v11: DNS Express &ndash; Part 2

George Watkins, 2011-18-10

Introduction

In our last Tech Tip, v11: DNS Express – Part 1, we discussed configuring DNS Express as an authoritative slave DNS server. We also discussed the advantages of using DNS Express in place of a pool of BIND servers. In this part of the series we will be discussing using a Transactional SIGnatures (TSIG) to secure zone transfers form our BIND server to the GTM. By implementing TSIGs for our zone transfers, we can ensure that no one could potentially poison the zone date of our DNS Express virtual server by impersonating our master server.

First of all, let's talk a little bit about what TSIGs accomplish as well as what they are and what they're not. Transactional signatures are used to sign DNS messages being exchanged between two hosts. This ensures that no middleman can manipulate a message sent from one host to another. A TSIG uses a hash-based message authentication code (HMAC-MD5), which comprises a one-way hash and shared secret, to sign a message sent to the destination host. The destination host can then compared their computed hash with that of the sender and ensure the request originated from that friendly host.

While a TSIG can verify the authenticity of a DNS message, it does not provide encryption. All records can be read off the wire in plain text, but cannot be manipulated by a third party without causing an error on the host receiving the message. This fine under almost all circumstances because DNS information does not need to be hidden, it just needs to maintain its integrity, which a TSIG provides.

For this Tech Tip, we will cover how to generate a TSIG, configure our BIND server and GTM to use it, do some testing and discuss some troubleshooting techniques.

Securing DNS Express Zone Transfers with a TSIG

This Tech Tip's walkthrough section depends on some of the configuration performed in Part 1 of the series. You should configure the master zone listed in Part 1 prior to proceeding with this section.

Generating The TSIG

1. We'll start by generating the symmetric key on our BIND server. The key generator will require some entropy to generate it, so grab an audio file (or something similarly random) and copy it over to the BIND server. This may not be necessary if there is a lot of activity occurring on the system, but an idle system will lack the necessary randomness to generate a key. The dnssec-keygen utility will generate a 256-bit symmetric key and output the contents to two files.

```
root@f5-test-bind-master:~# dnssec-keygen -a HMAC-MD5 -b 256 -n HOST -r /tmp/Beethoven\'s\ 5th.mp3 ho
```

This command will output two files: Khost1_to_host2_test_tsig.+157+55433.key and Khost1_to_host2_test_tsig.+157+55433.private. If we crack these two files open, we'll see that they contain the same symmetric key.

```
/etc/bind/Khost1_to_host2_test_tsig.+157+55433.key
-----
host1_to_host2_test_tsig. IN KEY 512 3 157 q0MMaG8ayhBlDdIOdPuRf38MnV2uqO0xzCb/t7GQbso=



/etc/bind/Khost1_to_host2_test_tsig.+157+55433.private
-----
Private-key-format: v1.3
Algorithm: 157 (HMAC_MD5)
Key: q0MMaG8ayhBlDdIOdPuRf38MnV2uqO0xzCb/t7GQbso=
Bits: AAA=
Created: 20111012222810
Publish: 20111012222810
Activate: 20111012222810
```

2. Next we'll want to add this key to our BIND configuration so we'll create another file called tsig.key in /etc/bind/. This file will hold all the TSIG keys for this server. Inside this file we'll define the key and the BIG-IP's self-IP. Be sure and set the permission and ownership to 660 and root:bind, respectively. No ordinary user should ever be able to read the key files.

```
/etc/bind/tsig.key
-----
key host1_to_host2_test_tsig {
  algorithm hmac-md5;
  secret "q0MMaG8ayhBlDdIOdPuRf38MnV2uqO0xzCb/t7GQbso=";
};

# BIG-IP self-IP
server 192.168.1.245 {
  keys { host1_to_host2_test_tsig; };
};



root@f5-test-bind-master:~# chown root:bind /etc/bind/tsig.key; chmod 660 /etc/bind/tsig.key
```

Note: The key name must match the name we provided to the dnssec-keygen utility or else it will not work.

