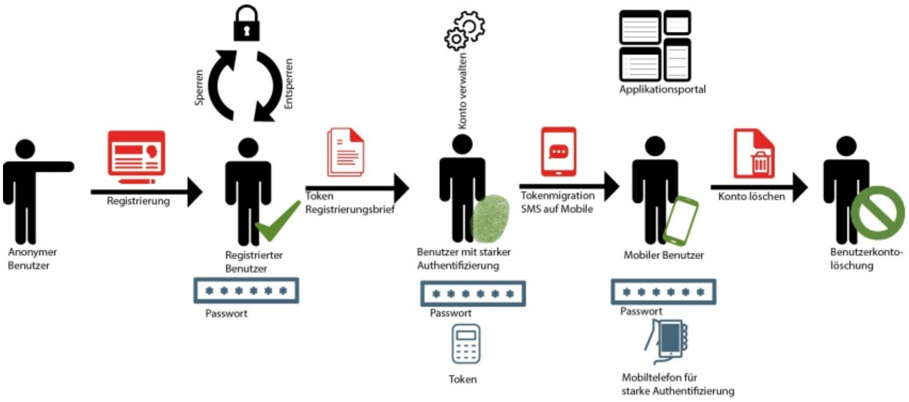
Test subject: Airlock Identity and Access Management 6

A powerful authentication solution with self-service functions

Dr. Götz Güttich

We recently took a close look at the Airlock Web Application Firewall, which is able to secure data traffic between users and applications within a company or the cloud. Many companies use the Web Application Firewall in combination with the Identity and Access Management solution from the same manufacturer. It makes sense to not only secure data transmission but also to check the actual identity of the users involved. This report takes a close look at the functions of the Identity and Access Management product.



The Airlock Identity and Access Management solution (IAM) doesn't have to be used with the Web Application Firewall (WAF), it can also be used as a stand-alone solution. But, for the purposes of our test, we ran both products together. Airlock IAM offers users a central authentication platform with enterprise functions. The solution supports a considerable number of authentication procedures, operates with standard protocols and also automates user administration. Comprehensive self-service functions ensure that users can complete every step in terms of user-account management, relieving the helpdesk considerably. The single sign on (SSO) product also enables SAML 2.0 IDP and SP, OAuth 2.0, OpenID Connect, Kerberos and NTLM.

The IAM solution combines user management with role management in operating mode and providing upstream authentication for sessions and requests. With the integration of LDAP, Microsoft Active Directory and database environments, radius servers and the like, users are able to continue using existing directories without any interruption. Integrated user management with token and role admin, reporting and password policy enforcement is also available. A powerful search function ensures that helpdesk staff are able to find user accounts and provide support for them quickly. If required, the search function can be extended via plugins where special functionalities are required in individual environments.

Logins run in operating mode via an integrated, freely configurable web application, which we will look at later in greater detail. A powerful REST API is also available. The Airlock IAM solution is also multi-client capable, offering fail-over and clustering functions and providing comprehensive logging features and statistics. Administration is via a central management console.

Authentication and identity propagation

The system supports a large number of authentication methods such as CrontoSign, mTAN/SMS, Vasco Digipass, Matrixcard, RSA SecurID and Kobil AST. In operating mode, the product separates authentication from identity propagation. The latter involves transmission of the user identity to the secured application – i.e. inwards. The way in which login data is transferred changes according to how the system is used. The system supports lots of different options in this respect. Since identity propagation is solved with the assistance of plugins, new applications can easily be added as required.

Its managers can also extend the IAM solution at any time and incorporate it in a non-standard environment. But such adaptations don’t turn it into a custom solution – it remains flexible and is compatible with regular update files provided by the manufacturer.

Assembly

Airlock IAM is made up of three components. The first is the login application with registration, self-service functions, for example, to reset forgotten passwords, captchas and the like. The second is the administration interface for managing the system and the third is the ‘service container’. This encompasses background services, synchronise databases, automatically print letters with authentication codes and much more.

