Design and Implementation of RADIUS – An Network Security Protocol

GJCST Computing Classification
C 2.0, D 4.6

Md. Hashmathur Rehman[1] Dr.A. Govardhan[2] T. Venkat Narayana Rao[3]

***Abstract*-RADIUS (Remote Authentication Dial in User Service) is a protocol used for authentication, authorization and accounting of network objects in networking environment. The protocol has set of weaknesses due to its implementation. First the overview presents the basic operation and functioning of RADIUS protocol. Then analysis part focuses on Vulnerability issues such as Security, transport and implementation. Finally, how to minimize or resolve various issues of the RADIUS protocol using deployment best practices and extensions are discussed.**

Keywords-Radius; authentication; authorization; accounting; security; extension, Network Access Server.

I. Introduction

Remote Authentication Dial in User Service (RADIUS) is an AAA (authentication, authorization and accounting) protocol for applications such as network access or IP mobility. It is commonly used for network devices such as routers, modem servers, switches, etc. RADIUS is currently the de-facto standard for remote authentication and accounting. There are many Vulnerability issues with RADIUS; these issues can be viewed as security issues, transport issues and implementation issues. RADIUS consistently provides some level of protection against a sniffing, active attacker, but it indicates that the RADIUS protocol still exist several leaks. It makes up the security shortcomings of RADIUS protocol to a certain extent and makes the RADIUS system meet the requirement of application. The rest of the paper is presented as follows. Section II gives the overview of RADIUS protocol and presents basic security mechanism used by the protocol. Section III analyses the RADIUS protocol, focusing in security, transport and implementation issues. However it does not cover RADIUS protocol‘s accounting functionality. Section IV presents the concrete implementation of the extended RADIUS. Finally, the paper concludes in section V

II. Protocol Overview

A. Basic Information

The newest RADIUS protocol is described in RFC 2865[1], "Remote Authentication Dial-in User Service (RADIUS)," and RFC 2866[2], "RADIUS Accounting". RADIUS protocol is used between two servers. RADIUS server is a shared authentication server that has a list of valid clients. There is a shared secret between the RADIUS server and these clients. This secret cannot be empty, but otherwise it is not defined by the protocol standard how strong it must be. It is only recommended that it is 16 octets minimum and unguessable. This secret is used for to authenticate the RADIUS server to the NAS and to hide the user password. For these purposes the secret is part of value that is hashed and the hash value is sent. [5]. RADIUS server also has a database of users containing their passwords, possible other requirements for these users to gain access and configuration data. According to information in this database the RADIUS server accepts or rejects the request or sends a challenge to user. RADIUS server can also act as a proxy relaying requests to other RADIUS server and to NAS [5], [15]. When acting as proxy RADIUS server relays messages between the NAS and other RADIUS server. There can be many RADIUS servers as proxies between the NAS and the RADIUS server that finally handles the authentication and authorization of the request. Network Access Server (NAS) acts as a client to the RADIUS server. Users call in and NAS prompts for needed authentication information, for example user name and password. The NAS then can use RADIUS server for user authentication. When doing so the NAS sends request to the RADIUS server containing attributes that have information about user that the RADIUS server needs. When sending request containing user password, the password is not sent as clear-text, instead it is encrypted as described in section 4. NAS then waits for reply from the RADIUS server. Server can accept or reject the request or present a challenge for the user to respond. If request is accepted the server can also provide the NAS with configuration data and type of service granted for the user. If RADIUS server does not response in given time, NAS can retransmit the request or it can also use possible alternate RADIUS servers. In figure 1 basic RADIUS system architecture operation is shown in Figure 1 [5], [6], and [9].

RADIUS protocol uses UDP as its means of transport. UDP port assigned for RADIUS protocol is 1812. Previously RADIUS used UDP port 1645, but usage of this port conflicted with datametrics service. Choice for using UDP instead of TCP is mainly for the reason that UDP is lighter protocol than more reliable TCP. RADIUS is a stateless protocol that does not carry much data as maximum size for UDP packet 4096 octets. As RADIUS is used for user authentication, few seconds delay is acceptable. In addition to complete the authentication and authorization does not

About-[1]Cisco Systems(i) Pvt Ltd, Hyderabad A P,India (e-mail: hasmoham@cisco.com)
About-[2]Jawaharlal Nehru Technological University, Hyderabad, A P, India, (e-mail: govardhan_cse@yahoo.co.in)
About-[3]Hyderabad Institute of Technology and Management (HITAM),R.R District , Hyderabad, A P, India (e-mail: tvnrbobby@yahoo.com)

need many RADIUS messages to be sent. Therefore choice for UDP over TCP is justified [7]. [8], [5].

