Policy-based Authentication and Authorization: Secure Access to the Network Infrastructure

Jeff Hayes
Alcatel IND

Overview

A gaping hole in many of today's networks is the weak security surrounding the network devices themselves--the routers, the switches, and the access servers. In all public networks and in some private networks, the network devices are shared virtually among different user communities. Access to the configuration schemes and command lines is most often an "all or nothing" proposition--the network administrator gets either read-only privileges or read / write privileges. In this case, authentication equals authorization. Herein lies the problem.

Security policies may mandate certain administrators have read-only capabilities for all device parameters and read / write capabilities for a certain subset of commands. Each administrator may have a unique access profile. Authentication verifies identity. Authorization verifies privileges. This paper will address the value of using a centralized provisioned management structure that disseminates network policies and administration privileges to all the devices that make up the network infrastructure.

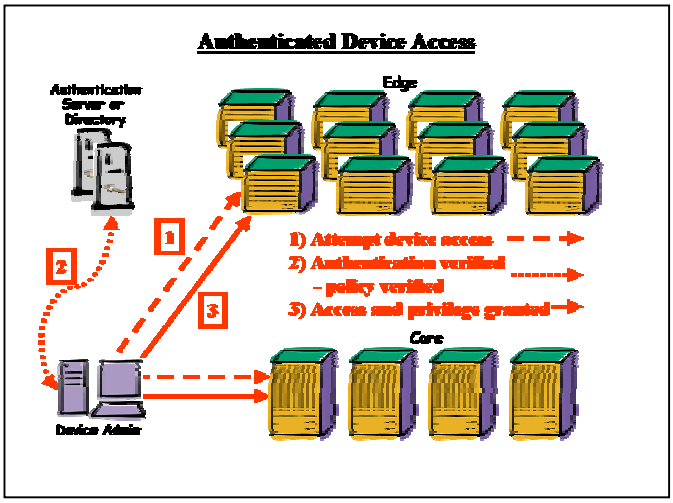
Authentication

With the mission critical nature of today's LANs, businesses cannot afford to have their data networks compromised by unauthorized users. Until now, the main security device or implementation for network access has been the firewall or router-based access control lists. Providing a barrier between untrusted networks (like the Internet) and internal, trusted networks is important but it does not end there.

Security experts today warn that while the external threat to networks is real, the largest threat often comes from inside the company. Authentication of internal users has long been established as the primary security device for file servers, network operating systems, and mainframes. There are also authentication requirements for routing tables (RIP, OSPF, BGP4), switch ports, router and switch configuration files, and web servers, to name but a few.

Traditional authentication--userID and password submitted in clear text--are typically not adequate for most security policies. People tend to use simple passwords or write them down in view of a potential perpetrator. Passwords can be stolen, sniffed, guessed, attacked via freeware dictionary tools or brute force attacks, compromised though insecure password files, and obtained through social engineering. Some passwords never expire. Some expire every 60 to 90 days without allowing the user to reuse an old password. Some are short, simple alpha characters only. Others are a combination of alpha, numeric, and special characters. Some password files are stored in clear text; others are encrypted. Many are transmitted in cleartext; others as cipher text. Whatever the method, some identification is better than none.

An area often overlooked is the authentication associated with the network infrastructure--the routers, switches, and access servers. The same issues associated with network operating systems and user applications have bearing in the infrastructure. Some methods are strong while other are weak. The idea presented here is to distribute device and user authentication to these devices to a standalone authentication server, as opposed to storing the information on each device independently.



For the past decade, the IT industry has seen an evolution in authentication techniques. Though most users rely on a user ID and password to establish their identity, more reliable authentication schemes involve multiple factors, insuring a great chance of accurate identification. These factors include:

Something you are, a biometric characteristic that is unique to every human. Fingerprints, hand prints, faces, retinas, voices, and keystroke timing can all be tied to a unique individual.

Something you know, the user ID and password method. It is currently the most widely used form of identification.

Something you possess, which typically involves external security devices including banking / ATM cards, tokens, and smart cards.

Advanced multiple-factor authentication techniques are needed to provide assurance that the user desiring connectivity is who she claims to be. There are a number of key methods for implementing this level of authentication.

One-time password schemes provide authentication over unsecured networks. The schemes can be based on one of two systems: 1) passwords stored both on a client device and on a central server; or 2) passwords kept on a central system and requested on demand by users. Because each password is only used once, most are sent in clear, unencrypted text, for example SASL methods like S/KEY.

