ORACLE ADAPTIVE ACCESS MANAGER: WHAT, WHY, HOW

Dan Norris, dnorris(at)piocon.com, Piocon

INTRODUCTION

You're a hacker. You want data. Shouldn't be too hard--just guess a password or two and you can find lots of confidential information. If you're lucky, you'll find a weakly-secured web application and watch network traffic as someone logs in to find their username and password. Now that you have the credentials, all you need to do is login. When you attempt login, you find that you're denied access or maybe you're asked to answer a secondary authentication question that wasn't asked to the user you were trying to hack (so you couldn't capture it in your capture of network traffic).

This scene was brought to you by Oracle Adaptive Access Manager (OAAM), a product Oracle acquired with its acquisition of Bharosa in October 2007. OAAM fills the need for a real time fraud detection capability in the Oracle Identity Management Suite. OAAM uses a database of heuristics and pattern matching to find the "bad" guys. What happens to them is up to you, the OAAM administrator.

Typically, fraud detection and strong authentication solutions were only interesting to those industries that handled the most sensitive information or where legislation or competition required it. However, in today's internet-based economy, consumers are asking how stores or businesses are going to protect them from being "hacked" and are likely to choose the more secure business over the one using standard username and password without any additional checks. In order to remain competitive, online businesses (retail stores, banks, healthcare providers, insurance companies, to name a few) must walk the fine line of making the consumer be and feel more protected without making the overall experience more complicated. For example, most consumers wouldn't be willing to carry an authentication token around with them everywhere just so that they could order a book from an online bookstore (not to mention the bookstore's expense and hassle of issuing tokens to customers and potential customers). However, they might be willing to answer a secondary authentication question from time to time when they login from a hotel while on a trip instead of their normal login location (at home).

In the sections that follow, you will learn about:

- Adaptive strong authenticators
- Adaptive risk management engine
- Response options
- Deployment considerations

ADAPTIVE STRONG AUTHENTICATORS

OAAM includes adaptive strong authenticators that can be easily deployed and used by any web application. To achieve traditional multi-factor authentication, companies invested in significant infrastructure, often token-based, to ensure security for web-based applications. OAAM provides the same strength with much lower entry and maintenance costs, plus it enables multi-factor authentication to be employed for individuals that typically were not considered candidates for corporate token rollouts such as customers or short-term partners. As a result, the systems protected with OAAM's adaptive strong authenticators are safer and more accessible to more people than traditional multi-factor methods. The adaptive strong authenticators are so powerful that some OAAM deployments only use these features and don't use the risk management features—at least not at first. Since OAAM is relatively easy to deploy with existing applications and has few prerequisites, it's got a good business case with a relatively short break-even ROI. And while it is difficult to measure the perceived value from the end user or customer perspective, it's logical to conclude that there is some positive impact and may offer some competitive edge over other, less secure sites.

Let's review the technology behind the adaptive strong authenticators and the built-in authenticators available with OAAM.

TECHNOLOGIES EMPLOYED

So, how do they do it? Well, it turns out that there are some very smart and creative individuals on the development teams. While the specific details are obviously secret and some technologies are patented or patent pending. The most impressive part of the product to me is the fact that they use the same technologies that most other web-based applications employ, but they also maintain the highest security levels. The product complies with many security standards including FIPS 140-2 and has been reviewed by many security-conscious companies and agencies of the U.S. federal government.

