Software Defined Perimeter (SDP)[1] A Primer for CIOs

White Paper

By: Juanita Koilpillai, Waverley Labs LLC

December 2017

Abstract

The software defined perimeter (SDP) is a new approach to cyber security that mitigates network-based attacks. It protects both legacy IT assets and cloud services of all classification levels. It works by hiding critical IT assets within an undetectable, invisible, black cloud, whether the assets are on premise or in a public or private cloud, a DMZ, a server in a data center, or even inside an application server.

SDP uses a combination of tried and true security protocols that were previously unconnected until the Department of Defense (DoD) announced them working in concert. The Cloud Security Alliance adapted the generalized DoD workflow but modified SDPs for commercial use and made it compatible with existing enterprise security controls.1 Where applicable, SDP has followed NIST guidelines on cryptographic protocols and securing applications in the cloud. DHS has funded the development of an open source version of the SDP for both public and private organizations to defend against distributed denial of service (DDoS) attacks. Other uses of the SDP beyond DDoS protection will be covered in future white papers.

This white paper will inform Chief Information Officers (CIOs) of large organizations and agencies how the software defined perimeter (SDP) works, map the technical design and workflow, describe all its features, identify the protections gained, and introduce benchmarks and monitoring.

Introduction

The software defined perimeter (SDP) is a new approach to security using tried and true protocols that mitigates network-based attacks by creating dynamically provisioned perimeters anywhere in the world, including clouds, demilitarized zones (DMZs), and data centers.

The traditional networking model provides visibility and connectivity with a 'need to know' access model and then adds a number of point controls to prevent access from untrusted systems. SDP provides zero visibility and zero connectivity, only establishing connectivity after end points prove they can be trusted to allow legitimate traffic. This approach prevents essentially all network-based attacks.

The SDP architecture consists of five components of security protections: single packet authorization, mutual transport layer security, device validation, dynamic firewalls, and application binding. Together, these protocols make it very difficult for attackers to access, let alone modify protected applications and data.

An open source SDP (Open SDP) specifically for DDoS attacks has been completed with funding from DHS S&T[2] and can now be used by the community at large. Proprietary commercial versions have been implemented and proven at Coca Cola, Mazda and other large companies.[3]

The Problem

Traditional cyber security constructs and implementations have become ineffective and unscalable by any measure and the reason for this is the slow evolution of cyber security that has not caught up with the sophistication of hackers. Various implementations to bring applications and systems 'online' enable machine to machine connections that have forced the requirement that these machines need to be secured. This practice has spawned a whole industry to find, fix and manage vulnerabilities and malware, update patches and monitor for rogue activities. Vulnerability and malware management or systems have proved expensive and hard to manage.

Access to these machines and the services on them are typically allowed prior to authentication and authorization. This construct has given rise to access control and other gateway products and mechanisms to restrict and control access. These gateways and firewalls are static making management of these products difficult. Access control and management of firewalls in larger organizations is complicated while non-existent in smaller organizations. Integrating identity and key management is also complicated and often configured insecurely.

Because authentication is enforced after connection to systems, organizations are forced to understand who the legitimate users are, what packets are good and resources are spent on threat intelligence. Experience has shown that even understanding the threat and impending breaches has not galvanized organizations to prioritize and allocated the appropriate resources to protect themselves effectively.

Mandates and regulations have complicated matters as the already scarce resources are focussed on compliance. Moreover, regardless of the various efforts to secure systems and infrastructures, the hacker community has outpaced these efforts as described by the highly publicized and expensive cyber attacks that are ever increasing in volume and velocity.

The idea for the Software Defined Perimeter came about in response to the urgency to upend the current state of cybersecurity. The following key components are incorporated into SDP implementations.

The Concept

Traditional cyber security constructs and implementations have become ineffective and not scalable by any measure and the reason for this is the slow evolution of cyber security that has not caught up with the sophistication of hackers. Various implementations to bring applications and systems ‘online’ enable machine to machine connections that have forced the requirement that these machines need to be secured. This practice has spawned a whole industry to find, fix and manage vulnerabilities and malware, update patches and monitor for rogue activities. Vulnerability and malware management or systems have proved expensive and hard to manage.

