One Time Passwords via an SMS Gateway with BIG-IP Access Policy Manager

Jason Rahm, 2011-08-02

One time passwords, or OTP, are used (as the name indicates) for a single session or transaction. The plus side is a more secure deployment, the downside is two-fold—first, most solutions involve a token system, which is costly in management, dollars, and complexity, and second, people are lousy at remembering things, so a delivery system for that OTP is necessary. The exercise in this tech tip is to employ BIG-IP APM to generate the OTP and pass it to the user via an SMS Gateway, eliminating the need for a token creating server/security appliance while reducing cost and complexity.

Getting Started

This guide was developed by F5er Per Boe utilizing the newly released BIG-IP version 10.2.1. The "-secure" option for the mcget command is new in this version and is required in one of the steps for this solution. Also, this solution uses the Clickatell SMS Gateway to deliver the OTPs. Their API is documented at http://www.clickatell.com/downloads/http/Clickatell_HTTP.pdf. Other gateway providers with a web-based API could easily be substituted. Also, there are steps at the tail end of this guide to utilize the BIG-IP's built-in mail capabilities to email the OTP during testing in lieu of SMS. The process in delivering the OTP is shown in Figure 1.

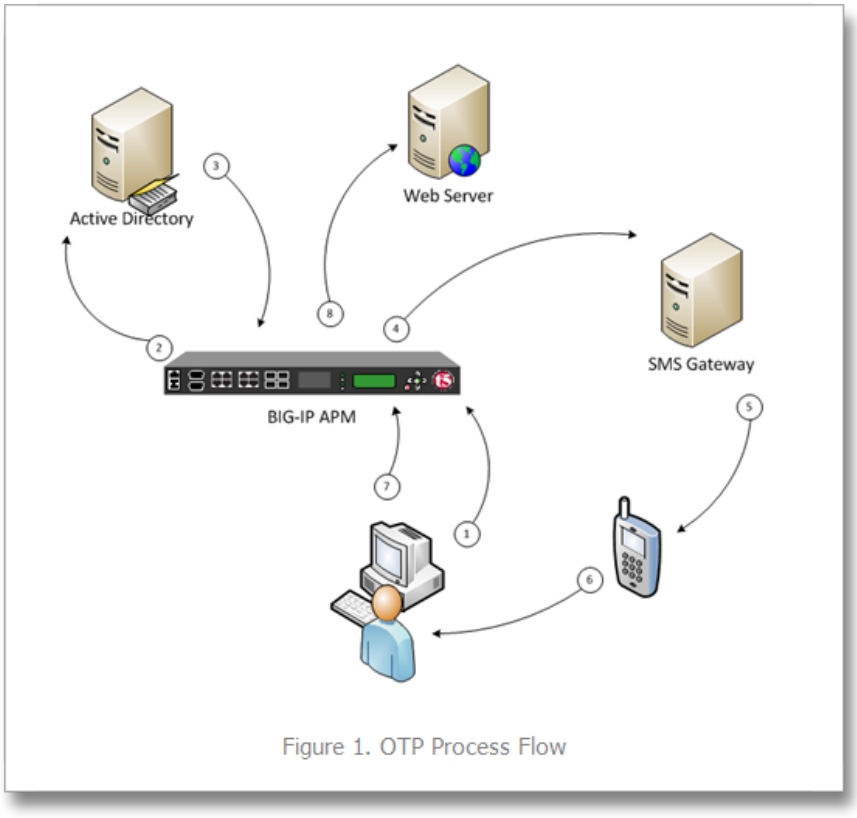


Figure 1. OTP Process Flow

First a request is made to the BIG-IP APM. The policy is configured to authenticate the user's phone number in Active Directory, and if successful, generate a OTP and pass along to the SMS via the HTTP API. The user will then use the OTP to enter into the form updated by APM before allowing the user through to the server resources.

BIG-IP APM Configuration

Before configuring the policy, an access profile needs to be created, as do a couple authentication servers. First, let's look at the authentication servers

Authentication Servers

Authentication Servers

To create servers used by BIG-IP APM, navigate to Access Policy->AAA Servers and then click create. This profile is simple, supply your domain server, domain name, and admin username and password as shown in Figure 2.