Time-based passwords are based on both a password and an external security device. Users desiring access possess a hand-held device or token. When prompted to log in they identify themselves with an ID and a one-time password that is displayed on the token. The resulting password is a combination of a PIN and the number generated by the device's LCD. The users' temporary LCD number is synchronized with a central authentication server. If they match, the user is authenticated. An example is an RSA' SecurID token and ACE/Server.

Challenge and response systems are also two-factor authentication systems that leverage hand-held devices. The initial login request causes the authentication server to generate a random numeric challenge. Users unlock their hand-held device by punching in a PIN on the card's keypad. Users enter the challenge into the card as it was received. The card uses a known algorithm, like Data Encryption Standard (DES), to calculate and display the response to the challenge. Users enter the card's response to complete the login process. CRAM is the common technology used for this.

Smart cards are similar to the aforementioned token systems but contain more intelligence and processing power--small microprocessors with embedded encryption. Smart cards communicate directly with the authentication server through a card reader. Users provide the initial PIN and the card does the rest--exchanges keys and agrees on the encryption algorithms to be used.

These authentication technologies are typically complemented by services and / or servers that facility user profile management. The following authentication services add the element of authorization to the authorization process, something not provided by the above solutions.

Remote Access Dial In User Service (RADIUS) systems use a client or agent to facilitate users' login requests. The client passes the information to a RADIUS server, which authenticates the user. All communication between the client and server is authenticated and not sent in clear text. Servers can support a variety of authentication methods including PAP, CHAP, UNIX login, time-based tokens, and challenge / response systems. RADIUS is widely used by ISPs and other dial-up applications. RADIUS is also being used for user authentication, authorization, and accounting beyond dial-up applications, including LAN-based applications. As such, a new standard known as DIAMETER is being proposed that attempts to expand on RADIUS' known shortcomings, resulting in a broader protocol.

X.500 Directory Servers with X.509 using either simple (passwords) or strong (public-key) authentication accessible via the Lightweight Directory Access Protocol (LDAP) are becoming a critical information repository for end user profiles. Most of X.509 security elements are provided by RSA's Public Key Cryptography Standards (PKCS), although other methods exist. As network administrators see the value of minimizing the number of directories, there will be a move to consolidate directories and / or to utilize the meta-directory concept. The Burton Group defines a meta-directory service as being a class of enterprise directory tools that integrate existing, or "disconnected," directories by addressing both the technical and political problems inherent in any large-scale directory integration project. A big challenge albeit a worthy one.

Kerberos is a strong, single sign-on authentication system with a goal of validating a principal's identity. A principal can be a client, a program, or a service. For each service a client needs, a server generates a ticket for that client / server session. Each ticket contains a number of components: client, server, address, time stamp, lifetime, key (c/s), and key (s). Kerberos is a published standard and is a true single sign-on technology--user logs in once and gains access to all pre-authorized resources without requiring a new or re-entered password. Kerberos is in use in many environments, namely North American colleges and universities. With Windows 2000 using Kerberos v5 as its default network authentication protocol, Kerberos may now become mainstream, albeit Microsoft's version of mainstream.

For many, especially in the enterprise, the idea of single sign-on is a network panacea. But given efforts by standards groups (GSS-API and CDSA) and individual companies like Novell (NICI), Microsoft (SSPI and CryptoAPI), and Sun (Java 2), it appears it will be some time before homogenous authentication will be a reality.