The test

For the purposes of our test we imported the IAM solution to a virtual machine with Centos 7. All you need to run the product is a working Java environment, the manufacturer recommends at least a JRE with Java 8, ideally version 1.8u91 or above from Oracle. In terms of hardware, a computer with a 2 GHz CPU, 4 gigabytes of RAM and 10-gigabyte hard disk is required. Having completed the setup we set the IAM up on our network so that, working alongside the Airlock WAF, it would protect access to the web interface of the Paessler PRTG network monitoring product used by us in the test lab. We also used a test environment provided by the manufacturer with a bookstore as a test application to take a close look at the functionality of the IAM solution via a number of use cases.

Installation

Production installation is relatively easy. All you need to do is ensure the installation file, as downloaded from the manufacturer’s website, is executable with Linux and then access it. The setup routine then asks for the password for the IAM administrator and ports set up for data communication. We left these with standard values for our test installation. The installation then ran and we were able to log into the system via the URL: http:// localhost: 8443/ auth-admin/ login. Having done that, we then uploaded our licence.

In order to configure the IAM solution, in order that it would work with the WAF, which was running on our network on another virtual machine, the next stage involved entering the IAM in the WAF as backend host and set up the appropriate mapping. The manufacturer provides a template for the mapping, making it a simple task. The whole procedure is described clearly in the documentation, so it is straightforward to adapt the configuration to individual situations.

The configuration interface

We will now take a brief look at the IAM configuration tool. When a user logs in with the tool, they land on a page showing the current login details. They can upload the licence here and they can also manage the various administrator accounts that need to have access to the product, adapt the solution configuration, set up maintenance alerts that the system displays on the login screen and manage users. When the responsible employee moves to user admin, they land at a search function that they can use to look through all existing user accounts. When they access a user account the solution shows them an overview page with the user name, company, user token (if there is one), number of failed logins, date and time of last login, etc.

The ‘Profile’ tab is of even greater interest here. This enables you to change the user name, enter address details, set account validity periods, delete accounts and add the user to certain roles. This might be ‘Customer’, ‘Employee’ or ‘Administrator’. Everything can be configured at any time and all settings can be modified on the fly.

The next tab relates to authentication methods. This is where the relevant employees set the active authentication mechanism. Authentication is nearly always via user name and password. The security level can then be increased via other methods such as mTAN/SMS or Kobil AST. IT managers are also able to add more authentication methods such as Matrixcard, email OTP, OATH OTP and CrontoSign. There is also an option to initiate the migration of the authentication methods and compel the relevant users to switch to the new authentication method by a certain time. We will take a closer look at these functionalities later.

The other user admin tabs configure the active authentication methods. This is where administrators determine whether password changes are to be compulsory and how the long the codes on activation letters sent in the post are valid, etc. Finally, the user-admin function includes an activity overview that shows when the relevant user has logged in or whether there have been failed login attempts.

Thanks to the many different functions of the IAM solution, administrators have a wide range of user management options for controlling user authentication and automating management processes. There is an option to allow users to activate an authentication method like CrontoSign, for example. If IAM has been configured appropriately, as soon as the user is active the system prints an activation letter with an authentication code and sends it to the relevant user. When they receive the letter they can check that the address details are correct. When they scan the code with their smart phone, the new authentication method is launched and can then be used. As already mentioned, administrators can also dictate that the code has to be scanned within a certain period to compel the switch to the different authentication method. There is also the option to directly allow users who have logged in to activate a new authentication method without any further action being required. This makes the system highly flexible.

Configuring the login application

Also of interest: login application configuration. This is where IP address restrictions can be added and language settings changed. If required, the relevant employees can also decide that logins have to be delayed to make brute-force hacking a lot more difficult to attempt. When configuring the login application, the relevant employees also have the option to change all settings to self-services, allowing users to change login information and their password, etc.

