Global Information Assurance Certification Paper

SANS SECURITY ESSENTIALS GSEC PRACTICAL ASSIGNMENT
Version 1
Author: Peter Huckins

Vulnerabilities of Router Management Utilizing TACACS+ for Authentication, Authorization, and Accounting (AAA).

What is the history of TACACS?

TACACS has its origin in the military community ARPANET (Advanced Research Projects Agency Network) and documentation can be found relating to this matter in RFC 1492. Networks are getting larger all the time and as might be expected, many of these administrative dilemmas were first recognized and addressed by the military community. BBN made the first versions of TACACS for MILNET and or ARPANET. At that time TACACS primarily was a UDP access based protocol that orchestrated user access. Since then, Cisco has issued a propriety version of TACACS and thereby keeping some propriety control of the protocol. Cisco owns XTACACS and the latest revision TACACS+. XTACACS, released in 1990, by Cisco was named as such since it included extensions for additional features and options to the original TACACS protocol. Later, circa 1998, the TACACS+ version 1.78 supported by IOS from11.2 on allow for the separation of the 3 control features of TACACS, namely Authentication, Authorization, and Accounting to be used on respectively independent servers. TACACS+ surpasses TACACS and XTACACS and furthermore is not compatible with its 2 predecessors which are considered end of maintenance by Cisco and should therefore probably not be considered for implementations.

Why is TACACS needed?

Access control has long been an issue to be reckoned with for Administrators and users alike. As networks emerge and converge, administrators increasingly need flexibility to determine and control access to resources under their care. Administrators have been increasingly faced with new situations pertaining to access. Remote users need to access their LANs just as though they were sitting at their desk at work. This creates a significant need for an administrator to be able to effectively and flexibly give those users seamless access yet be able to provide for security and user resource accountability. Also, within the commonly maintained network, different administrators have varying responsibilities that require varying levels of access privileges (authority).

TACACS may or may not be a necessary security scheme to the administrator. If only a few access servers are managed, then the need for uniformity of policy pertaining to the issues of authentication, authorization, and accounting is diminished. However, (and for the sake of this research paper) we are under the assumption that several access servers are in use and that users of varying levels of privilege are attempting to utilize the network via a Network Access Server.

Keeping the security data of users, passwords, and privileges centralized tends simplify the maintenance of that data. This is under the premise that it is easier to update

centralized records as opposed to updating a multitude of Network Access Servers, e.g. on an occasion when an employee users needs a password change. Further, centralization provides for uniformity of access policies on all the servers depending on the overall security policy of the company.

What is TACACS+ ? (Terminal Access Controller Access Control System+ [protocol])

TACACS+ is the third generation of Terminal Access Control, which is a Cisco proprietary client/server protocol. TACACS+ uses TCP (transmission control protocol). Its use originates from the need to manage and control terminal access. It's functions are based on classic server/client relationship using request and response to determine in an algorithm format whether or not users are authenticated, authorized, and to record each user's actions as needed (accounting). The requests are being sent to TACACS+ server on which the TACACS+ daemon is running. Responses are sent to the NAS where privileges need authentication and actions need authorization and the whole shooting mach can be accounted for if need be. The client sends a request and the server sends a response.

The real control facet of TACACS+ is determined by who is authorized to run and edit the TACACS+ daemon, usually the super user in a UNIX OS.

How does TACACS+ work ?

To accomplish goals of the Administrator's security policy, TACACS+ may use any or all of the 3 arenas of Authentication, Authorization, and or Accounting. Fundamentally the protocol achieves this through TCP packets and encryption (MD5). A vulnerability to this is that integrity checking in not within the protocol. Integrity checking is not necessary but if the correct sequence number were acquired, then bit flipping within the data packet could become possible. The sequences and exchanges of these packets between the TACACS+ server (usually a UNIX server) and the NAS (a router, in this instance) are further broken down into what can be referred to as "sessions". TACACS+ uses a "session" numbers 1,2, or 3 to denote the differences between the AAAs. The sessions can be further reduced to simple TCP packets and their related components that make up the packet. Depending on which of the AAA sessions is in progress the packet structure is appropriately determined. Some of the components of a packet header are the protocol version, type, sequence number, flags, session ID, and the packet length. The session ID number is cryptographically determined and if enough servers reboot and or there is adequate amount of packet traffic with identical session ID #s then this could become a vulnerability. Networks that pass vast quantities of traffic would therefore increase the probability that the randomness of the session ID # would decrease. The session number is the key determinant of which of the AAA sessions is taking place.