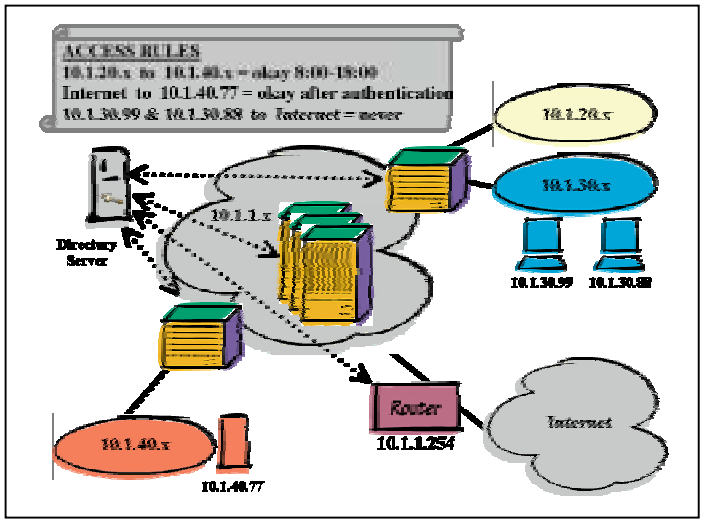
Authorization

Authorization is the granting of privileges associated with an authenticated user, device, or host. The traditional way authorization is granted is exemplified by common operating systems like UNIX. A super-user is the all-powerful system owner or administrator. That individual has the authority to grant privileges to other users. These privileges can be "read," "write," "append," "lock," "execute," and "save"-- individually or in combination. Less traditional although analogous are the privileges granted to network administrators--those that manage the network infrastructure.

A network is comprised of many different devices--from host machines to application, file, web, DNS, and communication servers; from remote access servers to hubs and workgroup switches on the edge; and from WAN-oriented routers to LAN- or ATM-oriented core switches and routers. There is a growing need to grant privileges to these systems on a need-to-know basis.

In order to permit this, the network devices must be able to support a ***provisioned management structure***. The privileges that can be granted can be broken down into devices, services, and configuration parameters.

Device access security is analogous to tradition access control list or firewall rules. The security administrator creates specific rules that limits access to the network devices based on characteristics of the device requesting access, for example, source and / or destination IP address or source MAC address. This traditional access control concept keeps all of the authorization on the device itself. The provisioned management structure proposed here ties on-device authentication and authorization rules to an external directory server. Access is granted provided the policy allows for it.



For example, an IP source (host or network) attempts to access an IP destination (host or network). The network device recognizes a policy exists for this request--it has a matching rule. It queries the directory to determine what to do with it. The appropriate policy is returned to the device and implemented accordingly. Besides this implicit application, the policy could also be associated with explicit information like time-of-day or -month.

Services-based access involves management protocols like telnet, FTP, TFTP, HTTP, and SNMP. Much like what is listed above for device security, it may be prudent to allow certain users access only to specific services based on pre-defined policies. An example of how privileges can be allocated within a group of administrators is shown below.

Service Privileges

Name	User	HTTP	Telnet	FTP	TFTP	SMTP	Console	Custom
Robert Ivins	X	X	X	X	X	X	X	X
Julio Lopez	X	X	X	X	X	X	X	
Alain Chadoin	X	X	X	X	X		X	
Steve Allison	X					X		
Barb Wheeler	X	X	X	X	X			
Marie Roth	X					X		
Jay Ng	X	X	X	X	X		X	

Configuration parameters are the tasks administrators are allowed to perform once they have been authenticated and granted access to the device. Similar to what is described above for service privileges, policy may exist that only allocates management privileges to certain individuals based on job descriptions. For example, policy may dictate that only the super user and the two security managers have the ability to add, remove, or change user security profiles or privileges.

Configuration Privileges

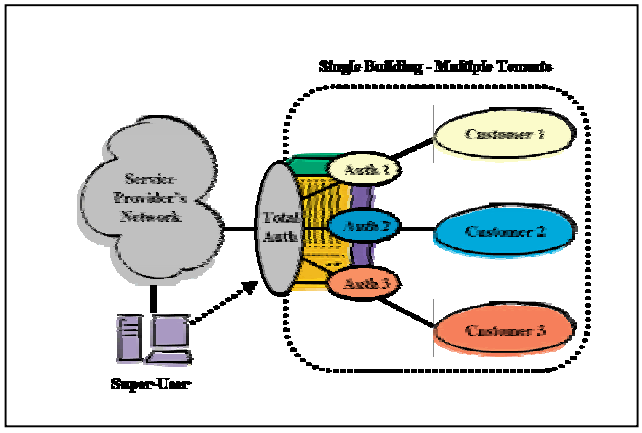
Name	Super User	Security	Routing	VLAN	WAN	QOS	ATM	Custom
Robert Ivins	X	X	X	X	X	X	X	X
Julio Lopez		X						X
Alain Chadoin					X	X	X	
Steve Allison			X	X				
Barb Wheeler		X	X	X	X	X	X	
Marie Roth					X		X	
Jay Ng			X	X				

These configuration submenus or individual command privileges are allocated and stored in a common directory. These ***access accounts*** can contain both implicit and explicit rules ranging from device identifiers, network IDs, TCP or UDP ports, as well a configuration submenu command (CLI) privileges.