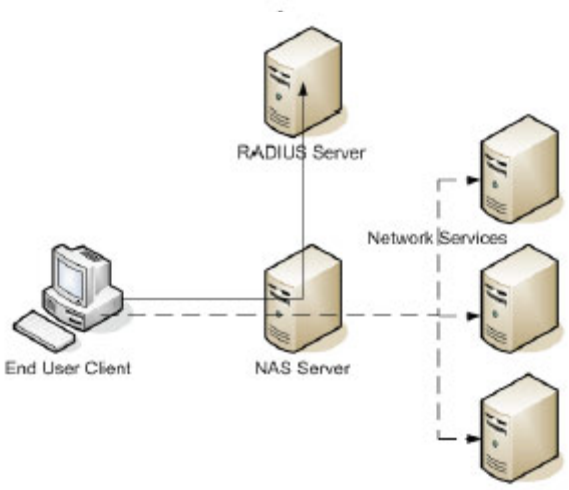

Figure 1. AAA client-to-RADIUS server relationship

References [1] and [2] define the following RADIUS message types: Access-Request, Access-Accept, Access-Reject, Access-Challenge, Accounting-Request, and Accounting- Response. Figure 2 shows a typical sequence diagram of RADIUS protocol when a user accesses the network through NAS and disconnects itself.

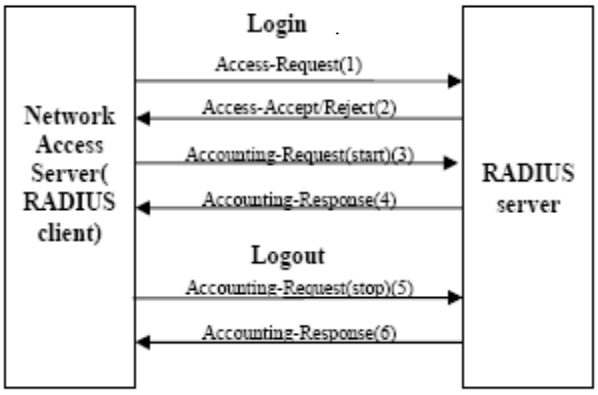

Figure 2. Typical RADIUS sequence diagram.

A RADIUS message consists of a RADIUS header and RADIUS attributes. Each RADIUS attribute specifies a piece of information about the connection attempt and is described by variable length attribute-length-value 3-tuples. RADIUS attributes are described in RFCs, [1], [2], 2867[3], 2868[4], 2869[5], and 3162[6].

A summary of the RADIUS packet is below (from the RFC):