Access to these machines and the services on them are typically allowed prior to authentication and authorization. This construct has given rise to access control and other gateway products and mechanisms to restrict and control access. These gateways and firewalls are static making management of these products difficult. Access control and management of firewalls in larger organizations is complicated while non-existent in smaller organizations. Integrating identity and key management is also complicated and often configured insecurely.

Because authentication is enforced after connection to systems, organizations are forced to understand who the legitimate users are, what packets are good and resources are spent on threat intelligence. Experience has shown that even understanding the threat and impending breaches has not galvanized organizations to prioritize and allocated the appropriate resources to protect themselves effectively.

Mandates and regulations have complicated matters as the already scarce resources are focused on compliance. Moreover, regardless of the various efforts to secure systems and infrastructures, the hacker community has outpaced these efforts as described by the highly publicized and expensive cyber attacks that are ever increasing in volume and velocity.

The idea for the Software Defined Perimeter came about in response to the urgency to upend the current state of cyber security. The principles behind SDPs are not entirely new. Software Defined Perimeter (SDP) offers a radically new approach to protecting networked applications that is more affordable and effective as less man power is required for support. The Software Defined Perimeter, if implemented as specified, deems applications both in the cloud and on premise impenetrable.

The Software Defined Perimeter architecture and associated components is evolutionary in that it builds upon known controls such as the ‘need to know’ access model verified in the DoD, device verification proven by NSA and Mutual Transport Layer Security promoted by NIST. 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. SDPs maintain the benefits of the need-to-know model but eliminate the disadvantages of requiring a remote access gateway appliance.

The Software Defined Perimeter is also revolutionary in that it extends the protection to the perimeter that is changing with the advent of mobile devices and the Internet of Things (IoT). 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.

Technical Design

Based upon our experience with implementing the open source SDP for the distributed denial of service use case and being involved with the SDP specification from its inception, the key takeaways are that:

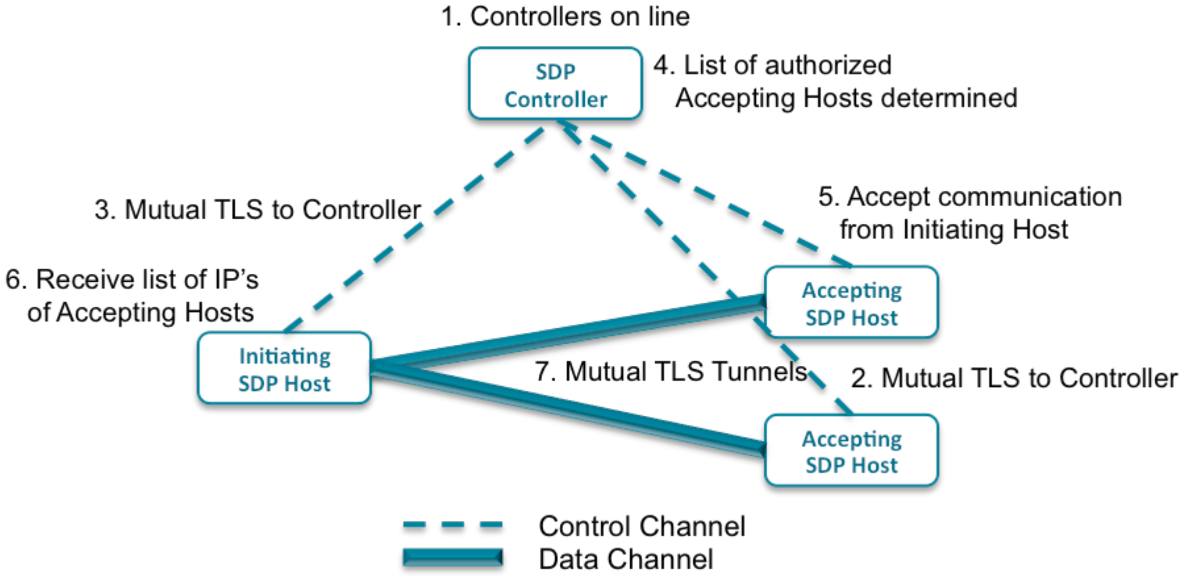
- SDP enables risk reduction by reducing the attack surface and therefore the exposure to cyber attacks
- SDP protects critical assets/infrastructures by separating the access control and data planes to render them ‘black’ and blocking potential network-based attacks
- SDP provides an integrated security architecture that is hard to achieve today with the various security point products. It integrates:
    - Applications that are user-aware
    - Devices that are client-aware
    - Firewalls/Gateways that are network-aware
- SDP provides a connection-based architecture (as opposed to an IP-based one) that is designed for the scalability necessary for today’s explosion of IPs and loss of the perimeter with the use of cloud environments
- SDP allows you to ‘control’ all connections as it is aware of who connected, from what device, to what service/infrastructure and other parameters

