Introduction

Welcome to my course on Software-Defined Perimeter: Leveraging Zero Trust to Create a New Network and Security Architecture. I'm Matt Conran from Network Insight, and I'm a network and security architecture based in Ireland. I'm excited to have you join me on this course, where we are going to discuss new methods to make your network more robust and secure from both insider and external security threats. In this course, we are going to discuss why there is a need, and how to create the new network and security architecture called Software-Defined Perimeter. Software-Defined Perimeter leverages the zero trust principles, where no user, device, or application is trusted until certain security controls are passed; essentially, SDP creates a dark network, protecting your critical network infrastructure and applications. We're all aware that our environment has changed a lot. We have a big push to the cloud, BYOT, IoT, remote workers, and phishing threats are all putting pressure on existing architectures. We have to accept that traditional methods to network and its security are no longer effective, creating a lot of challenges. We will address these challenges, and how SDP can help alleviate them, and go through some of the steps used to create an SDP. The Cloud Security Alliance has released a number of architectural guidelines. We will start this course discussing an SDP architecture that closely follows these guidelines, and utilizes a protocol called Single Packet Authorization. We will finish the course with a hands-on demonstration of a multi-cloud scenario, using an SDP architecture that is a variant of the CSA guideline. In this module, we're going to dive into why we need Software-Defined Perimeter. We mentioned that our environment has changed. We will highlight these changes and how they affect today's current approaches to networking and security. The common issues we are seeing is at the fixed perimeter to networking and security is designed for a different era, certainly not for today. The perimeter is now dissolved, yet traditional methods are still being used. IP addresses lack any type of user knowledge to assign and validate the trust of the device, yet the IP address are the base for security policy, and the anchor for network location. We will also examine how SDP can solve these issues, and demonstrate the challenges when using traditional firewall for security enforcement in the cloud. We will finish up this module with a quick overview of Software-Defined Perimeter. Hacking is getting more and more sophisticated, resulting in a pervasive sense of unease. Why is our network infrastructure and applications wide open to such severe security risks? We will discuss the reasons why this has occurred, and how it can be prevented. Essentially, we give devices IP addresses to connect to the internet. Within this connectivity model, we have global reachability, which signposts three pathways. Firstly, you need to know the bad actors who are attempting to connect to your machine. This gave rise to threat intelligence. Secondly, you must lock down a machine so it's airtight, with vulnerability, patch, and configuration management. The third component is the implementation of a firewalling device, which has no user context. More than often, firewalling is based on static configurations. None of these techniques ensure that attacks will not happen. They are like a preventive medicine, but really what's needed is total immunization to ensure that attacks cannot even touch your network infrastructure and its applications. Software-Defined Perimeter is a connection- based security architecture designed to stop attacks. It doesn't expose the infrastructure and its applications. It enables you to know who the authorized users are by authenticating and authorizing plus validating the device of the users before they connect to the protected resource. SDP allows you to operate while vulnerabilities, patches, and configurations are in progress. Essentially, it cloaks the application, or groups of applications, so they're invisible to attack.

