Setting Up and Installing GOAD or GOAD-Light on VMware ESXi

netsecfocus.com/infosec/walkthrough/2024/08/21/Setting_up_and_Installing_GOAD_or_GOAD-Light_on_VMware_ESXI.html

tjnull August 21, 2024

August 21, 2024 - tjnull

Table of Contents

Introduction

Over the years, I've been refining and automating vulnerable Active Directory environments in my homelab for testing. However, as these setups I created grew in complexity, managing the tools, scripts, and resources became a challenge. I needed a more efficient way to quickly spin up and tear down these environments without the hassle. That's when I discovered GOAD by Orange Cyberdefense. In my experience, it is a game-changer for anyone serious about Active Directory security testing.

GOAD is a comprehensive Active Directory (AD) lab environment designed for security testing, training, and learning purposes. It allows pentesters or security researches to simulate real-world AD environments to practice various attack and defense techniques. The lab is highly customizable, enabling users to configure different scenarios, user accounts, policies, and network topologies to mirror a production AD setup.

In this guide, I will walk you through the steps needed to install and configure GOAD on VMware ESXi.

A word of Caution:

This lab environment is intentionally vulnerable. Do not reuse this setup for production environments, and ensure it is isolated from any production networks. Never deploy it on the internet without strict isolation measures.

A word of Advice:

As of the time of writing, the Orange-Cyberdefense team has not yet merged the contributions from viris and fsacer into the main GOAD project. Their forked versions of GOAD include the vmware_esxi provider, which we'll be using for deployment. If you want to review and compare the enhancements they've made, you can check out their work:

Viris version of GOAD: https://github.com/viris/GOAD Fsacer version of GOAD: https://github.com/fsacer/GOAD

Requirements to deploy GOAD:

Before deploying the GOAD environment on your ESXi server there are a few things we need to configure:

1. Disk Space: A minimum of 125GB or more is needed to build the lab.
2. Memory: A minimum of 6-8GB should be allocated for each system in the lab.
3. Networking: We will need the ability to create port groups to ensure our GOAD builder can create the necessary connections and interfaces to seperate it from our personal network.

Operating System:

This lab is designed to be installed from a Linux host, and all testing has been conducted using Linux! While some users have successfully set up the lab from a Windows OS, this requires a slightly different approach VM Creation: VMs can be created on Windows using Vagrant.

Ansible Provisioning: The provisioning part must be executed from a Linux machine. If you choose to use a Windows OS for VM creation, ensure the following for the Linux machine used in provisioning:

Network Configuration: The Linux machine must have two network adapters.

1. One set to “VM Network” and the other connected to the same virtual private network or network port group as the lab. This ensures proper communication between the VMs and the provisioning scripts.

It is up to you to decide which operating system you want to use to deploy GOAD but in this situation we are going to use Linux to deploy it.

Current ESXI Setup:

Here is current setup that I will be using to deploy the GOAD environment:

Hardware: CPU: 24 CPUs x Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz Memory: 400GB DDR4 Storage: 4TB of lab storage ESXI Version: 8.0 Update 2

Configure GOAD network Group

Before we setup our linux system to build the GOAD environment we need to create a network port group that will use the virtual private network for the lab.

Once you login into your ESXI console, on the left-hand menu, click on “Networking” under the “Navigator” pane.

Under the “Port groups” tab, identify the virtual switch (vSwitch) where you want to create the new port group. Click on the “Add port group” button.

Name: Enter GOAD as the name of the new port group. VLAN ID: If required, specify a VLAN ID. Leave it as 0 if no VLAN tagging is needed. Security: Optionally configure settings like Promiscuous mode, MAC Address changes, and Forged Transmits as needed.

Save the Configuration:

Click “Add” to create the port group.

Ensure that the new port group GOAD appears in the list of port groups under the selected vSwitch.

Obtain required packages to deploy GOAD with our Linux machine.

Note: While deploying the GOAD environment, I used my Kali Linux system to accomplish this. The instructions I am providing should also work for other Linux Operating systems. You may have to install some other packages and plugins to make sure the lab can be deployed properly in your ESXI server.

