SYSTEM HARDENING

Provisioning and Hardening Two Digital Ocean Droplets with DISA STIGs Using Ansible Automation Tool

CORY SEARCY

COURSE: PROJECT & PORTFOLIO 7

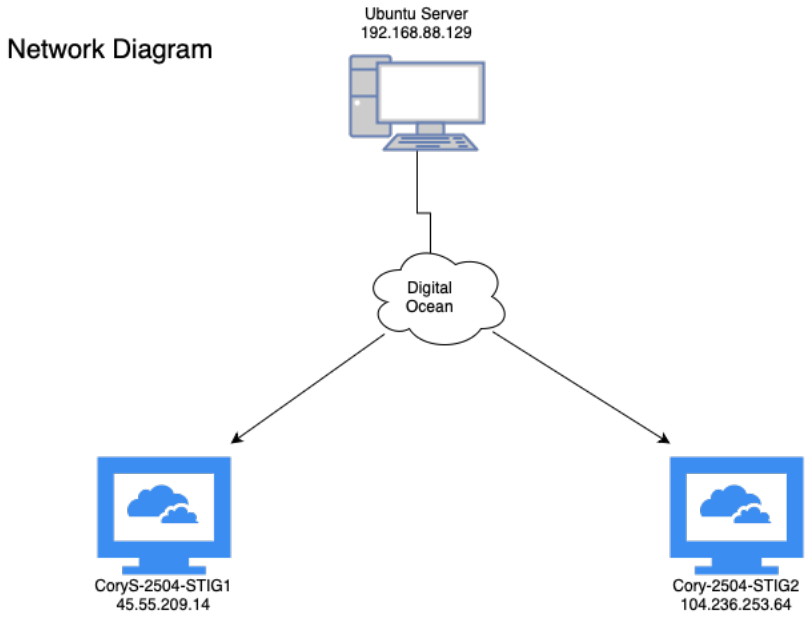
TABLE OF CONTENTS

Scope of Work

PROJECT SCOPE OVERVIEW

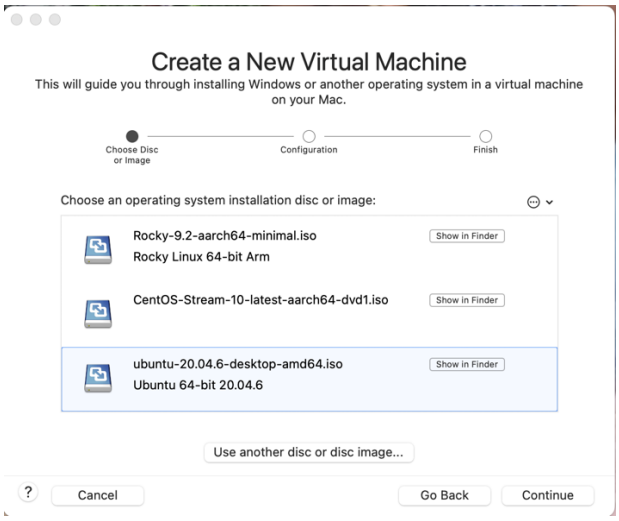
This project involves the creation of an Ubuntu VM. This project also involves installing and configuring Ansible to automate the provisioning and hardening of two Digital Ocean (DO) instances (droplets). SSH trust will be established so that Ansible can manage the droplets. The automation will apply lamp and 20-30 DISA STIGs to ensure continuous compliance, proper system configuration, and security hardening.

NETWORK TOPOLOGY DIAGRAM WITH IP ADDRESSES



CLOUD AUTOMATION PROCEDURAL INSTRUCTIONS

- Create a new Ubuntu VM using VMware.



- Install and configure Ubuntu Server.
- Now log in to the terminal on the Ubuntu VM.

```
cory@cory-virtual-machine:~$
```

- Run sudo apt update & sudo apt upgrade -y

```
cory@cory-virtual-machine:~$ sudo apt update
Reading package lists... Done
E: Could not get lock /var/lib/apt/lists/lock. It is held by process 42388 (apt-get)
N: Be aware that removing the lock file is not a solution and may break your system.
E: Unable to lock directory /var/lib/apt/lists/
cory@cory-virtual-machine:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  ieee-data python3-argcomplete python3-dnspython python3-libcloud python3-netaddr
  python3-pycryptodome python3-requests-toolbelt python3-selinux python3-simplejson
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  linux-headers-6.8.0-58-generic linux-hwe-6.8-headers-6.8.0-58 linux-hwe-6.8-tools-6.8.0-58
```