Weak Security Foundation

TCP has been around for decades, and has a weak security foundation. When it was created, security was out of scope. TCP can detect and retransmit error packets, but left to its default, communication packets are not encrypted. This poses security risks. TCP operates with a connect first, authenticate second model of operation. When a client wants to communicate and have access to an application, they first set up a connection. Then, only once the connect stage has been carried out successfully, can the authentication stage occur; and once the authentication stage has been carried out, only then can we begin to pass the data. With the process of connect first, authenticate second, we are essentially opening up the door of the network and the application without knowing who is on the other side. Unfortunately, with this model, we have no idea who the client is until they have carried out the connect phase, and once they have connected, they are already in the network. Maybe the requesting client is not trustworthy and has bad intentions. If so, once they connect, they have the opportunity to carry out malicious activity. With Software-Defined Perimeter, before the client is allowed to set up the connection, we must first establish trust between the client and the application. Trust is bidirectional, and happens between the client to the SDP and the application to the SDP. It's not a once off check, it's a continuous mode of operation. It is only once sufficient trust has been established that we then move to the next stage, which is the authentication. Once authentication has been carried out, can we then connect the user to the application. The process of events that SDP offers flips the entire security model, and makes it more robust. We have gone from connect first, authenticate second, to authenticate first, connect second. The user cannot see or know where the applications are located. SDP hides the application, and creates a dark network by using Single Packet Authorization, SPA, for the authorization. The goal of SPA is to overcome the opened and insecure nature of TCP IP, which follows a connect first, then authenticate model. SPA is a lightweight security protocol that validates a device or user's identity before permitting network access to the SDP. The purpose of SPA is to allow a service to be darkened via a default-deny firewall. SDP is an extension of Zero Trust. Zero Trust essentially does not grant access until sufficient trust is assessed, and security controls are passed. Only then can the client see and access the requested resource. For more information on Zero Trust concepts, check out my course, Zero Trust Networking: The Big Picture. The traditional world of networking started with static domains. Networks were initially designed to create internal segments that were separated from the external world by a fixed perimeter. The classical network model divided clients and users into two groups, trusted and untrusted. The internal network was deemed trustworthy, whereas the external was considered hostile. The perimeter approach to network and security has a number of zones. We have, for example, internet, DMZ, trusted, and then privileged. From there we have public and private address spaces, which separates network access. Private addresses were deemed more secure than public addresses, as they were not reachable on the internet; however, this trust assumption that all private addresses are safe is where all of our problems started. The digital threat landscape is concerning. We are getting hit by external threats from all over the world. Threats also come internally within your network. We have insider threats within a user group, and insider threats across user group boundaries. These types of threats need to be addressed one by one. One issue with a fixed perimeter approach is that it assumes a trusted internal network and a hostile external network; however, we really need to assume that the internal network is as hostile as the external network. Over 80% of threats are from internal malware or a malicious employee. The fixed perimeter approach to networking and security is still the foundation for most professionals, even though a lot has changed since the inception of the design. We are getting hacked every day, and major networks with skilled staff are crashing. Unfortunately, the perimeter approach to networking has failed to provide adequate security in today's digital world. It works to an extent by delaying an attack; however, with enough patience and skill, a bad actor will eventually penetrate your guarded walls. If your house is guarded by large walls and a gate, you would feel safe and think that you were fully protected while inside the house; however, as large and as thick as the perimeter protecting your house may be, there is still a chance that someone could climb the walls and access your front door, and gain entry to your property. However, if a bad actor cannot even see your house, they will not have the ability to go to the next step, and try to breach your security. The environment has changed. With the introduction of cloud, BYOD, machine-to-machine connections, the rise in remote access, and phishing attacks. We have a lot of internal devices, and a variety of users, such as onsite contractors, that need access to network resources. There is also a trend for corporate devices to move to the cloud collocated facilities, and offsite to customer pattern locations. IT is becoming more diversified, with hybrid architectures. These changes are causing major security problems for the fixed perimeter. For example, with the cloud, the internal perimeter is stretched to the cloud, but traditional security mechanisms are still being used, but the cloud introduces a completely new paradigm, also remote workers. We have an abundance of remote workers working from a variety of devices and places. Again, traditional security mechanisms are still being used. As our environment evolves, security tools and architectures must evolve along with them. Let's face it, the network perimeter has dissolved, as your remote users, things, services, applications, and data are literally everywhere. As the world moves to the cloud, mobile, and IOT; the ability to control and secure everything in a network is no longer available. We're also witnessing an increase in phishing attacks. That can result in a bad actor landing right on your local LAN. A phishing attack is a type of social engineering that can be used to steal data. It comes in the form of an email, and once a user opens the email and downloads a malicious file, the bad actor has broad level access, and can move laterally throughout the network. Recently, I was a victim of a phishing email. If you're not educated about them, it's very easy to click and download the malicious file. In my case, the particular file was a .WAV file, it looked safe, but it's not. So we have broad level access and lateral movements. Keep in mind with traditional network and security mechanisms, when a bad actor lands in a particular segment, a VLAN, which is known as zone-based networking, they can see everything on that segment. This is what gives them broad level access. Also, the bad actor can easily move laterally from one segment to another. Lateral movement is a common technique that bad actors use to navigate between or within segments with the intention to compromise valuable assets. This is made possible as traditionally, a security device does not carry out filtering this low down in the network inside a VLAN, which would be known as intra VLAN filtering. A phishing email can easily lead the bad actor to the LAN, with broad level access and the capability to move laterally throughout the network. For example, a bad actor can initially access an unpatched file server. They then move laterally between segments to go to, for example, the web developers machine, and use a key logger to get the credentials to then access a database server. They can then carry out data exfiltration with using DNS, or even social media accounts, such as Twitter. DNS is generally not checked by firewalls as a file transfer mechanism, so data exfiltration using DNS would often go unnoticed.

