Deployment Documentation

1. Prerequisites

a. Target Infrastructure – This project deployment is made to run on 3 Ubuntu Server VMs which is then managed by a main Ubuntu Server VM.
    i. SSH Enabled with openssh-server package installed and running on each machine
    ii. A user account called cyberrange that must be capable of passwordless sudo on the 3 non main VM.
    iii. All 4 VMs must be on the same network/subnet or reachable to eachother

b. Main VM or Control Node package requirements - The main VM that acts as a control node must be configured with these packages
    i. Ansible installed via apt
    ii. Your main machine (non vm) running VS code
    iii. VS Code must be configured with Remote SSH and Ansible extensions for remote management of the project
    iv. VS code must then be connected to the main VM over remote SSH
    v. This project repository must be cloned to the main VM. (Can be done through VS code)

c. Pre-Deployment Configuration – before running the automation for the deployment you must edit inventory.ini within vs code
   i. Edit the placeholder IP addresses with the actual IP addresses of the 3 Ubuntu VMs you'll be deploying the automation on
   ii. Also ensure the user account with passwordless sudo permissions matches the placeholder or change it to what you set your account on all 3 VMs. All 3 Vms must have the same username and password
   iii. Screenshot of Inventory.ini with labeled IP's and credentials

inventory.ini

```
# Ansible Inventory File
# This file tells Ansible about all servers and how to group th

# server groups

# Zone 1 (Public Facing)
[gateway]
# Hosts the vsftpd service with ananymous access enabled
dmz-ftp-gateway ansible_host=100.65.4.128

# Zone 2 (The Internal Network Servers)
[internal_servers]
# simulates an internal application server for FTP. Target for
internal-app-serv ansible_host=100.65.6.14
# Simulates a backend database and target for credentials being
internal-db-serv  ansible_host=100.65.3.234

# Global configuration for all VMs/Machines
[all:vars]
# remote management login
ansible_user=cyberrange
ansible_password=Cyberrange
```