There are significant reasons for this fine-grained provisioning profile model. Most network devices have a combination of read-only and read / write privileges. In many cases, especially on tightly controlled enterprise networks, this is adequate. But as networks become more complex and as autonomous systems begin to share the same devices, there is a need to segment the administrative privileges into groups. This is magnified when network provisioning and management is out-sourced.

Many organizations look to external resources for assistance at managing their network edges and access. Managed services are a multi-billion dollar (U.S.) business. Managed service providers are offering high-speed access to corporate Intranets, connecting common business partners via Extranet designs, or providing direct connection to the Internet to multiple tenants from a single device.

In the case of multi-tenant network access, the service provider may want to give each customer some basic troubleshooting capabilities, but for their subnetwork only. One tenant should not be able to see anything relative to anyone else's network. In fact, they should have no idea that other tenants are sharing the same local access device. In addition, the service provider may not want to give its own employees free access to the device. For example, it may be proper to give most of the operations team read-only, routing, and VLAN configuration privileges. Other policy may only give the privilege of changing QOS parameters to a few individuals.



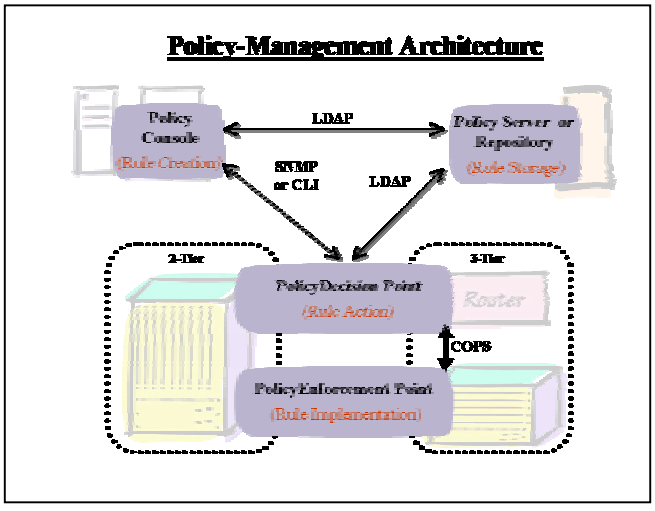
In order for authorization policies to be disseminated to the network devices, the network must be able to support a central repository for these policies. The network devices must be able to access those policies based on some event or pre-provisioned rule. The network can then decide what to do with the event in question. The device must also have the ability to enforce the policy. This policy deployment scenario is referred to as policy-based network management.

Policy Management

The power of this provisioned management structure is magnified when authentication and authorization are coupled with a centralized directory or policy server. Conceptually, when an administrator authenticates to the network, he / she is granted the ability to access all of the devices, services, and configuration parameters he / she has be pre-authorized to access. Each time the administrator attempts to access a network device, that device will query the policy server. The policy server will send an acknowledgement to the device granting authorizations for the requested service.

Policy-based network management leverages directories, the central repositories for policies. This is done for a very good reason. Instead of configuring each device with specific privileges, the devices consult the central directory for this information. This simplifies administration--instead of changing authentication and authorization information on dozens or hundreds of devices, it is done at a central location.

Policy-based management implementations, that leverage directory and policy servers, are offered by many vendors including Alcatel, Cisco, Lucent, and Nortel. All share a common design. They are all based on the concept of a policy console, a policy server or repository, a policy decision point (PDP), and a policy enforcement point (PEP).