The admin application

Let's now go back and take another brief look at the IAM configuration interface. This offers various roles for individual administrator accounts, via which access rights to functions within the administration application can be attributed in a granular way. The roles 'Useradmin', 'Tokenadmin', 'Helpdesk', 'Sysadmin' and 'Superadmin', and various combinations, are available, for example. This makes it easy to ensure that only users who really need it have access to individual functions in the IAM solution.

Life cycle of a user account

Let's take the example of the management of a user account throughout its entire life cycle. Users have the option, if the system has been configured appropriately, to set up their own account via the login application by signing in with their user name or email and password, for example. This worked in the test without any problems. Self-registration is freely configurable, as all organisations have different requirements for their user accounts. Many of them also require an address or company name etc. in addition to the name.

Once an account has been set up, users only have basic access to start with for a typical configuration. At the same time, the IAM sends them a letter with a code to scan. Once this scan has confirmed that the address is correct, they have the option of using their account to gain full access to all of the applications on offer. All of the stages can be completed automatically without the help of the helpdesk.

The self-unlock function is also of interest here. Users can unlock another login attempt at entering their previously forgotten password with the second authentication factor. This enables a locked account to be unlocked without the helpdesk having to be involved – even after the password has been entered incorrectly three times. If a password policy is changed, managers can compel users to change their password at any time after that to comply with the new policy. As mentioned before, there is also the option to compel users to switch authentication methods or give them the opportunity to add their own authentication methods to their account. Token migration functions are also available if required. This means that accounts can always be modified to current situations and nearly all activity happens without bothering the helpdesk team. The self-service portal also offers users many more functions, for changing their personal details or even deleting their account, for example. There were no problems with this during the test.

Other major functions

Let's now take a closer look at the major features of the IAM solution for administrators. The tool provides a range of different configuration contexts. This means that different practices can be set up within the configuration. For a user login with email address and password, for example, it is possible to add the user's company name as a suffix and make the login method dependent upon this information – either with or without a second authentication stage. If required, a step-up authentication can also be added. If various applications with varying authentication requirements are available via a WAF, administrators can ensure that, after login, the system permits access to applications without special protection. The others don't appear in the interface until the user in question has completed a further stage of authentication.