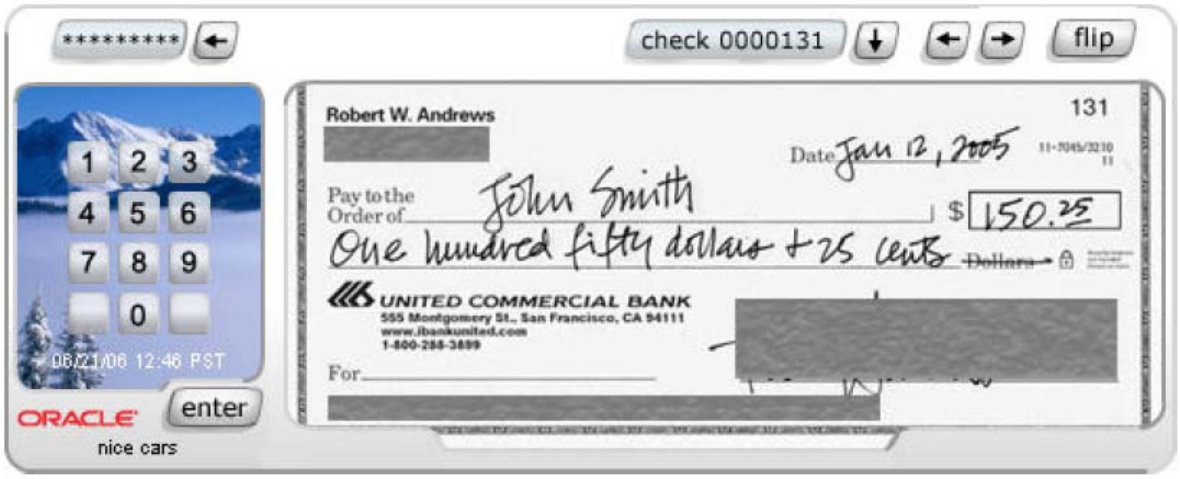
The authenticators are supported in browsers running on Windows, Linux, MacOS, and Solaris. Not all authenticators use all technologies, but some of them do use multiple technologies from the following list:

- DHTML
- Java applets
- Flash
- Adobe SVG

AUTHENTICATION METHODS

In this section, we'll review each authenticator, what it does, and typical usage scenarios. All the authenticators guard against phishing by providing some server authentication through the background image and phrase displayed as part of the authenticator. Most of the authenticators (all but the slider) also include the current date and time, so they can't be captured and redisplayed.

CHECKPAD



The CheckPad authenticator specifically deals with Check21 legislation. This law allows banks to transmit scanned facsimiles of checks for payment instead of actually shipping the physical checks to the institution holding the account. While this change allowed banks to obtain payment of their checks more quickly, it also resulted in all checks being digitized and stored online which makes them much easier for the "bad" guys to obtain. Checks contain vital information like account numbers, check numbers, and signatures.

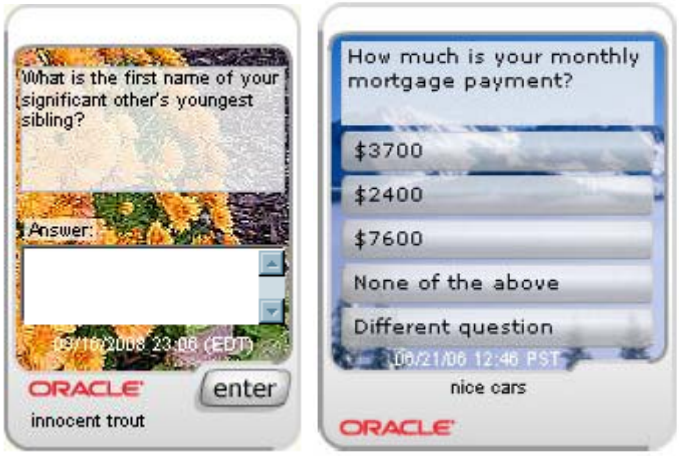
CheckPad presents check images with some sections obscured. Most often, the customer viewing their own checks don't need to see the account number, their own address, or the signature, so those fields are obscured. Additionally, when coupled with the other authenticators, even seeing the check image can be protected with a secondary authentication like the PinPad (as shown above).

DOCPAD

The DocPad authenticator is helpful for those cases where a sensitive document needs to be protected from unauthorized access. Some examples of sensitive documents include medical records, banking documents, mortgage documents (more often being archived online). The DocPad is a more generalized version of the CheckPad authenticator where a question must be answered or a PIN must be entered in order to view the document.

