable connectivity from any device to any infrastructure. Connectivity in an SDP is based on a need-to-know model, in which device posture and identity are verified before access to application infrastructure is granted. Application infrastructure is effectively "black" (a DoD term meaning the infrastructure cannot be detected), without visible DNS information or IP addresses. SDP mitigates the most common network-based attacks, including: server scanning, denial of service, SQL injection, OS & application vulnerability exploits, password cracking, man-in-the-middle, cross-site scripting (XSS), cross-site request forgery (CSRF), pass-the-hash, pass-the-ticket, and many others (see NIST, SANS, and more). From an end-point perspective, an SDP uses a lightweight access protocol to support deployment on mobile applications, networked sensors, and application servers.

1.1 The Changing Perimeter

Historically, enterprises deployed a perimeter security solution in their data center to protect against external threats to their application infrastructure. However, the traditional perimeter model is rapidly becoming obsolete for two reasons:

1. Hackers can easily gain access to devices inside the perimeter (for example via phishing attacks) and attack application infrastructure from within. Moreover, this vulnerability continues to increase as the number of devices inside the perimeter grows due to BYOD, on-site contractors, and partners.

2. Traditional data center infrastructure is being supplemented with external resources such as PaaS, IaaS, and SaaS. Subsequently, networking equipment used for perimeter security is topologically ill-located to protect application infrastructure.

The growth of devices moving inside the perimeter and the migration of application resources to outside the perimeter has stretched the traditional security model used by enterprises. Existing workaround solutions that involve backhauling users to a data center for identity verification and packet inspection do not scale well. A new approach is needed that enables application owners to protect infrastructure in a public or private cloud, a server in a data center, or even inside an application server.

1.2 SDP Concept

The SDP aims to give application owners the ability to deploy perimeter functionality where needed. SDPs replace physical appliances with logical components that operate under the control of the application owner. SDPs provide access to application infrastructure only after device attestation and identity verification.

The principles behind SDPs are not entirely new. Multiple organizations within the 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 (Section 2.4).

SDPs maintain the benefits of the need-to-know model described above but eliminate the disadvantages of requiring a remote access gateway appliance. SDPs require endpoints to authenticate and be authorized first before obtaining network access to protected servers, and then, encrypted connections are created in real time between requesting systems and application infrastructure.

2.0 SDP Architecture

In its simplest form, the architecture of the SDP consists of two components: SDP Hosts and SDP Controllers. SDP Hosts can either initiate connections or accept connections. These actions are managed by interactions with the SDP Controllers via a secure control channel (see Figure 1). Thus, in SDPs, the control plane is separated from the data plane to enable a completely scalable system. In addition, all of the components can be redundant for scale or uptime purposes.