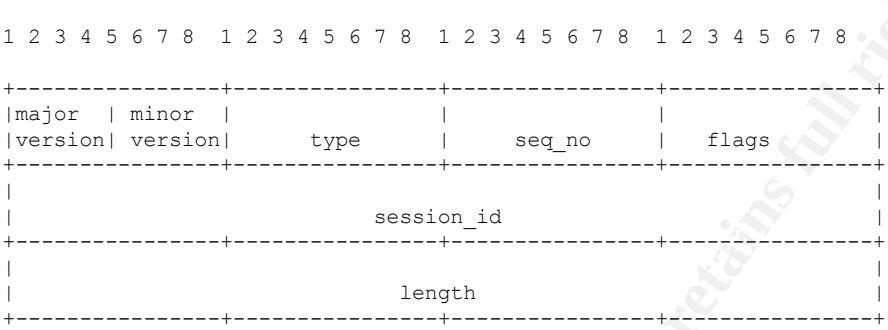
```
TAC_PLUS_AUTHEN := 0x01 (Authentication)

TAC_PLUS_AUTHOR := 0x02 (Authorization)

TAC_PLUS_ACCT   := 0x03 (Accounting)
```

There is some commonality in the packets irregardless of the session and is of a security concern, chiefly that all of the packets have a packet header, which is not encrypted. Only the body of the packet is MD5 encrypted.

Example of packet Header:



Authentication

In regards to TACACS+ and Authentication, the first A of AAA, Authentication determines whether or not a user has access to a network or resource. In TACACS+, the method of authentication is by a user inputting a unique password that of course must match the unique user ID. TACACS+ will support ASCII, PAP, CHAP, MS-CHAP, and ARAP authentication schemes. There can be two modes for the password exchange: a one-time password, which is only valid for one use by a particular user or a multi use password which can be valid for x number of sessions or a longer length of time. A multi-use password is more along the lines of user and password database (hopefully with reasonable password aging). Older Versions of TACACS prior to TACACS+ ver 1.78 used UDP to send authentication request and confirmation. This is not as reliable as connection based transmissions like TCP. TACACS+ listens for login request on TCP port 49, but this port is not required and another may be substituted as long as the client and server are both configured.

For TACACS+, authentication follows the similar sequence of TCP hand-shake. The packet sequence session always starts from the client. The TACACS+ server will not initiate a session. Like TCP's send , syn-ack, ack, TACACS+ sends Start, Continue, and Reply. The TACACS+ will always send the replies. Successful Authentication occurs when the Started packet or the Continue packet receives the Reply packet that indicates the login and password criteria are matches for the TACACS+ db's criteria. A vulnerability here could be that if a packet with a known sequence ID and a session ID were sent to a TACACS+ server, then the server would respond appropriately. The encryption could then be broken since the known clear text would snare the appropriate encrypted packet from the server. This is referred to as a frequency analysis attack.

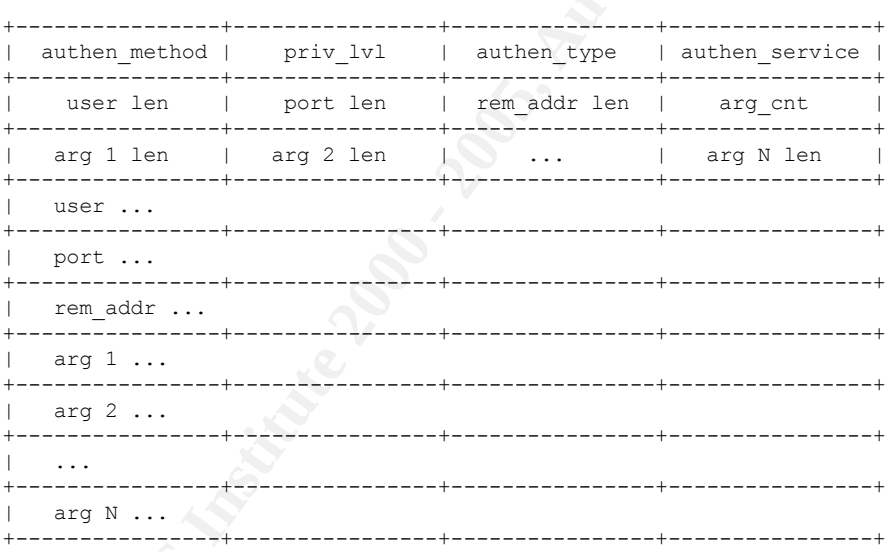
The Authentication packets for TACACS+ will include the following information components: action, priv_lvl, authen_type, service, user len, port len, rem_addr len, data len, user, port, rem_addr, and of course data. These are listed here primarily as knowledgeable reference and would be usually be transparent to the user since those

packet components are of the packet level. This same sequence and process is repeated when authentication for enable mode is requested.