When combined with another authenticator like the KeyPad or QuestionPad, the DocPad provides secondary authentication protection for the most sensitive documents.

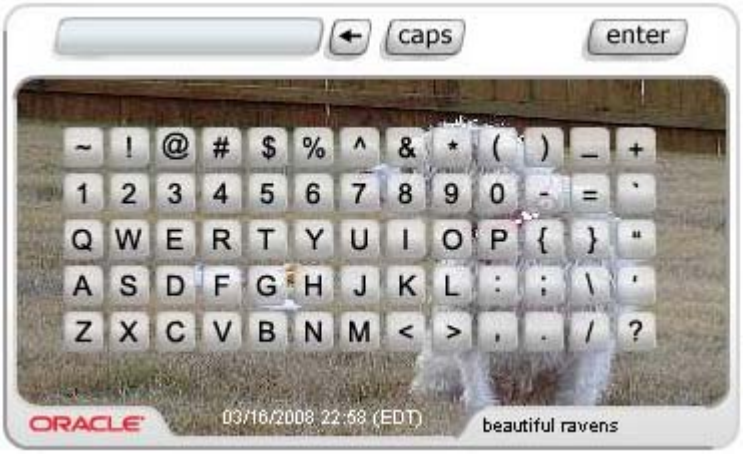
QUESTIONPAD AND QUIZPAD



The QuestionPad and QuizPad authenticators allow a question to be presented to the end user in order to ensure that they are who they say they are. Typically, these authenticators will be secondary authentication and maybe used to ask a user the name of the street they grew up on or their dog's name. In order to use these effectively, the question and answer pairs would have to be entered by the user when they enroll or be generated from some other data that the site would know about the user. For example, if the site is a credit card issuer, then the question could be to enter the CVV code (the 3-digit code) on the bank of the credit card. Or, the site may prompt the user for the amount of their last payment. The questions and sources for answers are configurable.

The difference between the two authenticators is that QuestionPad prompts for an answer in a text field while the QuizPad presents multiple choices for the user to choose from.

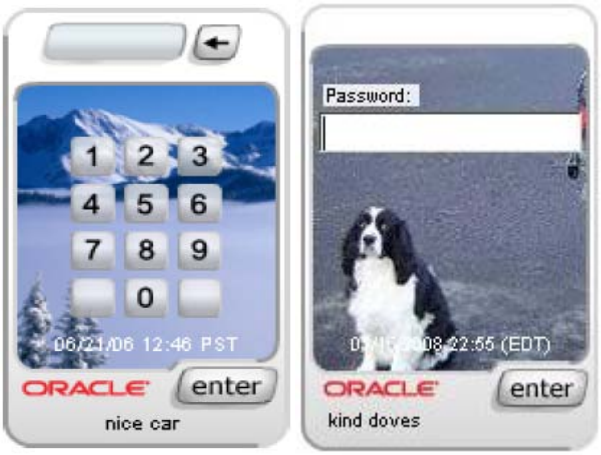
KEYPAD



The KeyPad authenticator presents a keyboard-like image to the user for entering alphanumeric information. The user is forced to click on the buttons in the image instead of using the keyboard. This defeats the keyboard loggers and because the image is presented in a different screen location each time, screen scraping is also much less effective as a hacking method.

While the most common uses for KeyPad are entering passwords or keys, any sensitive alphanumeric data can be entered with the KeyPad device. Given the complexity of clicking on each character, its use won't be widespread except for the most sensitive data.

PinPad and TextPad



The PinPad is similar to the KeyPad except that it just includes number keys. The TextPad allows standard password entry in an authenticator widget. It is still stronger than typical password entry because it authenticates the server to the client using the background image and key phrase that appears in the authenticator, making phishing more difficult. Additionally, the authenticators also encrypt the responses themselves, so when used in conjunction with SSL, it is effectively doubling the encryption, making attacks more difficult.

Slider

The Slider is perhaps the most unique authenticator consisting of symbols matched up with letters and numbers. This virtual device makes it nearly impossible for prying eyes to capture your password. Since the keyboard isn't used, keyboard loggers aren't a threat and since it's displayed in a different location with a different set of symbol/letter pairs each time, screen scrapers are also rendered useless.

