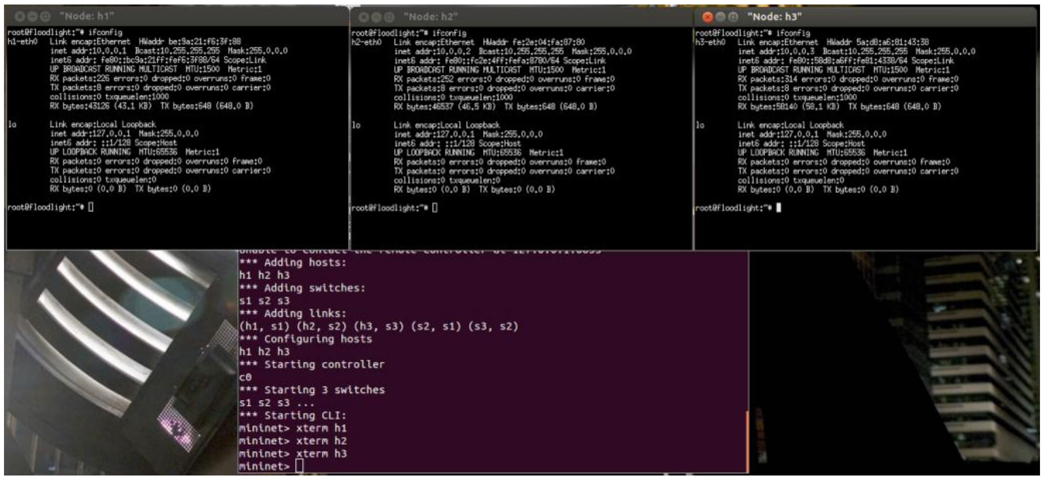
A Guide of ForenGuard Demo

1. This VM image (username & pwd: floodlight) contains the following things:
    a. Mininet + OpenVSwitch
    b. Community Version MongoDB
    c. Instrumented Floodlight Controller
    d. ForenGuard

2. This VM image can help demo the running case in our paper:

    Haopei Wang, Guangliang Yang, Phakpoom Chinprutthiwong, Lei Xu, Yangyong Zhang, Guofei Gu. "Towards Fine-grained Network Security Forensics and Diagnosis in the SDN Era." In *Proc. of the 25th ACM Conference on Computer and Communications Security (CCS'18)*, Toronto, Canada, October 2018.

3. Steps of demo:
    a. Start mongodb
    `mongod --dbpath data/db/ --smallfiles`
    b. Create the topology using Mininet
    For example: create a three-switch topology, each switch connects to a host
    `mn --controller=remote,port=6653 --topo linear,3`
    c. Start our instrumented Floodlight Controller
    `java -jar floodlight.jar`
    d. Prepare for the MAC spoofing attack
    Check all three hosts' network configuration



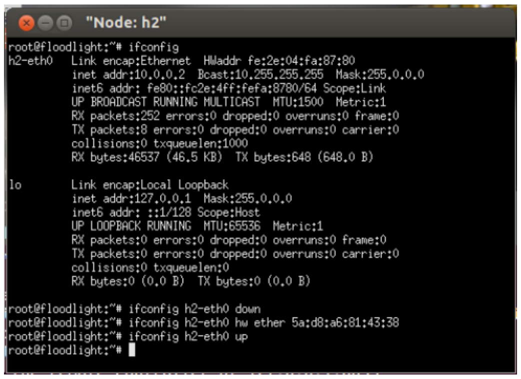
```
root@floodlight:~# ifconfig
h1-eth0   Link encap:Ethernet  HWaddr be:9a:21:f6:3f:88
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::bc9a:21ff:fef6:3f88/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:226 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:43126 (43.1 KB)  TX bytes:648 (648.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@floodlight:~# 
```

```
root@floodlight:~# ifconfig
h2-eth0   Link encap:Ethernet  HWaddr fe:2e:04:fa:87:80
          inet addr:10.0.0.2  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::fc2e:4ff:fefa:8780/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:252 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:46537 (46.5 KB)  TX bytes:648 (648.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@floodlight:~# 
```

```
root@floodlight:~# ifconfig
h3-eth0   Link encap:Ethernet  HWaddr 5a:d8:a6:81:43:38
          inet addr:10.0.0.3  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::58d8:a6ff:fe81:4338/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:314 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:58140 (58.1 KB)  TX bytes:648 (648.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@floodlight:~# 
```