SDP is built on proven, standards-based components such as mutual TLS, SAML and X.509 Certificates. Standards based technology ensures that SDP can be integrated with other security systems such as data encryption or remote attestation systems. The success and growing adoption of SDP has created the need to capture and codify the knowledge gained from the past few years of experience.

In its simplest form, the architecture (Figure 1) of the SDP consists of two components: SDP Hosts and SDP Controllers. SDP Hosts can either initiate connections (IH) or accept (AH) connections. These actions are managed by interactions with the SDP Controllers via a secure control channel (see Figure 2). 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

SDP Work Flows


1. Controllers on line
SDP Controller
4. List of authorized Accepting Hosts determined
3. Mutual TLS to Controller
5. Accept communication from Initiating Host
6. Receive list of IP's of Accepting Hosts
Accepting SDP Host
Initiating SDP Host
7. Mutual TLS Tunnels
2. Mutual TLS to Controller
Accepting SDP Host
Control Channel
Data Channel

The SDP workflow has the following steps:

1. One or more SDP Controllers are brought online and connected to the appropriate optional authentication and authorization services (e.g., PKI Issuing Certificate Authority service, device attestation, geo-location, SAML, OpenID, OAuth, LDAP, Kerberos, multifactor authentication, and other such services).
2. One or more AH's are brought online. These hosts connect to and authenticate the Controllers. However, they do not acknowledge communication from any other host and will not respond to any non-provisioned request.
3. Each IH that is brought on line connects with, and authenticates to, the SDP Controllers.
4. After authenticating the IH, the SDP Controllers determine a list of AH's to which the Initiating Host is authorized to communicate.
5. The SDP Controller instructs the AH's to accept communication from the IH as well as any optional policies required for encrypted communications.
6. The SDP Controller gives the IH the list of AH's as well as any optional policies required for encrypted communications.
7. The IH initiates an SPA to each authorized AH. It then creates a mutual TLS connection to those AH's.