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         Authenticator                         |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Attributes...
+-+-+-+-+-+-+-+-+-+-+-+-
```

The code establishes the type of RADIUS packet. The codes are:

Value	Description	Value	Description
1	Access-Request	11	Access-Challenge
2	Access-Accept	12	Status-Server (experimental)
3	Access-Reject	13	Status-Client (experimental)
4	Accounting-Request	255	Reserved
5	Accounting-Response	----------	

The identifier is a one octet value that allows the RADIUS client to match a RADIUS response with the correct outstanding request. The attributes section is where an arbitrary number of attribute fields are stored [11], [12]. The only pertinent attributes for this discussion are the User-Name and User-Password attributes. This description will

Concentrate on the most common type of RADIUS exchange: An Access-Request involving a username and user password, followed by either an Access-Accept, Access-Reject or a failure. I will refer to the two participants in this protocol as the client and the server. The client is the entity that has authentication information that it wishes to validate. The server is the entity that has access to a

database of authentication information that it can use to validate the client's authentication request.

B. Authentication and Authorization

When NAS wishes to authenticate user via RADIUS, the NAS sends Access-Request packet to RADIUS server. To this packet the NAS set appropriate attributes that describe the needed information about the user and the service required to the RADIUS server. User password in User-Password attribute is sent encrypted and not in clear-text. The NAS also generates unique Request Authenticator for this request and sets Identifier so that the NAS can connect the reply to this request. [10], [13].

Upon receiving this request the RADIUS server checks it's list of valid clients that it has a shared secret with. If the request does not come from a client in this list, the request is not handled and no error message is sent. If the client is valid, the RADIUS server decrypts the user password (if present) and checks it's user database for entry for requesting user and checks whether user passwords match.[6], [5].

If user is not found, passwords do not match or the user is not allowed to specific clients or ports that may be defined listed in user entry, the RADIUS server send Access-Reject packet to the client. If user is found, passwords are equal, user is allowed to access and no challenge/response is needed, then Access-Accept packet is sent to the client. [3], [4], [5].

For any response Response Authenticator is calculated for this packet and Identifier is identical to that of the request. The Access-Accept packet can have additional information about configuration values in attributes. Access-Reject packet in the other hand can only have attribute hat contains a text message to be shown to the user. Figure 3 shows RADIUS authentication and authorization procedure [11].

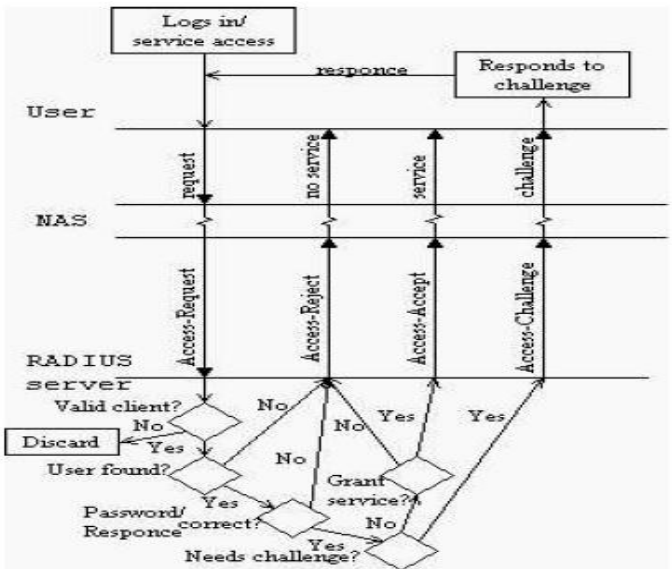

Figure 3: RADIUS authentication procedure.

When the NAS receives the reply, it matches the reply to the request using the Identifier. Then the NAS calculates Response Authenticator for received reply the same way the RADIUS server did and compares this value to Response Authenticator in the message. If these match the the RADIUS server is authenticated and the integrity of the reply is verified [4]. All the fields of the reply (with Request Authenticator in place of Response Authenticator that is being calculated) and attributes of the reply are concatenated with the shared secret and hash value of this concatenation is the Response Authenticator. The Response Authenticator can therefore be used for checking the integrity and to authenticate the RADIUS server, as any change in the message or mismatched shared secret, and the values would not match [5]. RADIUS protocol support also additional challenge/response authentication. In this method the RADIUS server, after receiving Access-Request and having checked the user information from user database, sends Access-Challenge packet to the client. This packet may have an attribute that is message to be displayed to the user. [5]

When the NAS receives Access-Challenge packet, it displays the message to the user (if message is present in the attributes) and waits for user's response to this challenge. After user has responded, the NAS resends the original Access-Request packet with new identifier and the users response encrypted in User-Password attribute. If the NAS does not support this challenge/response scheme, it will regard Access-Challenge as Access-Reject [4], [5].

