Fundamentals of a more secure SASE by abstracting the attack surface

EXECUTIVE SUMMARY

Secure access service edge, or **SASE**, is a cloud-based IT model that bundles software defined networking with network security functions and delivers them from a single service provider. Gartner, a global research and advisory firm, coined the term "SASE" in 2019.

Enterprises started adopting SASE in 2019 and now it has gone mainstream. However there are still a few problems in the model that concern security. First of all, the vulnerabilities are still the same as in on-premises solutions, except that it just moved into the cloud. Secondly, the respective controllers pose a single point of failure in terms of security and availability.

In this whitepaper, we are proposing a **framework where improvements can be made to any existing SASE infrastructure** in order t**o significantly improve resiliency, attack surface abstraction and eventually create a self-healing SASE in a seamless way.** This is done by using Kubernetes, orchestration strategies and a few lines of code.

WHAT IS SASE

SASE approach offers **better control** over and **visibility** into the user's activity, traffic, and data accessing a corporate network — vital capabilities for modern, globally distributed organizations. **Networks built with SASE are flexible and scalable**, able to connect globally distributed employees and offices across any location and via any device. In contrast to the old school, rigorous company perimeter approach, now the perimeter is infinite and always in motion.

SASE has the following components:

	SASE
Network as a service	**SD-WAN**, CDN, Multi Cloud, WANaaS
Network security as a service	**FWaaS**, **ZTNA, SWG, CASB**, DDos, RBI, DNS, WAAP

The highlighted components are mandatory, the rest are nice to have.

SASE combines software-defined wide area networking (SD-WAN) capabilities with a number of already established network security functions, all of which are delivered from a single cloud platform. In this way, SASE enables employees to authenticate and securely connect to internal resources from anywhere, and provides organizations better control over the traffic and data that enters and leaves their internal network.

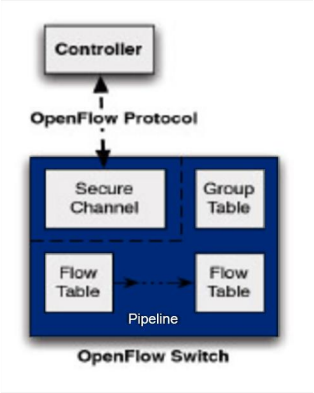
CROWN JEWEL OF SASE: CONTROLLERS (SD-WAN and ZTNA / SDP)

In terms of SASE, there are two main vulnerable areas: SD-WAN controller and SDP controller.

SD-WAN CONTROLLER

One of the most significant security risk factors would be an **attack at the control plane layer** via a compromised SDN controller. Due to the centralized design of SDN, the SDN controller becomes the brain of the SDN architecture. Attackers can focus on compromising the SDN controller in an attempt to manipulate the entire network. If the attacker successfully gains access, the compromised SDN controller can be used to direct the network devices it controls (e.g., switches) to drop all incoming traffic or launch serious attacks against other targets, such as sending useless traffic to a victim to deplete its resources. To mitigate this security risk, it is **critical to harden the operating system** that hosts the SDN controller and **prevent unauthorized access to the SDN controller**. Furthermore, the control plane layer is susceptible to a distributed denial-of-service (DDoS) attack. SDN switches may flood the SDN controller with excessive queries that may potentially cause a delay or drop of queries.

If attackers compromise the SDN controller, they can hack the SDN applications to manipulate security applications to reprogram the network traffic flow through the SDN controller, create new hosts, etc.



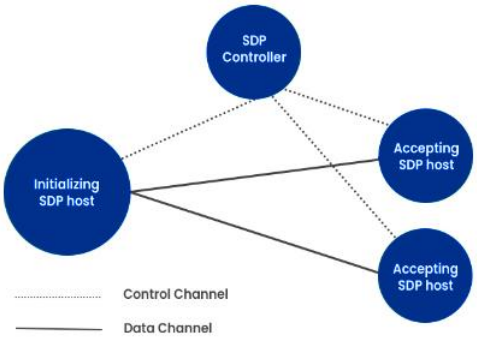
Communicating messages between the control plane layer and the data plane layer is subject to man-in-the-middle attacks. The attacker can potentially modify rules sent from the SDN controller to switches and take control of the switches. One of the **most effective solutions to such attacks is to encrypt the messages with the use of digital signatures** for securing and proofing the integrity and authenticity of the messages.

Real-time programmability is also vulnerable at the application plane layer. Specifically, if the attacker can hack the SDN security application, they can manipulate the flow of network traffic through the SDN controller. If the SDN applications are compromised, so is the entire network. To effectively mitigate this security risk, it is critical that **secure coding practices are enforced with comprehensive change management and integrity check processes** as part of the software development life cycle.

SDP

SDP hosts can communicate with each other as determined by an SDP controller. An SDP host can either initiate or accept a connection. To identify which hosts they can connect to, and initiate SDP host connects with an SDP controller. Only approved messages and connections from an SDP controller are accepted by an accepting SDP host.

Gateways are used in some SDP topologies to function as the accepting host between the two connected devices/users. All communications and users/devices are kept safe through encrypted connections – commonly a virtual private network (VPN) tunnel – between controllers, hosts, and gateways.



