FLEXIBLE AUTHENTICATION FRAMEWORK

Jan Hajny
Doctoral Degree Programme (1), FEEC BUT
E-mail: jan.hajny@phd.feec.vutbr.cz

Supervised by: Karel Burda
E-mail: burda@feec.vutbr.cz

ABSTRACT

This paper deals with user authentication which is a fundamental part of computer security. We identify main issues of present authentication systems which we find unsatisfactory mainly because of the lack of flexibility and security. As a reaction we provide an authentication framework model which gives more support for mobile nodes and simple devices. The framework is illustrated by a RADIUS protocol example. This example serves as a possible implementation of future realization but the framework is not restricted only to this particular protocol usage.

1 CURRENT STATE

User authentication is a process ensuring user identity verification. Without authentication we would be unable to build secure systems because we would not be able to distinguish valid users from attackers. This ability is very important in present networks. We use networks for very sensitive operations like money transactions or private data use. For all these activities there must be tools for reliable user identification because a mistake would make these systems completely untrustworthy and so useless. There are many authentication protocols to use and most of them are based on cryptography. We can mention RADIUS [1], Diameter [2] or EAP [3] as basic examples. These protocols basically work on the edge of a network. Valid users are permitted into private inside network and attackers are rejected. The decision is usually made after determining whether the user possesses some kind of secret given only to valid users. This secret could be a password, authentication key, authentication token or some smart card. But there are also some drawbacks of these protocols. The most important thing is that they should reflect the state of current networks, especially Internet. This state has changed a lot in recent time. There is a big effort to converge networks. This results in a multi-purpose network consisted of computers, mobile devices, GSM phones, sensors and so on. There are not only computers in computer networks but a very wide variety of other devices. With these devices a wide variety of hardware platforms, abilities and power restrictions comes together. Now we have strong computers with simple sensors side by side with a demand of authentication. Both these systems should be supported and authentication must stay both reliable and fast enough. The problem is that original authentication protocols don't have enough flexibility. There are solutions which can support computers but would be too complex for simple devices. On the other hand we can't simplify authentication protocols to support these devices because we would violate security. That's why there is a need for an authentication framework which

would be able to find out the type of a client and choose the right protocol for him. Appropriate rights management must be added to keep security. Here we sum up main issues of today's authentication methods:

- Flexibility - there is a need for wider support of all devices which can appear in present networks. Especially simple nodes must have the opportunity of reliable authentication.
- Security - security comes with flexibility. Although there must be good support of all kinds of devices this property can't go against security in the system.
- Modularity - we found that most of present protocols use traditional cryptographic methods to do authentication. It is often difficult to intruduce a new authentication method so there should be an easy way to extend the framework to support new techniques.

2 GENERAL AUTHENTICATION

We would like to provide information about general authentication scheme in this section. The reason is that we will come from this scheme during the new framework creation. Authentication protocols usually distinguish three entities - users, NAS (Network Access Servers) and AS (Authentication Servers). There is a communication scheme which could be illustrated by the Figure 1.

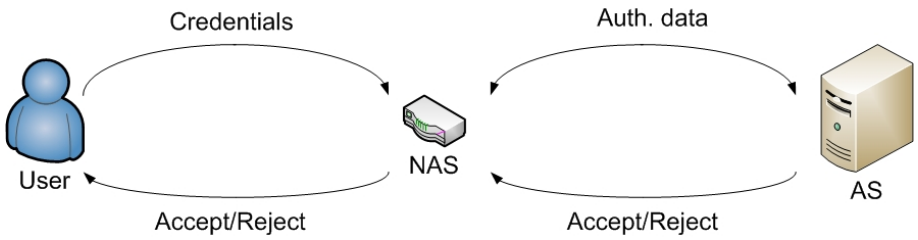

Figure 1: User authentication

- The user wants to get authenticated (wants to access the network or some service).
- The user sends credentials to NAS which is on the border of a network.
- NAS communicates with AS which has an access to the network object database and optionally with the user.
- After the communication a decision about the access is made and sent back to the user. NAS usually does the access control so it opens/closes communication ports.

During this process there is not much space where the client could have some influence on the authentication process. The client only provides credentials, eventually responds to a challenge. The protocol is more or less given by AS. Our goal is to give the client some rights to decide the way of authentication. There should be at least the possibility of a protocol choice. Without these client rights the scheme is too inflexible to distinguish the type of a user and so the authentication procedure won't fit the user as it could.