- Now install ansible using

- sudo apt install ansible -y

```
done
cory@cory-virtual-machine:~$ sudo apt install ansible -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ansible is already the newest version (10.7.0-1ppa~jammy).
The following packages were automatically installed and are no longer required:
  ieee-data linux-headers-6.8.0-57-generic linux-hwe-6.8-headers-6.8.0-57
  linux-hwe-6.8-tools-6.8.0-57 linux-image-6.8.0-57-generic linux-modules-6.8.0-57-generic
  linux-modules-extra-6.8.0-57-generic linux-tools-6.8.0-57-generic python3-argcomplete
  python3-dnspython python3-libcloud python3-netaddr python3-pycryptodome python3-requests-tool
  python3-selinux python3-simplejson
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
cory@cory-virtual-machine:~$
```

- Verify installation of ansible
  Run ansible –version

```
cory@cory-virtual-machine:~$ ansible --version
ansible [core 2.17.11]
  config file = /home/cory/ansible.cfg
  configured module search path = ['/home/cory/.ansible/plugins/modules', '/usr/share/ansible/p
modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/cory/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Feb  4 2025, 14:57:36) [GCC 11.4.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
cory@cory-virtual-machine:~$
```

- Install python
  Run sudo apt install python3

```
cory@cory-virtual-machine:~$ sudo apt install python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04.1).
python3 set to manually installed.
The following packages were automatically installed and are no longer required:
  ieee-data linux-headers-6.8.0-57-generic linux-hwe-6.8-headers-6.8.0-57
  linux-hwe-6.8-tools-6.8.0-57 linux-image-6.8.0-57-generic linux-modules-6.8.0-57-generic
  linux-modules-extra-6.8.0-57-generic linux-tools-6.8.0-57-generic python3-argcomplete
  python3-dnspython python3-libcloud python3-netaddr python3-pycryptodome python3-requests-tool
  python3-selinux python3-simplejson
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

- Next, we need to install python-pip
  Run sudo apt install python-pip

```
cory@cory-virtual-machine:~$ sudo apt install python-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  ieee-data javascript-common libexpat1-dev libjs-jquery libjs-sphinxdoc libjs-underscore
  libpython3-dev libpython3.10-dev linux-headers-6.8.0-57-generic linux-hwe-6.8-headers-6.8.0-
  tools-6.8.0-57 linux-image-6.8.0-57-generic linux-modules-6.8.0-57-generic
  extra-6.8.0-57-generic linux-tools-6.8.0-57-generic python3-argcomplete python
  python3-dnspython python3-libcloud python3-netaddr python3-pycryptodome python3-requests-too
  python3-selinux python3-simplejson python3-wheel python3.10-dev zlib1g-dev
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libpython2-stdlib libpython2.7-minimal libpython2.7-stdlib python-pkg-resources python-setup
```

LibreOffice Writer

- Now we must open the ansible.cfg file

  Use sudo nano ansible.cfg

```
cory@cory-virtual-machine:~$  sudo nano ansible.cfg
```

- Inside the ansible.cfg file add
  hostfile=hosts

  Save

```
  GNU nano 6.2                                ansible.cfg
[defaults]
hostfile=hosts
                                [ Read 2 lines ]
^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo
```

- Next, nano into the host file and add the following
  [localhost]

  Localhost ansible_connection=local

```
  GNU nano 6.2                                  hosts
[localhost]
localhost ansible_connection=local
```

- Now we need to generate ssh key
  Run ssh-keygen -t rsa

- Next run ssh-copy-id root@192.168.88.129

- Now cd to the ansible directory
  Run cd /etc/ansible

```
cory@cory-virtual-machine:~$ cd /etc/ansible
cory@cory-virtual-machine:/etc/ansible$
```

- Here we need to create an ansible playbook to configure 2 digital ocean droplets
  Run sudo nano droplet1.yml

```
cory@cory-virtual-machine:/etc/ansible$
```

- Now edit the Ansible playbook to create droplets.

