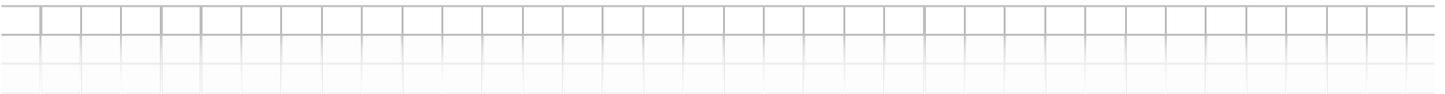


Network Security

Why **Authentication** Matters



For most organizations physical security is a given. Whether it is video surveillance, access control, motion detectors, or alarms; commercial businesses have embraced the concept of protecting their physical property and assets. As more and more business is being conducted via the internet and users are logging into their company's networks remotely, the importance placed on network security and protecting intellectual assets has risen dramatically.

The purpose of network security is to protect the network and it's elements from unauthorized access breaches that may lead to the loss of data, revenue, and/or productivity. This whitepaper will discuss some of the security measures that are currently being employed within networking equipment to limit infiltration from malicious users and potentially damaging electronic content.

What are HTTPS, SSL, and SSH?

Like a passport or a driver's license, an SSL Certificate or SSH Key is issued by a trusted source, known as the Certificate Authority (CA). Many CA's will verify the domain name and the existence of your ownership of the domain name, by issuing digital certificates that contain a public key and the identity of the owner. This certificate is also an attestation by the CA that the public key contained in the certificate belongs to the person, organization, server or other entity noted in the certificate.

HTTPS stands for Hypertext Transfer Protocol Secure. It is server software which provides the ability to "secure" transactions that take place on the World Wide Web (www). If a website is running off of an HTTPS server you can type in HTTPS instead of HTTP in the URL section of your browser to enter into the "secured mode". HTTPS is often offered by financial institutions to support their online banking offerings in order to verify that the information being passed between your PC and their server is "secure".

SSL stands for Secure Socket Layer and is a standards-based encrypting method that enables HTTPS to function securely. SSL is widely implemented to help secure internet transactions and is the preferred way for most financial institutions to conduct business through the web. It is based specifically on a public key certificate of authentication, which uses a self-generated private key and a password. An organization looking to implement SSL can purchase a public certificate from a few Certificate Authorities (CAs), such as: VeriSign, Comodo, and Go Daddy.



To make an SSL connection from a personal computer to an online banking server for example, the personal computer asks to connect and advertises the encryption methods that it is capable of. The banking server will pick the highest encryption method that both components are capable of and then will respond with its name, the Certificate Authority providing the Key, and the Public Encryption Key. At this point the customer could contact the Certificate Authority to verify the Key's authenticity. The personal computer will respond by randomly generating a number, encrypting the random number using the public key provided by the banking server, and then re-transmits to the banking server. At this point, the only device that should be able to decrypt the message is the banking server using its Private Key. This will initiate a unique "session" between the two network components in which the information passing between them is encrypted.

SSH works in a very similar fashion to SSL but it covers different layers of the OSI model in order to encrypt the exchanged data twice. The first encryption happens at the transport layer of the OSI model once a physical connection is made. At this point, the connection is authenticated using a Public Key that can be generated by most equipment. This verification of Public Keys creates a 128bit encryption that allows for a login and password to be given through an encrypted message between a client and a server. Once the password is verified, a unique session is created and a Private Key is used to digitally sign the session and encrypt it again, this time at the Session Layer of the OSI model.

In the event of an attack on an SSH system, an attacker must break the first 128bit encryption in order to intercept the password and then the attacker must break into the private key which is an immensely long series of numbers to "impersonate" a session. Otherwise, if they were logged-on (even at the exact same time as 'you') the sessions would be digitally signed with a different Private Key ID thus verifying the source of each session.

Many of Transition's newer models like the ION Chassis, SM24-100SFP-AH, and SM24-100SFP-AH all come with Secure Sockets Layer (SSL) and Secure Shell (SSH) termination for end-to-end security.

What is RADIUS and why would a Network Manager implement it with the ION Management Module (IONMM)?

RADIUS is an acronym that stands for Remote Authentication Dial In User Service, which is a networking protocol designed to help centralize the management authentication process. RADIUS has been widely adopted by service providers in order to limit the number of management rules that need to be implemented at every network element. The RADIUS protocol is designed to function in a client-to-server relationship. Meaning that all of the authentication rules and monitoring are done at a centralized server (or servers) and each client will access those rules before allowing any management to be done to the client.