3. BIND will need to know about the tsig.key file so we'll include this file in our named.conf. Add the include line below the other three usual suspects.

```
/etc/bind/named.conf
-----
include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
include "/etc/bind/tsig.key";
```

4. Next we'll want to enable zone transfers using our TSIG key. We'll want to modify our named.conf.local file from Part 1 and change our allow-transfer directive from the BIG-IP's source IP address to the TSIG key.

```
/etc/bind/named.conf.local
-----
zone "example.com" {
  type master;
  file "/etc/bind/db.example.com";
  allow-transfer {
    key host1_to_host2_test_tsig;
  };
};
```

5. Finally we'll want to reload our BIND configuration and make sure everything loads successfully.

```
root@f5-test-bind-master:~# rndc reload
  server reload successful
```

Testing and Troubleshooting TSIG Functionality

1. Now we'll move over to the command line on the GTM, so SSH to it and fire up a Bash shell. If you're in TMSH, use 'run util bash' to spawn a Bash shell.

2. From the shell on the BIG-IP we can test our zone transfer using the TSIG key.

```
[f5-test-gtm:Active] # dig @192.168.1.200 -y host1_to_host2_test_tsig:q0MMaG8ayhBlDdIOdPuRf38MnV2uqO0

; <<>> DiG 9.7.3-P3 <<>> @192.168.1.200 -y host1_to_host2_test_tsig AXFR example.com
; (1 server found)
;; global options: +cmd
example.com.               3600    IN      SOA     ns1.example.com. admin.ns.example.com. 2011100608
example.com.               3600    IN      NS      ns1.example.com.
example.com.               3600    IN      A       10.0.0.10
ns1.example.com.           3600    IN      A       10.0.0.1
www.example.com.           3600    IN      CNAME   example.com.
example.com.               3600    IN      SOA     ns1.example.com. admin.ns.example.com. 2011100608
host1_to_host2_test_tsig.  0       ANY     TSIG    hmac-md5.sig-alg.reg.int. 1318446000 300 16 khe1GP
;; Query time: 2 msec
;; SERVER: 192.168.1.200#53(192.168.1.200)
;; WHEN: Wed Oct 12 12:00:00 2011
;; XFR size: 6 records (messages 1, bytes 269)
```

This is what a successful zone transfer using transactional signatures should look like. Note the TSIG record returned from the BIND server. This record was not defined in our zone, but is attached so that the receiving host can verify the zone it received. The TSIG record can be broken down into nine fields:

```
Field Name       Contents
-------------------------------------
Algorithm Name   hmac-md5.sig-alg.reg.int.
Time Signed      1318446000
Fudge            300
MAC Size         16
MAC              khe1GPFDLFIcU0H6veTpww==
Original ID      62118
Error            0 (NOERROR)
Other Len        0
Other Data       empty
```

Source: RFC2845 - http://tools.ietf.org/html/rfc2845

The four that we care most about are the algorithm, the timestamp, the MAC, and the error (or lack thereof). The algorithm and MAC provide the necessary elements to verify the zone transfer, the timestamp ensures that a middle man cannot replay the transaction at some later date, and finally the error code of zero lets us know that everything is working as expected.

Troubleshooting Common TSIG Errors

Clocks Out Of Sync

If you receive an error from dig that looks like this, your clocks are out of sync on the hosts and the timestamp on the TSIG record cannot be verified. Note the 'BADTIME' error in the TSIG record.

```
[f5-test-gtm:Active] # dig @192.168.1.200 -y host1_to_host2_test_tsig:q0MMaG8ayhBlDdIOdPuRf38MnV2uqO0
;; Couldn't verify signature: clocks are unsynchronized

; <<>> DiG 9.7.3-P3 <<>> @192.168.1.200 -y host1_to_host2_test_tsig AXFR example.com
; (1 server found)
;; global options: +cmd
```

```
host1_to_host2_test_tsig.  0       ANY     TSIG    hmac-md5.sig-alg.reg.int. 1318420986 300 16 wcB3bv
;Transfer failed.
```

If you see this message you'll need to enable NTP on both hosts. You can nail down the perpetrator by using ntpdate to query a time server.

```
[f5-test-gtm:Active] # ntpdate -q some-time-server.net
12 Oct 12:00:00 ntpdate[26853]: step time server 4.2.2.2 offset 24959.255333 sec
```

If you see an offset of more than a second or two (the test machine was off by almost 7 hours), you've found the culprit. You can add a time server to the BIG-IP by navigating to System > Configuration > Device > NTP. Then define an NTP server, add it, and update. Also note that if you are using a host name here, then you'll need to configure DNS resolution for the BIG-IP as well.