3 CLIENT PREFERENCES

This problem could be solved by client preferences. Only a slight modification of the scheme is needed. We need to provide NAS with not only credentials but also with information about a client. At least we should give information about abilities and computing power. AS could do a better decision with these data and choose among protocols to fit user's needs. The intention is to choose a faster protocol for weak devices and complex protocols for strong ones. There is still the problem with security. Faster protocols are usually weaker so simple node authentication would not be as strong as server authentication. To solve this we need to bind the authentication with authorization. We result from the fact that simple nodes like sensors usually do not require complex access rights and their access can be quite restricted. On the other hand e.g. computers which do the full complex authentication can have full access to the system. So the information about user capabilities does not go only to the authenticator to choose a suitable protocol but also to the authorizator who sets appropriate rigts. The result is a flexible system where the user can decide whether the authentication process will be fast and easy with low permissions or complex and safe with wide permissions. With such a dynamic system we are able to support wide range of devices from sensors to fast computers without loss of security.

4 UNIVERSAL AUTHENTICATION FRAMEWORK

We present a universal authentication framework model in this section. This model implements previously mentioned scheme with client preferences. There are four critical points which must be fulfilled – flexibility, security, modularity (as mentioned in the first paragraph) plus efficiency. The efficiency of the model must be present because of the practical implementation. There are three main stages of the framework:

- Handshake – the first stage where information is sent from the client to AS. A proper authentication protocol must be chosen by the server. There can be a further communication between the client and the server if needed.
- Service execution – a protocol run. In this phase a concrete protocol is executed and authentication result (accept/reject) is obtained.
- Record – this phase is designed to store information about user history. These data could be later used for a protocol choice.

The service execution and the record are easy to implement because we use already developed protocols and the record is trivial. Handshake is the most critical phase. Here the right choice must be made. For this we use a decision model illustrated in the Figure 2.

This model chooses a protocol based on input data from the client and on information about available protocols from a table which is set by the administrator. Client information is divided into critical parameters and non-critical ones. Critical parameters must be satisfied completely (like e.g. proof factor) and non-critical ones serve as a decision criteria among remaining protocols. With this scheme we are able to use many protocols and also include new ones.

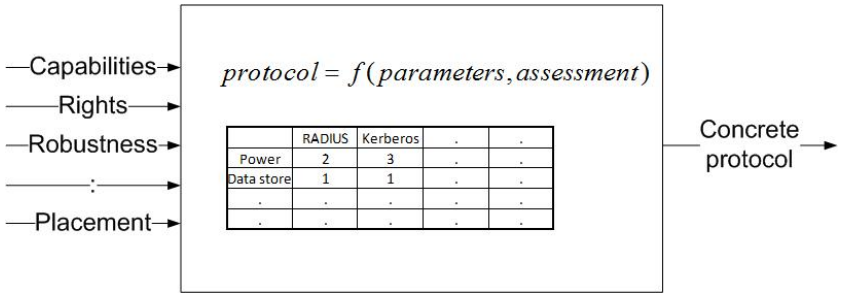

Figure 2: Protocol choice

5 PRACTICAL FRAMEWORK PLACEMENT

We present a possible placement of the framework in the real-life protocol. This practical example uses a RADIUS (Remote Authentication Dial In User Service) protocol which is one of the most used protocols on the Internet. Almost every border node accepts RADIUS as an authentication protocol. RADIUS also supports many authentication methods from default user/password to e.g. Kerberos. This flexibility is achieved by the EAP (Extensible Authentication Protocol) support [4]. EAP is a framework which associates many authentication techniques to a unified environment. The framework is also extensible – a new method could be easily added. This functionality is essential for us – for our flexibility requirement we need this wide technique support. As RADIUS can transfer EAP messages the basic authentication works like depicted in the Figure 3.