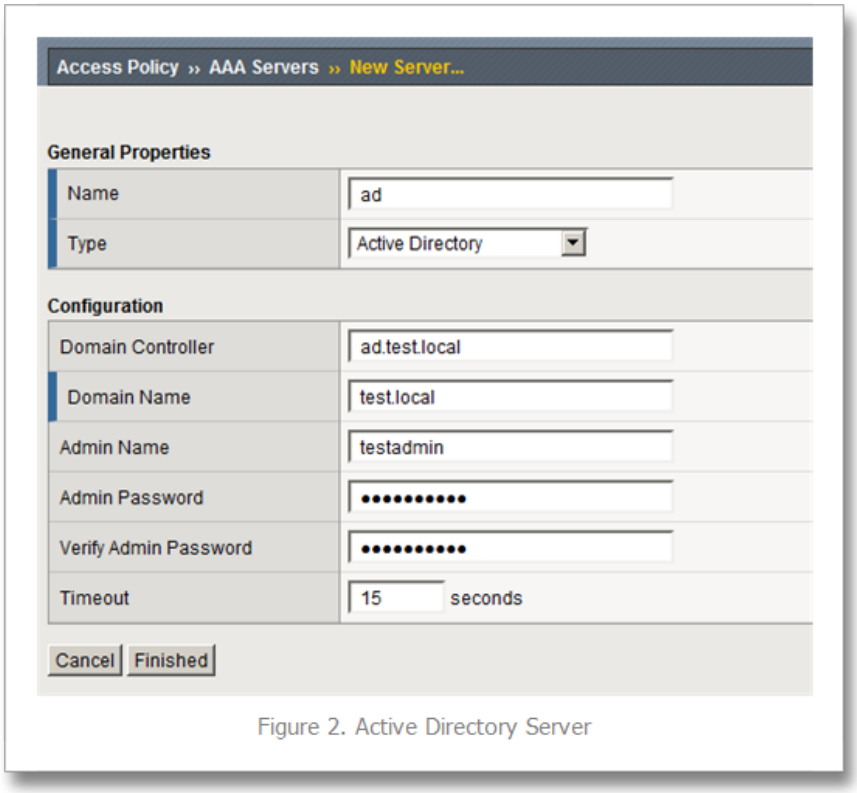


Figure 2. Active Directory Server

The other authentication server is for the SMS Gateway, and since it is an HTTP API we're using, we need the HTTP type server as shown in Figure 3.

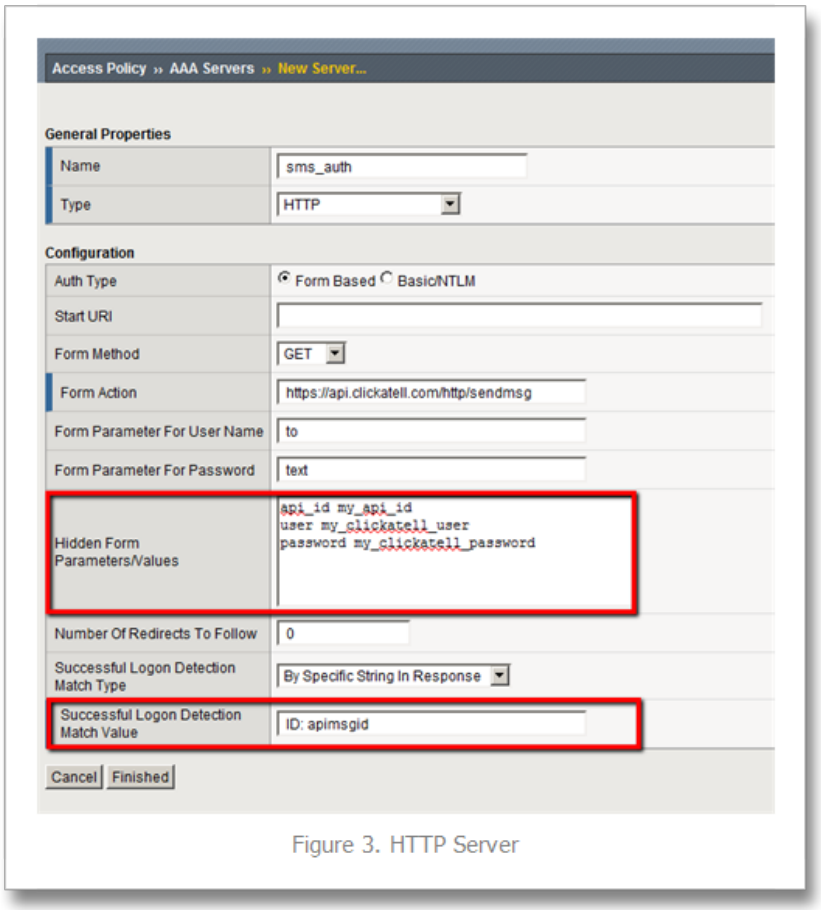


Figure 3. HTTP Server

Note that the hidden form values highlighted in red will come from your Clickatell account information. Also note that the form method is GET, the form action references the Clickatell API interface, and that the match type is set to look for a specific string. The Clickatell SMS Gateway expects the following format:

> https://api.clickatell.com/http/sendmsg?
> api_id=xxxx&user=xxxx&password=xxxx&to=xxxx&text=xxxx

Finally, successful logon detection value highlighted in red at the bottom of Figure 3 should be modified to response code returned from SMS Gateway. Now that the authentication servers are configured, let's take a look at the access profile and create the policy.