Incorrect Key and/or Key Name

If you run your AXFR test and get a message like this. It can be one of a numberof things going wrong.

```
[f5-test-gtm:Active] # dig @192.168.1.200 -y host1_to_host2_test_tsig:q0MMaG8ayhBlDdIOdPuRf38MnV2uqO0
;; Couldn't verify signature: tsig indicates error

; <<>> DiG 9.7.3-P3 <<>> @192.168.1.200 -y host1_to_host2_test_tsig AXFR example.com
; (1 server found)
;; global options: +cmd
host1_to_host2_test_tsig.  0       ANY     TSIG    hmac-md5.sig-alg.reg.int. 1318461445 300 0  63868
; Transfer failed.
```

In this case we've obviously got a bad key, but nailing down the problem can be frustrating. Go through these steps to remedy the issue:

1. Look back at the 'Khost1_to_host2_test_tsig.+157+55433.key' file in /etc/bind. Make sure that the key name (trailing dot is optional and will not cause an issue) and the key you're using for your test are the same. Make any adjustments, test again.

2. If you're still having an issue and you're testing the right BIND server then there is probably a problem on the BIND server. Make sure that you are including the TSIG keys file in named.conf and that they are being loaded properly. Look in '/var/log/syslog' for any errors that may have been thrown on loading the keys. Verify everything, reload, test again.

3. If you're still having issues, come back to the BIND server, run a dig @localhost with the same key and zone AXFR query and see what you get. If this doesn't work, crank up debugging on BIND and keep troubleshooting.

Configuring DNS Express To Use TSIG Keys For Zone Transfers

If your tests worked and you saw the zone you were expecting, you can now move on to configuring the GTM to initiate zone transfers. The good news is that this is the easy part.

1. Open up your browser and login to the web interface for the GTM

2. Navigate to the DNS Express TSIG tab by opening Local Traffic > DNS Express Zone > DNS Express TSIG Key List.

3. Once in the DNS Express TSIG Key List tab, create a new TSIG key by clicking "Create" in the upper right corner.

4. Now we'll fill in the Name and Secret fields with the values we used to test the zone.

```
Name: host1_to_host2_test_tsig
Algorithm: HMAC-MD5
Secret: q0MMaG8ayhBlDdIOdPuRf38MnV2uqO0xzCb/t7GQbso=
```

5. Click "Finished" to save the key information.

6. Next we'll associate the TSIG key with the example.com. zone. You can either click the 'DNS Express Zone List' tab at the top or navigate to the zone list via Local Traffic > DNS Express Zone > DNS Express Zone List.

7. Select (or create, see Part 1) the example.com. zone.

8. In the zone properties, we'll want to assign the 'host1_to_host2_test_tsig' TSIG key from the drop down and then click "Update."

9. Take a look at '/var/log/ltm' either from the web interface of command line. If you see a line that looks like this, everything went as planned.

```
Oct 12 12:00:00 f5-test-gtm notice zxfrd[3800]: 01531025:5: Serials equal (2011100402); transfer for
Oct 12 12:00:00 f5-test-gtm notice zxfrd[3800]: 01531023:5: Scheduling zone transfer in 3600s for exa
```

Another confirmation can be found when the zone status light in the DNS Express Zone List turns green. That means the zone transferred successfully.

Conclusion

We hope that this two-part series on DNS Express gives you all the tips and tools that you need to get it up and running in your environment. If you’ve got a GTM running version 11 in your environment, you’ve got to give DNS Express a go. It offers features to improve performance and security of your existing DNS infrastructure at zero additional cost above that of the GTM. We hope you found this series informative and useful. Until next time, happy queries!

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