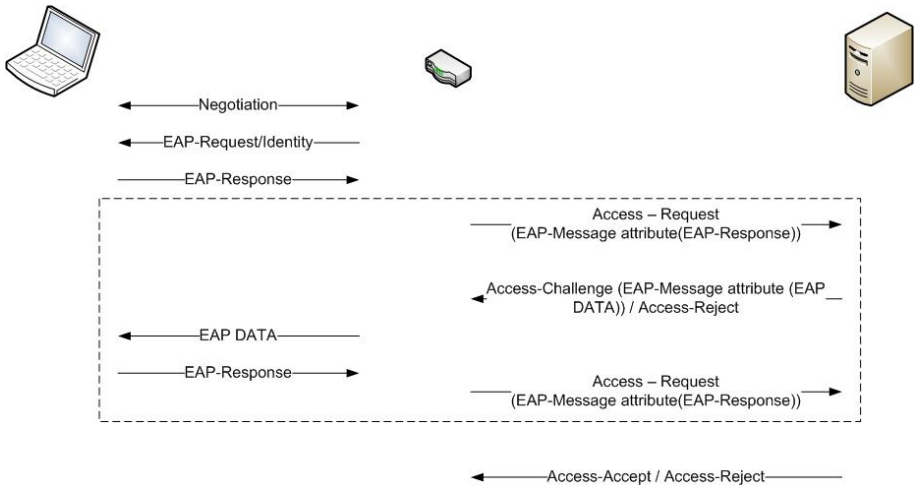

Figure 3: EAP over RADIUS

The authentication process works very similarly to the general one described in the beginning of the paper. There is a communication among user, NAS and AS. The most important part is the one inside the dash-and-dash box. EAP data can be transferred directly between the user and AS here. So any information can be sent from the user to AS and any EAP authentication method can follow. This is a perfect place for the handshake phase placement. By putting this phase before the EAP authentication process we can implement the Universal Authentication Framework described in the previous section. The result shown in the Figure 4 is then able to use any EAP authentication method (modularity satisfied) based on client preferences (flexibility satisfied) within an efficient and well-established protocol (efficiency satisfied).

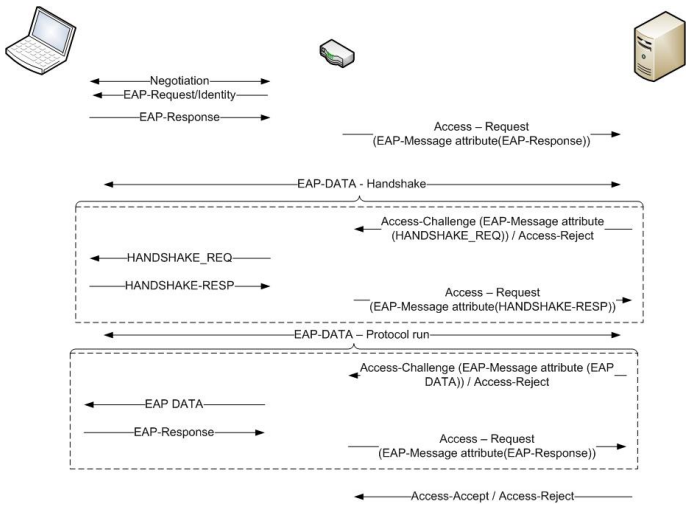

Figure 4: Added handshake to RADIUS/EAP

6 FUTURE PLANS

We presented a basic model for a flexible user authentication. Our goal was to add more flexibility to present authentication protocols to support wider variety of devices and that's why reflect modern trends in computer networks. This task was accomplished by the introduction of the universal framework which works with client parameters in the handshake phase. This is a major change in comparison to previous solutions. There are still some open problems left. We must bind authentication with authorization to reflect different security levels. This should be solved by creating more roles in the system. A practical model is another intention. A NS2 [5] model should give us more information about practical behavior of this system.

ACKNOWLEDGEMENTS

Sponsored under the National Program of Research II by the Ministry of Education, Youth and Sports of the Czech Republic in 2C08002 Project - KAAPS Research of Universal and Complex Authentication and Authorization for Permanent and Mobile Computer Networks.

REFERENCES

[1] Rigney, C.: Authentication dial in user service (RADIUS), 1997. [Online]. Available: <http://www.ietf.org/rfc/rfc2865.txt>

[2] Calhoun, P.: Diameter base protocol, 2003. [Online]. Available: <http://www.ietf.org/rfc/rfc3588.txt>

[3] Aboba, B.: Extensible Authentication Protocol (EAP), 2004. [Online]. Available: <http://www.ietf.org/rfc/rfc3748.txt>

[4] Aboba, B.: RADIUS Support For Extensible Authentication Protocol (EAP), 2004. [Online]. Available: <http://www.ietf.org/rfc/rfc3579.txt>

[5] The network simulator, 2009. [Online]. Available: <http://www.isi.edu/nsnam/ns/>