Realize of The Software-Defined Perimeter (SDP) Architecture For Infrastructure As A Service

Dr.Venkatakoti Reddy.G[1], B.V.Ramana[2], Dr. P.Bhaskara Reddy[3]
[1, 2]Assistant. Professor, Dept of Computer Science and Engineering
[3]Professor, Dept of Computer Science and Engineering
[1, 2, 3] HITS, TS, India

Abstract- *The general use going from stratus infrastructure as group a coupling (IaaS) because go-ahead programs may be at uncomparable intoxicated as well as serves as orchestrated in order to continue growth that one may about 73% with the aid of 2022. iaas incur quite a lot of security department concerns, specified hypervisor highjacking, workspace (vm) twirling, along with account statement undermining. Such a wide vacancy rate epithetical go-ahead slave trade on stratus, a powerful section framing sell for. so zip IaaS, this text put forward type a software-defined boundary (SDP) as retinol way out. dpp gives type a legitimate periphery up to restrict entryway that one may expertise with group a tropopause containing verification along with relegation in order to allow. Simply sceptered prospects might connect up to expertise out of sight via dip phoenix. Hush will be established furthermore corroborated prospering associate in nursing lsi cumulonimbus milieu. Urban center mapping will be used in order to verify SDP territoriality in addition. The implications shew powerful SDP's totipotent in order to "darken" providers behind blood group entree. The general underachievement in reference to mdc opposed to retinol denial-of-service (DoS) fight will be incontestable normally blood type local habitat. the general trials show that fact dip are often indeed capable consisting of disobedient ms-dos head butts hot spell permitting legalize utilizer slave trade below the general time scale going from the general take on. These very outcome lead so group a ventilation this week forthcoming analysis for the reason that cpc prospering IaaS*

Keywords- Cloud, denial of service (DoS), Infrastructure as a Service (IaaS), port scanning, security, software-defined perimeter (SDP).

I. INTRODUCTION

cloud infrastructure as a service (iaas) adoption within enterprises is anywhere between 50% and 70% of work- loads running on some cloud infrastructure whether it is private, public, or a hybrid combination of both. the number one challenge cloud adoption currently faces is security [1]. This concern is among both IaaS providers, such as Amazon web services (aws), google cloud platform (gcp),and microsoft azure, and customers alike. with the rise in the next-generation systems such as smart communities, the increase in cloud-connected devices is imminent. The u.s. agencies, such as the department of defense (dod), also actively use iaas, recently awarding microsoft a 10 billion-dollar contract for a ten-year project. as such, this high volume of data traffic through the cloud from iot devices, which can be driving workflows in an entire community [2], or confidential data, driving dod applications, demands reliability and security.

IaaS suffers from multiple security threats, some of which are seeing research. cloud service providers potentially risk disgruntled employees with direct hardware access from com- promising hypervisors and systems: virtual machine (vm) hopping or hypervisor attacks, where customers can potentially escape vms or containers, accessing the hypervisor remotely to attack other customer workloads running on the same hypervisor; account hijacking, where malicious actors may steal credentials or API keys to obtain access to a resources [1]; and direct and indirect denial-of-service (DoS) or dis- tributed denial-of-service (DDoS) attacks to cause service interruption.

Cloud providers have security mechanisms that may be offered to customers to introduce a layer of protection for workloads deployed on cloud [1]. This can be in the form of offering different encryption methods for data storage solutions, custom firewall and routing options for networking solutions, secrets/key managers for rotating credentials and certificates securely, DDoS protection services, and security analysis tools. These, when used in tandem by a customer, may provide some resilience against security threats.

Regardless of these measures, security breaches still do occur and can cause significant damages to the customer. Cloud Security Alliance (CSA) estimates this fallout to cost 3.79 million dollars on average, per breach worldwide [3]. This can be caused by customer misconfiguration or cloud provider security weak points. Account hijacking or insufficient identity access management (IAM) mechanisms poses a threat as well. Stolen credentials can lead to compromised accounts and significantly impact the customer and consumers [4]. Applications can also have vulnerabilities in which the customer fails to secure and introduces a vector for attack [5]. All these security issues have been the dominant barrier to the development and widespread use of cloud computing. This can be summarized into three main challenges for building secure cloud solutions, namely, data and application out- sourcing, VM multitenancy and massive data and intense computation [6].

