1

The 3 "A"s: Authentication, Authorization, Accounting

For the road travelers in the United States, especially the parents who take their children in the family car on the long road trips, the letters AAA stand for a peace of mind. They feel that any time their car breaks down, they can call the number for the American Automobile Association and ask for roadside assistance. Even though this book is not about that sort of AAA, the 3 "A"s that we talk about here, when designed properly, can bring the same peace of mind to the network operator and its customers. Authentication, authorization, and accounting are three important blocks used in the construction of a network architecture that helps protect the network operator and its customers from fraud, attacks, inappropriate resource management, and loss of revenue.

In this chapter, we describe each of the "A"s in the AAA first as a separate topic, and then as a piece that interacts with the other "A"s in an effort to justify why all the 3 "A"s should be treated by the same framework and servers. At the end of the chapter, we provide a model for a generic AAA architecture.

1.1 Authentication Concepts

According to the dictionary, the word "authentic" refers to something that is not false, or a fake imitation, but is worthy of acceptance as a truth or a fact. From the times of early civilizations, where people have run 26 miles only to deliver a message and then fall over and die, to today, when information can travel across the globe in fractions of a minute with a mouse click, proof of authenticity is the first thing the receiver of a message checks.

Authentication consists of two acts: first, the act of providing proof of authenticity for the information that is being delivered or stored, and second, the act of verifying the proof of authenticity for the information that is being received or retrieved. In the early ages, an emperor would use his personal seal on his letters to provide assurance for the authenticity of the letter. The letter could then be carried by any messenger, whose identity was not important. The local lord would recognize the emperor seal and trust authenticity of the letter. He would break the seal, read the letter, start an attack or collect taxes accordingly. In the days of digital information delivery, delivering proof of authenticity is equally important but poses its own challenges, as we will see.

The message delivery example above presents one type of authentication problem where authenticity of the information is important, while the identity of the messenger is not. However, in most of the cases, the identity of a person we are dealing with is an important factor in how we handle that interaction. When we go to a bank or through customs into a new country, we have to show identification to prove our identity. At first, the problem of identification does not seem to be related to the authentication. However, when one thinks about the possibility of a person lying about her identity or privileges, verification of authenticity of the provided identity becomes an authentication problem as well. Stating a name is typically not enough for identification, while showing a sort of identification issued by a trusted authority typically is. The acts of providing proof and verifying the authenticity of the identification presented are again the two acts of authentication.

Today, the two mentioned forms of authentications, i.e. providing information integrity and identity verification, are among the most fundamental security mechanisms required for providing access to network users and clients. In this introductory section, we provide a relatively short overview of various authentication concepts to allow the reader to understand the distinction between the constantly confused types of authentication. In Chapter 2, we will delve into more details of various authentication procedures.

1.1.1 Client Authentication

Client authentication means that a client wishing to gain access and connect to the network presents its identity along with a set of credentials. As proof of authencity for the presented identity. The credentials are then used by the network to verify that the identity actually belongs to the client.

We intentionally used the term client, since it can be interpreted both as a device as well as a human user, who is a consumer of a network service. For that reason, the client authentication needs to be further refined into two categories: user authentication and device authentication. Until recently, very few network security designs made a visible distinction between user authentication and device authentication. In the following we will explain the reason. Traditional architectures dealing with network access control could be divided into two categories:

- Architectures accommodating users that arrive at a fixed location, such as a local area network with fixed devices, such as data terminals, already connected to an infrastructure. The user needs to use its personal credentials to log into the network through a device (terminal), which itself typically resides in a computer room and is trusted through its wired connections. A good old world college campus terminal room scenario! The student simply trusts the network set up by the campus, as long as the college is an accredited one and the terminal is not asking for credit card numbers as login credentials! In this basic scenario, the distinction between device and user authentication although very clear for a human, is not important. The device is not authenticated at all. The user credential with a central server is the main criterion for allowing network access to the user.
- Architectures accommodating mobile users carrying their personal devices to gain access to a wide area network. A perfect example is cellular phone systems. The user registers

with a cellular service provider and purchases a cellular phone that works with operator’s network. The operator creates a set of credentials specific for the user. The cellular phone, that the user carries, is then programmed with such credentials to access the network. Many times, the user is unaware of these credentials and the actual process of authentication during connection establishment. The device is the entity that interacts with the network and presents the credentials needed to perform authentication. The idea is: since the user always carries the same cellular phone (as long as she is loyal to her service provider) no distinction between the user and device credentials has to be made. The downside is that if the device was lost, stolen, or even cloned, the rouge or unintended user could use the device to gain access to the network without having her real identity exposed, and this could go on until the legitimate user would report a lost or stolen phone or illegitimate entries in her monthly statement from the service provider.

With the proliferation of public local network providers, such as wireless hot spots serving passing customers, many sorts of vulnerabilities will appear at various corners of the architecture:

- The long-term customer–operator business and legal relationship no longer exists, which means the network operator and the user cannot trust each other as.
- The one-to-one mapping of user–device does not exist. Even if the operator could trust a user, the operator would not know what device the user may use every time they try to access the network. In other cases, such as in service organizations, government agencies, or police department, users that belong to a team can share their devices with each other.

Such refinements require more precise definition of the network usage and security policies that in turn means the architecture must be designed more carefully. The network operator may need to make sure that both device and user are authentic, possibly using separate processes. In some cases, the device may have to even be manufactured and configured with credentials for access to the network. (For instance, cable modems for cable Internet Service Provider (ISP) networks are produced this way.) In such cases, proper protection must be in place, so that the credentials on the device can not be tampered with. The network operator also needs to make sure the user presents accurate identity and proper credentials that can identify her at the time of use, so that the various users can be distinguished even when they are using a shared device.

Device authentication credentials can be certificates or cryptographic keys that are loaded in the devices either by the manufacturer in the factory or by the network operator at the time of service initiation. When designed properly and in a modular manner, the device authentication process should be transparent to the user. In fact, the user should not even be allowed to access device authentication credentials. On the other hand, user authentication credentials are personalized, typically given to the user in an out-of-band method. Examples are over a phone call by the user stating some secrets about her identity or through a face-to-face meeting after presentation of a driving license, company badge, and so on. Upon identification of the user, the system operator issues the authentication credentials for the user. The credentials must be carried by the user at all times either memorized (password) or in the form of a token such as a secure ID card, a certificate on some sort of cryptographic module. The user applies her authentication credentials on the device provided for access to the network to connect to the network.

It should be noted that the security architecture may require both device authentication as well as user authentication in various steps of a network access process. An example would be the case of IP networks: in order to communicate to the IP network, the device needs to acquire an IP address. The IP address is not only an important resource for the network, but also allows the user to gain access to many other network services. Furthermore, the IP address can be used as a backdoor to launch active or passive attacks. Hence, IP address acquisition should be tightly controlled. From a security standpoint, it means a device needs to first authenticate itself to the network before being able to gain an IP address from the network. Once the device is authenticated, has gained IP address from the network, and is registered with various agents, it allows the user to present her credentials to the network and gain access to the services to which she is entitled. The latter brings another point and that is the user credentials may be used to determine authorization levels for the users based on their pre-configured service profile. We will discuss the topic of authorization later on.

1.1.2 Message Authentication

Device or user authentications deal with ensuring that the end points of the communications are legitimate and who they claim they are. Message authentication, on the other hand, ensures and verifies the integrity of the data at hand (remember the example of sealed letter from the emperor). When message authentication is performed, the receiver of the message can be sure that the information included in the message has been produced by a legitimate source and not been altered by other parties in transit. This is why message authentication is usually considered as a data integrity protection mechanism. Unlike device or user authentications that are typically performed at the beginning of a session and require their own messaging mechanism, message authentication may have to be done quite often during the session and for a variety of traffic, such as control messaging, important data packets, and sometimes even data session.