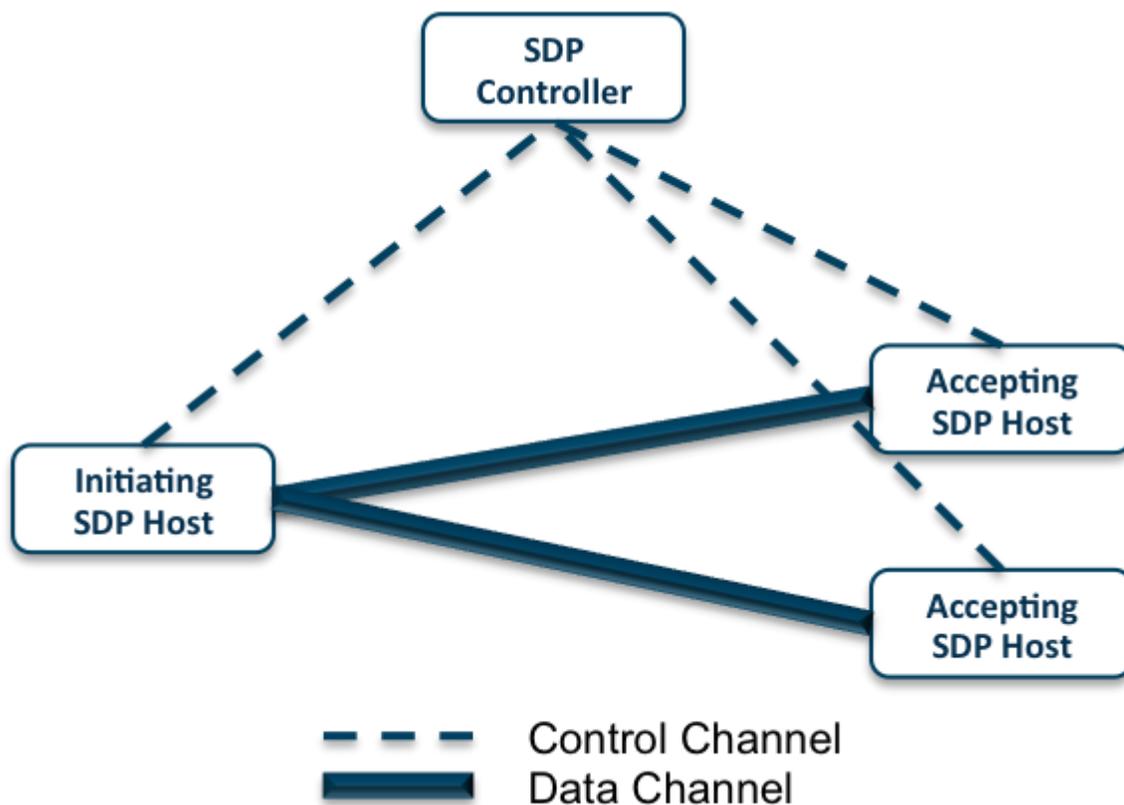


Figure 1: The architecture of the Software Defined Perimeter consists of two components: SDP Hosts and SDP Controllers

2.1 SDP Controller

The SDP Controller determines which SDP Hosts can communicate with each other. The Controller may relay information to external authentication services such as attestation, geo-location, and/or identity servers.

2.2 Initiating SDP Hosts

Initiating SDP Hosts communicate with the SDP Controller to request a list of accepting Hosts to which they can connect. The Controller may request information such as hardware or software inventory from the Initiating Host before providing any information.

2.3 Accepting SDP Hosts

Accepting SDP Hosts reject all communication from all hosts other than the SDP Controller. The Accepting Host accepts connections only at the request of the Controller.

2.4 SDP Workflow

The SDP framework has the following workflow (see Figure 2).

1. One or more SDP Controllers are brought online and connected to the appropriate optional authentication and authorization services (e.g., PKI, device fingerprinting, geolocation, SAML, OpenID, OAuth, LDAP, Kerberos, multifactor authentication, and other such services).
2. One or more Accepting SDP Hosts are brought online. These hosts connect to and authenticate to the Controllers. However, they do not acknowledge communication from any other Host and will not respond to any non-provisioned request.
3. Each Initiating SDP Host that is brought on line connects with, and authenticates to, the SDP Controllers.
4. After authenticating the Initiating SDP Host, the SDP Controllers determine a list of Accepting Hosts to which the Initiating Host is authorized to communicate.
5. The SDP Controller instructs the Accepting SDP Hosts to accept communication from the Initiating Host as well as any optional policies required for encrypted communications.
6. The SDP Controller gives the Initiating SDP Host the list of Accepting Hosts as well as any optional policies required for encrypted communications.
7. The Initiating SDP Host initiates a mutual VPN connection to all authorized Accepting Hosts.