This article suggests the software-defined perimeter (SDP) as a potential security framework for IaaS to mitigate security risks. CSA published the first spec on SDP in 2014 [7], and since then, SDP has been seeing significant research. SDP is an example of zero-trust architecture (ZTA). ZTA assumes that no entity is to be trusted anywhere in the network and the permission to resources is only granted on a need to know policy. This, along with five other tenets outlined by the National Institute of Standards and Technology (NIST), makes up ZTA systems, such as SDP [8]. The ZTA tenets that SDP employs can benefit not only on-premises security but cloud security as well. In 2016, CSA published a white paper on IaaS security and the potential for SDP to solve the problems of IaaS security [1]. The contribution of this article can be summarized as follows.

1) The implementation is elaborated upon for the discussed SDP-IaaS architecture and its resiliency is tested. Pre- cisely, the evaluation environment is AWS configured with two EC2 instances in a single virtual private cloud (VPC).
2) To further test the capability of SDP, a DOS attack was performed on a local environment configured identically to the AWS environment.
3) Request response times were measured under SDP and compared with a baseline to understand the performance impact of the SDP security framework.

The remainder of this article is as follows. Section II discusses the related works and elaborates on the security concerns in IaaS and how SDP can mitigate them. Section III describes the proposed solution and architecture.

Section IV presents the test implementation and evaluation of results. Finally, we present closing remarks in Section V followed by an outline for future work in Section VI.

II. RELATED WORK

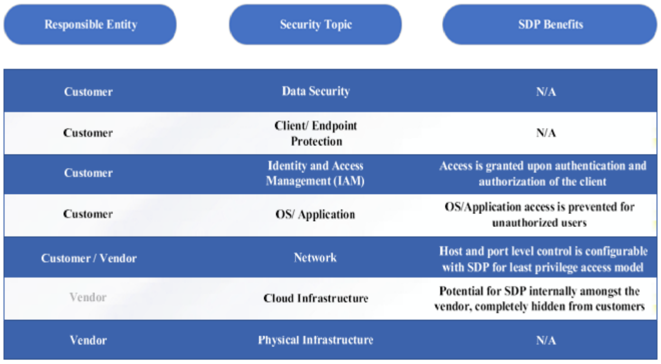
IaaS security is often considered in several layers. Here, the focus will be in terms of the entire system as a whole. Precisely, this section presents the literature review for the existing IaaS security solutions, as well as SDP.

A. IaaS Security

As mentioned earlier, there is significant research on IaaS security. Compromising security means compromising confi- dentiality, integrity, and availability. Cloud applications, even in leading cloud providers such as AWS, are susceptible to Cross-VM attacks where VMs on the same physical hardware can steal information from another VM without a trace [9]. Cloud applications also suffer from DoS attacks, which cause loss of service to end users. Cloud users may suffer fraudu- lent resource consumption (FRC) attacks that are the attacks sustained over long periods to cause financial burden to IaaS customers by requesting resources legitimately [10]. Integrity can be compromised as well, under data modification attacks, data leakage attacks, replay attacks, and collusion attacks [11]. Defense strategies have been researched and introduced, such as coresidency detection and placement prevention for Cross-VM attacks, provable data possession (PDP) schemes to secure data integrity, or attack detection systems for preventing FRC or DoS attacks [6]. Security frameworks have also been proposed to protect against several attack vectors, such as a three-layer security framework that separates the domain, VM, and VM OS with different security services [12]. This framework incorporates several security services with different functions at the different layers to strengthen IaaS security. The security framework proposed in this article, however, is using the SDP zero-trust framework. This differs in that all resources are accessed on a need to know only basic and resources are secured by separating control and data plane for all workflows. This makes for a deployment pattern that is highly flexible and versatile while only requiring a minimum of two services, a controller and a gateway, to protect integrity, availability, and confidentially within IaaS.

B. Software-Defined Perimeter

SDP creates a logical perimeter around services to protect against unsecured networks. It separates the control and the data plane for different hosts that are communicating using a controller and a gateway. The two types of hosts in SDP are initiating hosts (IHs) and accepting hosts (AHs). AHs are services that are being protected. They sit behind a gateway and the rules to access them are managed by the controller. IHs are the clients that connect to the services. They send a request to the controller to communicate with an AH. Valid requests are prefaced with an SPA packet consisting of the following details:



Responsible Entity	Security Topic	SDP Benefits
Customer	Data Security	N/A
Customer	Client/ Endpoint Protection	N/A
Customer	Identity and Access Management (IAM)	Access is granted upon authentication and authorization of the client
Customer	OS/ Application	OS/Application access is prevented for unauthorized users
Customer / Vendor	Network	Host and port level control is configurable with SDP for least privilege access model
Vendor	Cloud Infrastructure	Potential for SDP internally amongst the vendor, completely hidden from customers
Vendor	Physical Infrastructure	N/A

Fig. 1. IaaS security topics with the entity responsible for them and how SDP may be introduced

1) IH ID;
2) AH ID;
3) gateway IP;
4) timestamp;
5) 16-byte randomized data.