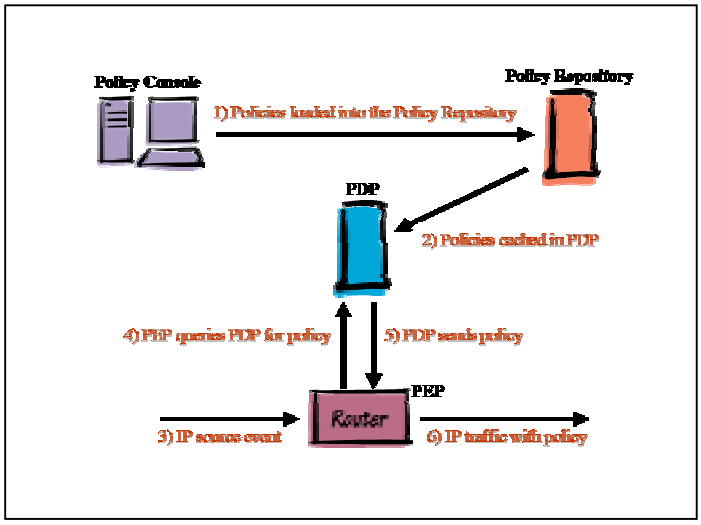


Policies can be recalled via some triggered event or it can be provisioned. In the case of the former, an event can be the arrival of a frame that the network device is unsure how to treat. For example, an IP source or destination address, MAC address, or IP multicast membership record can be the trigger. If the network device has no cached policy for that event, it must query the PDP. The PDP receives its policies from the directory server which are configured and stored there by a policy administrator via policy console downloads. The PDP informs the network device--which is the PEP--to follow specific policy instructions. The PEP implements the policy for that frame and related session flow.

These policy management architectures are either a two-tiered or three-tiered design. A two-tiered method combines the PDP and PEP in the same network device. The three-tiered method has the PDP and PEP running in separate devices. The protocols used to communicate policies will depend on the newness of the products. For example, in newer gear, a separate PDP communicates to the PEP via the Common Open Policy Service (COPS) protocol. In an integrated PDP / PEP, the policy is communicated from the policy repository via LDAP. In older networking gear, the policy communication may be SNMP or CLI.

When the questions of availability and scalability are asked, the provisioned device management structure provides a positive response. Depending on the value placed on the network and its availability, repositories and servers can be redundantly implemented. In addition, based on the number of devices that will be accessing policies and the volume of policy decisions that will need to be made, scalability can be designed into the implementation.

An example of how a policy-based management implementation works is presented below. In this scenario, the triggered event (#3) is an IP source address. The router has a rule (ACL) that states it must check with the policy server (#4) in order to know how (or if) it should be forwarded. The PDP compares the request with the policy (obtained previously in #1 and #2). Once it knows this information, it informs the router how the policy should be enforced (#5). The traffic is then forwarded based on the policy (#6).



The most effective manner this provisioned management structure can operate in is when the policy server, PDP and / or PEP, understand the concept of ***"state."*** State is best described as an awareness of network communications and the rules that are regulating it. State tables contain information like who logged on, when, and which resources are being accessed. Few policy or directory protocols understand the concept of state (COPS however does). For wide-scale usage, maintaining authentication and authorization state is a pre-requisite.

Policy Security

Communications between the policy console, policy repository, PDP and PEP must address security. It is becoming unacceptable practice to communicate device configuration profiles and parameters across the network in clear text. Adequate technology exists to allow this communication to be secured.

Secure Socket Layer (SSL) and its cousin ***Transport Layer Security (TLS)*** are widely used transport protocols that, when couple with public-key cryptography, provide a secure communications tunnel between clients and servers or between network devices. However, there is no assurance the user behind the client computer is an authenticated user.

Simple Authentication and Security Layer (SASL) is a standards-based, general authentication mechanism which can be used by any connection-oriented protocol, like SNMP, LDAP, and S/KEY. ***Digest Authentication*** is also a SASL mechanism used to secure HTTP communications, albeit less secure than others like SSL.

The best method for secure communications to / from / between the devices that make up the network infrastructure will be a fully-implemented ***Public Key Infrastructure (PKI)*** based on X.509 authentication foundations and a standards-based family of encryption capabilities like RSA's PKCS. The issue with this model is it relies too heavily on a single vendor, RSA. However, because the RSA protocols are platform independent and considered technically sound, their appeal is wide. There is plenty of activity by other vendors to attempt to standardize PKI, without forcing vendors and the end users to pay RSA fees.

Conclusion

How much security is enough? How much is not enough? This proposal about how one can use a provisioned management structure for the network infrastructure is only useful to the organization that understands the value of its network and the information contained therein. For many, this management model is overkill. For others, it is well suited. Whatever the desire, organizations must understand the value of their networks and calculate the cost to the business if the network were unavailable.

The result of this assessment should be a corporate security policy document. This document will be the plan that a company will follow for all its security issues. It will clearly spell out the business values (strengths) and weaknesses (vulnerabilities). It will delineate what is important and what is not. From this, the corporate security budget, procedures, technologies, actions, and awareness programs can be deployed. Hopefully, requiring secure access to the network infrastructure will be part of the corporate information security agenda.

References

Enterprise Security, M. Kabay; 1996.

The Burton Group; various *Network Strategy Reports*; 1999.