Deployment Options

OAAM can be deployed in one of two different ways. The first option is to configure an existing application to use the libraries from OAAM to perform the necessary encoding and decoding for handling the authenticators. This option may be favored by environments only using the authenticators or those not inclined to perform a complete deployment. Often, especially when using OAAM with one or a small number of applications, the additional overhead of another complete environment isn't justified in terms of cost of the additional hardware and/or maintenance requirements.

The second deployment option is the more traditional method of hosting OAAM components in their own middle tier environment with the OAAM web front end deployed as an application. The OAAM interfaces can be customized for look and feel consistency and branding with appropriate logos and the like. This configuration incurs the additional overhead of deploying and maintaining a new application server environment, but also offloads some of the OAAM processing to a centralized, dedicated platform instead of having it be co-deployed with each application that is OAAM-integrated.

Adaptive Risk Manager

The Adaptive Risk Manager is the OAAM component that provides real-time fraud detection using a method of scoring that accounts for many factors and allows many different types of responses. The risk to the application and the data in the application is scored based on such unique and complex factors that attacks are much more difficult and ultimately less successful than traditional methods.

RISK SCORING

Every access to the application, at logon or during normal application navigation and use, is scored. The score includes whatever factors you configure it to consider for each request or request type. Once a score is assigned, another set of rules takes over to determine what will be done with the request. The response options are reviewed later in this document. Next, let's review some of the criteria that make up the assigned score.

RISK EVALUATION CRITERIA

The unique and proprietary fraud detection scoring is based on many factors. In this section, we'll review each of the factors and how they could be used to identify fraudulent activity.

LOGON LOCATION

One criterion used to compute a risk score is the logon location. The logon location takes in to account many factors such as IP address, IP routing type, ASN, top-level domain, information from Geo databases, and ISP flag. Some of these attributes are also matched with a confidence factor to signify the likely accuracy of the information. After all, an IP address registered to a US-based organization may be used by a network in Egypt.

This criteria allows you to assess risk based on factors such as a user that consistently access from one of two IP address ranges (maybe home and work), but suddenly makes access from eastern Europe at 3am. That type of access wouldn't necessarily cause their request to be rejected, but might trigger some other action to re-verify the user's identity.

DEVICE FINGERPRINTING

The risk management engine also "learns" what devices you use to access the applications by gathering information about the browser, operating system, and display you use to view the application's pages. These attributes can be combined with secure cookies and Flash objects to create a unique signature for your computer so that it can be identified and tracked as an input to the risk management engine. Any variation in the device fingerprint such as a different browser, different version of Flash, different IP address, different video resolution, or different operating system version will be detected as a new device. Rules can then be created in the risk manager to address what happens when the user is accessing the application from a new or different device. It may be common to introduce additional challenges to the user in order to confirm their identity.

APPLICATION ACTIVITY

The risk manager can also be natively integrated with the application such that it provides advanced protection for certain application activities such as balance transfers for a banking application or stock trading for an online trading application. When operating in proxy mode, it is more difficult for OAAM to detect certain application actions, so native integration is best when intending to have OAAM review specific application activity.

RESPONSE OPTIONS

Along with the scoring configuration, you also configure responses based on the resulting score.

ADDITIONAL OR SECONDARY AUTHENTICATION

The most common response from a rule being triggered would be to request additional or secondary authentication. While a username and password may have been used to confirm identity initially, the secondary authentication may consist of submitting a certificate to the application or supplying an alternate PIN number. This secondary authentication may or may not use one of the secure authenticators that come with OAAM. Other authentication mechanisms can be added with some API programming.

MESSAGING VIA SMS OR EMAIL

It is possible to configure OAAM to send messages via email or SMS in order to ensure proper authentication. Specifically, using SMS may be stronger than some other mechanisms due to the need to possess a cell phone to receive the message. Since most people keep fairly close watch on their cell phone, this method almost becomes a 2-factor authentication mechanism.

