DevSecOps: Secure Cloud Enclaves

Sprint 3: *March 7 - Mar 28*

Ethan McIlhenny, Avantika Dasgupta, Josh Manning, Dharmit Dalvi

Recap

- Sprint 2:
  - Learned how to implement secure buckets
    - Disable Public Read/Write
    - Object Versioning
    - Only SSL Requests
  - Used the S3 API to access MOC Ceph
    - Created Buckets
    - Read/Write to Buckets
  - Configured console access to our MOC Account
- Sprint 3 Goals:
  - Write Scripts to automate Object storage processes
  - Begin Logging
    - Configure Logging
      - OS Level Events
      - Services on the VM
      - Object Level Logging

Scripting

- Created a Shell script to automate:
  - Initial VM setup (installing dependencies)
  - Configuring environment variables
  - Running a Python script to create S3 buckets (to store data and logs)
  - Enabling versioning on the buckets
- When CONS3RT spins up a new VM, this script will be put on it and run automatically

```
NOVARC=$(readlink -f "${BASH_SOURCE:-${0}}" 2>/dev/null) || NOVARC=$(python -c 'import os,sys; pr
int os.path.abspath(os.path.realpath(sys.argv[1]))' "${BASH_SOURCE:-${0}}")
NOVA_KEY_DIR=${NOVARC%/*}
export EC2_ACCESS_KEY=[redacted]
export EC2_SECRET_KEY=[redacted]
export EC2_URL=https://kaizen.massopen.cloud:13788
export EC2_USER_ID=42 # nova does not use user id, but bundling requires it
export EC2_PRIVATE_KEY=${NOVA_KEY_DIR}/pk.pem
export EC2_CERT=${NOVA_KEY_DIR}/cert.pem
export NOVA_CERT=${NOVA_KEY_DIR}/cacert.pem
export EUCALYPTUS_CERT=${NOVA_CERT} # euca-bundle-image seems to require this set

alias ec2-bundle-image="ec2-bundle-image --cert ${EC2_CERT} --privatekey ${EC2_PRIVATE_KEY} --use
r 42 --ec2cert ${NOVA_CERT}"
alias ec2-upload-bundle="ec2-upload-bundle -a ${EC2_ACCESS_KEY} -s ${EC2_SECRET_KEY} --url ${S3_U
RL} --ec2cert ${NOVA_CERT}"

######################## CREATING BUCKETS ######################
python connection.py > output.txt
errMsg=`cat output.txt`
echo $errMsg
```

```
import boto
import boto.s3.connection

access_key = os.environ['EC2_ACCESS_KEY']
secret_key = os.environ['EC2_SECRET_KEY']


conn = boto.s3.connection.S3Connection(
    aws_access_key_id=access_key,
    aws_secret_access_key=secret_key,
    port=443,
    host='kzn-swift.massopen.cloud',
    is_secure=True,
    calling_format=boto.s3.connection.OrdinaryCallingFormat())

bucket = conn.create_bucket('data-storage-bucket-001')
bucket.configure_versioning(True)
bucket2 = conn.create_bucket('log-storage-bucket-001')
bucket2.configure_versioning(True)
```

Logging

- OS Level Events :
  - *sudo* actions and by whom
  - Logins : attempted, successful, failed
  - Changes to networks or firewalls
  - Changes to user accounts
- Service Level:
  - Logs of a service running on top of the VM
  - Changes to the service state
- Object Level:
  - Remote access to buckets
  - Operations within buckets

OpenStack APIs

- Keystone → Identity
  - Changes to user accounts
  - Logins
- Nova → Compute
  - Servers, Instances
- Glance → Images
  - Image create, change
- Neutron → Networking

OpenStack API Configuration

- Install OpenStack CLI API Clients
- Modify “APIt.conf” files
  - Add a couple lines
    - debug = True ----> Sets the log level to DEBUG (more verbose)
    - log_dir = /var/log/API ----> Log directory for easy collection
    - log_file = /var/log/API/API.log ---> convenient log file name
    - use_stderr = false ---> logs do not go to stderr

Elastic Stack

- Filebeat
  - Collects specified logs
  - Forwards them to Logstash
- Logstash
  - Filters log inputs
  - Distributes filtered logs to an output destination

Filebeat Configuration

Filebeat.yml

- Inputs
  - Where filebeat looks for logs
- Output
  - Where filebeat forwards the logs to
- Logging
  - Where the logs for filebeat itself get sent

```
############################ Filebeat ######################################
filebeat:
  # List of prospectors to fetch data.
  prospectors:
  #inputs:
  - type: log
    enabled: true
    paths:
      - /var/log/*.log
      - /var/log/secure
      - /var/log/messages
      - /var/log/neutron
      - /var/log/nova
      - /var/log/cinder
      - /var/log/glance
      - /var/log/horizon
      - /var/log/keystone
    registry_file: /var/lib/filebeat/registry
output:
  logstash:
    enabled: true
    hosts: ["localhost:5044"]
shipping:
logging:
  to_files: true
  json: true
  files:
    path: /var/log/filebeat
    rotateeverybytes: 10485760
```