Access Profile & Policy

Before we can create the policy, we need an access profile, shown below in Figure 4 with all default settings.

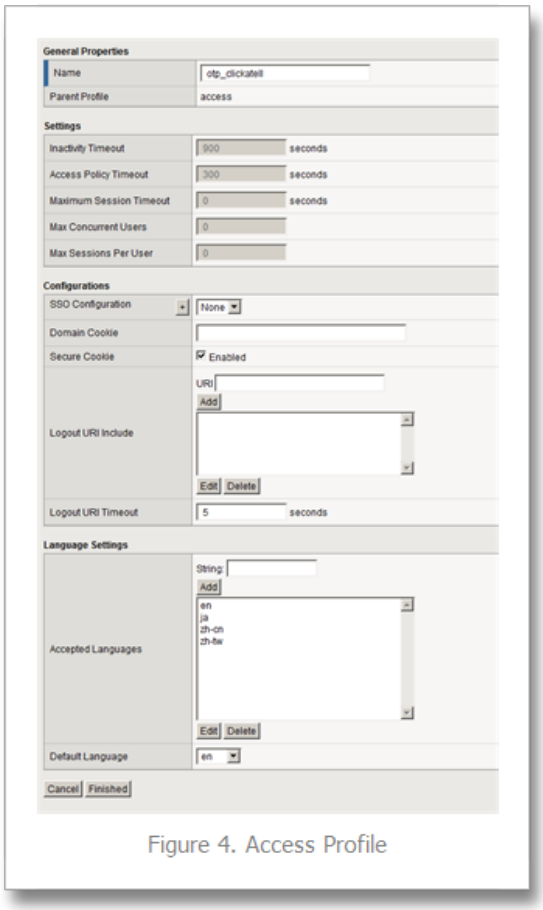


Figure 4. Access Profile

Now that that is done, we click on Edit under the Access Policy column highlighted in red in Figure 5.

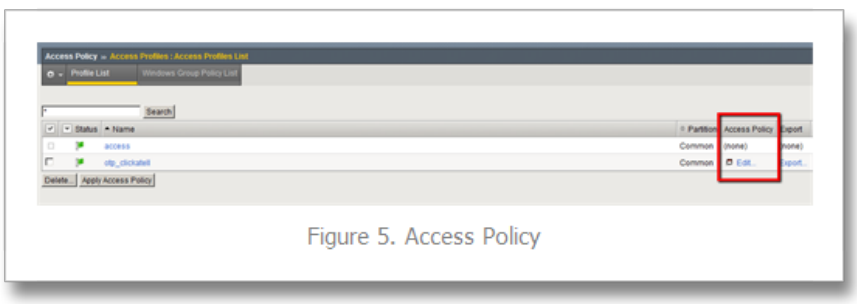


Figure 5. Access Policy

The default policy is bare bones, or as some call it, empty. We'll work our way through the objects, taking screen captures as we go and making notes as necessary. To add an object, just click the "+" sign after the Start flag. The first object we'll add is a Logon Page as shown in Figure 6. No modifications are necessary here, so you can just click save.