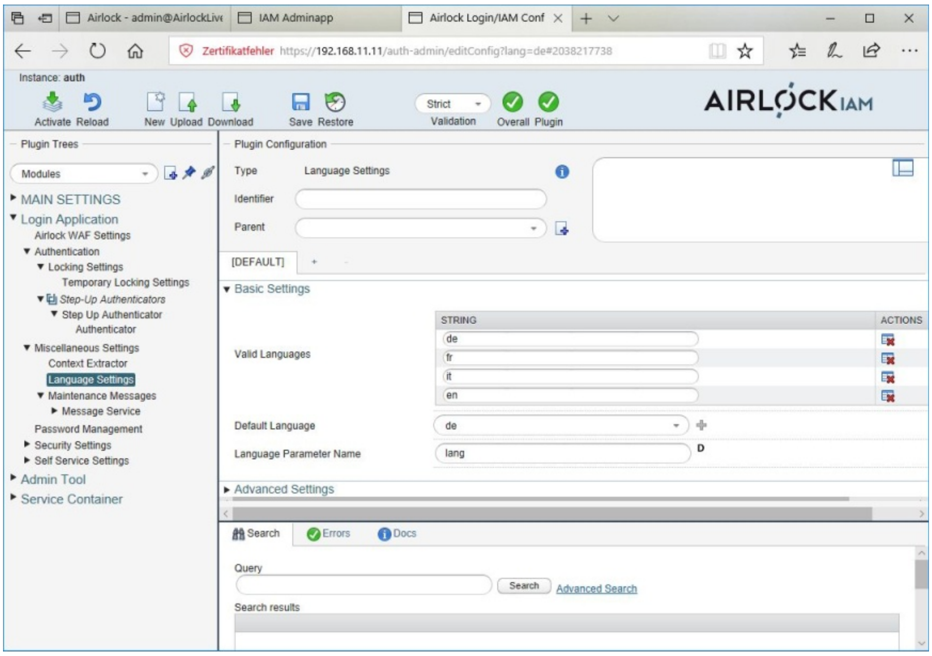
Risk-based authentication

Risk-based authentication is also of interest. The software gathers a lot of data during login procedures, such as geolocation, browser used, cookies, the time and lots more. This information can be used to make logins even more secure. For example the system, if appropriately configured, adds a second login stage to a sign-in if the last successful login was only a short while before but on another continent.

Also of interest: stealth mode. Login applications often provide hackers with useful information, involuntarily. For example, if a hacker tries to log into a web interface with a valid email address and a false password, the login fails and in many instances the interface issues a 'false password' notification. The hacker then knows that this email address exists in the systems and can focus on that one in their attempts to log in. This is why IAM has a 'stealth mode' that, when activated, will not give out any information at all for failed login attempts. This might not be exactly user-friendly but it does raise the security level another notch. This worked perfectly in the test.

Environment staging

Environment staging enables administrators to use various configurations with identical parameters. This makes sense where an existing configuration needs to move from a test environment to a production environment. In this instance, most of the parameters can stay the same but a couple of settings usually change, such as URLs and database server addresses, as they are not the same in the test environment and the production environment. Environment staging makes the job a lot easier, as there is no need to alter the configuration parameters manually. If need be, there is even the option to lock any parameters that are not to be modified, making changes impossible. In this way the system removes sources of error, increasing the level of security. There is also the option to export entire configurations as zip files and upload them in other locations. We didn't have any problems working with environment staging in the test.



https://192.168.11.11/auth-admin/editConfig?lang=de#2038217738

Instance: auth

Type: Language Settings

Identifier:

Parent:

	STRING	ACTIONS
Valid Languages	de	
	fr	
	it	
	en	

Default Language: de

Language Parameter Name: lang

Query:

Search results

The configuration interface for the login application

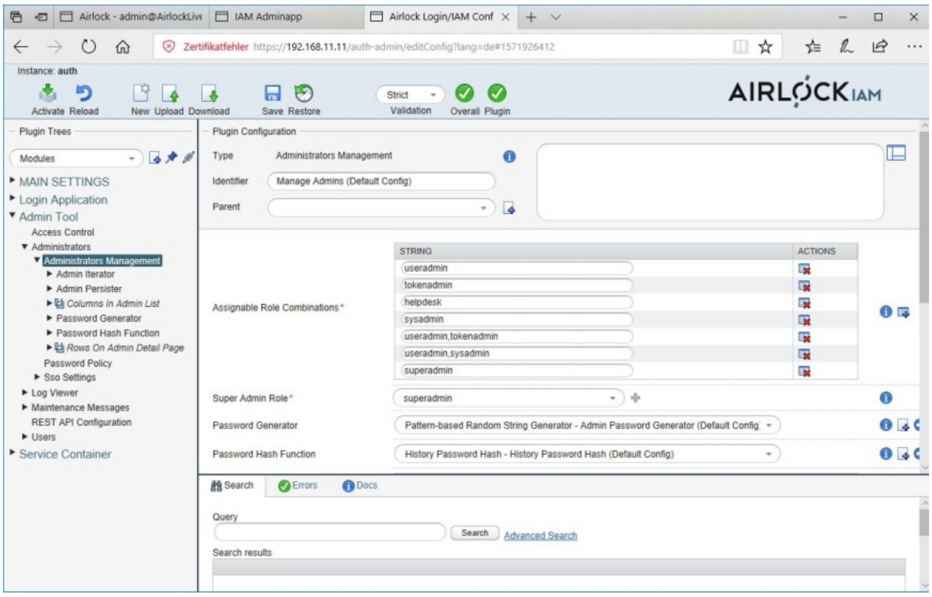
Use cases

For the next stage of the test we used the demo installation with the 'BuggyBook' online bookstore provided by the manufacturer. For our first use case we went to: https://{IP address of the server}/buggybook-withlogin in the test installation and this enabled us to log into the bookstore via the IAM login application. We then tried to log in with a user name and password, using a test account that we had already configured for this specific login method via the web interface. This worked perfectly.