RADIUS server then again checks from it's user database if the response to the challenge was correct. If not, then Access-Reject packet is send. If response was correct, the RADIUS server can send Access-Accept or new Access-Challenge packet. In figure 3 can RADIUS challenge/response operation be seen. [5]. With this challenge/response method RADIUS protocol can use special devices such as one-time-password generators or smart cards to enforce stronger authentication for dial-in users. This enhances RADIUS authentication strength because new innovation in this field can be added as part of RADIUS user authentication process [4].

III. ACCOUNTING

RADIUS accounting is done almost the same way as RADIUS authentication and authorization.

There are some differences. RADIUS accounting uses the UDP port 1813. There are also two RADIUS message codes and 12 attributes for RADIUS accounting. In addition Request Authenticator is calculated differently when using RADIUS accounting [6]. Accounting starts with the NAS sends RADIUS packet with code Accounting-Request having Acct-Status-Type attribute for Start to the RADIUS server. In the accounting start request, the attributes containing information about the user and service being used. All attributes that can be used in Access-Request can also be used in Accounting-Request with five exceptions. These are User-Password, CHAP-Password, Reply-Message, State and CHAP-Challenge attributes [6].When the NAS wishes to stop the accounting, it sends RADIUS

packet with code Accounting-Request having Acct-Status-Type attribute for Stop to the RADIUS server. This packet can have attributes containing information about the service that was used and statistics of the use [6], [3]. Having received a request packet the RADIUS server then records the Accounting-Request and after successfully recording the packet acknowledges it by sending Accounting-Response packet to the NAS. If the request is not successfully recorded, then no acknowledgment is sent. If the NAS does not get acknowledgment for its request, it will retransmit the request or transmit the request to other RADIUS server. In Accounting-Response packet there are no attributes, except for possible Proxy-State and Vendor-Specific. Figure 4 shows operation of RADIUS accounting [6].

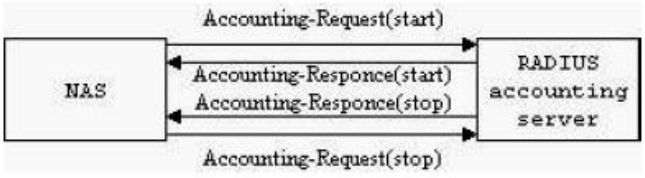

Figure 4: RADIUS accounting.

In Accounting-Request packet the generation of Request Authenticator is different from the generation of Request Authenticator in Access-Request packet. In Access-Request the Request Authenticator is random number, but in Accounting-Request packet the Request Authenticator a hash value so that it will protect the integrity of the request. Accounting-Request Request Authenticator is MD5 hash over concatenation of Code, Identifier, Length, 16 zero octets, attributes in the request and secret shared between the NAS and the RADIUS server[3], [6] . Response Authenticator in Accounting-Response packet is calculated the same way as described for Response Authenticator in section 2.2 [6].

A. Security Measures

Firstly, to provide security for RADIUS messages, the RADIUS client and the RADIUS server are configured with a common shared-secret, which is never sent over the network...

Secondly, the RADIUS protocol adopts Authenticator mechanism. The Authenticator authenticates the reply from the RADIUS server to the NAS and is also used in encryption of User-Password attribute. Two different kinds of Authenticator fields are defined. Request Authenticator is the name of the Authenticator field in Access-Request type packets. It is a random number that the NAS generates in order to be able to authenticate that the reply is intended exactly for the request that the Request Authenticator was generated for. Therefore it must be unique and unpredictable. NAS also uses Request Authenticator when encrypting User-

Password attribute. Response Authenticator is the name of the Authenticator field in Access-Accept, Access-Reject and Access-Challenge type packets. Its value is calculated by the RADIUS server. Equation (1) shows formula for Response Authenticator.

Response Authenticator=MD5(Code+Identifier+Length+Request Authenticator+Attributes+Shared Secret)

…… (1)

Thirdly, user password in User-Password attribute is encrypted by stream-cipher. Encryption is done as follows. First password is divided to 16 octet segments, padded is not multiple of 16. So a number of 16 octet segments are gained, now denoted as p1, p2... Then MD5 [13], [14] hash is calculated over concatenation the secret shared between the NAS and the RADIUS server and the Request Authenticator result now denoted as b1. Then p1 and b1 are XORed, the result denoted as c(1). c(1) is the placed as the first 16 octets of the User-Password attribute. For all values of pi, i greater than 2, MD5 hash is calculated over concatenation the secret shared between the NAS and the RADIUS server and c(i-1) and h(i) is gained. Then pi and bi are XORed and c(i) is the result. Finally the User-Password attribute contains concatenation of all c(1) to c(i) values.

b1=MD5(S+RA) c(1)=p1 xor b1
b2=MD5(S+c(1)) c(2)=p2 xor b2
…
bi=MD5(S+c(i-1)) c(i)=pi xor bi
c=c(1)+c(2)+c(i) (2).

IV. RADIUS ISSUES

The RADIUS protocol has a set of vulnerabilities that are either caused by the protocol or caused by poor implementation and exacerbated by the protocol [7].

