Mitigating threats with Zscaler Workload Segmentation

Harry Sverdlove

December 1, 2020

Table of Contents

Threat Frameworks

When doing threat modelling and cyber defense planning, it's more important to understand how attacks occur than to defend against any specific attack. This is as true in the physical world as it is in the virtual. For example, if you wanted to protect the valuables in your home, you would not just focus on the one or two known criminals in your neighborhood. Rather, you would consider the steps *any* criminal must take to steal your property and focus on preventing any one, or many, of those steps. In that way, you are defending against anyone who might choose to take your property, not just a few specific individuals.

In cyber, there are a number of models that outline the stages or lifecycle of an attack. By understanding each stage, and the specific techniques and tactics used by attackers at each stage, defenders can better detect and defend against each step of an intrusion. If you can effectively stop (or detect and respond) at any given phase, you can neutralize the entire attack.

In 2011, Lockeed Martin introduced the Cyber Kill Chain to describe the phases of a cyberattack. In 2015, Mitre introduced the ATT&CK Framework, a more detailed breakdown of the stages of a cyberattack and their tactics/techniques. ATT&CK is perhaps the most popular model, but there are others as well, such as the Diamond Model of Intrusion Analysis. These models are not exclusive or incompatible; they are simply different ways to frame the behaviors of an attack and, consequently, different ways to evaluate the defensive controls you put in place.

The evolution of our understanding of cyberattacks (not to mention the evolution and growing sophistication of the attackers) led to the evolution of security technologies that better align with attack stages and attacker TTPs (tactics, techniques, procedures). For example, instead of antivirus tools that look for very specific signatures of *known* malware, we have technologies that look for sequences of malicious behavior; instead of network firewalls that scan for known signatures or block specific ports, we have segmentation technologies that detect or prevent communications between *any* unauthorized software. These advancements enable cyber defense programs to build defenses against new and unknown attacks, not just already discovered ones with very specific signatures.

Zscaler Workload Segmentation and Mitre ATT&CK Framework

Zscaler Workload Segmentation (ZWS) is a microsegmentation solution that is focused on securing east-west traffic based on the identities of the applications or software communicating. This both detects and prevents unauthorized lateral movement within a network (e.g. a data center or a private/public/hybrid cloud). Lateral movement, in its various different forms, is a critical step in every one of the attack frameworks. Stopping this behavior can effectively neutralize a cyberattack even if the technical details of that attack are not yet fully understood.

As just one example, let's take a closer look at how ZWS maps to the MITRE ATT&CK framework. For enterprise organizations, the framework has 14 tactics (originally 11), each with specific techniques that describe how the attacker might achieve their intent. These tactics are:

(1) Reconnaissance (TA0043)

(2) Resource Development (TA0042)

(3) Initial Access (TA0001)

(4) Execution (TA0002)

(5) Persistence (TA0003)

(6) Privilege Escalation (TA0004)

(7) Defense Evasion (TA0005)

(8) Credential Access (TA0006)

(9) Discovery (TA0007)

(10) Lateral Movement (TA0008)

(11) Collection (TA0009)

(12) Command and Control (TA0011)

(13) Exfiltration (TA0010)

(14) Impact (TA0040)

Of course, ZWS maps directly to Lateral Movement, but when diving deeper into the actual techniques used by these tactics, ZWS serves as a primary defense against many aspects of the following tactics: Discovery, Lateral Movement, Command and Control, and Exfiltration. ZWS secures network communications, so any technique that involves using unauthorized (even if legitimate) software to make a network connection (even local or loopback connections) will be detected or prevented (depending on policy settings). The following mapping is just an initial example:

Discovery:

This is where an adversary is trying to understand your environment. Because ZWS blocks all unauthorized communications, including attempted communications that occur when scanning, it prevents the attacker from gaining further insight into other services running in your network.

- Account Discovery (T1087): ZWS will prevent any new or unexpected software from communicating with your LDAP/Active Directory servers or other external systems to gain additional credentials.
- Network Service Scanning (T1046): When ZWS is protecting a host or application, it prevents that entity from receiving inbound connections from unauthorized sources, even if those attempts are simply connects or pings (ZWS uses *pre-connect* validation which prevents the connection from even occurring, versus traditional firewalls which may tear down the connection if it is determined to be malicious but often that is already too late -- as the exploit or simple discovery can occur on the initial connect handshake).
- Network Sniffing (T1040): ZWS prevents any unauthorized software from attaching to the network, whether to "sniff" or make a specific connection.
- Remote System Discovery (T1018): As described earlier, ZWS effectively hides protected hosts and services from unauthorized pings or scanning.
- As well as any aspect of the other Discovery techniques if they involve using software to make network connections

Lateral Movement:

This is where an adversary is moving through your environment. Movement is critical for most cyberattacks because an adversary rarely lands on the actual system(s) they are trying to compromise.