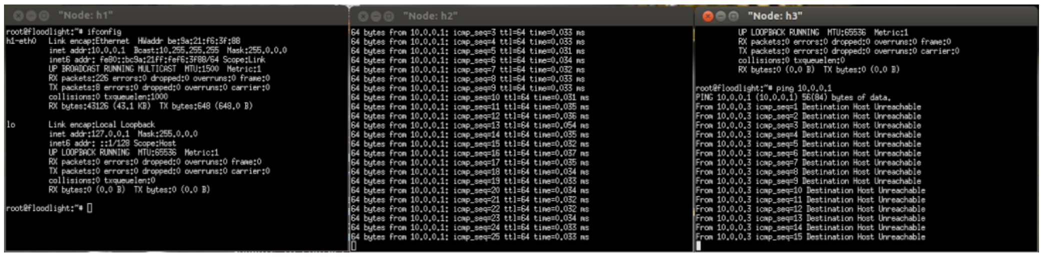
```
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1 s2 s3
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (s2, s1) (s3, s2)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet> xterm h1
mininet> xterm h2
mininet> xterm h3
mininet> 
```

    e. Use h2 to spoof h3's MAC address



```
root@floodlight:~# ifconfig
h2-eth0   Link encap:Ethernet  HWaddr fe:2e:04:fa:87:80
          inet addr:10.0.0.2  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::fc2e:4ff:fefa:8780/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:252 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:46537 (46.5 KB)  TX bytes:648 (648.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@floodlight:~# ifconfig h2-eth0 down
root@floodlight:~# ifconfig h2-eth0 hw ether 5a:d8:a6:81:43:38
root@floodlight:~# ifconfig h2-eth0 up
root@floodlight:~# 
```

f. Launch the spoofing attack. Use h2 to send network packets to h1. Also at the same time, try to use h3 to make connection with h1 as well.



```
root@floodlight:~# ifconfig
h1-eth0   Link encap:Ethernet  HWaddr be:9a:21:f6:3f:88
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::bc9a:21ff:fef6:3f88/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:226 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:43126 (43.1 KB)  TX bytes:648 (648.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@floodlight:~# 
```

```
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.033 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.033 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.031 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.034 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=0.032 ms
64 bytes from 10.0.0.1: icmp_seq=8 ttl=64 time=0.033 ms
64 bytes from 10.0.0.1: icmp_seq=9 ttl=64 time=0.033 ms
64 bytes from 10.0.0.1: icmp_seq=10 ttl=64 time=0.031 ms
64 bytes from 10.0.0.1: icmp_seq=11 ttl=64 time=0.035 ms
64 bytes from 10.0.0.1: icmp_seq=12 ttl=64 time=0.036 ms
64 bytes from 10.0.0.1: icmp_seq=13 ttl=64 time=0.054 ms
64 bytes from 10.0.0.1: icmp_seq=14 ttl=64 time=0.035 ms
64 bytes from 10.0.0.1: icmp_seq=15 ttl=64 time=0.032 ms
64 bytes from 10.0.0.1: icmp_seq=16 ttl=64 time=0.037 ms
64 bytes from 10.0.0.1: icmp_seq=17 ttl=64 time=0.035 ms
64 bytes from 10.0.0.1: icmp_seq=18 ttl=64 time=0.034 ms
64 bytes from 10.0.0.1: icmp_seq=19 ttl=64 time=0.033 ms
64 bytes from 10.0.0.1: icmp_seq=20 ttl=64 time=0.034 ms
64 bytes from 10.0.0.1: icmp_seq=21 ttl=64 time=0.032 ms
64 bytes from 10.0.0.1: icmp_seq=22 ttl=64 time=0.032 ms
64 bytes from 10.0.0.1: icmp_seq=23 ttl=64 time=0.034 ms
64 bytes from 10.0.0.1: icmp_seq=24 ttl=64 time=0.033 ms
64 bytes from 10.0.0.1: icmp_seq=25 ttl=64 time=0.033 ms
```

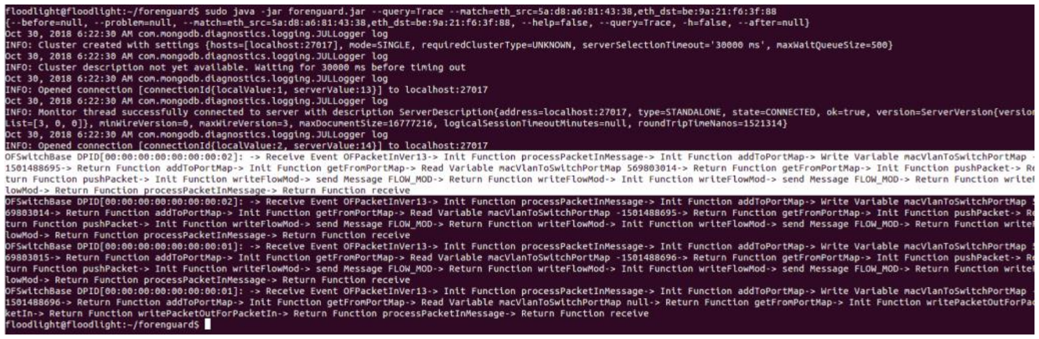
```
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@floodlight:~# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
From 10.0.0.3 icmp_seq=1 Destination Host Unreachable
From 10.0.0.3 icmp_seq=2 Destination Host Unreachable
From 10.0.0.3 icmp_seq=3 Destination Host Unreachable
From 10.0.0.3 icmp_seq=4 Destination Host Unreachable
From 10.0.0.3 icmp_seq=5 Destination Host Unreachable
From 10.0.0.3 icmp_seq=6 Destination Host Unreachable
From 10.0.0.3 icmp_seq=7 Destination Host Unreachable
From 10.0.0.3 icmp_seq=8 Destination Host Unreachable
From 10.0.0.3 icmp_seq=9 Destination Host Unreachable
From 10.0.0.3 icmp_seq=10 Destination Host Unreachable
From 10.0.0.3 icmp_seq=11 Destination Host Unreachable
From 10.0.0.3 icmp_seq=12 Destination Host Unreachable
From 10.0.0.3 icmp_seq=13 Destination Host Unreachable
From 10.0.0.3 icmp_seq=14 Destination Host Unreachable
From 10.0.0.3 icmp_seq=15 Destination Host Unreachable
```

