Logout

My Workspace | CS-6250-O01 FALL14 | CS-6300-O01 FALL14

ASSIGNMENTS

Home
Syllabus
Announcements
Resources
Assignments
Gradebook
Email Archive
Roster
Site Info
Section Info
Piazza
Help

Assignment 8 - Applying SDN to DoS Attack Mitigation - Returned

Title	Assignment 8 - Applying SDN to DoS Attack Mitigation
Student	Telfort, Dominique Gayot
Submitted Date	Nov 13, 2014 10:29 pm
Grade	9.5 (max 10.0)

Instructions

Assignment 8: Applying SDN to DoS Attack Mitigation

Overview

In previous assignments, you have seen how SDN makes dynamic network configuration incredibly easy. In this assignment, you will apply those techniques to a real world application: mitigating Denial of Service (DoS) attacks.

In a DoS attack, an attacker consumes the victim's resources to prevent them from serving their normal customers. The DoS attack could consume network resources by flooding the victims network with data. In more complicated attacks, the attacker can consume the victim's CPU or memory resources without flooding the link. A common example of this is a TCP SYN attack, where an attackers sends a large number of TCP SYNs to start a connection with the victim, but then drops the connection on the attacker side. By moving the victim into the TCP state machine, the attacker forces the victim to retain state and consume resources for variables and timers.

In this assignment, you will use several new tools to monitor the network and respond to events as they happen. In particular, you will use a monitoring tool called SFlow to get notifications about network events and an state machine implementation on top of Pyretic called PyResonance to change the network configuration in response to notifications.

sFlow

sFlow is an SDN standard for collecting network measurements from switches on the network. For this assignment, sFlow-RT will be used, a product created by InMon instrument switches for network measurement.

To collect performance information, sFlow-RT has each switch send data back to an aggregator. To access this data, applications can register for notifications when their defined thresholds are exceeded. In sFlow-RT, applications register using Javascript and view callbacks by making repeated GET requests of the aggregator. This allows for powerful constructions, but it would be a bit overwhelming to learn Javascript for this assignment. Therefore, all code using sFlow has already been written for you for assignment 8.

Please see these sites for more info about sFlow and sFlowRT.

PyResonance

PyResonance is an application framework which uses state machines to alter network configurations in response to network events.

Within PyResonance, each flow has its own state machine and a routing policy associated with each state. By defining the state behavior and the transitions between states, PyResonance has limitless potential for defining custom network behavior. When coupled with tools to report on network performance, PyResonance becomes even more powerful. In addition to traditional applications like firewalls and Intrusion Detection Systems, PyResonance allows users to define their own behaviors and implement custom functionality. To learn more, please go to PyResonance wiki.

Assignment Steps

In this assignment, you will set SFlow to notify you when a DoS attack is happening and then respond to the notification in PyResonance. If you are interested in learning more about either tool, please see below. Additionally, due to some complications with SFlow, this assignment will only be monitoring host h1, 10.0.0.1, to see if it launches a DoS attack.

Note: It is possible to configure SFlow and PyResonance to respond to

events from arbitrary hosts, but only monitoring 1 host significantly simplifies the code. This is relevant to you because you could get incorrect results, likely nothing, if you fail to launch the commands from the specific hosts listed below. (If you are not getting any results, please make sure that you have exactly followed the directions below)

1. Install and test PyResonance
    1. Download PyResonance by running `$cd /home/mininet/pyretic/pyretic` and `$git clo`
    2. Verify that PyResonance downloaded correctly by verifying that the directory exists `$ls /home/mininet/pyretic/pyretic/pyresonance`
    3. Verify that PyResonance works by running the following example
        1. Start PyResonance by running `$cd /home/mininet/pyretic` and then `$pyretic.py pyretic.pyresonance.main --confi`
        2. In a separate terminal, start the mininet topology by moving to the PyResonance directory `$cd /home/mininet/pyretic/pyretic/pyresc` running `$sudo mn --controller=remote,ip=127.0.0.1`
        3. In the default application, hosts should not be able to send traffic unless they are authenticated. This means that hosts should not be able to send traffic yet. Verify this by running `mininet>pingall`
        4. In a third terminal window, you will send some json events to authenticate the hosts. Move to the PyResonance directory as shown above and send authentication and IDS events for the hosts:

```
$python json_sender.py --flow='{srcip=10.0.0.1}' -e auth -s authenticated -a 127.0.0.1 -p 50001
$python json_sender.py --flow='{srcip=10.0.0.2}' -e auth -s authenticated -a 127.0.0.1 -p 50001
$python json_sender.py --flow='{srcip=10.0.0.1}' -e ids -s clean -a 127.0.0.1 -p 50002
$python json_sender.py --flow='{srcip=10.0.0.2}' -e ids -s clean -a 127.0.0.1 -p 50002
```

5. In the mininet terminal, verify that the hosts can now communicate by running the pingall command again.

4. For more information about PyResonance, please see the PyResonance wiki.

2. Install Java on the VM (this may take a few minutes):

```
sudo apt-get update
```

```
sudo apt-get install openjdk-7-jre
```

3. Download SFlowRT
    1. Download and unzip SFlow to the directory of your choice on the VM:

```
$wget http://www.inmon.com/products/sFlow-RT/sflow-rt.tar.gz
$tar -xvzf sflow-rt.tar.gz
```

    2. Test that sflow works by moving into the SFlow directory and running `$./start.sh`. If you see something along these lines, then SFlow is working correctly.

```
2014-02-18T15:26:16-0800 INFO: Listening, sFlow port 6343
2014-02-18T15:26:16-0800 INFO: Listening, http://localhost:8008
2014-02-18T15:26:16-0800 INFO: init.js started
2014-02-18T15:26:16-0800 INFO: init.js stopped
```

    3. Note: it is possible to retrieve graphs of your traffic from SFlowRT but it may not work on the VM using SSH Port Forwarding. If you find a solution, please post it on the forums.
    4. For more info about SFlowRT, please see the InMon description or download a copy of SFlowRT as shown above to your workstation, not the VM, then run SFlow. By navigating to http://localhost:8008 on your workstation, you should be able to see more documentation
4. Implement the section marked with a TODO in the assignment code
    1. Update the assignment code:

```
git commit -a -m "Saving work"
```

    2. PyResonance and SFlow have a steep learning curve, so we

2. Pyresonance and Cplow have a steep learning curve, so we spent some time finding an appropriate task with these tools