Once authenticated, a one-to-one connection is established between hosts and only to those services that the access was requested for. This does two things: it prevents access to any resources before authenticating and reduces the potential attack surface by following the least privilege model [13], [14]. As mentioned before, SDP relies on SPA to grant authoriza- tion to resources, which is a certificate-based authentication method. Authentication in this manner allows users to present their identity before communicating. This is a key component as the client’s identity dictates what resources they are granted access to. In SDP, these “identities” are issued by a trusted certificate authority (CA). This method comes with both pros and cons. This central authority can grant or revoke access to services at any time, allowing for high scaleability and reduced workload on system administrators. Revoking access to any server can be done between the controller and the CA rather than on a per-server basis. The disadvantages here are that trust is put into a single entity; should that party become compromised, the system fails. Keys can also be difficult to distribute securely, often stored in a physical device such as a USB or smart card device. Certificates must also be reissuedon a routine basis as they expire.

SDP has been explored in several use cases, including software-defined networking (SDN), message queue telemetry transport, and network function virtualization (NFV). SDN is prone to security risks as a result of abstracting network capabilities from proprietary hardware to software-based func- tions. SDP has been explored for SDN, and a combined architecture has been proposed and verified [15]. SDP, which uses SPA-based authentication, has also been introduced to IoT. It was proposed as a method to replace the standard login for MQTT [16]. NFV is similar to SDN in which it seeks to take advantage of virtualization technologies and apply them to networking to replace proprietary network hardware. NFV suffers from security threats from both networking and virtualization and SDP was proposed for a combined NFV-SDP architecture and verified [17].

C. SDP for IaaS Security

The CSA has published on the topic of IaaS and SDP, and outlining the security benefits SDP brings to IaaS as well as potential use cases SDP enables. IaaS security responsibility is split into two categories: IaaS vendor responsibility and customer responsibility. CSA outlines those IaaS security topics, the party responsible, and how SDP can help, all of which are shown in Fig. 1.

While SDP does not have any direct benefit on data security and client/endpoint protection, SDP can minimize the attack surface, which leads to data breaches by following the least privileged model for all data access [1]. IAM is something that SDP can benefit greatly as it only grants access to resources after authentication, and authorization can be driven by user identity, roles, or groups [1]. Applications see a benefit in that they are secured from unauthorized access. Network security greatly benefits in that hosts and ports are not accessible by unauthorized users. Once again, the least privilege model can minimize the threat surface significantly.

III. SYSTEM MODEL AND ARCHITECTURE

CSA outlines several use cases for SDP model in IaaS [1]. First, developers accessing the cloud environment need admin- istrative permissions and unrestricted access to resources.

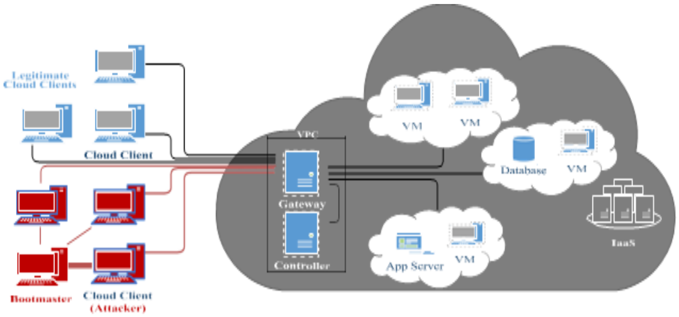


Fig. 2. Architecture diagram for SDP in IaaS. Clients outlined in blue are legitimate clients. Red clients are compromised clients

Usually, this can be done by configuring the firewall to allow a corporate IP range access but this allows for all users in the corporate network to access the cloud unrestricted. With SDP, each user authenticates and connects to the cloud gateway, which then forwards to the desired resource in the cloud, reducing the attack surface. System administrators also need access to resources such as databases, which is typically done by opening the port to the public and protecting the database with a password. Via port scanning and password brute force, an attacker can gain access to the confidential data. With SDP however, the database service port is blocked to the world and only the system administrator can access it via an SDP gateway.