Authorization

Authorization is the process by which it is determined what privileges a given user may or may not use and can be conditional as well. E.g. said users must issue a specific command (a privilege) so long as they are at a specific IP address. Authorization follows a simpler exchange than the Authentication process but has more options. There are many options under Authorization since one of the goals of the TACACS+ protocol is to give administrators, and thereby the security policy, granularity in determination of users privileges. The descriptors in this case are values that are conveyed from the TACACS+ to the client when requested. They are called AVP (attribute value pairs). For the most part these AVPs are present only in within the content of the packet's header and body.

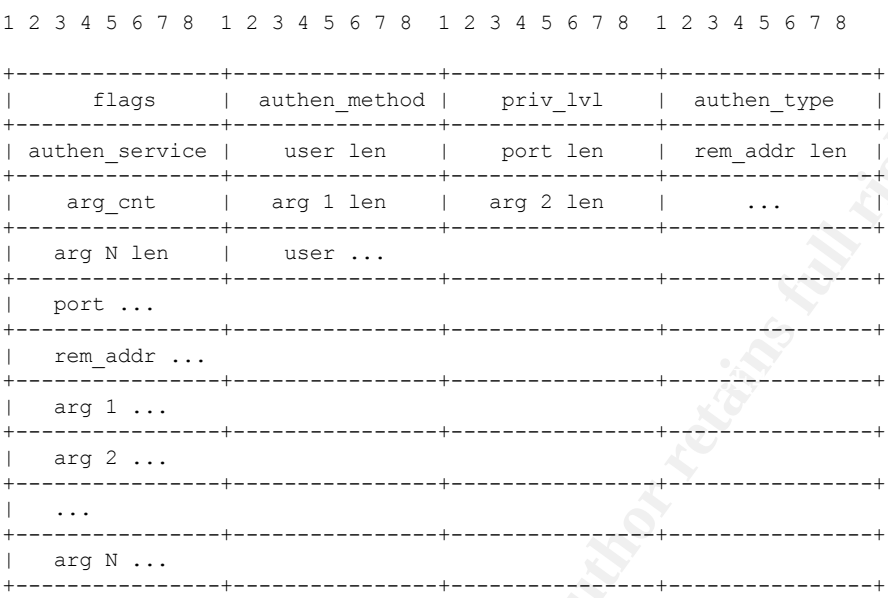
Example of Authorization Request Packet:



Accounting

TACACS+ allows a super user administrator to track a user's actions. When users perform certain actions, if accounting is enabled, then if the Administrator reviews the log files, accountability becomes a possibility. This will option will give details of when an action took place, how long this process ran, and of course who initiated the action. On the packet level, this option is similar to Authorization and necessitates additional AVPs to detail when, who, and how long an event occurred. It also will require more space for logging on the TACACS+ server. Like the Authorization packet exchanges, Accounting packets are a one to one exchange. This make the transfer vulnerable to what is referred to as a replay attack. The odds would be good that the correct sequence number could be successfully sent to the server since it is on a 1 to 1 packet exchange.

Example of Accounting Request Packet:



References:

Cisco Support Website "TAC-RFC 1.78" D. Carrel, Lol Grant, January, 1997
ftp://ftpeng.cisco.com/pub/tacacs/tac-rfc.1.78.txt

"TACACS FAQ" Kiessling, Robert, 1998-04-23 V1.03
http://www.de.easynet.net/tacacs-faq/tacacs-faq.html

Cisco Support Website "TACACS+ Freeware for First-time Users"
http://www.cisco.com/warp/customer/480/tacplus.shtml

Cisco Support Website "Single-User Network Access Security TACACS+"
http://www.cisco.com/warp/customer/614/7.html

"TACACS+", Welcher, J. Peter
http://www.mentortech.com/learn/welcher/papers/tacacs.htm

"TACACS+ Protocol Flaws Vulnerabilities", Designer, Solar May 30, 2000
http://www.securityfocus.com/bid/1294