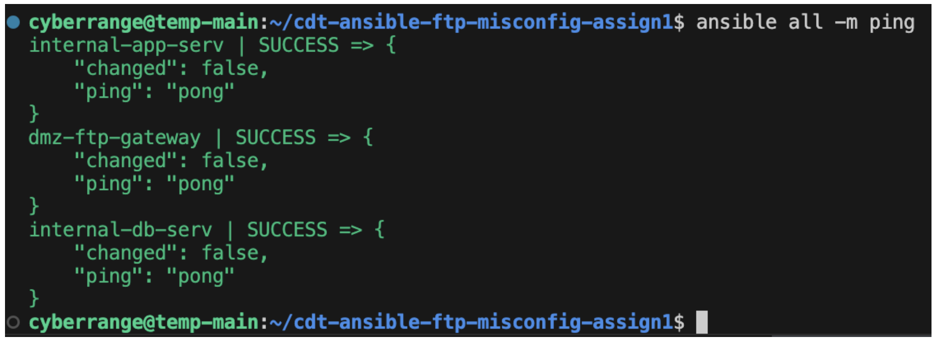
2. Installation & Configuration

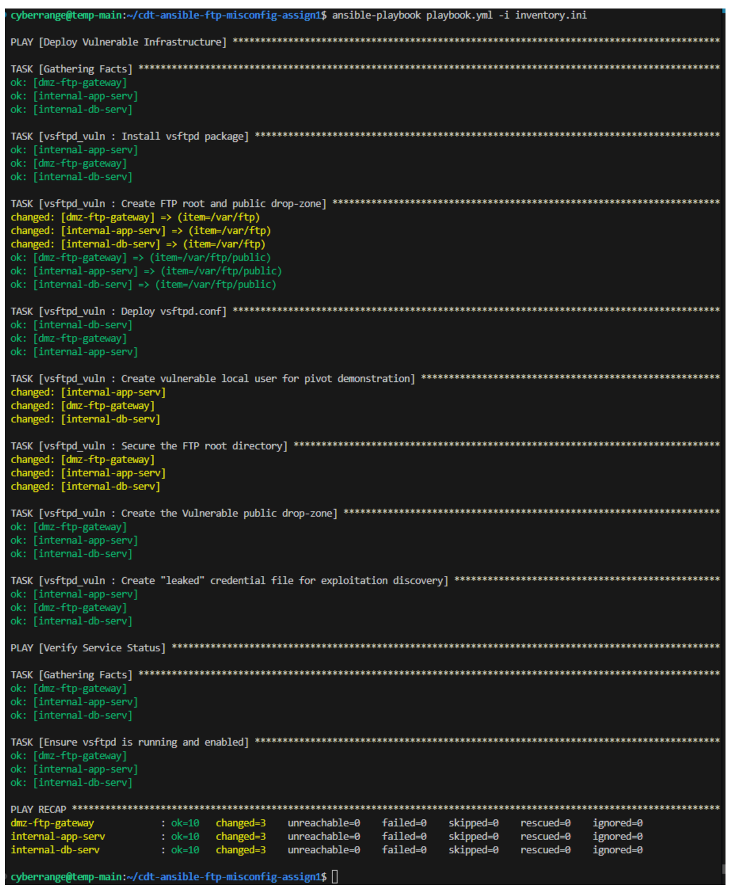
a. Establish SSH Connectivity with your VMs

i. Run the included setup script in the terminal for the Main VM Control Node to ensure the Main VM or Control Node can communicate with the 3 hosting VMs without having to do manual prompts

1. ./setup-ssh.sh
2. You should get an output as shown below (It may ask you to enter the password for the ssh users to test if not using a key)

```
cyberrange@temp-main:~/cdt-ansible-ftp-misconfig-assign1$ ./setup-ssh.sh
==========================================
  Ansible Demo - SSH Key Setup
==========================================

Step 1: Checking for SSH key...
✓ SSH key already exists at /home/cyberrange/.ssh/id_ed25519

Step 2: Reading server IPs from inventory...
Found servers:
100.65.4.128
100.65.6.14
100.65.3.234

Step 3: Configuring SSH for jump host...
✓ SSH config already configured

Step 4: Copying SSH key to servers...
(You may see some warnings - that's normal)

  Copying key to 100.65.4.128... (may already be configured)
  Copying key to 100.65.6.14... (may already be configured)
  Copying key to 100.65.3.234... (may already be configured)

Step 5: Testing connections...

cyberrange@100.65.4.128's password:
✓ Connected!
cyberrange@100.65.6.14's password:
✓ Connected!
cyberrange@100.65.3.234's password:
✓ Connected!

==========================================
  All servers connected successfully!
==========================================

You can now run Ansible commands:
  ansible all -m ping

cyberrange@temp-main:~/cdt-ansible-ftp-misconfig-assign1$
```

b. Connection Verification with VMs and Ansible
   i. Run the ansible command below to verify that ansible can communicate with the VMs in the infrastructure. You'll get an output similar to what's shown below to show successful pong message
      1. "ansible all -m ping"
      2. 

c. Execution of the Deployment
   i. Now you'll run the main deployment playbook by using the command below. This playbook installs the vsftpd service and applies the configurations and sets up the local user. Below is a screenshot of what you will see when it runs.
      1. "ansible-playbook playbook.yml -I inventory.ini"
      2. 

3. Verification Steps

a. Automated Audit

i. This verifies the deployment is running but using the validate.yml playbook. It audits the security state of the VMs to ensure the configurations are correct and the vulnerabilities are enabled. You'll run this command below and see this output.