A. Protocol Dependent Issues

- According to the protocol, a RADIUS server will not validate Access-Request packet really originated by RADIUS client before (and even after, if packet has no User-Password attribute) decoding all attributes, that is to say, RADIUS access requests need not be authenticated and integrity protected. It opens a packets. And it will make User-Password based password attack possible.
- The RADIUS hiding mechanism uses the RADIUS shared-secret, the Request Authenticator, and the MD5 hashing algorithm to encrypt the User-Password and other sensitive attributes. This is a well-known issue stated in [1]. MD5 is not designed to be a stream cipher primitive; it is designed to be a cryptographic hash [13], [14]. This sort of misuse of cryptographic primitives often leads to subtly flawed systems. It makes User-Password attribute based shared-secret attack and user-password based password attack easy.
- RADIUS encrypts only the password in the access-request packet, from the client to the server. The remainder of the packet is in the clear. Other information, such as username, authorized services, and accounting, could be captured by a third party.

The matter will be even worse when dealing with plaintext password authentication (such as Password Authentication Protocol, PAP).

- There is also problem with some vendor-specific RADIUS authentication implementation. For example Microsoft has its specific attributes defined in RFC 2548[8]. These attributes allow MS-CHAP and MS-CHAPv2 authentication via RADIUS. Microsoft doesn't use some cryptographic schema for its MS-CHAP-Challenge, MS-CHAP-Response and MS-CHAP2-Response attribute [9]. This opens possibility of both replay and spoof attack against MS-CHAP and MS-CHAPv2 authentication. There is design flow in this scenario which makes it vulnerable against Man-in-the-Middle attack.
- The RADIUS protocol does not offer replay attack prevention. An old packet can be replayed without detection by a malicious NAS impersonator. This can result in denial of service(DoS) if the server limits concurrent sessions for a user. Duplicate accounting messages can also create havoc. A malicious RADIUS server impersonator can replay response message to NAS too.
- The RADIUS protocol offers only hop-by-hop security and has no facility for securing Attribute-Value pairs between the NAS and the RADIUS server. This offers proxy servers the opportunity to collect confidential information or modify messages without detection by the endpoints.
- The RADIUS protocol does not allow a server to send unsolicited messages to the NAS. Where server initiated actions are needed, vendors are forced into solutions outside of the RADIUS protocol or solutions involving proprietary extensions to the RADIUS protocol in ways that often compromise interoperability.
- RADIUS runs on UDP, with no defined retransmission or accounting record retention policy, and according to the protocol, the NAS cannot distinguish the cause of the failure from RADIUS server. After several times (can be configured) retransmissions, the RADIUS protocol specifies that messages are silently discarded for a variety of error conditions. If the messages are Accounting-Request, silent discarding may cause RADIUS server lose user's accounting message. In addition, the connectionless nature of UDP means that one spoofed UDP packet is often enough for an attacker.

B. Implementation Dependent Issues

- Request Authenticator in Access-Request packet is a 128-bit quantity intended to be unpredictable and pseudo-random. But bad implementations do not create Request Authenticators that are sufficiently random. It opens possibility to spoof RADIUS client Access-Request. It is possible for an attacker with the ability to capture traffic between the RADIUS client and server, and to attempt network access to create a dictionary of RADIUS Request Authenticators and the corresponding key stream used to encrypt the User-Password and other attributes. And it can lead to Access-Accept/Reject replay and bring DoS.
- In many RADIUS installations, the same shared secret is used to protect many RADIUS client-server pairs, and many implementations only allow shared secrets and user-passwords that are ASCII characters, and less than 16 characters resulting the RADIUS shared-secrets and user-passwords does not have sufficient randomness to prevent a successful offline dictionary attack. For a guess of the RADIUS shared-secret, the Response Authenticator field and the contents of the Message-Authenticator attribute are easily computed. Offline dictionary attack on RADIUS shared-secrets can be easy.
- In some cases RADIUS may allow privilege escalation: for example user can try to login to NAS via telnet and change Service-Type attribute of Access-Request packet from login to framed to be authenticated by RADIUS. Everything depends on RADIUS and NAS configuration (the good practice for RADIUS server is always send back Service-Type attribute). If Access-Accept doesn't contain Service-Type attribute or NAS doesn't check it, user can login via telnet.
