Assignment Eight – AAA and TACACS+

CITA370

Brian Dwyer

Morrisville State College

Brian Dwyer
CITA370
2/1/2010
Assignment Eight: AAA and TACACS+

When designing a network, one aspect which must be considered is security design. It is difficult to discuss network security design thoroughly since network security design goals are driven by organizational requirements. However, all security technology seeks to achieve the same goals; confidentiality, integrity and availability. When designing a network, perimeter security is a vital consideration. Part of perimeter security is the authentication and authorization of users attempting to gain network access, whether for administrative purposes or otherwise.

Network management architecture can be broken into five components. The five components together are often referred to as FCAPS, or Fault, Configuration, Accounting, Performance and Security management. FCAPS is an ISO network management standard. Each of these elements of network management performs a unique function and is reliant upon specific technologies. Fault management deals with events & alarms, problem identification, troubleshooting, problem resolution and fault logging. Configuration management deals with change control, inventory of hardware and software, software deployment and configuration information. Accounting management deals with device and service usage tracking, billing and/or invoicing, resource and asset management, as well as cost control. Performance management deals with capacity planning, availability, response time, throughput and utilization. Security management handles policy, authority & authentication, access level, data integrity and logging.

Several technology models attempt to address all elements of FCAPS in a single solution. One such model utilized in large-scale networks is the Telecommunications Management

Network (TMN) model. This model breaks a network down into five layers; business management, service management, network management, element management and the element layer. The element layer is composed of all physical network infrastructure components including switches, routers, servers, storage devices, terminals, firewalls and printers. The second layer is the element management layer. This layer manages the individual elements in an attempt to detect equipment errors, measure environmental variables (temperature, power consumption, etc.), measure resource utilization, the logging of statistical data and the firmware update process. The third layer is the network management layer which serves to create a logical network map, create dedicated paths through the network using QoS, modify routing tables, monitor link utilization, optimize network performance and detect network faults. The fourth layer is the service management layer which handles network services and users. This layer performs QoS management, accounting, user account management, IP address assignment and maintenance of multicast group addresses. The fifth layer is the business management layer which handles functions related to the business process. At this layer, policies are defined and data from the network and service management layers are utilized to analyze trends and control quality of network service. (Javvin Technologies, Inc., 2007)

The TMN management model paints a picture of how large organizations manage their networks. Smaller organizations may have different needs or cannot afford a comprehensive solution. Smaller organizations take the aspects they consider most important to their operation and address only them, usually for reason of cost. One commonly implemented solution required by most all organizations is a form of identity management. Identity management coincides with a process model called AAA or Authentication, Authorization and Accounting. Authentication serves to verify a user or device. Authorization grants access according to privileges and policies. Accounting tracks usage for billing, planning and auditing which may be used to evaluate system usage or for forensic analysis. Many government regulations require the implementation of technology auditing, therefore the implementation of AAA is seen in most all organizations in some shape or form. For example, the Sarbanes-Oxley Act of 2002 requires event retention information for a minimum of one year. Although most of these regulations focus more on the financial aspects of business operations, the responsibility of the auditing and logging often lands on the desk of IT personnel. Some regulations are more geared toward the IT field. For example, the Payment Card Industry Data Security Standard blatantly states the requirement to audit network access. "Requirement 10: Track and monitor all access to network resources and cardholder data. Logging mechanisms and the ability to track user activities…" (PCI Security Standards Council, 2010)

Ensuring regulatory compliance is a vital aspect of network design. There are many different regulations with many requirements, but for the purposes of this paper I will be sticking to how AAA services can help with auditing regulations and how AAA technology works. Authentication is the primary function of AAA. Without authentication, you cannot have authorization, nor can you have accounting. If you don’t know who someone is, you cannot assign permissions without identification. Likewise, if you do not know who someone is, you cannot provide enough accounting information to generate a valid audit report. For this reason, the authorization and accounting functions of AAA are reliant upon that first element of authentication. Collectively, the three functions of AAA provide a robust means to control who can access your network, limit what they can do, and also provide a logging mechanism for all user interactions with the system. The purpose of regulatory compliance is to ensure the proper measures are in place so that proper legal documentation can be provided with regard to system access. You must be able to prove in court who was logged onto what system and what it is that they did. This is especially prevalent in the financial sector where the utmost concern is placed upon the integrity of financial data. If someone changed some numbers around, a financial organization needs to know who did what, and when. In addition to protecting organizational data, the integrity of systems configuration data is a concern.