CHALLENGE QUESTIONS

Using challenge questions is a common practice in many online websites today. The process involves a registration event during which the user is required to choose 3-5 questions from a question pool and then submit answers for each question. The questions should generally be things that are not publicly known, not easily derived, and the answers to which are known

only by the user. These question/answer combinations can be used later to confirm the user's identity by making sure that they can submit matching answers to those submitted during registration. While 3-5 questions are commonly gathered from the user during registration, it is customary to choose one of those questions at random to display during a challenge.

Combinations

As you might expect, since each of these methods alone provides a strong chance that fraudulent activity will be thwarted, combining two or more of these actions provides even more protection. For example, when accessing the application from a new device (as detected by the device fingerprint), a challenge question may be presented in order to enter the application (after initial authentication). If that challenge is successful, then later the user may attempt a funds transfer which requires a secondary authentication using a PIN code.

In such a case, the user experience would be slightly disrupted, but if the users are educated about why they're registering and later being forced to provide multiple authentications, they will realize that if they aren't presented with these additional challenge events, their account may be more susceptible to fraudulent activity.

Deployment Considerations

When considering OAAM deployment options, one of the first choices you'll need to make is whether your deployment will be tightly integrated with the application or if it will utilize a proxy-based mechanism, known as the Universal Install Option or UIO, to minimize the application integration in favor of a quicker and likely easier deployment. In most cases, you'll gain more flexibility and potentially some security by using a deployment integrated with the application, known as a native integration.

Native Integration

OAAM offers two types of native integration. The first type uses SOAP messages and a wrapper to abstract the actual API calls from the application. The application then just has to make the appropriate SOAP call to interact with OAAM components. This integration is best when the application administrators do not want to include the OAAM libraries in the application directly and is also beneficial since the application programmers do not have to learn the specific OAAM API calls. However, there are two distinct disadvantages to this approach: a) SOAP has significant XML overhead and b) there is a dependence on the ARM web service to be available to handle SOAP calls.

The second type of native integration is known as static integration. This involves linking the OAAM API library with the application (.NET, Java, and C++ APIs are available). The application programmers must first learn the OAAM API calls and responses and then properly embed those into the application at the appropriate locations. The benefits of this type of integration are that you avoid the SOAP and XML overhead as well as remove the dependence on the web service so that static native integration can operate even when the ARM web service endpoint is not available.

Universal Install Option Proxy Deployment

If native integration is not an option, then the universal install option (UIO) proxy deployment is the other method to employ OAAM with an application. This deployment consists of placing the UIO proxy module on a proxy server that is situated in "front" of the application and requiring all access to the application to pass through the UIO proxy. The UIO proxy evaluates each request and also employs the other features discussed earlier like device and location fingerprinting to determine a risk score. The UIO proxy functions as any other deployment by challenging the user with a secondary authentication, challenge questions or other actions when rules dictate.

The UIO deployment can be used with almost any application. While OAAM can be customized to provide some authentication, it is not normally used for authentication. Instead, OAAM is typically configured to authenticate users using the normal authentication mechanism used prior to OAAM deployment. For UIO deployments, OAAM can be configured to post the username and password (which OAAM can gather from the user) to the web-based login page of the application which it is protecting. The UIO proxy will be configured to recognize a successful authentication and allow the user access accordingly.

References

- Oracle US Commercial Price List dated January 6, 2006, http://www.oracle.com/corporate/pricing/pricelists.html
- Oracle Adaptive Access Manager – Business White Paper, October, 2007, http://www.oracle.com/technology/products/id_mgmt/oaam/pdf/wp_oaam.pdf
- Oracle Adaptive Access Manager 10gR3 Workshop (10.1.4.0.1), March, 2008

FROM THE LAWYERS

The information contained herein should be deemed reliable but not guaranteed. The author has made every attempt to provide current and accurate information. If you have any comments or suggestions, please contact the author at dnorris(at)piocon.com.