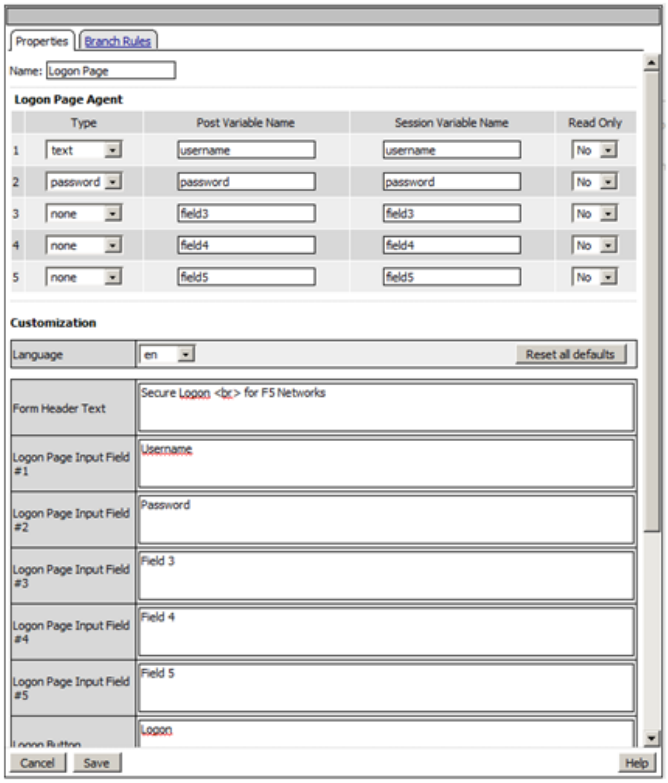


Figure 6. Logon Page

Next, we'll configure the Active Directory authentication, so we'll add an AD Auth object. Only setting here in Figure 7 is selecting the server we created earlier.

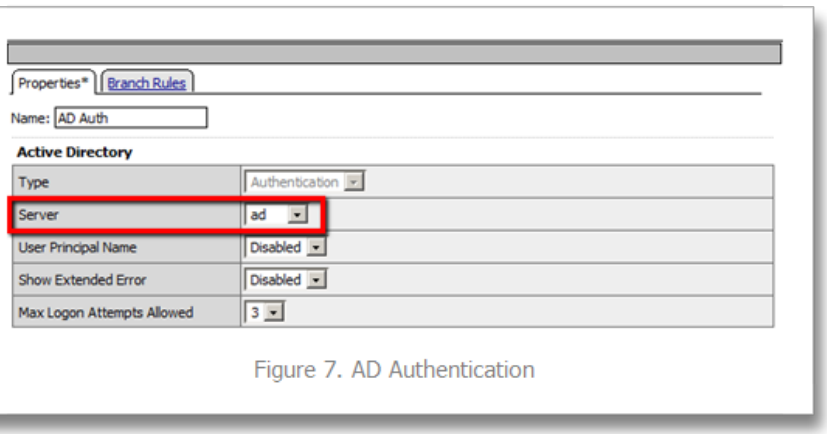


Figure 7. AD Authentication

Following the AD Auth object, we need to add an AD Query object on the AD Auth successful branch as shown in Figures 8 and 9. The server is selected in the properties tab, and then we create an expression in the branch rules tab. To create the expression, click change, and then select the Advanced tab.

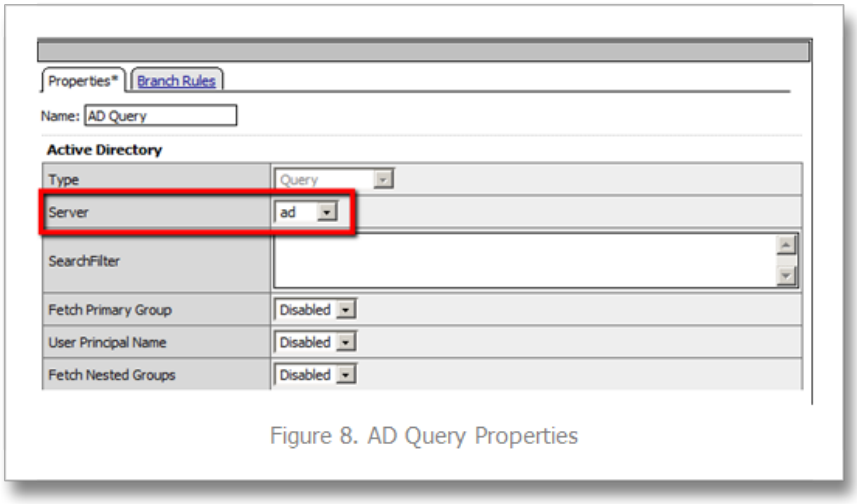


Figure 8. AD Query Properties

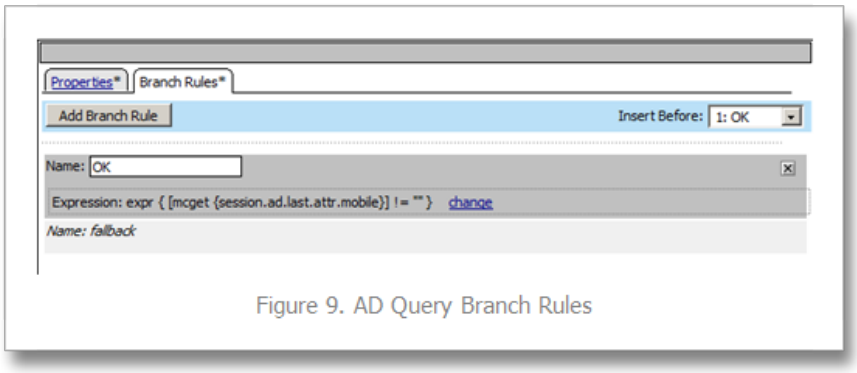


Figure 9. AD Query Branch Rules

The expression used in this AD Query branch rule:

```
expr { [mcget {session.ad.last.attr.mobile}] != "" }
```

Next we add an iRule Event object to the AD Query OK branch that will generate the one time password and provide logging. Figure 10 Shows the iRule Event object configuration.