ii. "ansible-playbook validate.yml -i inventory.ini"

```
cyberrange@temp-main:~/cdt-ansible-ftp-misconfig-assign1$ ansible-playbook validate.yml -i inventory.ini

PLAY [Validate FTP Gateway Vulnerabilities] ***************************************************************

TASK [Gathering Facts] ************************************************************************************
ok: [dmz-ftp-gateway]

TASK [Gather service status] ******************************************************************************
ok: [dmz-ftp-gateway]

TASK [CHECK: vsftpd service is running] *******************************************************************
ok: [dmz-ftp-gateway] => changed=false
  msg: 'PASS: vsftpd service is running on temp-1'

TASK [Verify anonymous access configuration] **************************************************************
ok: [dmz-ftp-gateway]

TASK [CHECK: Anonymous FTP is enabled (Intentional Vulnerability)] ****************************************
ok: [dmz-ftp-gateway] => changed=false
  msg: 'PASS: Anonymous access is ENABLED (Vulnerable state confirmed)'

TASK [Check FTP drop-zone permissions] ********************************************************************
ok: [dmz-ftp-gateway]

TASK [CHECK: FTP drop-zone has weak permissions (0777)] ***************************************************
ok: [dmz-ftp-gateway] => changed=false
  msg: 'PASS: Drop-zone is world-writable (Exploitation vector confirmed)'

PLAY [Validate Pivot Targets and Credentials] *************************************************************

TASK [Gathering Facts] ************************************************************************************
ok: [internal-app-serv]
ok: [internal-db-serv]

TASK [Check for vulnerable developer user] ****************************************************************
ok: [internal-db-serv]
ok: [internal-app-serv]

TASK [CHECK: Vulnerable user account exists] **************************************************************
ok: [internal-app-serv] => changed=false
  msg: 'PASS: User ''dev_user'' exists for pivot demonstration'
ok: [internal-db-serv] => changed=false
  msg: 'PASS: User ''dev_user'' exists for pivot demonstration'

TASK [Final Success Summary] ******************************************************************************
ok: [internal-app-serv] =>
  msg:
  - '  ALL CHECKS PASSED – Your Vulnerable FTP Stack is Active!'
  - ''
  - '  [gateway]  vsftpd service ........... PASS'
  - '  [gateway]  Anonymous Access .......... PASS'
  - '  [gateway]  Weak Permissions (0777) ... PASS'
  - '  [targets]    Local User Account ........ PASS'
  - ''
  - '  Test locally from your control node using:'
  - '  ftp -v 100.65.4.128'
  - ==========================================================

PLAY RECAP ************************************************************************************************
dmz-ftp-gateway            : ok=7    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
internal-app-serv          : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
internal-db-serv           : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

cyberrange@temp-main:~/cdt-ansible-ftp-misconfig-assign1$
```

iii.

b. Manual service confirmation

   i. To manually confirm the status of the gateway service, log into the gateway VM via ssh and run the command below. You should see something similar to the screenshot below.

      1. “systemctl status vsftpd”

      2.
```
cyberrange@temp-4:~$ systemctl status vsftpd
● vsftpd.service - vsftpd FTP server
     Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: >
     Active: active (running) since Thu 2026-02-05 12:06:23 EST; 8min ago
    Process: 20233 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exit>
   Main PID: 20235 (vsftpd)
      Tasks: 1 (limit: 2322)
     Memory: 708.0K (peak: 964.0K)
        CPU: 6ms
     CGroup: /system.slice/vsftpd.service
             └─20235 /usr/sbin/vsftpd /etc/vsftpd.conf

Feb 05 12:06:23 temp-4 systemd[1]: Starting vsftpd.service - vsftpd FTP server>
Feb 05 12:06:23 temp-4 systemd[1]: Started vsftpd.service - vsftpd FTP server.
lines 1-13/13 (END)
```

c. Network Connectivity test

   i. Install and Use Nmap to scan from. The control node to confirm that port 21 for FTP is open on the gateway VM. The command you can use as example is listed below. Screenshot shows the output you should see.

      1. “sudo apt install nmap -y || nmap -p 21 100.65.4.128”

      2.
```
cyberrange@temp-main:~/cdt-ansible-ftp-misconfig-assign1$ nmap
-p 21 100.65.4.128
Starting Nmap 7.94SVN ( https://nmap.org ) at 2026-02-05 13:27
EST
Nmap scan report for 100.65.4.128
Host is up (0.010s latency).

PORT   STATE SERVICE
21/tcp open  ftp

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
cyberrange@temp-main:~/cdt-ansible-ftp-misconfig-assign1$
```

d. Expected Access

   i. Now there are the following access points available for exploitation

      1. On gateway VM, FTP server
      2. Application server for ssh with weak cred
      3. Database server with ssh reuse