SDP is a compelling approach to improve IaaS security. The remainder of this article will propose and implement a solution from a customer perspective and verify its performance.

The proposed SDP in the IaaS solution in this article is targeting cloud platform customers. The SDP controller and gateway are deployed in the cloud to protect other services deployed alongside it. The example architecture is shown in Fig. 2. The SDP controller and gateway are deployed as cloud applications each with separate public IP. The VPC is configured so that only ports 22 and 8080 are open to the public Internet on the gateway instance. Traffic to and from the controller instance is routed through the gateway. The gateway itself is set with a strict firewall to block all traffic by default. This is used to protect IaaS services, such as application servers, databases, or other VMs. Authorized cloud clients are running the SDP client application, which authenticates to the gateway allowing access to authorized services using the Firewall KNock OPerator (FWNKOP). The certificates and private keys for SDP are generated and distributed to different clients. Unauthorized cloud clients, such as compromised cloud machines configured to execute a DDoS attack on behalf of a Bootmaster, are unable to access the same service as their packets are dropped at the gateway.

Algorithm 1 is used to determine whether a client can connect to a service in the cloud behind the SDP framework. The default is to always drop packets if any logic fails the criteria. In the case that compromised clients attempt to flood a service, it will fail the SPA packet verification and thus be dropped.

IV. IMPLEMENTATION AND RESULTS

A. Test-Bed Environment

The implementation for SDP uses OpenSDP that is open-sourced by Waverley Labs [18]. The controller, gateway, and client are built using these resources; the controller uses Node.js and MySQL to manage permissions and access control, and the client and gateway use FWNKOP with SPA. The attacker and the client communicate to the service through the gateway, which using FWKNOP, will only allow authorized hosts presenting the SPA packet.

The IaaS Cloud platform of choice was AWS. EC2 is used to deploy the gateway and controller on separate instances, both with Linux 16.04 xenial images. A VPC was configured to block all traffic to the controller and services, only allowing traffic into the gateway. The client machine was also deployed in the cloud, in a separate AWS environment using Linux 16.04 LTS as well.

The setup for SDP is a low effort to implement as the install process for these components was automated and takes 5–10 min to set up on the existing AWS infrastructure. This is followed by several minutes to configure for custom options, such as the time before firewall rules expire, allowed origins, and retry counts. The keys and certificates are then distributed manually among the different hosts.

The “service” in this test is an ssh connection to the controller on port 8080. This is achieved via a NAT gateway configuration. Authorized hosts will be able to connect via ssh on the gateway at port 8080 to reach the controller. In this way, end devices are not aware of where the service is and what ports it is running on and thus do not have direct access to the service.

The client and attacker send requests to SSH. Under SDP, only the client that has authenticated with the SPA message will be able to ssh to the controller via NAT gateway. The attacker will not be able to access this ssh service, even if in possession of the correct private key used to SSH.

The gateway firewall rules are set to drop all traffic unless authenticated and authorized via SDP. These rules will prevent any incoming traffic from all unauthorized hosts, while the FWNKOP service will modify the firewall to allow autho- rized clients, which sends the appropriate SPA message first, to connect to the service.

V. CONCLUSION

This article demonstrates the implementation of SDP within IaaS for protecting cloud services. The security issues of IaaS were discussed in tandem with the protections SDP providers to mitigate them. The SDP in the IaaS deployment pattern was described in detail and then implemented. This was followed by several tests. The results shown in this article are twofold. First, the port scanning attack demonstrates how SDP can "black-out" services behind gateways even in cloud environ- ments, preventing unauthorized access to them. Second, the resiliency of SDP was verified against DoS attacked. This was done in a local environment due to restrictions from the cloud provider.

CSA has highlighted the potential for SDP in IaaS, detailing the different security threats that IaaS suffers from and how SDP can mitigate them. This article verifies those benefits in a real AWS environment. The architecture requires minimal time and effort to introduce to the existingsolutions. The automated setup process used in the test bed required under 10 min to introduce the system and another 5–10 min to configure. The added overhead of connection time was also found to be 3.419 s on the initial connection with minimal overhead on response times. This combination of minimally invasive setup and added security benefits makes SDP a strong candidate for protecting IaaS.