```yaml
---
- name: Create Digital Ocean Droplets
  hosts: localhost
  connection: local
  gather_facts: false

  vars:
    droplet_name_prefix: "CoryS-2504-STIG"
    region: "nyc3"
    size: "s-1vcpu-1gb"
    image: "ubuntu-22-04-x64"
    ssh_key_name: "Cory-Searcy-2504"
    number_of_droplets: 2
    droplets:
      - CoryS-2504-STIG1
      - CoryS-2504-STIG2

  tasks:
    - name: Create SSH Key in Digital Ocean
      digital_ocean_sshkey:
        name: "{{ ssh_key_name }}"
        ssh_pub_key: "{{ lookup('file', '~/.ssh/id_rsa.pub') }}"
      register: ssh_key_result

    - name: Create Droplets
      digital_ocean_droplet:
        name: "{{ droplet_name_prefix }}{{ item }}"
        region: "{{ region }}"
        size: "{{ size }}"
        image: "{{ image }}"
        ssh_keys: "{{ ssh_key_result.data.ssh_key.id }}"
        unique_name: yes
        state: present
      loop: "{{ range(1, number_of_droplets + 1) | list }}"
      register: droplet_result

    - name: Wait for droplets to be created
      pause:
        seconds: 30

    - name: Add new droplets to host group
      add_host:
        name: "{{ item.networks.v4[0].ip_address }}"
        groups: droplets
      loop: "{{ droplets_info.data }}"
      when: item.networks is defined and item.networks.v4 | length > 0
```

- Now, create the playbook, adding a lamp to droplets.
- Run sudo nano lamp.yml

```
cory@cory-virtual-machine:/etc/ansible$ sudo nano lamp.yml
```

- Edit the lamp.yml playbook.

```yaml
- name: Install LAMP stack on new droplets
  hosts: droplets
  become: yes
  gather_facts: yes
  remote_user: ubuntu
  tasks:
    - name: Update apt cache
      apt:
        update_cache: yes

    - name: Install Apache
      apt:
        name: apache2
        state: present

    - name: Install MySQL server
      apt:
        name: mysql-server
        state: present

    - name: Install PHP and common modules
      apt:
        name:
          - php
          - php-mysql
          - libapache2-mod-php
          - php-cli
          - php-curl
          - php-gd
          - php-mbstring
          - php-xml
          - php-xmlrpc
        state: present

    - name: Enable Apache mod_rewrite
      apache2_module:
        name: rewrite
        state: present

    - name: Restart Apache
      service:
        name: apache2
        state: restarted

    - name: Ensure Apache is enabled on boot
      systemd:
        name: apache2
        enabled: yes
```

- Next, we can create the hardening playbook adding DIA STIGs.

```
cory@cory-virtual-machine:/etc/ansible$ sudo nano diastigs1.yml
```

```
- name: Apply DISA STIG Hardening to Ubuntu Systems
  hosts: all
  become: yes
  vars:
    min_password_length: 14
    max_password_age: 60
    password_warn_age: 7

  tasks:

    - name: Ensure auditd is installed
      apt:
        name: auditd
        state: present
        update_cache: yes

    - name: Ensure auditd is enabled and started
      systemd:
        name: auditd
        enabled: yes
        state: started

    - name: Ensure libpam-pwquality is installed
```

```
    apt:
      name: libpam-pwquality
      state: present

  - name: Set password maximum age
    lineinfile:
      path: /etc/login.defs
      regexp: '^PASS_MAX_DAYS'
      line: "PASS_MAX_DAYS {{ max_password_age }}"

  - name: Set password minimum length
    lineinfile:
      path: /etc/security/pwquality.conf
      regexp: '^minlen'
      line: "minlen = {{ min_password_length }}"

  - name: Set password warning age
    lineinfile:
      path: /etc/login.defs
      regexp: '^PASS_WARN_AGE'
      line: "PASS_WARN_AGE {{ password_warn_age }}"

  - name: Disable core dumps
    sysctl:
      name: fs.suid_dumpable
      value: '0'
```

```
    state: present
    reload: yes

- name: Set sticky bit on /tmp
  command: chmod +t /tmp

- name: Ensure permissions on /etc/shadow are correct
  file:
    path: /etc/shadow
    owner: root
    group: shadow
    mode: '0640'

- name: Ensure permissions on /etc/passwd are correct
  file:
    path: /etc/passwd
    owner: root
    group: root
    mode: '0644'

- name: Ensure permissions on /etc/group are correct
  file:
    path: /etc/group
    owner: root
    group: root
    mode: '0644'
```

```yaml
- name: Disable USB storage (if applicable)
  lineinfile:
    path: /etc/modprobe.d/disable-usb.conf
    create: yes
    line: "install usb-storage /bin/true"

- name: Configure sysctl for IP spoofing protection
  sysctl:
    name: net.ipv4.conf.all.rp_filter
    value: '1'
    state: present
    reload: yes