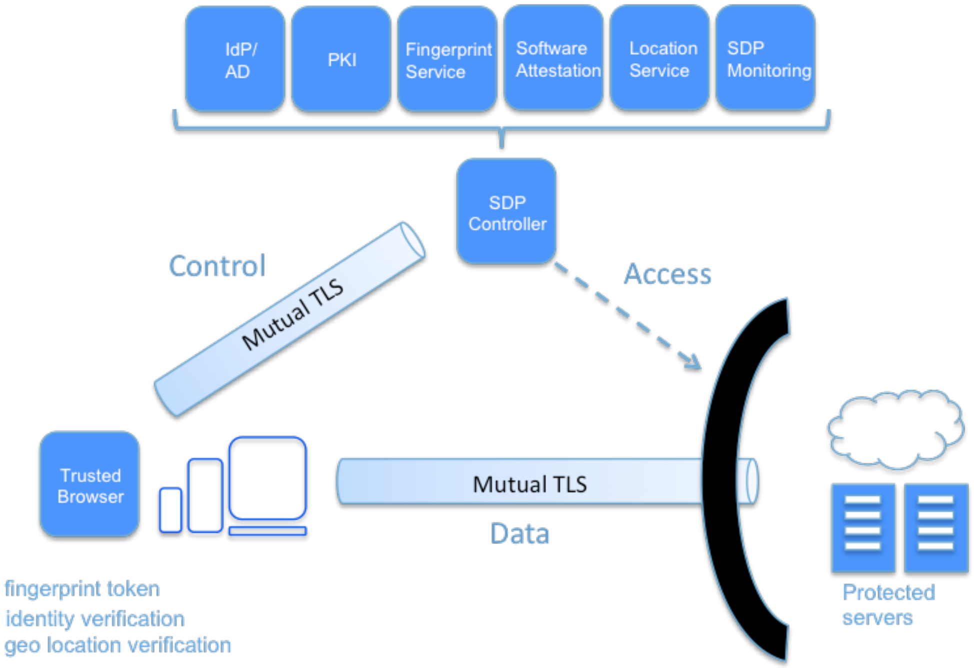
SDP Features

The main features of the SDP are the ability to authenticate devices, authenticate and authorize users and to dynamically provision services. Device authentication is provided by protocols such as Single Packet Authentication (SPA), Host-specific Firewalls, Diffie Hellman Mutual Transport Layer Security, Device Fingerprinting, Software Verification and Geo Location. User Authentication and Authorization is provided by a Trusted Brower, SAML, authentication to a Identity Provider (IdP) and Membership in Groups and Dynamic Provisioning of applications is provided by Firewalls, Gateway Addresses, Diffie Hellman Mutual Transport Layer Security and Whitelisted Applications.

Figure 2 lists the full-blown feature set of the SDP. The features include multiple layers that protect applications from DDoS, Man in the Middle, and the OWASP Top 10 attacks observed over the past decade.

Figure 2

Security Features Provided by the SDP



SDP Protections

The SDP architecture consists of five components: single packet authorization, mutual transport layer security, device validation, dynamic firewalls, and application binding. Together, these protocols make it very difficult for attackers to access protected applications.

1. Hidden Services & Infrastructure

Hackers cannot attack what they can't see. One of the primary objectives of the software defined perimeter is to make the application infrastructure effectively "black," or undetectable, showing no domain name system (DNS) information or IP addresses. SDP protected assets are considered "dark" as it is impossible to port scan for their presence. Some SDP vendors completely lock down the DMZ-LAN firewall closing any incoming ports to the application even after authorization and authentication. Other vendors open the firewall after authorization and authentication, although the SDP specification requires completely preventing any inbound traffic even for authorized clients. Benefits of this feature are:

- **Servers are "Blackened":** The server will not respond to any connections until the clients have been authenticated and authorized with the use of protocols such as single packet authorization (SPA),
- **Mitigates Denial of Service attacks:** Internet-facing servers running the https protocol are highly susceptible to Denial-of-Service (DoS) attacks. The use of protocols like SPA or similar mitigate these attacks because it allows the server to discard all bad packets and only accept good packets.
- **Attack detection:** The first packet to an AH from any other host is a SPA or similar construct. If an AH receives any other packet, it is viewed as an attack. Therefore, SDP determines an attack based on a single malicious packet which is a highly effective way to detect network-based attacks.

For the SPA protocol, a single SPA packet is sent from the client to the server. The server does not reply until the SDP Controller verifies authentication and authorization. The format of the packet is:

IP	TCP	AID (32-bit)	Password (32-bit)	Counter (64-bit)

After receiving the packet, the server must enable the client to connect via mutual TLS on port 443.

2. Mutual Two-way Encrypted Communications

The software-defined perimeter requires mutual, two-way cryptographic authentications. Transport layer security (TLS), also known as secure sockets layer (SSL), was designed to provide device authentication prior to enabling confidential communication over the Internet. The standard was originally designed to provide mutual device authentication. However, in practice, TLS is typically only used to authenticate servers to clients, not clients to servers. The software-defined perimeter uses the full TLS standard to provide mutual, two-way cryptographic authentications. The following benefits are afforded with this approach.