Controllers play a critical function in an SDP design because they connect devices to protected resources. It is difficult to connect to resources if controllers are not available (DDoS, ransomware). Furthermore, SDPs are not the same as typical network security measures because attackers could modify policies of all devices and apps, implementing an attack that might create network and infrastructure interruptions in large companies.

ATTACK VECTORS

We use the STRIDE framework to show the attack vectors on SASE applications. STRIDE is a model of threats, used to help reason and find threats to a system. This includes a full breakdown of processes, data stores, data flows, and trust boundaries.

	Threat	**Property Violated**	**Threat Definition**
S	**Spoofing identity**	Authentication	Pretending to be someone other than yourself
T	**Tampering**	Integrity	Modifying something on disk
R	Repudiation	Non-repudiation	Claiming that you didn't do something
I	Information disclosure	Confidentiality	Providing information to someone not authorized
D	**Denial of service**	Availability	Exhausting the resources that are needed to provide service
E	Elevation of privilege	Authorization	Allowing someone to do something they are not authorized to

Additionally we assume that:

- Attackers are coming from outside the network
- Adversaries have multiple network resources to use for scanning, planning, etc.

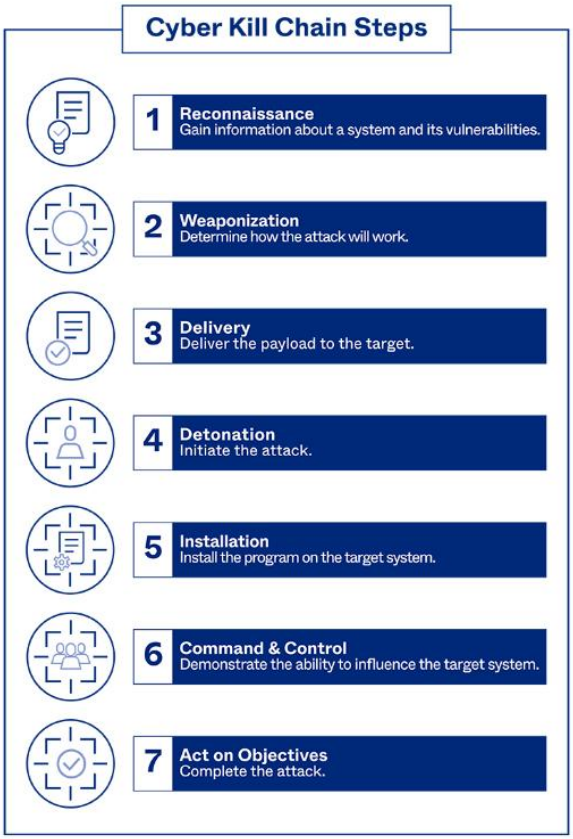
Self-healing SASE

We are using Moving Target Defense (MTD) that leverages a simple, potent, and proven strategy to prevent cyber attacks: a **moving target is harder to hit** than a stationary one. An analogy from physical security explains how moving target defense works.

Every bank has locks on the doors (NGAV - next generation antivirus) to keep crooks out, and perhaps video cameras (EPP - endpoint protection or EDR - endpoint detection and response) to deter perpetrators and record illicit activity once criminals have penetrated defenses. MTD takes a prevention-first approach to complement more reactive fortifications that may miss sophisticated zero-day threats by constantly shifting and hiding entry points from criminals to prevent them from getting in. Even better, it sets a trap to capture threat actors' movements to further secure against future attacks.

Most attacks follow a prescribed roadmap to reach their intended target. Therefore, if attackers can't find what they expect –as in a door or window into an organization– they will fail. Entry points kept in motion, rather than left at rest, are significantly more secure because they are unpredictable and unknown by nature. **With Moving Target Defense, attackers must find their way forward and fight their way through. Given the added effort and cost of perpetuating these attacks, most attackers move on to other, easier targets.**

Basically it hinders attackers, **increases their costs**, and makes their task nearly impossible at each step of the cyber kill chain:



Put simply, **MTD hides vulnerabilities**, **weaknesses and critical assets from threat actors without disrupting current NGAV, EPP, or EDR functionality**. This ensures that zero-day, ransomware, and other advanced attacks are stopped before they can do damage. Furthermore, attacker dwell time can be limited (right now it is an average of 287 days [according to Ponemon Institute.]

We even **brought this further by creating a self-healing network** where we use the immutability of Kubernetes (or Docker) to perpetually recreate a pristine state hence kicking attackers and their malware out and recreate edited policies on the controller level. This immutability **ensures that spoofing and tampering are impossible to hold up**. Ransomware and denial of service can be tackled almost immediately.

Furthermore, by using this strategy **we eliminate the usual attack surface** by scrambling it even at the usual locations (PoP - points of presence) where external attackers get in.

CONCLUSION

SASE is a dramatic improvement on traditional data center security stacks. They are distributed to be close to remote users and offer better protection against attacks on those users. Yet, SASE solutions could be compromised to either take over control of data flows or deny access to users.

To counter this vulnerability we propose deploying MTD to SDN and SDP controllers. In this way, attackers will not have time to execute simple or sophisticated attacks against SASE infrastructure.

We invite you to give it a test drive of our moving target defense SDN controller. Caveat - we are SASE vendor agnostic but the current nature of SASE implementations - multiple vendors, etc - would require customizing.