Introducing SDP

Before, we had clear network demarcation points that were set by a physical central firewall, creating an inside and an outside zone of trust. Anything on the outside was considered hostile, and anything on the inside was considered trusted. SDP flips this model around, and considers everything untrusted. To do this, there is no longer predefined fixed network demarcation points. The network perimeter that was initially set in stone is now fluid and software-based. SDP is a connection-centric, not network-centric. Each user on a specific device that connects the network gets an individualized connection to a specific service hidden by the perimeter. Instead of having one perimeter that every user has to use, SDP creates many small perimeters that are purposely built for a user and an application. These are known as microperimeters. Clients are cryptographically signed into these perimeters. The perimeter is now based on the user and the device context, and can dynamically adjust to changes in the environment, so as a user moves to different locations or different devices, the SDP can detect this and set the proper security controls based on the new context. The data center is no longer the center of the universe, the user on their devices, along with the service request is now the new center of the universe. SDP does this by decoupling the user and device from the network. To remove the user from the network, SDP provides a separation of the data plane from the control plane. The control plane is where the authentication happens first, and then the data plane, which is the client's application connection, is used to transfer the data. Therefore, the user's don't need to be on the network to gain application access. As a result, they have least privilege access, and not broad level access.

The Challenges with IP Addresses

Everything today relies on IP addresses for trust, but there is a problem; IP addresses lack any type of user knowledge to assign and validate the trust of a device. There is simply no way for an IP address to do this. IP addresses simply provide connectivity, and do not get involved in validating the trust of the endpoint or of the user. Also, IP addresses should not be using an anchor for network location, as they are used today, because when a user moves from one location to another, the IP address changes. So at this point, you're probably saying, okay Matt, this is not a big deal. When I move locations from say, my internet café to a public hotspot, my IP address changes. So my TCP sessions will drop for a short time, but they will start back up again, and I'll be back working as normal within no time. But what about security policy that is assigned to the old IP address? This gets lost too when you change IPs. Anything tied to IP is ridiculous, as we don't have a valid hook to hang things on for security policy enforcement. When you examine policy, there are a number of facets. User access policy that touches on authorization, network access policy touches on what to connect to, and user account policy, which touches on authentication. With either one, there is no policy visibility with IP addresses. This is also a major problem for traditional firewalling, which just displays static configurations. For example, a static configuration may state that this particular source can reach this destination, using this port number. This has no meaning. There is no indication of why that rule exists, and under what conditions should a packet be allowed from one source to one destination. There is no contextual information taken into consideration. We need to look at more than ports and IP addresses when it comes to creating a robust security posture. For a robust security posture, you need full visibility into the network to see who, what, when, and how they are connecting, with what device. Today's firewall is static and only contains information about the network. On the other hand, SDP enables a dynamic firewall that has user and device context to open a firewall for a single secure connection. The firewall remains closed at all times, thereby creating a black cloud stance, regardless of whether the connections are made to the cloud or on prem. Next generation firewalls are more advanced than traditional firewalls, and they use information layers five to seven to perform additional functionality. They can provide advanced features, such as intrusion detection and VPN, but they are still IP-based systems, offering limited identity and application- centric capabilities. Next-gen firewalls are still static firewalls, and they often mandate traditional perimeter-centric network architectures, with site-to-site connections, and don't offer flexible network segmentation capabilities. Similar to traditional firewalls, their access policies are typically course grade, providing users broader network access than what is strictly necessary.