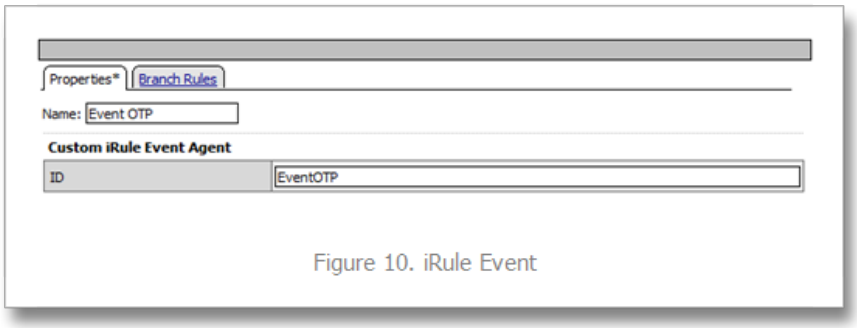


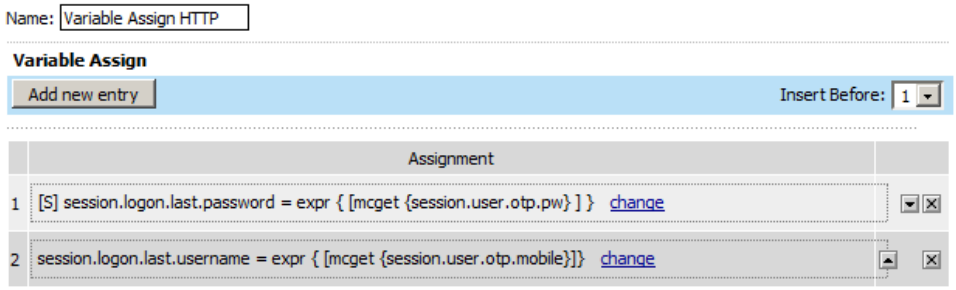
Figure 10. iRule Event

The iRule referenced by this event is below. The logging is there for troubleshooting purposes, and should probably be disabled in production.

```
1: when  ACCESS_POLICY_AGENT_EVENT {
2:     expr srand([clock clicks])
3:     set otp [string range [format "%08d" [expr int(rand() * 1e9)]] 1 6 ]
4:     set mail [ACCESS::session data get "session.ad.last.attr.mail"]
5:     set mobile [ACCESS::session data get "session.ad.last.attr.mobile"]
6:     set logstring mail,$mail,otp,$otp,mobile,$mobile
7:     ACCESS::session data set session.user.otp.pw $otp
8:     ACCESS::session data set session.user.otp.mobile $mobile
9:     ACCESS::session data set session.user.otp.username [ACCESS::session data get "session.logon.last.username"]
10:    log local0.alert "Event [ACCESS::policy agent_id] Log $logstring"
11: }
12:
13: when ACCESS_POLICY_COMPLETED {
```

```
14:     log local0.alert "Result: [ACCESS::policy result]"
15: }
```

On the fallback path of the iRule Event object, add a Variable Assign object as show in Figure 10b. Note that the first assignment should be set to **secure** as indicated in image with the [S].



The expressions in Figure 10b are:

> session.logon.last.password = expr { [mcget {session.user.otp.pw}]}
>
> session.logon.last.username = expr { [mcget {session.user.otp.mobile}]}

On the fallback path of the AD Query object, add a Message Box object as shown in Figure 11 to alert the user if no mobile number is configured in Active Directory.