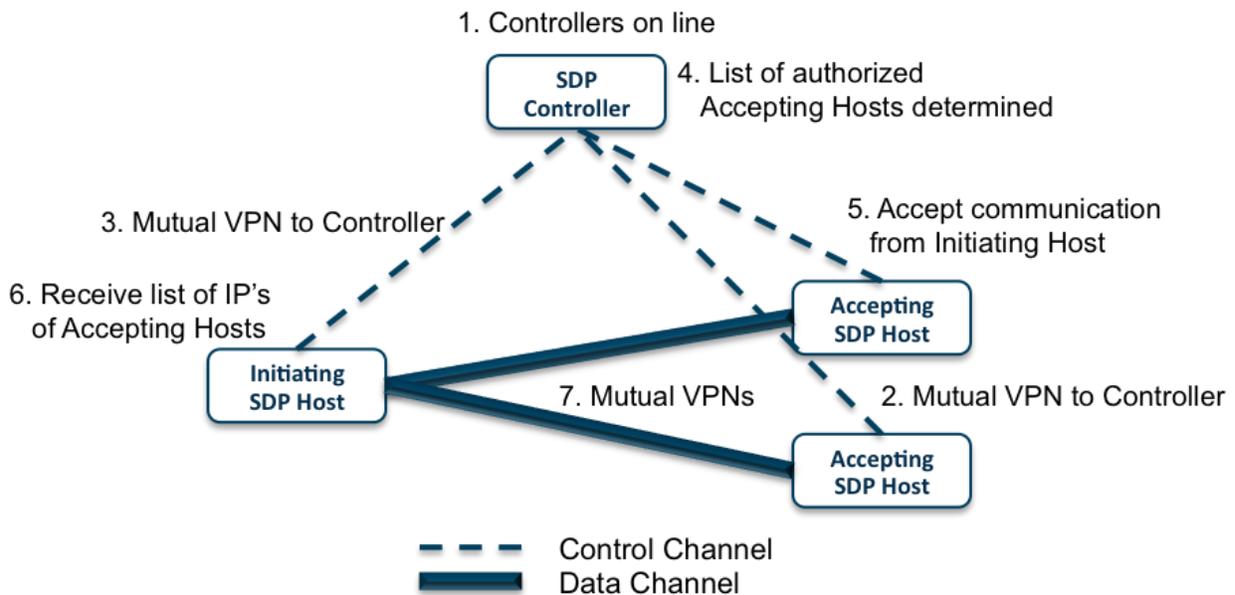


Figure 2: Workflow of the architecture of the Software Defined Perimeter

3.0 SDP Implementations

While the general workflow remains the same for all implementations, the application of SDPs can favor certain implementations over others.

3.1 Client-to-Gateway

In the client-to-gateway implementation, one or more servers are protected behind an Accepting SDP Host such that the Accepting SDP Host acts as a gateway between the clients and the protected servers. This implementation can be executed inside an enterprise network to mitigate common lateral movement attacks such as server scanning, OS and application vulnerability exploits, password cracking, man-in-the-middle, Pass-the-Hash (PtH), and many others. Alternatively, it can be implemented on the Internet to isolate protected servers from unauthorized users and mitigate attacks such as denial of service, SQL injection, OS and application vulnerability exploits, password cracking, man-in-the-middle, cross-site scripting (XSS), cross-site request forgery (CSRF), and many others.

3.2 Client-to-Server

The client-to-server implementation is similar in features and benefits to the client-to-gateway implementation discussed above. However, in this case, the server being protected will be running the Accepting SDP Host software instead of a gateway sitting in front of the server running that software. The choice between the client-to-gateway implementation and the client-to-server implementation is typically based on number of servers being protected, load balancing methodology, elasticity of servers, and other similar topological factors.

3.3 Server-to-Server

In the server-to-server implementation, servers offering a Representational State Transfer (REST) service, a Simple Object Access Protocol (SOAP) service, a remote procedure call (RPC), or any kind of application programming interface (API) over the Internet can be protected from all unauthorized hosts on the network. For example, in this case, the server initiating the REST call would be the Initiating SDP Host and the server offering the REST service would be the Accepting SDP Host. Implementing an SDP for this use case can significantly reduce the load on these services and mitigate a number of attacks similar to the ones mitigated above. This concept can be used for any server-to-server communication.

3.4 Client-to-Server-to-Client

The client-to-server-to-client implementation results in a peer-to-peer relationship between the two clients and can be used for applications such as IP telephone, chat, and video conferencing. In these cases, the SDP obfuscates the IP addresses of the connecting clients. As a minor variation, a user can also have a client-to-gateway-client configuration if the user wishes to hide the application server as well.

4.0 SDP Applications

The SDP can protect servers of all types from network-based attacks. Some of the more interesting applications of SDP are described below.

4.1 Enterprise Application Isolation

For data breaches that involve intellectual property, financial information, HR data, and other sets of data that are only available within the enterprise network, attackers may gain entrance to the internal network by compromising one of the computers in the network and then move laterally to get access to the high value information asset. In this case, an enterprise can deploy an SDP inside its data center in order to isolate high-value applications from other applications in the data center and isolate them from unauthorized users throughout the network. Unauthorized users will not be able to detect the protected application, and this will mitigate the lateral movement these attacks depend on.

4.2 Private Cloud and Hybrid Cloud

While useful to protect physical machines, the software overlay nature of the SDP allows it to be easily integrated into private clouds to leverage the flexibility and elasticity of such environments. Also, SDPs can be used by enterprises to hide and secure their public cloud instances in isolation, or as a unified system that includes private and public cloud instances and/or cross-cloud clusters.

4.3 Software as a Service

Software-as-a-Service (SaaS) vendors can use SDP architecture to protect their services. In this application, the software service would be an Accepting SDP Host, and all users desiring connectivity to the service would be the Initiating Hosts. This allows a SaaS to leverage the global reach of the Internet without its associated security concerns.