- **Device Authentication:** The connections between all hosts must use mutual authentication to validate the device as an authorized member of the SDP prior to further device validation and/or user authentication. All weak cipher suites and all suites that do not support mutual authentication is discouraged.
- **Disallows forged certificates:** Mutual authentication schemes pin certificated to a known valid root and will not consist of the hundreds of root certificates trusted by most consumer browsers. This mitigates impersonation attacks whereby an attacker can forge a certificate from a compromised certificate authority. The methodology by which the root certificate is on boarded to the IH, AH, and Controller is outside the initial protocol specification. Typical methods would be via Chef or Puppet or their hosting service equivalents (e.g., RightScale, AWS CloudFormation, etc.).
- **Disallows Man-in-the-middle attacks:** These attacks use forged or obsolete online certificate status protocol (OCSP) response stapling as defined by the IETF working draft “X.509v3 Extension: OCSP stapling Required draft-hallambaker-muststaple-00”, which references the stapling implementation in RFC 4366 “Transport Layer Security (TLS) Extensions”. The mutual handshake protects from man-in-the-middle attacks that exploit OSCP responses before the server certificate is revoked.

3. “Need to Know” Access Model

SDP grants access to specific applications and systems based on granular role based access control rules, enforcing “Need to Know” and “Least Privilege” access. Both pre-authentication and pre-authorization of users and devices is verified before connectivity is granted. Users are provisioned access only to applications and infrastructures servers that are appropriate for their role. Pre-authentication can include using identity systems that utilizes a SAML assertion to inform the SDP Controller of the hosts’ privileges or similar mechanisms. Device identity is determined via a multi-factor token that is embedded in the connection information. Users are associated with with devices that are validated based upon organizational policy. Only connections to the specifically requested service is enabled and no other connection is allowed anywhere else; the connection is removed upon transaction completion.

- **Makes forensics easier:** All bad packets can be analyzed and tracked for forensics activities. Established connections can be recorded with information about who made the connection, from what device and to what service; information that is very hard to gather today during forensics activities.
- **Enforces device validation:** Mutual two-way encrypted communications proves that the device requesting access to the software-defined perimeter possesses a private key that has not expired and that has not been revoked, but it does not prove that the key has not been stolen. Device validation proves that the key is held by the proper device. In addition, device validation attests to the fact that the device is running trusted software and is being used by the appropriate user. Device validation has been

successfully implemented in organizations such as Google among others and but will be addressed in future versions of the SDP specification in detail.

- **Mitigates credential theft:** The objective of device validation is to prove that the proper device holds the private key and that the software running on the device can be trusted. In an SDP, the Controller is assumed to be a trusted device (because it exists in the most controlled environment) and the IHs and AHs must validate to it. Device validation mitigates credential theft and the resultant impersonation attacks.
- **Protects from compromised devices:** Because users are only granted access to authorized applications and to no other ones, the threat of lateral movement from compromised devices is eliminated.

4. Dynamic Firewalls

Most people are familiar with traditional firewalls that use static configurations to limit incoming and outgoing traffic based on the address information in the IP packet (that is, based on the quintuplet of protocol, source IP address and port, and destination IP address and port). Most enterprise firewalls have ten, hundreds, or even thousands of firewall rules. Unlike traditional firewalls, dynamic firewalls have only one firewall rule: deny all. Communication with each device is individually enabled by dynamically inserting “Permit <IP quintuplet>” into the firewall policy. In the software defined perimeter architecture, gateways incorporate this dynamic firewall security control.

- **Allows membership-based enclaves dynamically:** With the physical perimeter disappearing because of the influx of mobile and IoT devices, firewalls play a more important role to dynamically extend the perimeter to allow connections from these devices. More specifically, the software-defined perimeter dynamically binds users to devices, and then dynamically enables those users to access protected resources by dynamically creating and removing firewall rules in the SDP gateways.