- name: Disable ICMP redirects
  sysctl:
    name: net.ipv4.conf.all.accept_redirects
    value: '0'
    state: present
    reload: yes

- name: Enable ExecShield (if supported)
  sysctl:
    name: kernel.exec-shield
    value: '1'
    state: present
```

```
    reload: yes
  ignore_errors: yes

- name: Set permissions on /etc/cron.d
  file:
    path: /etc/cron.d
    state: directory
    owner: root
    group: root
    mode: '0700'

- name: Restrict cron to authorized users
  copy:
    dest: /etc/cron.allow
    content: "root\n"
    owner: root
    group: root
    mode: '0600'

- name: Disable IPv6 if not needed
  sysctl:
    name: net.ipv6.conf.all.disable_ipv6
    value: '1'
    state: present
    reload: yes
```

```
    - name: Ensure default umask is 027
      lineinfile:
        path: /etc/profile
        regexp: '^umask'
        line: 'umask 027'

    - name: Lock inactive user accounts after 35 days
      lineinfile:
        path: /etc/default/useradd
        regexp: '^INACTIVE='
        line: 'INACTIVE=35'

    - name: Enforce PAM password complexity
      lineinfile:
        path: /etc/pam.d/common-password
        regexp: '^password\s+requisite\s+pam_pwquality.so'
        line: 'password requisite pam_pwquality.so retry=3 minlen={{
min_password_length }} ucredit=-1 lcredit=-1 dcredit=-1 ocredit=-1'

    - name: Ensure rsyslog is installed
      apt:
        name: rsyslog
        state: present

    - name: Ensure rsyslog is enabled and running
```

```
  systemd:
    name: rsyslog
    enabled: yes
    state: started

- name: Ensure permissions on rsyslog.conf are correct
  file:
    path: /etc/rsyslog.conf
    owner: root
    group: root
    mode: '0644'

- name: Ensure telnet is not installed
  apt:
    name: telnet
    state: absent

- name: Ensure rsh-server is not installed
  apt:
    name: rsh-server
    state: absent

- name: Ensure xinetd is not installed
  apt:
    name: xinetd
    state: absent
```

```yaml
- name: Disable Ctrl+Alt+Del reboot
  copy:
    dest: /etc/systemd/system/ctrl-alt-del.target
    content: ''
    owner: root
    group: root
    mode: '0644'


- name: Remove .netrc files (insecure)
  find:
    paths: /home
    patterns: '.netrc'
    recurse: yes
  register: netrc_files

- name: Delete .netrc files
  file:
    path: "{{ item.path }}"
    state: absent
  loop: "{{ netrc_files.files }}"
  when: netrc_files.matched > 0

- name: Disable SSH root login
  lineinfile:
```

```
    path: /etc/ssh/sshd_config
    regexp: '^PermitRootLogin'
    line: 'PermitRootLogin no'
    create: yes
    backup: yes
  notify: restart ssh

- name: Set SSH idle timeout to 10 minutes
  lineinfile:
    path: /etc/ssh/sshd_config
    regexp: '^ClientAliveInterval'
    line: 'ClientAliveInterval 600'
    create: yes
  notify: restart ssh

- name: Set SSH session disconnect after 3 failures
  lineinfile:
    path: /etc/ssh/sshd_config
    regexp: '^MaxAuthTries'
    line: 'MaxAuthTries 3'
    create: yes
  notify: restart ssh

- name: Set login banner (DoD warning)
  copy:
    dest: /etc/issue
```

```
        content: |
          You are accessing a U.S. Government (USG) Information System
(IS)
          that is provided for USG-authorized use only.
        owner: root
        group: root
        mode: '0644'


  handlers:
    - name: restart auditd
      service:
        name: auditd
        state: restarted

    - name: restart ssh
      service:
        name: ssh
        state: restarted
```

- Now we are ready to run all of our playbooks.

Run ansible-playbook -i inventory.ini droplet1.yml

- Run ansible-playbook -i inventory.ini lamp1.yml

```
cory@cory-virtual-machine:/etc/ansible$ ansible-playbook -i inventory.ini lamp1.yml

PLAY [Install LAMP stack on new droplets] ***************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************************
[WARNING]: Platform linux on host CoryS-2504-STIG1 is using the discovered Python interpreter at /usr/bin/python3.10, but future
installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [CoryS-2504-STIG1]
[WARNING]: Platform linux on host CoryS-2504-STIG2 is using the discovered Python interpreter at /usr/bin/python3.10, but future
installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [CoryS-2504-STIG2]