Note that the goal of data integrity protection is to prevent malicious and intended corruption of data by the so-called men in the middle (MITM), trying to tamper with the message contents. This is different from the information-theoretical codes and cyclic redundancy checks designed to mitigate the random and natural data corruptions caused by physical communications media imperfections. Aside from the cause of corruption being different, calculations required for message authentication for security protection is also different from its signal processing counter parts; an attacker can always alter the data and re-calculate the information-theoretical checks to lure the receiver, while the attacker cannot re-calculate the message authentication data added to the message, since message authentication is typically performed on the basis of knowledge of a shared secret.

More details are provided on message authentication in Chapter 2, so we will not go into any details in this introductory chapter. In short, this is how message authentication works: the sender provides proof for data integrity by running a so-called secret hash algorithm over the contents of the message and adds the results of the algorithm (called digest or hash) to the end of the message. Hash algorithms are mathematical one-way functions. In other words, while it may be straightforward to calculate the output of a hash function (digest) from an input data packet, it is extremely hard to determine what input packet has been used to create the output digest. However, hash algorithms are rather well known, and hence if no secrets are used, it is possible for an attacker to change the packet in flight and re-calculate a new digest and then to replace both the original packet and digest with the altered packet and the new digest. In order to prevent the MITM from running the same algorithm over an altered version of the message, the sender uses a secret when creating the hash. The secret is only known to the sender and the receiver. This process is often referred to as running a secret hash algorithm, even though the algorithm itself is usually known. Once the receiver receives the message, it runs the same hash algorithm over the same portion of the data and compares the result of its calculation with the hash provided by the sender and if there is a match, the receiver considers the data authentic, otherwise the receiver needs to discard the data.

The secret hash algorithms are examples of symmetric key authentication methods. Another method of providing message authentication is to create the so-called digital signatures. Digital signatures are typically based on public key cryptography. We will go through the details of these procedures in Chapters 2 and 3.

1.1.3 Mutual Authentication

We finalize the description of various authentication concepts with a short note on mutual authentication. Client authentication that was explained earlier is a unilateral authentication, where only one end of a communications channel proves that the identity it has presented to the other end is authentic. It would make sense that when establishing communications both ends authenticate to each other. However, the reason this type of unilateral authentication is so widespread is that in many cases, the client simply trusted the network. So the network does not have to prove to the client that it is also authentic. In the earlier cellular phone example, due to the large expense involved with setting up a cellular network, it would be hard to imagine that attackers will go through the trouble of setting up a whole network and its high towers simply to hijack a cellular phone subscription. For those reasons until very recently, neither the device nor the user required any authentication from the network.

With a proliferation of large number of access providers competing for customers, a user may be confronted with a number of competing networks with which she does not have any business or trust relationships. Take a mom and pop coffee shop for example: a customer entering the coffee shop does not know whether the mom and pop that own a coffee shop actually know how to operate a wireless local access network (WLAN) access point (AP) or how to protect their access points from being loaded with viruses and Trojan horses. For all a user knows, another customer may have found the WLAN AP behind the coffee grinder machine, disconnected it and simply installed another AP to re-route all the traffic in the shop to some place else, or simply passively copy traffic to a server. Imagine, if the customer was planning to do some Internet shopping while sipping coffee and was entering her credit card number at an e-commerce web site not using proper encryption methods. In this case, it makes sense for the coffee shop AP to authenticate itself to the user's device as well. As we can see the unilateral client authentication is not sufficient and has to be upgraded to a mutual authentication where the server also authenticates to the client.

The mutual authentication between the client and the server is a special case of a more generic case of mutual authentication, where the two parties are simply peers as opposed to client and server. In that case, each peer authenticates to the other, either sequentially, or in parallel.

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.