5. Application Layer Binding

Users are only granted access to specific applications at an application layer, and not to a broad network segment or set of ports. Additionally SDP solutions may whitelist applications on the user’s device. SDP can be used to protect application as well as system-level resources. After authenticating and authorizing both the device and the user, the software-defined perimeter creates two-way encrypted communications to the protected applications.

- **Provides encrypted application communication:** Application binding constraints authorized applications so they can only communicate through those encrypted tunnels, and, simultaneously, blocks all other applications from using those tunnels.

SDP Benchmarks & Monitoring

The SDP benchmark and monitoring builds on the following measures.

Measure of Weakness (or classes of vulnerabilities)

Vulnerability is an artifact of a weak design and implementation and information about vulnerabilities can lead to the development of exploits using a variety of attack vectors. Armed with the knowledge about critical assets and how they behave in the presence of exploits or threat actors we can measure weaknesses based upon failure scenario analysis. In addition, we can recognize potential attack patterns based upon data from the output of the SDP.

Measure of Impact

The benchmark will be based upon automated analysis of the technical impacts of the various SDP controls to the application it supports and potential failures against attacks (DDoS, OWASP top ten etc.). DDoS attacks will be measured by understanding the various vectors such as a) rapidly opening multiple connections and keeping them open to exhaust resources or b) rapidly sending "irrelevant data" to open connections to tie up resources and prevent/slow down servicing of legitimate traffic.

SDP DDoS Use Case

Protecting cloud and critical infrastructure applications against DDoS attacks is one of the most difficult challenges facing organizations today. The current approach to protect Internet accessible or cloud based applications is to throttle traffic a the network layers therefore blocking legitimate traffic. SDP will allow legitimate traffic while disallowing all other traffic.

Software Defined Perimeter (SDP) offers a radically new approach to protecting networked applications. SDP is based on a strong security model that only allows TCP connections from pre-authorized devices. Moreover SDP issues user level access at the port/protocol level (after user authentication) to ensure connections cannot be re-tasked or hijacked. SDP provides Federal agencies and organizations a new approach to securing cloud and critical infrastructure applications by allowing them to customize a SDP implementation to their unique requirements - this includes all aspects of security from certificates, cyphers, identity systems, monitoring, management etc. Organizations can then wrap applications within their SDP implementation to ensure both security as well as secure workflow. As the SDP gateway component can be deployed on the same physical server as the application, unauthorized access as well as the exfiltration of data is completely blocked.

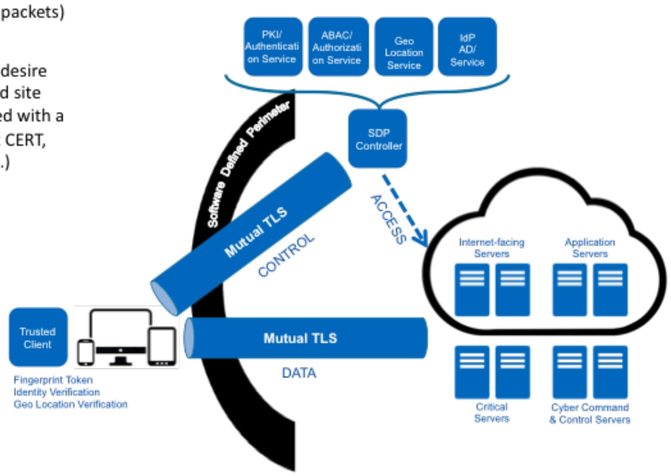
For pubic-facing websites the SDP ensures that all Internet-facing web servers are blocked by the SDP and the firewall protecting these web servers denies all traffic. Internet users who desire access to a protected site will be routed to an on-boarding site (for eg. one that accepts only approved IP address spaces). Users can on-board themselves if they are coming from a valid IP address and have a phone for one-time authentication as an option or use PKI verification for higher levels of authentication. The mobile phone or device used for access can be verified as belonging to the user. The user's geo location can also be logged by the SDP and can be used as a multi-factor authentication attribute.