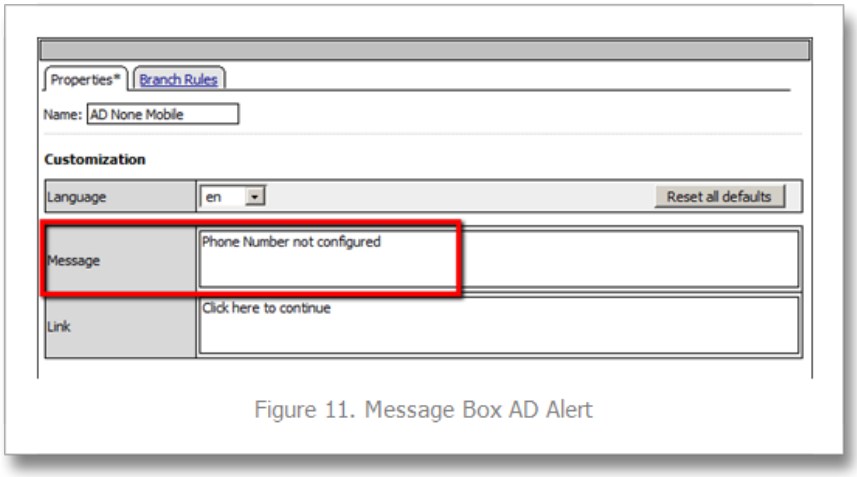


Figure 11. Message Box AD Alert

On the fallback path of the Event OTP object, we need to add the HTTP Auth object. This is where the SMS Gateway we configured in the authentication server is referenced. It is shown in Figure 12.

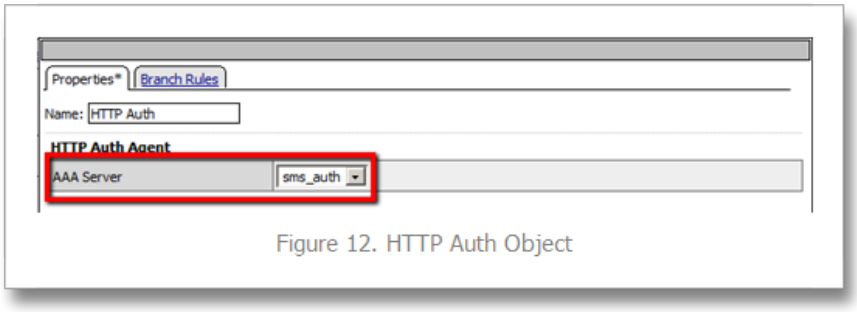


Figure 12. HTTP Auth Object

On the fallback path of the HTTP Auth object, we need to add a Message Box as shown in Figure 13 to communicate the error to the client.

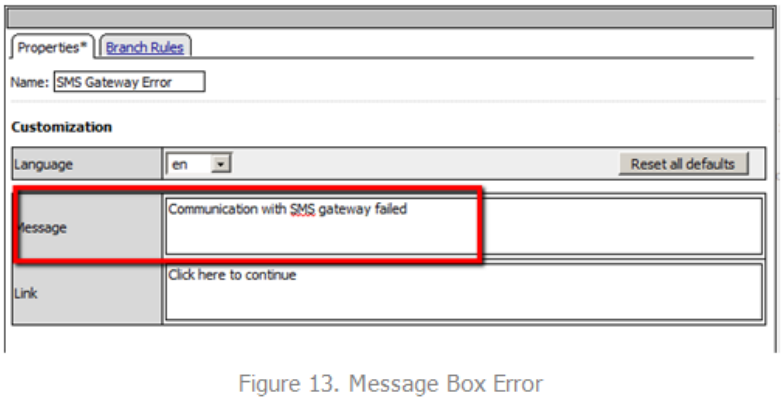


Figure 13. Message Box Error

On the Successful branch of the HTTP Auth object, we need to add a Variable Assign object to store the username. A simple expression and a unique name for this variable object is all that is changed. This is shown in Figure 14.

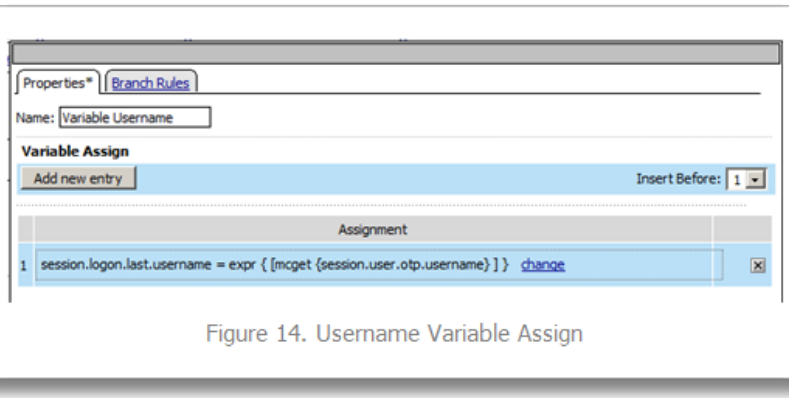


Figure 14. Username Variable Assign

On the fallback branch of the Username Variable Assign object, we'll configure the OTP Logon page, which requires a Logon Page object (shown in Figure 15). I haven't mentioned it yet, but the name field of all these objects isn't a required change, but adding information specific to the object helps with readability. On this form, only one entry field is required, the one time password, so the second password field (enabled by default) is set to none and the initial username field is changed to password. The Input field below is changed to reflect the type of logon to better queue the user.