TASK [Update apt cache] *********************************************************************************************************************
changed: [CoryS-2504-STIG1]
changed: [CoryS-2504-STIG2]

TASK [Install Apache] ***********************************************************************************************************************
changed: [CoryS-2504-STIG1]
changed: [CoryS-2504-STIG2]

TASK [Install MySQL server] *****************************************************************************************************************
changed: [CoryS-2504-STIG1]
changed: [CoryS-2504-STIG2]

TASK [Install PHP and common modules] *******************************************************************************************************
changed: [CoryS-2504-STIG1]
changed: [CoryS-2504-STIG2]

TASK [Enable Apache mod_rewrite] ************************************************************************************************************
changed: [CoryS-2504-STIG2]
changed: [CoryS-2504-STIG1]

TASK [Restart Apache] ***********************************************************************************************************************
changed: [CoryS-2504-STIG2]
changed: [CoryS-2504-STIG1]

TASK [Ensure Apache is enabled on boot] *****************************************************************************************************
ok: [CoryS-2504-STIG2]
ok: [CoryS-2504-STIG1]

PLAY RECAP **********************************************************************************************************************************
CoryS-2504-STIG1           : ok=8    changed=6    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
CoryS-2504-STIG2           : ok=8    changed=6    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

cory@cory-virtual-machine:/etc/ansible$
```

- Run ansible-playbook -i inventory.ini diastigs1.yml

```
ok: [CoryS-2504-STIG2]

TASK [Ensure auditd is installed] ***********************************************************************************************
changed: [CoryS-2504-STIG1]
changed: [CoryS-2504-STIG2]
fatal: [localhost]: FAILED! => {"changed": false, "msg": "Failed to lock apt for exclusive operation: Failed to lock directory /var/lib/apt/l
ists/: E:Could not get lock /var/lib/apt/lists/lock. It is held by process 42388 (apt-get)"}

TASK [Ensure auditd is enabled and started] *************************************************************************************
ok: [CoryS-2504-STIG2]
ok: [CoryS-2504-STIG1]

TASK [Ensure libpam-pwquality is installed] *************************************************************************************
changed: [CoryS-2504-STIG1]
changed: [CoryS-2504-STIG2]

TASK [Set password maximum age] *************************************************************************************************
changed: [CoryS-2504-STIG2]
changed: [CoryS-2504-STIG1]

TASK [Set password minimum length] **********************************************************************************************
changed: [CoryS-2504-STIG2]
changed: [CoryS-2504-STIG1]

TASK [Set password warning age] *************************************************************************************************
changed: [CoryS-2504-STIG2]
changed: [CoryS-2504-STIG1]

TASK [Disable core dumps] *******************************************************************************************************
changed: [CoryS-2504-STIG2]
changed: [CoryS-2504-STIG1]

TASK [Set sticky bit on /tmp] ***************************************************************************************************
changed: [CoryS-2504-STIG1]
changed: [CoryS-2504-STIG2]

TASK [Ensure permissions on /etc/shadow are correct] ****************************************************************************
ok: [CoryS-2504-STIG2]
ok: [CoryS-2504-STIG1]

TASK [Ensure permissions on /etc/passwd are correct] ****************************************************************************
ok: [CoryS-2504-STIG2]
ok: [CoryS-2504-STIG1]

TASK [Ensure permissions on /etc/group are correct] *****************************************************************************
ok: [CoryS-2504-STIG2]
ok: [CoryS-2504-STIG1]

TASK [Disable USB storage (if applicable)] **************************************************************************************
changed: [CoryS-2504-STIG2]
changed: [CoryS-2504-STIG1]

TASK [Configure sysctl for IP spoofing protection] ******************************************************************************
changed: [CoryS-2504-STIG2]
changed: [CoryS-2504-STIG1]

TASK [Disable ICMP redirects] ***************************************************************************************************
changed: [CoryS-2504-STIG2]
changed: [CoryS-2504-STIG1]

TASK [Enable ExecShield (if supported)] *****************************************************************************************
fatal: [CoryS-2504-STIG2]: FAILED! => {"changed": false, "msg": "Failed to reload sysctl: fs.suid_dumpable = 0\nnet.ipv4.conf.all.rp_filter =
 1\nnet.ipv4.conf.all.accept_redirects = 0\nsysctl: cannot stat /proc/sys/kernel/exec-shield: No such file or directory\n"}
...ignoring
fatal: [CoryS-2504-STIG1]: FAILED! => {"changed": false, "msg": "Failed to reload sysctl: fs.suid_dumpable = 0\nnet.ipv4.conf.all.rp_filter =
 1\nnet.ipv4.conf.all.accept_redirects = 0\nsysctl: cannot stat /proc/sys/kernel/exec-shield: No such file or directory\n"}
...ignoring