Imagine that you are a network architect for a large service provider that has 20,000 network elements (switches, routers, NIDs, etc) and that there are 50 management stations with authorized access to the management network. Each management station might have different authorization levels (i.e. laptop A might be able to read/write NIDs and access switches, but might only have read access to core equipment). Before RADIUS, each network element would need to be programmed with the rules, as would every new network element. With a RADIUS server, the rules are in one location and each client device essentially “learns” the rules from that server.



In the case of a RADIUS application, the ION management module (IONMM) would act like a Network Access Server in that a user would send a management request to the IONMM and it would forward the request directly on to a RADIUS server for authentication and authorization. This feature will help network managers ensure that access to the IONMM management information is only given to authorized personnel -without worrying about installing the rules on all IONMMs deployed in their network. It also ensures that if a mistrusted user were to gain access to an ION management module, that user would not get access to all of the network rules and authentication codes without having to dig further into the network.

Many of Transition’s models, like the ION Chassis, SM24-100SFP-AH, and SM24-100SFP-AH all come with RADIUS built in.





What is an ACL rule and what does it give/prevent access to?

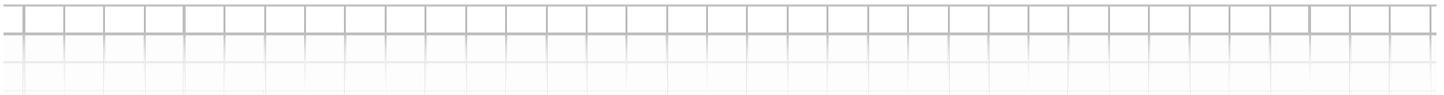
An Access Control List (ACL) is literally a list that permits or denies access to networking equipment. In terms of the ION management module (IONMM), it is best to think of an access control list in terms of having an 'allow' or a 'deny' statement with no middle ground. You are either allowed in, or you are out.

In the case of the IONMM, an access control list is used to control access to the management of the ION chassis and subsequently to the management interfaces of the cards inserted into that chassis. The IONMM has three layers of access control lists; in terms of the OSI model, the highest layer access control list for the IONMM is at layer 4. At layer 4, the IONMM can filter access based on UDP port, TCP port, or ICMP type. This filter is granting or denying access based on the type of transport protocols that could be attempting to gain access to the management interface. The next highest layer of access control operates at layer 3 and is based on the IP addresses of stations attempting to access the management controls. While the lowest layer ACL works at layer 2 with MAC addresses. By default, each of these ACL rules has an implicit allow command given. That means that everything that plugs in is allowed access with the proper username and password.

This implicit allow is important to have as a default because at set-up, it is important that a network operator be able to gain access to the device from any work station. Once deployed however, it represents a potential security breach point that could possibly have a negative effect on an organization's network and revenue. This makes ACLs important to configure in order to ensure not only that someone with the correct passwords is logging in, but also that a machine which you gave access to is logging in.

The most basic form of an ACL is to specify MAC addresses to allow/deny. This ACL type works well if connected on the same broadcast network or if you want to allow management from a router interface while also allowing the router to police who has access to its interface. The MAC ACL default state is to implicitly allow all MAC addresses with the correct username and password to access the management information. If a network operator only allows for management to be configured by a few devices and they are both in the same broadcast domain as the management module, entering the MAC address(es) directly into the management module is a very effective way of ensuring that only registered users are allowed access to the management interface. Once a MAC address is entered as allow, an implicit deny statement is automatically activated below it. This means that only the specified MAC address(es) are allowed access to the management interface. In the case of the ION management module, the IONMM must have at least one allow statement or management will become inaccessible via the web interface or Telnet.





If the management module is in a different broadcast domain than the management stations, the MAC table would need to use the MAC address of its router interface(s) since that is where the “source” MAC address will be associated to. If that happened, any piece of equipment with access to the routers’ interface would by default have access to the management interface of the IONMM with the correct username and password. This potential security problem can be worked around by assigning an ACL based on IP address(es).

Assume that a network admin has three individual laptops and two workstations that they use for all management applications. Simply entering those five IP addresses into the IONMM ACL with a permit statement will allow them to each access the management prompt-for-password. As soon as a “permit” statement is entered for an IP address, an implicit deny is also automatically entered which ensures that only authorized IPs have access to the management interface. For example, if a network operator wanted to permit 192.168.100.100 to manage the IONMM, the operator would not need to type in a denial statement for every other IP address, it would be implicitly applied to that interface after the allow statement was entered. It is possible to enter an IPv4 range of addresses or an entire IPv4 network using a single command. The IONMM must have at least one allow statement or management will become inaccessible via the web interface or Telnet.

Conclusion

Through the course of this whitepaper, we have covered some of the security technology that is being integrated into today’s networking elements to help ease network management and increase network security. The new ION Platform is evidence that Transition Networks has embraced these measures wholeheartedly - delivering a next generation family of intelligent optical networking devices built around a multitude of new security features and a robust management framework.