Cloud Environments

Today we are very cloud-focused, but when moving applications to the cloud, we need to be very security focused. The cloud environment is less mature in providing the traditional security controls that we're used to having in our legacy environment. So when you are putting applications in the cloud, you shouldn't leave security to its defaults. Why? Well firstly, we are operating in a shared model where the tenant next to you can steal your encryption keys or data. There has been a lot of cloud breaches. In terms of cloud protection, we have firewalls with static rule sets, authentication, and key management issues. One of the biggest problems is that when you move to cloud-based applications, the perimeter is also moved. Servers are no longer under your control, mobile and tablets add to the problem, as they can be located everywhere. So trying to control the perimeter is very difficult. One of the biggest problems is that when you move to cloud-based applications, the perimeter has also moved. Servers are no longer under your control. Mobile and tablets add to the problem, as they could be located everywhere. So trying to control the perimeter is very difficult. More importantly, firewalls only have access to network information, yet they are controlling a lot of information and should be armed with more content. Defining this perimeter is what SDP is doing. Firewalls are now managed by cloud users who are moving the application to the cloud, not the IT teams within the cloud providers. Vulnerability and patch management is taken over by the cloud providers for systems and networks that they own. However, the vulnerabilities in your application will still remain, and you are the one who's responsible for that. Also, most authentication systems provided by cloud providers could be a third-party, typically, not implemented using strict guidelines or tied to access control. Finally, key management poses problems. There could be too many keys; user keys, device keys, and encryption keys. Who is managing them, and more importantly, where are they stored? So when you're moving applications to the cloud, even though cloud providers provide security tools, it's a cloud consumer that has to integrate the security.

Static Firewalls

Traditional firewalls typically control access to network resources based on source IP address. This creates the fundamental challenge with securing access, namely, we need to solve the problem of user access, but we only have the tools to control access based on the IP address. As a result, you have to group users, some of which may work in different departments and have different roles to access the same service, and with the same IP address. The firewall rules are also static, and don't dynamically change based on the levels of trust of a device, they only provide network information. Maybe the user moves to a more risky location, such as an internet café, its local firewall, or even antivirus software has been turned off by malware, or even by accident. A traditional firewall cannot detect this, as they live in a limited world of the five tuple. Traditional firewalls are only capable of expressing static rule sets, and cannot express or enforce rules based on identity information. Upon examining the cloud, let's make the comparison of a public parking space. A public cloud is a place where you can put your car publicly, in comparison to putting your car in your private garage. In a public parking space, we have multiple tenants that can take your space or damage your car. With cloud environments, we have security groups that provide basic filtering and isolation, but once configured in the cloud, they cannot distinguish between users coming in on the same IP address, and must grant the same access to everyone. In some cases, this could be full access, that may not be requirement for some of the users. This also means that malicious users, attackers, or malware can traverse from on-premise to the cloud network unimpeded. On the bottom left, we have a company headquarters that consists of a number of users, all of which have different roles. Even though these users have different roles and responsibilities within the organization, they are mapped to a single IP address when assessing resources in the cloud. In the middle, we have users working from a coffee shop. The coffee shop is external to the organization, and should be deemed as insecure. It is also natted to an IP address that could change. Network access from these insecure locations will extend to any malicious users on the same network. It's difficult to manually adjust network access and keep up with users changing locations and access needs. IP addresses are dynamically assigned, and are subject to change on a regular if not daily basis. Users can also access the cloud from multiple devices. So either the IT operations teams continually update security group rules, imposing a delay on business, or simply supply security policy, fully open to the cloud, leaving network resources at a significant risk. While no individual user requires access to all servers in a data center, this environment effectively forces the firewall to have a rule set that permits all IPs in the office LAN to all IPs in the datacenter network. Organizations can only think they are secure by using VPNs from unsecured locations. But they are not. The VPN connection terminates at a point in a network, from where, if the bad actor hijacks the VPN connection, they can try to get into services on the network segment that the VPN ends.

Overview of SDP

The principles behind SDP are not entirely new. Multiple organizations within the Department of Defense and intelligence communities have implemented a similar network architecture based on authentication and authorization prior to network access. The process ensures that users are authenticated and devices are validated prior to establishing secure connections to hidden services. This is in comparison to today's security projects that are designed to allow access to servers prior to authenticating the users or their devices. SDP creates a dark network and hides the network resources from unauthorized users. This reduces attack surface to an absolute minimum, as each authorized connection is individualized for that user. So, if an attacker gains a foothold in the organization through, for example, a phishing link as previously described, they will have a limited view of the network. Also, current firewalls only have network information and are static, while SDP has user context, and is dynamic. Applications typically have awareness and information about users that the networks don't have. Firewalls and gateways are network ware, and information about the network that applications do not have. Clients have all the information about the devices, but none of the users and the network. SDP is in a unique position, so the data provides all the necessary integration of these various pieces of information to become more secure. In addition, SDP separates the data plane from the control plane, and as a result, this architecture supports all cloud and hybrid architectures.