TASK [Set permissions on /etc/cron.d] *******************************************************************************************
```

FINAL SCRIPT

- *Droplet1.yml*

```yaml
---
- name: Create Digital Ocean Droplets
  hosts: localhost
  connection: local
  gather_facts: false

  vars:
    droplet_name_prefix: "CoryS-2504-STIG"
    region: "nyc3"
    size: "s-1vcpu-1gb"
    image: "ubuntu-22-04-x64"
    ssh_key_name: "Cory-Searcy-2504"
    number_of_droplets: 2
    droplets:
      - CoryS-2504-STIG1
      - CoryS-2504-STIG2

  tasks:
    - name: Create SSH Key in Digital Ocean
      digital_ocean_sshkey:
        name: "{{ ssh_key_name }}"
        ssh_pub_key: "{{ lookup('file', '~/.ssh/id_rsa.pub') }}"
      register: ssh_key_result

    - name: Create Droplets
      digital_ocean_droplet:
        name: "{{ droplet_name_prefix }}{{ item }}"
        region: "{{ region }}"
        size: "{{ size }}"
        image: "{{ image }}"
        ssh_keys: "{{ ssh_key_result.data.ssh_key.id }}"
        unique_name: yes
        state: present
      loop: "{{ range(1, number_of_droplets + 1) | list }}"
      register: droplet_result

    - name: Wait for droplets to be created
      pause:
        seconds: 30

    - name: Add new droplets to host group
      add_host:
        name: "{{ item.networks.v4[0].ip_address }}"
        groups: droplets
      loop: "{{ droplets_info.data }}"
```

```
        when: item.networks is defined and item.networks.v4 |
    length > 0
```

- *Lamp.yml*

```
---
- name: Install LAMP stack on new droplets
  hosts: droplets
  become: yes
  gather_facts: yes

  tasks:
    - name: Update apt cache
      apt:
        update_cache: yes

    - name: Install Apache
      apt:
        name: apache2
        state: present

    - name: Install MySQL server
      apt:
        name: mysql-server
        state: present

    - name: Install PHP and common modules
      apt:
        name:
          - php
          - php-mysql
          - libapache2-mod-php
          - php-cli
          - php-curl
          - php-gd
          - php-mbstring
          - php-xml
          - php-xmlrpc
        state: present

    - name: Enable Apache mod_rewrite
      apache2_module:
        name: rewrite
        state: present

    - name: Restart Apache
      service:
        name: apache2
        state: restarted
```

```
  - name: Ensure Apache is enabled on boot
    systemd:
      name: apache2
            enabled: yes
```

- *Diastigs1.yml*

```
---
- name: Apply DISA STIG Hardening to Ubuntu Systems
  hosts: all
  become: yes
  vars:
    min_password_length: 14
    max_password_age: 60
    password_warn_age: 7

  tasks:

    - name: Ensure auditd is installed
      apt:
        name: auditd
        state: present
        update_cache: yes

    - name: Ensure auditd is enabled and started
      systemd:
        name: auditd
        enabled: yes
        state: started
```

```
- name: Set password maximum age
  lineinfile:
    path: /etc/login.defs
    regexp: '^PASS_MAX_DAYS'
    line: "PASS_MAX_DAYS {{ max_password_age }}"

- name: Set password minimum length
  lineinfile:
    path: /etc/security/pwquality.conf
    regexp: '^minlen'
    line: "minlen = {{ min_password_length }}"

- name: Set password warning age
  lineinfile:
    path: /etc/login.defs
    regexp: '^PASS_WARN_AGE'
    line: "PASS_WARN_AGE {{ password_warn_age }}"

- name: Disable core dumps
  sysctl:
    name: fs.suid_dumpable
    value: '0'
    state: present
    reload: yes

- name: Set sticky bit on /tmp
```

```
  command: chmod +t /tmp

- name: Ensure permissions on /etc/shadow are correct
  file:
    path: /etc/shadow
    owner: root
    group: shadow
    mode: '0640'

- name: Ensure permissions on /etc/passwd are correct
  file:
    path: /etc/passwd
    owner: root
    group: root
    mode: '0644'