Use Case

Anti-DDos SDP

1 All Internet facing servers are hidden by SDP gateway. (ie. default drop all packets)

2 Internet users who desire access to a protected site would be on-boarded with a unique ID (eg. client CERT, encryption keys, etc.)



3 When users wish to access a protected site they would click on the SDP client on their personal device

4 Info in the unique SPA packet must match id of user. This is the key that opens the gateway to the client (ie. port on firewall)

5 If the device and user identity are valid the users will be granted access. (IP address can be verified to match the stored location for dedicated clients)

When users wish to access a protected site, users will use an SDP client on their personal device. If the device and user identity are valid AND the IP address matches the stored location the users will be granted access. All these checks are done transparent to the user to ensure security.

With these mechanisms in place, the following list of recent attacks can be stopped.

- **False credential – IRS (stealing tax refunds)**
  If a hacker tried to impersonate a tax filer their mobile phone id would not match the filers name – thus no access would be granted for theft using false credentials.

- **Stolen credential – OPM (stealing employee files)**
  If an attacker stole a credential it would not work, as the device id would be different. Hackers could try to re onboard themselves but their mobile phone id would be wrong – thus no access will be provided with stolen credentials.

- **APT – Titan Rain (device breach)**
  SDP does not stop APT data theft from devices. However SDP can be used to ensure that encrypted data is only accessible on the users device if the key management system was only accessible via a SDP.

- **Denial of Service**
  SDP will allow legitimate traffic by dropping all other unauthorized packets thereby protecting applications from DDoS attacks.

- **Remote Surveillance**
  SDP would make it impossible for foreign spies and hackers to conduct remote surveillance on critical infrastructure. Even if foreign governments could do a APT attack on a single user, their visibility would be limited to only what the user could see.

The combination of the following concepts rolled into one package is what makes the SDP unique.

1. Device Authentication – Issuance and the check is made with the Central Authority
2. User Authorization – User signs in and SDP says what the user can or cannot do
3. Dynamic Connection – SAML federation or Active Directory is used to check IDs

This multi-factor authorization scheme is truly integrated into the network while most other schemes do not.

The hard work is to issue certificates for devices for the secure connections and to maintain an identity system that has to have all the roles defined and kept up to date. These two activities are par for the course and over time will become second nature to maintaining secure and resilient applications or infrastructures.

About the Author

Juanita Koilpillai is Founder and CEO of Waverley Labs a cyber risk engineering company. She has spent 30 years developing systems in computer security, network management and distributed software. She is currently pioneering the field of digital risk management and is the technical advisor to the Digital Risk Management Institute. As part of that effort, she is leading the open source software-defined perimeter (SDP) effort for "black" apps in the cloud with the Cloud Security Alliance. The SDP is a prescriptive five layer security model that stops all network-based cyber attacks.

She is an active contributor to NIST and led the creation of a security risk index system for moving apps to the cloud (NIST 500-299). She was a key member of FEMA's Enterprise Security Management Team and has served as Principle Investigator for several Department of Defense initiatives. She co-founded CyberWolf, one of the most advanced automated attack sensing and warning systems that was deployed by government and later acquired by Symantec.

About Waverley Labs

Waverley Labs is a leading a leading provider of digital risk management software and services that helps large organizations reduce their exposure to digital risk. Its products and services range from the industry's first open source software defined perimeters (SDPs) for large federal agencies, to the assessment, quantification, and mitigation of digital risk from the business perspective. Waverley Labs' automated analysis and visualization capabilities provide business leaders, risk officers and CISOs with an at-a-glance view of business risks prioritized according to business impact and recommended risk mitigations. Waverley Labs works closely with NIST and the Cloud Security Alliance to provide thought leadership in digital risk management. For more information visit http://www.waverleylabs.com.

References

1. https://cloudsecurityalliance.org/download/sdp-specification-v1-0/

2. https://gcn.com/articles/2015/09/29/waverly-ddos-single-packet-authorization.aspx

3. http://blogs.wsj.com/cio/2015/03/23/coca-cola-looks-to-secure-network-edge-for-age-of-cloud-mobility/