Network devices and their configurations lie at the heart of network operations. It is a good rule of thumb that if someone can obtain physical access to a device, they can gain control of it. Likewise, if someone attempts to gain remote access to a network device, it is important that the devices be configured in a secure manner and only authorized users are permitted access. This includes the usage of strong passwords and having unnecessary services turned off. All network infrastructure devices, perimeter routers in particular need to have administrative access protected. Unauthorized access to these devices, even accidental could bring network operations to a screeching halt. For this reason, it is important to utilize AAA functions on these devices whenever possible. Cisco network equipment supports the usage of AAA functions at both the local and remote level. This means that the devices can be configured to maintain a AAA database locally, or they can be configured to access a trusted remote database. For small operations with a low number of network devices, it would probably be more cost effective to utilize local AAA functions than to utilize a remote database. On Cisco devices, "AAA services provide a higher degree of scalability than the line-level and privileged EXEC authentication commands alone." (Paquet, 2009) By utilizing AAA functionality, you can limit user to running only commands which they are authorized to run. In addition, the username, time/date of logon and what they did are recorded by the accounting function.

Running AAA services locally presents a problem in a large enterprise with many devices in many locations. Imagine if a network administrator left the company on bad terms and they knew all of the passwords to your organizations infrastructure devices. If running AAA services locally, someone would have to change the username and password databases on each device individually. The utilization of centralized AAA services allows for a single change to a central database to be reflected throughout the network. In this situation, each network device polls the AAA server with user-supplied authentication credentials to determine validity. The AAA server then replies with authorization information which can range from deny access to allow full control. By centralizing the AAA function, we can better secure access to network devices, allow easier management of user accounts, and also allow more granular access to devices – even specific commands. In addition to improving manageability, we also improve scalability. New devices can be configured to authenticate to the AAA server as needed. There is no need to provision and populate each device with a user database manually. This allows for network build-outs to be completed faster due to the standing presence of AAA infrastructure.

Cisco devices primarily support two AAA protocols, TACACS+ and RADIUS. These protocols both provide similar AAA functionality, although there is a place and time for each. They have distinct differences which characterize exactly why each one is suited to a particular function. Both protocols are able to authenticate access to data link and network layers, as well as network infrastructure device virtual terminals. For the purposes of this paper, I will characterize TACACS+ and describe RADIUS by method of contrast.

TACACS+ stands for Terminal Access Controller Access Control System Plus. TACACS+ has two predecessors, TACACS and XTACACS. “The original Cisco TACACS was modeled after the original Defense Data Network (DDN) application.” (Knipp, et al., 2002) It was developed in the 1980’s for the DDN by MILNET developers. This original version of TACACS utilized communication over UDP port 49. In addition, this version only supported authentication, not authorization or accounting. In 1993, IETF RFC 1492 established the TACACS protocol as “An Access Control Protocol, Sometimes Called TACACS”. (Finseth, 1993) This protocol was also known as XTACACS, or Extended TACACS. This protocol extended upon the original TACACS protocol by including the element of accounting in addition to authentication. XTACACS also utilized UDP port 49 for communication. XTACACS set the precedent for TACACS+ by separating the authorization and accounting functions; they function separately in that the XTACACS server can utilize a separate database for each function. Neither TACACS nor XTACACS saw much usage, but the advent of Cisco’s TACACS+ brought about large-scale deployments and widespread industry penetration.

TACACS+ is a major enhancement to the TACACS protocol family. Its primary strength is the ability to perform all three AAA functions. TACACS+ brought about the ability to authorize users in addition to authentication and accounting. TACACS+ authorization allows granular authorization functionality including 16 levels of user permissions. TACACS+ is also much more reliable and secure than its two predecessors. TACACS+ utilizes Transmission Control Protocol (TCP), meaning that TACACS+ is a connection-oriented protocol. TCP is more reliable than User Datagram Protocol (UDP) which is sometimes referred to as “unreliable” datagram protocol. In addition to utilizing TCP, “TACACS+ can also encrypt the body of traffic traveling between the TACACS+ server and client. Only the TACACS+ header is left unencrypted.” (Knipp, et al., 2002) TACACS+ also allows for increased modularity by permitting individual AAA element configurations, in that each AAA function can utilize a separate database. These three functions combined made TACACS+ all the more appealing to network designers looking to centralize their AAA functions. The usage of TCP, encryption and authorization levels make TACACS+ a good candidate for AAA for administrative networks and network infrastructure device access.

Earlier, it was stated that the goal of security is confidentiality, integrity and availability. TACACS+ reinforces each of these ideals. TACACS+ delivers confidentiality via encryption, integrity by delivering AAA, and availability by leveraging TCP's reliability and the support of database replication. The encryption of TACACS+ requires each AAA client to be registered in a client database on the server and also requires a secret key to be identically configured for each client on both the server side and client side. This secret key is then used in a "combination of a hashing algorithm and an XOR function" to establish secure channel between server and client. "TACACS+ uses MD5 to hash using a secret key provided on both ends." (Carroll, 2004)

TACACS+ handles each element of AAA separately. First, TACACS+ authenticates each user to verify their identity. Second, TACACS+ performs authentication and determines what functions the user is entitled to use. Finally, TACACS+ performs accounting by keeping record of what actions were performed. I would like to reiterate that the authorization and accounting functions are reliant upon the authentication function; Authentication is valid without authorization or accounting but authorization and accounting are not valid without authentication.