- According to [1], each RADIUS packet can be up to 4096 bytes. It allows putting > 2000 attributes into a single packet. Most implementations of RADIUS servers allocate maximum attribute length for each attributes, it means for each attributes > 256 bytes of memory will be allocated. Therefore, it is possible to lock >512K of memory and amount of CPU time with a single 4K packet. It opens a possibility to spoof source IP for this kind of packets. This is a major weakness in RADIUS protocol rather then all hard-to-exploit cryptographic Man-in-the-Middle issues. Coupled with the spoofing of Access-Request packets with no Message-Authenticator attribute, this may be serious.
- Some of current RADIUS server implementations are derived from Cistron. And most of them have buffer overflow in digest calculation. Probably this overflow can only lead to DoS. Since overflow occurs before packet is checked, it can be exploited from spoofed IP.
- The RADIUS server has local root exploits if its configuration files had open write permissions. So the system is vulnerable to insider attacks. A common way to protect a static encryption key is to save it in a file with restricted access. However, it

is inadequate to prevent a super-user privilege from accessing the static key in its hosting file.

V. RADIUS IMPLEMENTATION AND EXTENSIONS

To address RADIUS issues when deploying a RADIUS solution, the following deployment can be used:

A. Deployment Best Practices

- Use cryptographically strong Request Authenticators. The Request Authenticator value must be changed each time a new Identifier is used.
- Allow the configuration and use of shared-secrets and user-passwords consisting of a random sequence at least 32 hexadecimal digits long or 22 keyboard characters long including a random sequence of upper and lower case letters, numbers, and punctuation. Ideally, the shared-secrets and userpasswords should be computer-generated. Use a different shared-secret for each server- client pair.
- Check the sizes of attributes or messages properly, and ignore inconsistent attributes result in the message. To solve the problem that a single RADIUS packet locks too much memory, have value-pair with variable size data.
- Use the Message-Authenticator attribute in all Access-Request messages. The access server must send Access-Request messages with the Message-Authenticator attribute and the RADIUS server must silently discard the message if the Message-Authenticator attribute is either not present or fails verification [11].
- Configure the NAS to send RADIUS Accounting-Interim-Update attribute. It can keep the accounting information up to the point of the last Accounting-Interim-Update. But this will bring periodic traffic between RADIUS client and RADIUS server [4].
- Set primary RADIUS server and secondary RADIUS server at the same time to realize disaster recovery. They can be all in one LAN or not.
- Limit the number of the same user attempt to login. If a user try to login too much times, use lockout mechanism to lock its accounting.
- Disable PAP authentication by default. Use a strong CHAP (Challenge Handshake Authentication Protocol) challenge when implement CHAP authentication. If implement MS-CHAP or MSCHAPv2 authentication, do not support LAN manager encoding of MS-CHAP challenge responses or password changes.
- Use dynamic user-passwords to gain access. RADIUS is capable of using a strong, two-factor form of authentication, in which users need to possess both a user ID and a hardware or software token to gain access. Token-based schemes use dynamic passwords. Every minute or so, the token generates a unique 4-, 6- or 8-digit access number that is synchronized with the security server. To gain entry into the system, the user must generate both this one-time number and user ID and password.

B. Extensions

- Design a two server billing mechanism to record accounting messages on two RADIUS servers at the same time. This makes it easy for ISPs (Internet Server Provider) or users to check up if two servers have the same accounting records, to avoid some cheating. Notice that just authenticate a user on one RADIUS server.
- Add a message to traditional RADIUS protocol to achieve the synchronization between RADIUS server and client. The message can be serverinitiated or client-initiated. It will be used when server or NAS want to synchronize with another, for example when RADIUS server wants to restart.
- Extend some vendor-specific attributes. For example, add Error-Info attribute to record error information from RADIUS server, add No-Account-Info attribute to declare that need not to record a user‘s accounting message because it is in monthly or yearly packet, add VLAN (Virtual Local Area Network) attribute to declare a user in a VLAN. If there are new requirements, it‘s convenience to add new attribute to the system, but pay attention not to bring insecurity factors.