g. Observation: when h2 is connecting with h1, h3 cannot make the connection with h1 any more. Attack succeeds!

h. Suppose you are the owner of h3. Let's use ForenGuard to diagnose why you lose network connection with h1.

i. We can use the following command to query for the control plane execution traces of how they forward the traffic from h3 to h1:
java -jar forenguard.jar --query=Trace --match=eth_src=5a:d8:a6:81:43:38,eth_dst=be:9a:21:f6:3f:88

j. Then we can check the diagnosis results:



```
floodlight@floodlight:~/forenguard$ sudo java -jar forenguard.jar --query=Trace --match=eth_src=5a:d8:a6:81:43:38,eth_dst=be:9a:21:f6:3f:88
{--before=null, --problem=null, --match=eth_src=5a:d8:a6:81:43:38,eth_dst=be:9a:21:f6:3f:88, --help=false, --query=Trace, -h=false, --after=null}
Oct 30, 2018 6:22:30 AM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500}
Oct 30, 2018 6:22:30 AM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
Oct 30, 2018 6:22:30 AM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:13}] to localhost:27017
Oct 30, 2018 6:22:30 AM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versio
List=[3, 0, 0]}, minWireVersion=0, maxWireVersion=3, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=null, roundTripTimeNanos=1521314}
Oct 30, 2018 6:22:30 AM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:14}] to localhost:27017
OFSwitchBase DPID[00:00:00:00:00:00:00:02]: -> Receive Event OFPacketInVer13-> Init Function processPacketInMessage-> Init Function addToPortMap-> Write Variable macVlanToSwitchPortMap -
1501488695-> Return Function addToPortMap-> Init Function getFromPortMap-> Read Variable macVlanToSwitchPortMap 569803014-> Return Function getFromPortMap-> Init Function pushPacket-> Re
turn Function pushPacket-> Init Function writeFlowMod-> send Message FLOW_MOD-> Return Function writeFlowMod-> Init Function writeFlowMod-> send Message FLOW_MOD-> Return Function write
lowMod-> Return Function processPacketInMessage-> Return Function receive
OFSwitchBase DPID[00:00:00:00:00:00:00:02]: -> Receive Event OFPacketInVer13-> Init Function processPacketInMessage-> Init Function addToPortMap-> Write Variable macVlanToSwitchPortMap 5
69803014-> Return Function addToPortMap-> Init Function getFromPortMap-> Read Variable macVlanToSwitchPortMap -1501488695-> Return Function getFromPortMap-> Init Function pushPacket-> Re
turn Function pushPacket-> Init Function writeFlowMod-> send Message FLOW_MOD-> Return Function writeFlowMod-> Init Function writeFlowMod-> send Message FLOW_MOD-> Return Function write
lowMod-> Return Function processPacketInMessage-> Return Function receive
OFSwitchBase DPID[00:00:00:00:00:00:00:01]: -> Receive Event OFPacketInVer13-> Init Function processPacketInMessage-> Init Function addToPortMap-> Write Variable macVlanToSwitchPortMap 5
69803015-> Return Function addToPortMap-> Init Function getFromPortMap-> Read Variable macVlanToSwitchPortMap -1501488696-> Return Function getFromPortMap-> Init Function pushPacket-> Re
turn Function pushPacket-> Init Function writeFlowMod-> send Message FLOW_MOD-> Return Function writeFlowMod-> Init Function writeFlowMod-> send Message FLOW_MOD-> Return Function write
lowMod-> Return Function processPacketInMessage-> Return Function receive
OFSwitchBase DPID[00:00:00:00:00:00:00:01]: -> Receive Event OFPacketInVer13-> Init Function processPacketInMessage-> Init Function addToPortMap-> Write Variable macVlanToSwitchPortMap -
1501488696-> Return Function addToPortMap-> Init Function getFromPortMap-> Read Variable macVlanToSwitchPortMap null-> Return Function getFromPortMap-> Init Function writePacketOutForPa
ketIn-> Return Function writePacketOutForPacketIn-> Return Function processPacketInMessage-> Return Function receive
floodlight@floodlight:~/forenguard$ 
```

k. We can observe that there are four events that are the root causes. When you further check these four events, they are triggered by h2's spoofing.

l. Don't forget to clean up MongoDB after the use
(mongodb shell) > db.dropDatabase()