4.4 Infrastructure as a Service

Infrastructure-as-a-Service (IaaS) vendors can offer SDP-as-a-Service as a protected on-ramp to their customers. This allows their customers to take advantage of the agility and cost savings of IaaS while mitigating a wide range of potential attacks.

4.5 Platform as a Service

Platform-as-a-Service (PaaS) vendors can differentiate their offering by including the SDP architecture as part of their service. This gives end users an embedded security service that mitigates network-based attacks.

4.6 Cloud-Based VDI

Perhaps the ideal location for virtual desktop infrastructure (VDI) is in an elastic cloud in which use of the VDI is paid for by the hour. Unfortunately, if the VDI user needs to access servers inside the corporate network, VDI can be challenging to use and can open up security holes. However, VDI coupled with an SDP solves both of these problems through simpler user interaction and granular access.

4.7 Internet-of-Things

A vast amount of new devices are being connected to the Internet. Back-end applications that manage these devices and/or extract information from these devices are often mission-critical and act as a custodian for private or sensitive data. SDPs can be used to hide these servers and their interactions over the Internet to maximize both security and up-time.

5.0 SDP's Relationship to IKE/IPsec and TLS

As discussed in the previous sections, SDPs may use protocols such as IKE/IPsec and TLS to create VPNs between Initiating Hosts to Accepting Hosts. However, SDPs are not the same as VPNs. The differences between them are outlined below:

- Different amounts of effort are required to create SDP-protected servers compared to VPN gateway-protected servers. In the SDP case, once the SDP Controller is online, the user can create as many protected servers as desired via software and can differentiate authorized from unauthorized users via LDAP associations.
- There are greater capital and operating costs associated with setting up VPN gateways to protect individual servers than an SDP. SDPs are a software construct that can be deployed in a cloud environment.
- SDPs can be used for both security and remote access simultaneously, while VPN gateways cannot. If one were to try to use a VPN client and VPN gateway within an enterprise to protect a server, then users could not use a remote access VPN to access the server (because the VPN client already connected to the remote access VPN gateway). However, SDP communication can occur over a remote access VPN.
- SDP protects against bandwidth denial of service attacks, while VPN gateways do not. Accepting SDP Hosts can be deployed in a topologically different location than the application server it is protecting, thereby hiding the true location even from authorized users.

6.0 Working Group Purpose

The working group intends to serve as a focal point for the definition of the protocols by which the SDP components communicate with one another. Our goal is to foster interoperability among multiple vendors of those components.

7.0 Core Values

It is the intention of the Software Defined Perimeter Working Group to:

- Build upon existing standards, research and other related work
- Be inclusive by seeking input and communicating effectively with all stakeholders
- Provide robust guidance by utilizing sound scientific research

We believe in one of the founding beliefs of the Internet as embodied in an early quote from David Clark: "We reject kings, presidents and voting. We believe in rough consensus and running code".

8.0 Scope

The scope of the working group includes, but is not limited to, the following topics:

- Definition of the command channel communication protocol between the SDP Host and the SDP Controller
- Definition of the SDP host-to-host communication protocol(s)

The goal is to enable interoperability at a communications level between SDP components developed by different vendors, provide extensions that can be used to add value to products from different vendors, and future-proof the standard to accommodate standardized enhancements.

9.0 Deliverables

Deliverables will be governed by CSA's intellectual property rights policy and will be defined at the CSA Conference meeting on December 6, 2013 in Orlando.

10.0 Membership and Structure of the Working Group

The working group will be composed of CSA volunteers who meet at least one of the criteria listed below. Individuals who meet more than one of the criteria will be given preference when adding new working group members. Ideally, working group members:

- Are highly motivated and willing to contribute to a non-profit working group
- Have documented experience in the domain either through their current or previous jobs or through conducting high-quality academic research
- Can provide references as to their credentials

The Software Defined Perimeter Working Group will have two co-chairs, who will provide updates and seek guidance from the advisory roles. The working group will require typical project management, online workspace and technical writing assistance.

11.0 Advisory Roles

Advisory roles will be held by subject matter experts to support the SDP efforts. Subject matter experts will help provide:

- Guidance on research projects conducted by the working group
- Suggestions for new projects relevant to the industry
- Input on current research projects, including peer review of materials
- Access to properly sanitized data for research purposes
- Access to cloud provider contacts for research interviews

Advisory roles can be filled through individual subject matter experts, as well as through liaisons with other groups.

12.0 Duration

The working group will operate until its chartered deliverables are complete.