Logstash Configuration

```
input {
        beats {
                port => 5044
        }
}
filter{
        if[source]=="/var/log/httpd/*.log"
        {
                mutate{
                        remove_tag=>["beats_input_codec_plain_applied"]
                        add_tag=>["httpd_logs"]
                        }
        }
}

output {
       # s3 {
       #         access_key_id =>
       #         secret_access_key => "
       #         bucket => "versioning-enabled-bucket"
       #         canned_acl => "bucket-owner-full-control"
        #        endpoint =>"http://kzn-swift.massopen.cloud:443"
        #}
        file {
                path => "/test_log/%{+YYYY-MM-dd}.txt"
        }
}
```

logstash.conf (pictured)

- Inputs
  - Filebeat forwarding to port 5044
- Filters
  - Example filter for a httpd service log
    - Can modify formatting
- Output
  - S3 Output
    - Eventual output destination
      - Troubleshooting
  - File Output

logstash.yml → Configured CPU cores limit

DEMO!

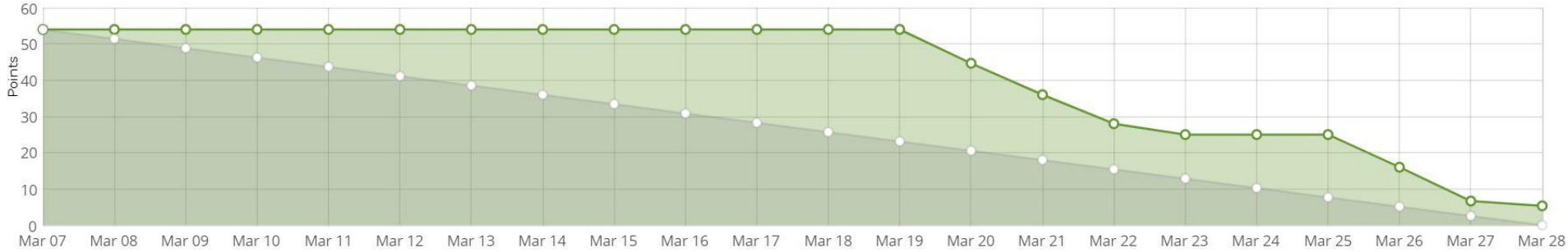
Sprint 3 Progress

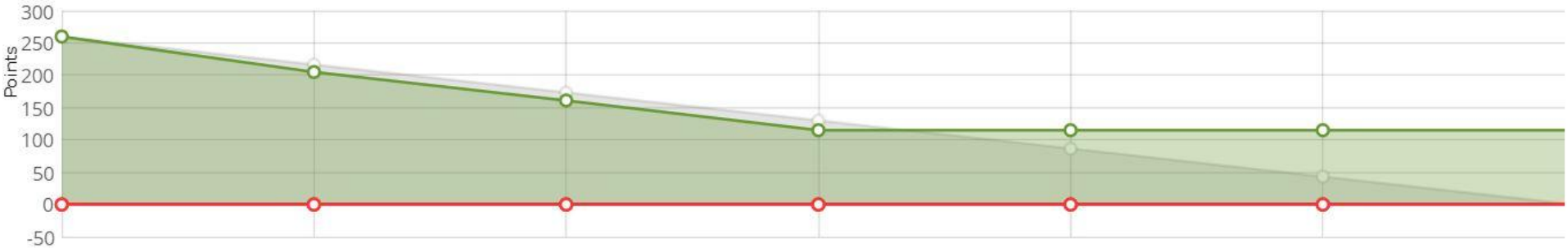
- Caught up with missed work from Sprint 2
  - Finished Scripting Object Storage Actions
- Categorized and Configured logging on VM
- Ran Filebeat and Logstash on VM to collect and process logs

Problems:

- S3 Output plugin not working → might be because not using real S3 buckets
- S3 bucket logging not working → might be because MOC has an older version of Ceph

Burndown Chart





Future Work

Sprint 4

- Finish Logging
  - Script Logging Configuration and Deployment on a VM
  - Enable Log file validation
- Accounts and Credentials
  - Implement system to rotate accounts and credentials from CONS3RT to OpenStack

Sprint 5

- Validating Security Features
  - Find a method to validate our security standards are implemented correctly
- Stretch Goals
  - Key Management and Encryption
  - Change Scripts to Java Workflows