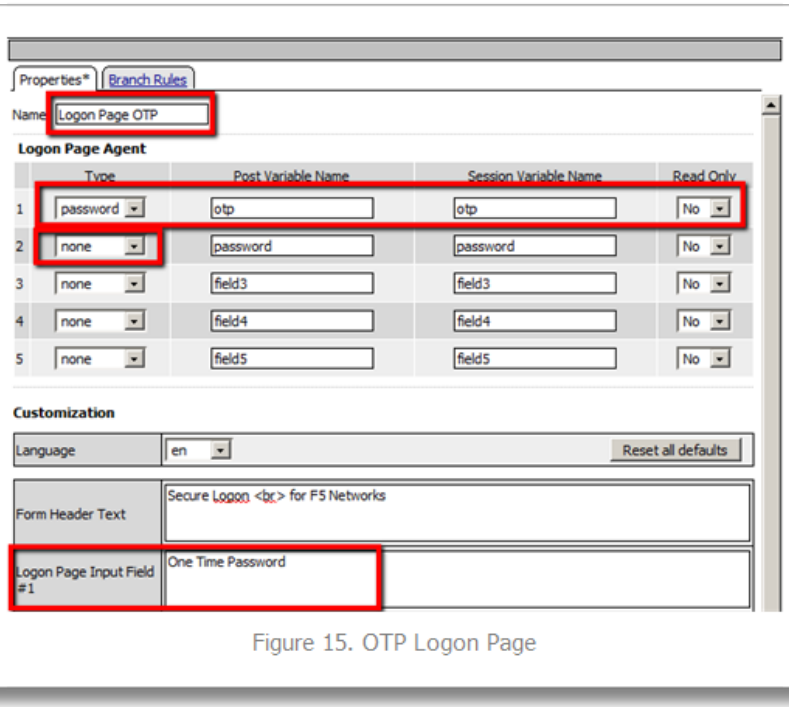


Figure 15. OTP Logon Page

Finally, we'll finish off with an Empty Action object where we'll insert an expression to verify the OTP. The name is configured in properties and the expression in the branch rules, as shown in Figures 16 and 17. Again, you'll want to click advanced on the branch rules to enter the expression.

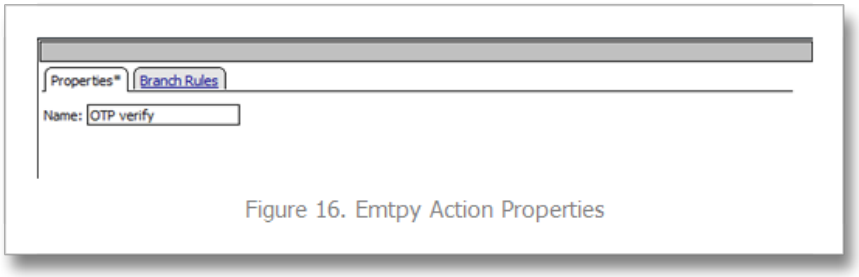


Figure 16. Emtpy Action Properties

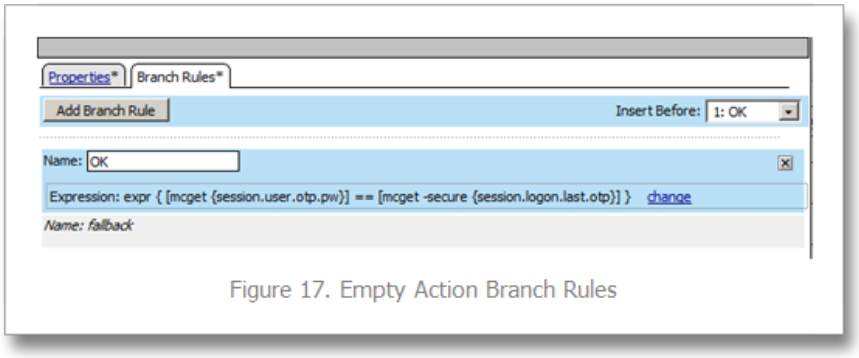
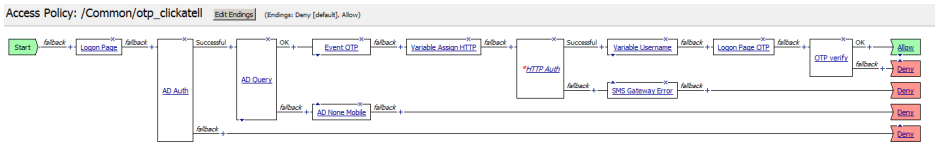