- name: Ensure permissions on /etc/group are correct
  file:
    path: /etc/group
    owner: root
    group: root
    mode: '0644'

- name: Disable USB storage (if applicable)
  lineinfile:
    path: /etc/modprobe.d/disable-usb.conf
    create: yes
```

```
    line: "install usb-storage /bin/true"

- name: Configure sysctl for IP spoofing protection
  sysctl:
    name: net.ipv4.conf.all.rp_filter
    value: '1'
    state: present
    reload: yes

- name: Disable ICMP redirects
  sysctl:
    name: net.ipv4.conf.all.accept_redirects
    value: '0'
    state: present
    reload: yes

- name: Enable ExecShield (if supported)
  sysctl:
    name: kernel.exec-shield
    value: '1'
    state: present
    reload: yes
  ignore_errors: yes

- name: Set permissions on /etc/cron.d
  file:
    path: /etc/cron.d
```

```
    state: directory
    owner: root
    group: root
    mode: '0700'

- name: Restrict cron to authorized users
  copy:
    dest: /etc/cron.allow
    content: "root\n"
    owner: root
    group: root
    mode: '0600'

- name: Disable IPv6 if not needed
  sysctl:
    name: net.ipv6.conf.all.disable_ipv6
    value: '1'
    state: present
    reload: yes

- name: Ensure default umask is 027
  lineinfile:
    path: /etc/profile
    regexp: '^umask'
    line: 'umask 027'

- name: Lock inactive user accounts after 35 days
```

```
    lineinfile:
      path: /etc/default/useradd
      regexp: '^INACTIVE='
      line: 'INACTIVE=35'

  - name: Enforce PAM password complexity
    lineinfile:
      path: /etc/pam.d/common-password
      regexp: '^password\s+requisite\s+pam_pwquality.so'
      line: 'password requisite pam_pwquality.so retry=3 minlen={{ min_password_length }} ucredit=-1 lcredit=-1
dcredit=-1 ocredit=-1'

  - name: Ensure rsyslog is installed
    apt:
      name: rsyslog
      state: present

  - name: Ensure rsyslog is enabled and running
    systemd:
      name: rsyslog
      enabled: yes
      state: started

  - name: Ensure permissions on rsyslog.conf are correct
    file:
      path: /etc/rsyslog.conf
      owner: root
```

```
    group: root
    mode: '0644'

- name: Ensure audit rules are configured
  copy:
    src: audit.rules
    dest: /etc/audit/rules.d/stig.rules
    owner: root
    group: root
    mode: '0600'
    notify: restart auditd

- name: Ensure telnet is not installed
  apt:
    name: telnet
    state: absent

- name: Ensure rsh-server is not installed
  apt:
    name: rsh-server
    state: absent

- name: Ensure xinetd is not installed
  apt:
    name: xinetd
    state: absent
```

```
- name: Disable Ctrl+Alt+Del reboot
  copy:
    dest: /etc/systemd/system/ctrl-alt-del.target
    content: ''
    owner: root
    group: root
    mode: '0644'

- name: Remove .netrc files (insecure)
  find:
    paths: /home
    patterns: '.netrc'
    recurse: yes
  register: netrc_files

- name: Delete .netrc files
  file:
    path: "{{ item.path }}"
    state: absent
  loop: "{{ netrc_files.files }}"
  when: netrc_files.matched > 0

- name: Disable SSH root login
  lineinfile:
    path: /etc/ssh/sshd_config
    regexp: '^PermitRootLogin'
    line: 'PermitRootLogin no'
```

```
    create: yes
    backup: yes
  notify: restart ssh

- name: Set SSH idle timeout to 10 minutes
  lineinfile:
    path: /etc/ssh/sshd_config
    regexp: '^ClientAliveInterval'
    line: 'ClientAliveInterval 600'
    create: yes
  notify: restart ssh

- name: Set SSH session disconnect after 3 failures
  lineinfile:
    path: /etc/ssh/sshd_config
    regexp: '^MaxAuthTries'
    line: 'MaxAuthTries 3'
    create: yes
  notify: restart ssh

- name: Set login banner (DoD warning)
  copy:
    dest: /etc/issue
    content: |
      You are accessing a U.S. Government (USG) Information System (IS)
      that is provided for USG-authorized use only.
    owner: root
```

```
    group: root
    mode: '0644'

handlers:
  - name: restart auditd
    service:
      name: auditd
      state: restarted

  - name: restart ssh
    service:
      name: ssh
      state: restarted
```

LESSONS LEARNED

One lesson learned from this project is that for some reason, I had to add my droplets to the inventory file to gain access to my droplets successfully.

APPENDIX A | DOCUMENT ERRORS & SOLUTIONS

1 - Error Title: Failed to connect to the host via ssh.