- Exploitation of Remote Services (T1210): ZWS prevents unauthorized connections to protected services, and because enforcement is done at the pre-connect stage, even protocol-level vulnerabilities (which can often be exploited with a simple connect) are prevented.
- Lateral Tool Transfer (T1570): ZWS can be used to apply protection around tools that may already be present (RDP, scp, rsync, sftp, etc.), as well as default blocking of connections from unknown software.
- Software Deployment Tools (T1072): Software deployment and administrative tools can be a problem for traditional firewalls because they require such ubiquitous access. ZWS will automatically model observed behavior of administrative tools and recommend policies that are directional and narrow in scope. So for example, with ZWS you can easily authorize specific systems to deploy software via SCCM across large swaths while preventing SCCM being used anywhere else.
- In addition, areas like SSH Hijacking, RDP Hijacking, and Remote Services can also be augmented in their protection with ZWS. Most backend servers should *not* be allowing outbound connections from tools like ssh (and in many cases they should not be allowing inbound either). Proper system and minimal OS configuration should be used in general for such systems, but ZWS can be used to effectively prevent use of such tools. If connections are required (e.g. for administrative use), ZWS can properly restrict the use of those tools to just authorized software on authorized hosts.

Command and Control

This is where the adversary establishes a means to communicate with, and control, the compromised system(s). In addition to protecting all east-west traffic, ZWS can protect against unauthorized inbound or outbound traffic from the internet, preventing most methods of C2.

- Application Layer Protocol (T1071): This is one of the areas ZWS is far more secure than a firewall. Traditional firewalls can only block based on address/port/protocol. NGFWs can use deep packet inspection (DPI) to block specific types of communications, like non HTTP traffic. But neither can distinguish between an authorized application, like a browser, from a malicious piece of software or the misuse of an existing piece of software like wget. If the traffic looks like normal web traffic, a firewall will just let it through. ZWS protects based on the identity of the software communicating, regardless of its address/port/protocol and regardless of its communication syntax. Therefore, unauthorized software can be blocked even if it is trying to “hide in plain sight” with its traffic by pretending to be a standard service.
- Encrypted Channel (T1573): Because ZWS is host-based, has its own control mechanism, and does not rely on deep packet inspection, it properly enforces policy regardless of the encryption used. There is no need for MITM certificates or other special configurations to support encrypted communications.

- Since ZWS is host-based and operates independent of layers 3 through 7, it is not subject to the limitations of centrally placed network security devices. Nearly all of the other techniques listed under C2 will be prevented using the default deny aspect of ZWS. If a piece of software is not authorized to communicate with an explicit external address (or an explicit internal application residing within the network), the communication is blocked. This includes techniques like:

  Fallback Channels (T1008), Ingress Tool Transfer (T1105), Multi-State Channels (T1104), Non-Standard Pot (T1571), Protocol Tunneling (T1572), Proxy (T1090), Remote Access Software (T1219), and Traffic Signaling (T1205)

Exfiltration

This is where the adversary is trying to steal or extract data. Often this involves similar communications channels as with Command and Control, so ZWS' applicability here is the same as with the previous tactic.

- As noted with C2, because ZWS is not a centrally located network device, it does not have the same limitations as traditional firewalls or other network-centric security tools. Policies are based on application and host identity, not based on size of packets, time of day, or network protocols. Therefore, there is no "hiding within normal traffic" or "avoiding threshold limits" with ZWS. This covers at least:

  Automated Exfiltration (T1020)

  Data Transfer Size Limits (T1030)

  Exfiltration Over Alternative Protocol (T1048)

  Scheduled Transfer (T1029)

- Exfiltration Over C2 Channel (T1041): As per previous points, any Command and Control protection applies to Exfiltration as well.
- Exfiltration Over Other Network Medium (T1011): ZWS works equally on WiFi as it does on wired network connections.
- Exfiltration Over Web Service (T1567), Transfer Data to Cloud Account (T1537): ZWS is typically deployed within servers and data centers. Since these are not generally end-user systems, access to common web services or cloud storage accounts can be effectively blocked with ZWS policies.

The above tactics are where ZWS can be applied as a primary mitigation technique. ZWS can also be applied as a secondary or additional mitigation for any other tactic that involves network communication. For example, ZWS can be used to prevent Initial Access by securing all inbound traffic from the internet (as most backend servers have very specific inbound allowances, or none at all).

In fact, MITRE details all the techniques where “network segmentation” can be used as a mitigation. See https://attack.mitre.org/mitigations/M1030/. Their table is useful as reference, but because ZWS is a host-based application identity solution for segmentation, not a network-based firewall, it applies to a broader set of techniques (areas within Application Isolation, Limit Access to Resource Over Network, Network Intrusion Prevention, to name just a few examples).

When developing a security program, it’s important to deploy overlapping processes and technologies to cover as many aspects of the attack lifecycle as possible. While ZWS is primarily applied for preventing lateral movement, and the related tactics of discovery, command and control, and exfiltration, it also serves as an additional layer to other adversary tactics to help establish a comprehensive security strategy.