Having ensured the operability of our test installation in this way, we moved on to more complex login methods. In the second use case we wanted to log in with a user name and password, where Google Authenticator protection was also in place. We installed the Google authenticator app on a Huawei P9 android smart phone. We then logged into the bookstore again but, as we were using an account that we had configured with Google authenticator for the purpose of the test, we didn’t land immediately in the bookstore but on a page with a QR code. We scanned it with our smart phone to upload the user in the Google authenticator app. This app then displayed a security code, which we entered into the web interface of the IAM solution and were then authenticated successfully. It isn’t entirely necessary to take the path via the QR code for every login. If the authenticator app knows the user already, the system asks for the security code following the request for the user name and password. In this instance, you still need to check the one that is valid in the app (it changes at brief intervals) and enter it on the website, then access to the bookstore will work.

The third use case was about logging in with mTAN and an activation letter. We used a different user account that we had already set up for this type of authentication. We had the activation letter as a PDF and logged in with user name and password. The system then asked us to provide a mobile phone number and click ‘Register’. An SMS with a code was sent to this mobile number in a production environment and the user was required to enter it later. As we didn’t have an SMS gateway in our test environment, we had to find another way of getting the code. We logged in as an administrator on the IAM configuration tool and checked for the code in the admin app log. We then went back into the login application. It then asked us for the code from the activation letter and the security code from the SMS. Our phone number was then registered in the system and we were able to log in as normal. We still had to log in with user name and password and enter the SMS code then issued by the system. We then had access to the bookstore. This functionality enables managers to check the validity of both the mobile number and the address.

The final use case was about a step-up authentication. This is a good idea if additional security is required for certain actions. In our case we used a step-up authentication with Google Authenticator for extra security with the checkout process with our bookstore virtual shopping cart. We then adapted the mapping within the WAF configuration, as this step-up authentication needs to start when the store’s checkout page is accessed. We then logged in to the bookstore, added a book to our purchases and went to the checkout. The system then asked us for a token. At this point, we checked for the valid token in the authenticator app and then entered it to purchase the book. These examples show that the IAM system can be used along with the WAF to secure many different actions when working with web applications and even to carry out extra authentication stages for critical activities. It all works without having to change the actual configuration of the protected applications.



https://192.168.11.11/auth-admin/editConfig?lang=de#1571926412

Instance: auth

Strict

Type Administrators Management

Identifier Manage Admins (Default Config)

Parent

Assignable Role Combinations*

STRING	ACTIONS
useradmin	
tokenadmin	
helpdesk	
sysadmin	
useradmin,tokenadmin	
useradmin,sysadmin	
superadmin	

Super Admin Role* superadmin

Password Generator Pattern-based Random String Generator - Admin Password Generator (Default Config)

Password Hash Function History Password Hash - History Password Hash (Default Config)

The configuration interface for the administration interface

Conclusion

The Airlock IAM system is very easy to work and implement. It supports a large number of authentication methods that can be combined in any way and also offers a wide range of functions. We were particularly impressed by the self-service functions, as they save the helpdesk a lot of work. The REST API support is also worth mentioning. Finally, automation is increasingly important in today’s IT environment and development is moving further and further away from graphic user interfaces to DevOps and from traditional applications to APIs. The IAM solution covers practically every scenario in this field, providing comprehensive application security when used with the WAF from the same manufacturer.