```
cory@cory-virtual-machine:/etc/ansible$ ansible-playbook -i inventory.ini lamp1.yml

PLAY [Install LAMP stack on new droplets] ***************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************************
fatal: [CoryS-2504-STIG1]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: root@159.65.187.82: Permission d
enied (publickey).", "unreachable": true}
fatal: [CoryS-2504-STIG2]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: root@159.203.100.33: Permission
denied (publickey).", "unreachable": true}

PLAY RECAP **********************************************************************************************************************************
CoryS-2504-STIG1           : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
CoryS-2504-STIG2           : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
```

Error explanation: Droplets are not being automatically added to inventory.ini

Solution: Manually add droplets to inventory.ini

```
 GNU nano 6.2                                        inventory.ini
[local]
localhost ansible_connection=local

[droplets]
CoryS-2504-STIG2 ansible_host=68.183.48.88 ansible_user=root ansible_ssh_private_key_file=~/.ssh/id_rsa
CoryS-2504-STIG1 ansible_host=159.203.139.197 ansible_user=root ansible_ssh_private_key_file=~/.ssh/id_rsa

[droplets:vars]
ansible_user=root
ansible_ssh_private_key_file=~/.ssh/id_rsa
ansible_ssh_common_args='-o StrictHostKeyChecking=no'
```

APPENDIX B | DISA STIGS LIST

List all the STIGs, and the risk category and a brief description of each. See example below.

STIG #	Risk Category	Description
UBTU-20-010007	(Cat 3) Low	The Ubuntu operating system must enforce 24 hours/1 day as the minimum password lifetime.
UBTU-20-010008	(Cat 3) Low	The Ubuntu operating system must enforce a 60-day maximum password lifetime restriction.

UBTU-20-010050	(Cat 3) Low	The Ubuntu operating system must enforce password complexity by requiring that at least one upper-case character be used.
UBTU-20-010051	(Cat 3) Low	The Ubuntu operating system must enforce password complexity by requiring that at least one lower-case character be used.
UBTU-20-010052	(Cat 3) Low	The Ubuntu operating system must enforce password complexity by requiring that at least one numeric character be used.
UBTU-20-010053	(Cat 3) Low	The Ubuntu operating system must require the change of at least 8 characters when passwords are changed.
UBTU-20-010055	(Cat 3) Low	The Ubuntu operating system must enforce password complexity by requiring that at least one special character be used.
UBTU-20-010070	(Cat 3) Low	The Ubuntu operating system must prohibit password reuse for a minimum of five generations.
UBTU-20-010075	(Cat 3) Low	The Ubuntu operating system must enforce a delay of at least 4 seconds between logon prompts following a failed logon attempt
UBTU-20-010410	(Cat 3) Low	The Ubuntu operating system must automatically remove or disable emergency accounts after 72 hours."
UBTU-20-010000	(Cat 2) Medium	The Ubuntu operating system must provision temporary user accounts with an expiration time of 72 hours or less."
UBTU-20-010004	(Cat 2) Medium	The Ubuntu operating system must retain a user's session lock until that user reestablishes access using established identification and authentication procedures.
UBTU-20-010010	(Cat 2) Medium	The Ubuntu operating system must uniquely identify interactive users.
UBTU-20-010013	(Cat 2) Medium	The Ubuntu operating system must automatically terminate a user session after inactivity timeouts have expired.

UBTU-20-010035	(Cat 2) Medium	The Ubuntu operating system must use strong authenticators in establishing nonlocal maintenance and diagnostic sessions.
UBTU-20-010036	(Cat 2) Medium	The Ubuntu operating system must immediately terminate all network connections associated with SSH traffic after a period of inactivity.
UBTU-20-010037	(Cat 2) Medium	The Ubuntu operating system must automatically terminate all network connections associated with SSH traffic at the end of the session or after 10 minutes of inactivity.
UBTU-20-010049	(Cat 2) Medium	The Ubuntu operating system SSH daemon must prevent remote hosts from connecting to the proxy display.
UBTU-20-010054	(Cat 2) Medium	The Ubuntu operating system must enforce a minimum 15-character password length.
UBTU-20-010056	Cat 2) Medium	The Ubuntu operating system must prevent the use of dictionary words for passwords.