Figure 17. Empty Action Branch Rules

The expression used in the branch rules above is:

```
expr { [mcget {session.user.otp.pw}] == [mcget -secure {session.logon.last.otp}] }
```

Note again that the –secure option is only available in version 10.2.1 forward. Now that we're done adding objects to the policy, one final step is to click on the Deny following the OK branch of the OTP Verify Empty Action object and change it from Deny to Allow. Figure 18 shows how it should look in the visual policy editor window.



Now that the policy is completed, we can attach the access profile to the virtual server and test it out, as can be seen in Figures 19 and 20 below.

Figure 19. User Logon

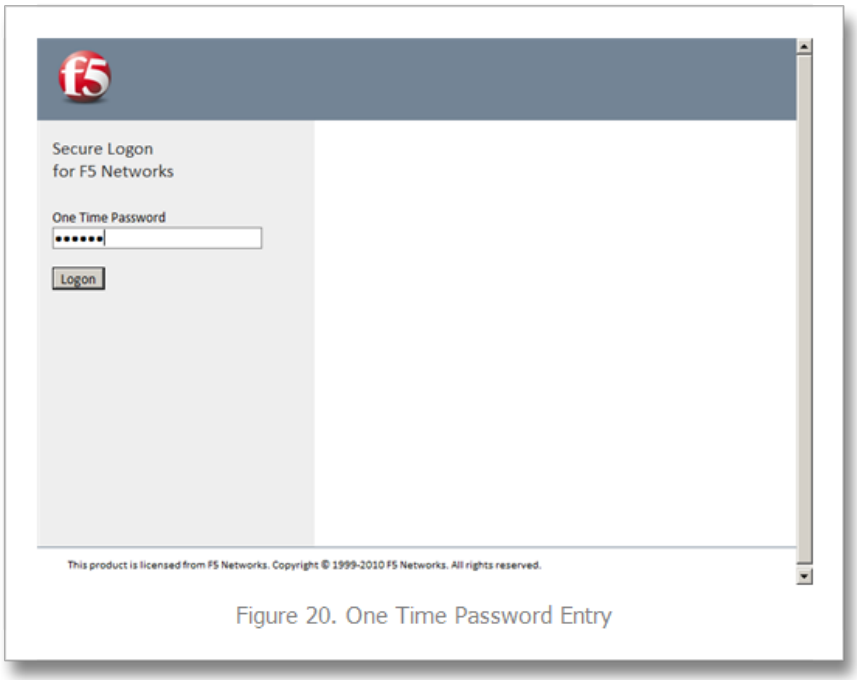


Figure 20. One Time Password Entry

Email Option

If during testing you'd rather send emails than utilize the SMS Gateway, then configure your BIG-IP for mail support (Solution 3664), keep the Logging object, lose the HTTP Auth object, and configure the system with this script to listen for the messages sent to /var/log/ltm from the configured Logging object:

```
#!/bin/bash
while true
do
  tail -n0 -f /var/log/ltm | while read line
  do
    var2=`echo $line | grep otp | awk -F'[,]' '{ print $2 }'`
    var3=`echo $line | grep otp | awk -F'[,]' '{ print $3 }'`
    var4=`echo $line | grep otp | awk -F'[,]' '{ print $4 }'`
    if [ "$var3" = "otp" -a -n "$var4" ]; then
      echo Sending pin $var4 to $var2
      echo One Time Password is $var4 | mail -s $var4 $var2
    fi
  done
done
```

The log messages look like this:

```
Jan 26 13:37:24 local/bigip1 notice apd[4118]: 01490113:5: b94f603a: session.user.otp.log is
mail,user1@home.local,otp,609819,mobile,12345678
```

The output from the script as configured looks like this:

```
[root@bigip1:Active] config # ./otp_mail.sh
Sending pin 239272 to user1@home.local
```

Conclusion

The BIG-IP APM is an incredibly powerful tool to add to the LTM toolbox. Whether using the mail system or an SMS gateway, you can take a bite out of your infrastructure complexity by using this solution to eliminate the need for a token management service. Many thanks again to F5er Per Boe for this excellent solution!

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com