- The perfect solution is to use it in conjunction with IPSec, if RADIUS traffic cross untrusted network. If the NAS can support IPSec, then the best thing to do is to forsake RADIUS application-layer security entirely and to just run RADIUS over IPSec ESP with a non-null transform. This is described in [6].Unfortunately, many embedded systems do not have the horsepower or headroom to run IPSec, so RADIUS/IPSec is not widely used today.

VI. WHY MODIFY RADIUS?

So, why attempt to modify RADIUS at all? Why not just go to another (presumably more modern, more secure) protocol? Well, for the most part, the answer is "Because such a protocol doesn't currently exist." In the near future, however, Diameter is likely to be released by the ETF. Diameter is the planned RADIUS replacement. The great majority of all the protocol work that has gone into Diameter has been directed to removing some of the functional limitations imposed by the RADIUS protocol. Effectively no work has been done as relates to the client/server security of the protocol [8], [9]. (CMS is defined, but this is a security layer for the proxy to proxy interaction, not the client to proxy/server interaction)

So, does this mean that they continue to use even RADIUS's ad hoc system? No, they removed ll security functionality from the protocol. They did the protocol designer's equivalent of punting. Section 2.2 of the current Diameter protocol spec says:

"Diameter clients, such as Network Access Servers (NASes) and Foreign Agents MUST support IP Security, and MAY support TLS. Diameter servers MUST support TLS, but the administrator MAY opt to configure IPSec instead of using TLS. Operating the Diameter protocol without any security mechanism is not recommended."

So, all security aspects of the protocol are handled by IPSec and/or TLS. From a security aspect, his strikes me as a very good idea. Both IPSec and TLS are fully featured (sometimes too fully featured) protocols that many people have reviewed. (That's already much better than RADIUS ever did).

Examining this from a slightly different angle gives me some cause for concern, however. It strikes me that the overhead imposed by a full TLS/IPSec implementation is very significant for many current-day embedded devices. This would seem to indicate that (at least in the near future) manufactures are going to either continue to use RADIUS or ignore the diameter standard and perform Diameter without TLS or IPSec.

VII. Conclusions

RADIUS is a widely used AAA protocol because it is simple, efficient and easy to implement. This paper provided an overview of RADIUS protocol and described how RADIUS security issues are addressed or minimized using implementation, deployment best practices and extensions. These include using strong shared-secrets, the Message-Authenticator attribute, cryptographic-quality values for the Request Authenticator, different shared-secrets for each RADIUS client/server pair, and IPSec to provide data confidentiality for RADIUS messages, and so on. At the same time, many new-world technologies are requiring a secure, peer-to-peer, and reliable framework that not only has the richness of RADIUS but also the flexibility and robustness of Diameter, the next-generation AAA protocol.

VIII. References

1) [C. Rigney, A. Rubens,W. Simpson and S.Willens. RFC 2865: Remote Authentication Dial In User Service (RADIUS). http://www.ietf.org/rfc/rfc2865.txt
2) C. Rigney. RFC 2866: RADIUS Accounting. http://www.ietf.org/rfc/rfc2866.txt
3) G. Zorn, B. Aboba, D. Mitton. RFC 2867: RADIUS Accounting Modi_cations for Tunnel Protocol Support.
4) http://www.ietf.org/rfc/rfc2867.txt
5) G. Zorn, D. Leifer, A. Rubens, J. Shriver, M. Holdrege, I. Goyret. RFC 2868: RADIUS Attributes for Tunnel Protocol Support. http://www.ietf.org/rfc/rfc2868.txt
6) C. Rigney, W. Willats, P. Calhoun. RFC 2869: RADIUS Extensions. http://www.ietf.org/rfc/rfc2869.txt
7) B. Aboba, G. Zorn, D. Mitton. RFC 3162: RADIUS and IPv6. http://www.ietf.org/rfc/rfc3162.txt
8) J. Hill. An Analysis of the RADIUS Authentication Protocol. On Bugtraq mailing list, 12 November 2001,
9) G. Zorn. RFC 2548: Microsoft Vendor-specific RADIUS Attributes. http://www.ietf.org/rfc/rfc2548.txt
10) S. Bruce. Cryptanalysis of Microsoft's PPTP Authentication Extensions (MS-CHAPv2). CQRE '99, Springer-Verlag, 1999, pp.192-203.
11) P. Calhoun, J. Loughney, E. Guttman, G. Zorn, J. Arkko. Diameter Base Protocol. http://www.ietf.org/internet-drafts/draft-ietf-aaadiameter-17.txt
12) Andrew S.Tanenbaum, Computer network (Fourth edition), 2007
The Handbook of Applied Cryptography, by Alfred J Menezes, Paul C. van Ooschot, Scott A. Vanstone.
13) Hans Dobbertin. The Status of MD5 After a Recent Attack, CryptoBytes, volume 2,number 2, summer 1996. ftp://ftp.rsasecurity.com/pub/cryptobytes/crypto2n2.pdf
14) R. Rivest, S. Dusse. RFC 1321: The MD5 Message-Digest Algorithm..http://www.ietf.org/rfc/rfc1321.txt
15) B.Aboba et al. RFC 2989: Criteria for Evaluating AAA Protocols for Network Access. http://www.ietf.org/rfc/rfc2989.txt