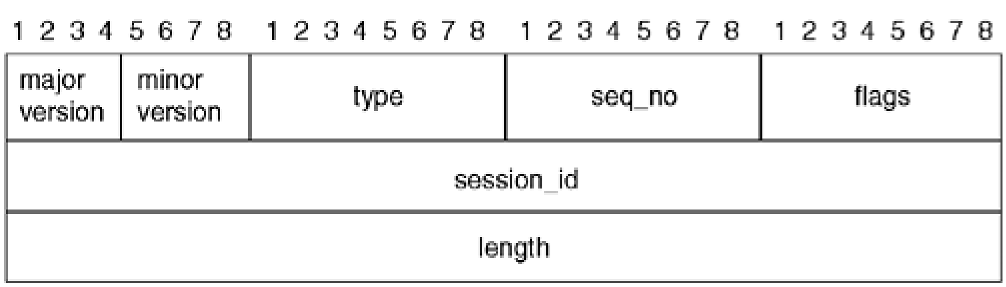

Figure 1 - TACACS+ Packet Header (*Cisco Systems*)

Since each of the three functions of AAA are performed separately, there are three packet types utilized by TACACS+. The header portion of a TACACS+ packet is not encrypted and the header contains identifying information about the transmission. This information is the *major version*, *minor version*, *type*, *sequence number*, *flags*, *session ID* and *length*. The ***major version*** is usually seen statically in the form of `TAC_PLUS_MAJOR_VER = 0xc`.
The ***minor version*** field allows for backwards compatibility with new revisions to the TACACS+ protocol. This allows the server or client to determine identify the capabilities of each other and adjust accordingly. The default value is `TAC_PLUS_MINOR_VER_DEFAULT=0x0`.
The ***type*** field determines the type of AAA function contained in the packet. There are only three valid ***type*** values, `TAC_PLUS_AUTHEN=0x01` for authentication, `TAC_PLUS_AUTHOR=0x02` for authorization, and `TAC_PLUS_ACCT=0x03` for accounting. These type values are what allow TACACS+ to differentiate between AAA functions and perform them as individual functions. The ***sequence number*** field indicates the order a packet falls in a transmission. Sequence number one always belongs to the initial transmission from the client to server requesting authentication. In addition, packets traveling from client to server will always be odd and server to client will be even because of this. It is also important to note that "the highest sequence number that can be reached is 2^8-1. After this value is reached, the session that is established between the AAA client and the AAA server is reset, and a new session is started." (Carroll, 2004) The ***flags*** field basically allows for TACACS+ options to be specified. TACACS+ currently supports two flag values, `TAC_PLUS_UNENCRYPTED_FLAG` and `TAC_PLUS_SINGLE_CONNECT_FLAG`. The first indicates whether encryption is turned on or not. This flag should only be set for debugging purposes as it disables encryption. The second indicates whether multiple TACACS+ sessions are supported over a single TCP session. This

value is set based upon the first two TACACS+ packets exchanged between client and server. The field ***session ID*** determines session ID and the ***length*** field specifies the overall length of the TACACS+ packet.

The TACACS+ authentication process is relatively simple and the data exchange between client and server falls into three packet types; *start*, *reply* and *continue*. A ***start*** packet is the first packet sent by the client to the server upon a user attempt to authenticate. Upon reception of this message, the server validates the client request and sends a ***reply*** packet asking the client for the username. Upon reception of the ***reply*** packet, the client sends a ***continue*** packet containing the user-supplied username. The TACACS+ server then responds with another ***reply*** packet, this time asking for a password. The client then sends another ***continue*** packet containing the password. Upon reception, the TACACS+ server checks the supplied credentials against its own database, then proceeds to check external databases if the account is not found. After determining the validity of the username & password combination, the server sends a ***reply*** with one of four values; *accept*, *reject*, *error* or *continue*. ***Accept*** means the supplied credentials are valid and if no authorization is required, the user is granted access. ***Reject*** means the credentials are invalid or were not found. ***Error*** means that a network error may have occurred or the TACACS+ daemon had a problem in its processing of the request. ***Continue*** responses are requests for more authentication information from the user.