VI. FUTURE WORK

SDP is a promising, complete security framework for IaaS; however, there is future work that remains. First, upon approval with the cloud provider, a controlled DoS or DDoS should be performed to verify the performance under known attack scenarios. Furthermore, the implementation in this article only takes the steps that a customer may take to secure their applications and services; however, the zero-trust SDP framework can be adapted for the cloud provider use as well. IAM services offered by cloud providers may be coupled with SDP at the hypervisor level to only allow users with access to the account and access to the respective VMs on which their services are deployed. SDP can be utilized in this case to "darken" VMs deployed on the same physical hardware to mitigate the potential of Cross-VM attacks. Providers may also use SDP as a security offering for API gateway authorization

REFERENCES

[1] J. Garbis, P. Thapliyal, B. Flores, and J. Islam, "Software defined perimeter for infrastructure as a service," Cloud Secur. Alliance, Seattle, WA, USA, Tech. Rep., 2016.

[2] M. J. Kaur and P. Maheshwari, "Building smart cities applications using IoT and cloud-based architectures," in *Proc. Int. Conf. Ind. Informat. Comput. Syst. (CIICS)*, Mar. 2016, pp. 1–5.

[3] *Top Threats to Cloud Computing: Deep Dive*, Cloud Security Alliance, Seattle, WA, USA, 2018.

[4] J. Ullrich, T. Zseby, J. Fabini, and E. Weippl, "Network-based secret communication in clouds: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1112–1144, 2nd Quart., 2017.

[5] Esposito, A. Castiglione, B. Martini, and K.-K.-R. Choo, "Cloud manufacturing: Security, privacy, and forensic concerns," *IEEE Cloud Comput.*, vol. 3, no. 4, pp. 16–22, Jul. 2016.

[6] Z. Xiao and Y. Xiao, "Security and privacy in cloud computing," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 843–859, May 2013.

[7] B. Bilger, "Software defined perimeter working group SDP specification 1.0," Cloud Secur. Alliance, Seattle, WA, USA, Tech. Rep., Apr. 2014.

[8] S. Rose, O. Borchet, S. Mitchel, and S. Connelly, "Zero trust archi- tecture," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. 09/23/19: SP 800-207 (1st Draft), 2019.

[9] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds," in *Proc. 16th ACM Conf. Comput. Commun. Secur.*, 2009, pp. 199–212.

[10] J. Idziorek and M. Tannian, "Exploiting cloud utility models for profit and ruin," in *Proc. IEEE 4th Int. Conf. Cloud Comput.*, Jul. 2011, pp. 33–40.

[11] S. Meena, E. Daniel, and N. A. Vasanthi, "Survey on various data integrity attacks in cloud environment and the solutions," in *Proc. Int. Conf. Circuits, Power Comput. Technol. (ICCPCT)*, Mar. 2013, pp. 1076–1081.

[12] X. Yin, X. Chen, L. Chen, G. Shao, H. Li, and S. Tao, "Research of security as a service for vms in iaas platform," *IEEE Access*, vol. 6, pp. 29158–29172, 2018.

[13] Moubayed, A. Refaey, and A. Shami, "Software-defined perimeter (SDP): State of the art secure solution for modern networks," *IEEE Netw.*, vol. 33, no. 5, pp. 226–233, Sep. 2019.

[14] M. Li, H. Tang, A. R. Hussein, and X. Wang, "A sidechain-based decen- tralized authentication scheme via optimized two-way peg protocol for smart community," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 282–292, 2020.

[15] Sallam, A. Refaey, and A. Shami, "On the security of SDN: A completed secure and scalable framework using the software- defined perimeter," *IEEE Access*, vol. 7, pp. 146577–146587, 2019.

[16] Refaey, A. Sallam, and A. Shami, "On IoT applications: A pro- posed SDP framework for MQTT," *Electron. Lett.*, vol. 55, no. 22, pp. 1201–1203, Oct. 2019.

[17] J. Singh, A. Refaey, and A. Shami, "Multilevel security framework for NFV based on software Defined perimeter (SDP)," *IEEE Netw.*, early access, Mar. 27, 2020, doi: 10.1109/MNET.011.1900563.

[18] J. Koilpillai, “Software defined perimeter (SDP) a primer for cios,” Waverley Labs LLC, Waterford, VA, USA, Tech. Rep., 2017.

[19] S. Feghhi and D. J. Leith, “An efficient Web traffic defence against timing-analysis attacks,” *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 525–540, Feb. 2019.