The TACACS+ authorization process is a bit more complicated, only because there are an increased number of possible server responses. The authorization process only involves two message types, *request* and *response*. ***Requests*** are sent by the client to the server, and ***response*** is then sent by the server back to the client. The requests are simply requests for authorization, but the response messages contain an Attribute-Value (AV) which can be used for configuration settings and/or permissions. For authorization of administrative access to the command line of network devices, the AV which specifies the authorized privilege level is `priv-lvl=X` where X equals the privilege level from 0-15 for a total of 16 levels. "There are 16 privilege levels, 0 to 15; level 0 is reserved for the user-level access privileges, levels 1 to 14 are levels you can customize, and level 15 is reserved for enable mode privileges." (Paquet, 2009) It is important to note that the privileged commands must be manually configured on the router if you choose to use this feature. Some other notable AV's related to command authorization are `cmd=X` and `cmd-arg=X`. By enabling command authorization on the local device, the device will submit the user-requested command and/or its arguments to the TACACS+ server for verification of authorization. This is easier than manually configuring privilege levels on each device and assigning commands to privilege levels. It is also important to note that command authorization does not apply to console requests unless manually configured to do so.

The TACACS+ accounting process is similar to the authorization process in that a record of an event is sent to the TACACS+ server in addition to an AV pair associated with accounting. Requests and responses are also utilized. Only three types of requests are associated with the TACACS+ accounting process, *start record*, *stop record* and *continue record*. The ***start record*** is sent from the client device to the server when an event begins and includes any authorization information utilized, as well as the identity from the authentication process. A ***stop record*** is sent when an event ends, such as a user logging off or leaving privileged exec mode. This also includes the authorization and authentication information like the previous record type. A ***continue record***, also known as a watchdog record can be sent in between the start and stop of an event. This basically provides incremental updates until a stop record is sent. Upon reception of a record, the TACACS+ server, the server sends one of three possible responses. It can send a *success*, *error* or *follow*. A ***success*** message is sent when the server successfully recorded the request. An ***error*** response is sent when the server failed to record the request. A ***follow*** response is sent when the server wishes for the client to utilize a different server for the request. This follow response also includes the IP information for the server to use.

With a clear understanding of how TACACS+ works, contrasting it to RADIUS is easy. They are both AAA protocols and perform AAA functions but do so in a different manner. TACACS+ utilizes TCP port 49, encrypts the data payload of each packet and allows separation of AAA services individually. TACACS+ is also a Cisco proprietary protocol, but it is supported on many devices made by other manufacturers such as HP ProCurve switches. In addition, TACACS+ allows for command authorization and it was designed for device management. RADIUS utilizes UDP ports 1645 and 1812 for authentication and authorization and 1646 and 1813 for accounting. RADIUS is an IETF-standardized protocol and does not encrypt its data payload, but only passwords which are limited to 16 bytes. Authentication and authorization are also combined as a single service under RADIUS. Radius was intended for user access control and “typically provides more complete accounting capabilities than TACACS+.” (Carroll, 2004)

Many environments utilize both RADIUS and TACACS+, with RADIUS for general user authentication and TACACS+ for network administration functions. Even Cisco realizes this as Cisco’s SecureACS houses both a TACACS+ and RADIUS daemon on a single server. RADIUS provides more complete accounting capabilities because vendors can create their own authentication AV’s for usage in addition to the already robust list of over 100 standard, non-proprietary AV’s reserved in the RADIUS IETF standard. TACACS+ provides more secure, reliable communication between client and server.

Works Cited

Angelescu, S., & Swerczek, A. (2010). *CCNA Certification All-In-One For Dummies.* Indianapolis, IN: Wiley Publishing, Inc.

Carroll, B. (2004). *Cisco Access Control Security: AAA Administration Services.* Indianapolis, IN: Cisco Press.

Doyle, J., & Carroll, J. (2006). *CCIE Professional Development - Routing TCP/IP, Volume 1, 2nd Edition.* Indianapolis, IN: Cisco Press.

Finseth, C. (1993, July). *An Access Control Protocol, Sometimes Called TACACS.* Retrieved February 20, 2011, from IETF: http://tools.ietf.org/rfc/rfc1492.txt

Hucaby, D., McQuerry, S., & Whitaker, A. (2010). *Cisco Router Configuration Handbook - Second Edition.* Chicago, IL: Cisco Press.

Javvin Technologies, Inc. (2007). Network Management Architecture and Technologies Map.

Knipp, E., Browne, B., Weaver, W., Baumrucker, C. T., Chaffin, L., Caesar, J., et al. (2002). *Managing Cisco Network Security - 2nd Edition.* Rockland, MA: Syngress.

Paquet, C. (2009). *Implementing Cisco IOS Network Security (IINS).* Indianapolis, IN: Cisco Press.

PCI Security Standards Council. (2010, October). *Data Security Standard Requirements and Security Assessment Procedures.* Retrieved February 14, 2011, from Payment Card Industry (PCI): https://www.pcisecuritystandards.org/documents/pci_dss_v2.pdf