Remember you want to have two network interfaces configured on your Kali Linux system. One interface should be connected to your VM Network and the other interface should be connected to the GOAD network. Otherwise GOAD will not deploy the systems and set the network configurations properly during the deployment.

By default the GOAD network has the boxes set to be on the 192.168.56.0/24 network. For the second interface that connected to the GOAD environment, I set my Kali Linux to 192.168.56.2 with a /24 subnet and the gateway set as 192.168.56.1.

1. Installing the following packages:

```
Vagrant:

sudo apt install vagrant
sudo apt install ansible
sudo apt install ansible core
sudo pip3 install pywinrm
```

1. Install Vagrant ESXI Plugins:

```
vagrant plugin install vagrant-vmware-esxi
vagrant plugin install vagrant-reload
vagrant plugin install vagrant-vmware-desktop
vagrant plugin install winrm
vagrant plugin install winrm-fs
vagrant plugin install winrm-elevated
```

1. Install VMware OVFTool

Thanks to broadcom for changing VMware URL structure it took me some time to find the latest version of OVFtool for Linux.

Source: https://developer.broadcom.com/tools/open-virtualization-format-ovf-tool/latest

Download the “OVF Tool for Linux Zip” package. The current version we used is 4.6.3.

Once you downloaded the zip file and extracted the files on your Kali Linux machine you can either echo the path in your environment variable or you can add the path to your .bashrc file.

```
tjnull@auto-kali:~/Downloads/VMware-ovftool-4.6.3-24031167-lin.x86_64/ovftool$
echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/games:/usr
/games

tjnull@auto-kali:~/Downloads/VMware-ovftool-4.6.3-24031167-
lin.x86_64/ovftool$export PATH=/home/tjnull/VMware-ovftool-4.6.3-24031167-
lin.x86_64/ovftool:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/u
sr/local/games:/usr/games
```

Once our path has been added and we verified ovftool can be loaded into our terminal, we can now start the process to deploy GOAD.

Stage 1: Deploying the GOAD Environment

Now we have the necessary packages, plugins, and tools installed, we need to make some changes to the configuration files in the vmware_esxi directory listed in the provider folder. To do this, we are going to do the following:

```
tjnull@auto-kali:git clone https://github.com/viris/GOAD
tjnull@auto-kali:cd GOAD/ad/providers/vmware_esxi
tjnull@auto-kali: ls -a
. .. .env inventory .vagrant Vagrantfile
```

In the .env file, make the necessary changes to make sure GOAD can connect to your ESXI Server:

```
tjnull@conops:/opt/GOAD/ad/GOAD/providers/vmware_esxi$ cat .env
export GOAD_VAGRANT_ESXIHOSTNAME='10.10.10.10'
export GOAD_VAGRANT_ESXIUSERNAME='root'
export GOAD_VAGRANT_ESXIPASSWORD='password'
export GOAD_VAGRANT_ESXINETNAT='VM Network'
export GOAD_VAGRANT_ESXINETDOM='GOAD'
export GOAD_VAGRANT_ESXISTORE='datastore1'
tjnull@conops:/opt/GOAD/ad/GOAD/providers/vmware_esxi$
```

You do not need to make any changes to the Vagrantfile or to the inventory file! To see if vagrant can reach the ESXI server you can type ‘vagrant up’ and vagrant will begin to deploy the GOAD virtual machines to your VMware ESXI Server. This does not run the ansible playbooks to deploy the vulnerable configurations to those machines.

The vagrantfile already has the systems set to the 192.168.56.0/24 network.

If you choose to not do this and you want to run the entire the deployment and ansible configuration setup at the same time we can use the GOAD script to do this. Deploying the environment this way can become very time consuming!

```
tjnull@auto-kali:/opt/GOAD/$ ./goad.sh -t check -l GOAD -p vmware_esxi -m local
```

From the script you should see in your output a list of checks that GOAD will run. Read through the results and if it passes you should then you can start the installation by running the following command:

```
tjnull@auto-kali:/opt/GOAD/$ ./goad.sh -t install -l GOAD -p vmware_esxi -m local
```

If you ran the vagrant up command and you see your virtual machines were installed in ESXI, then you can run the following command:

```
tjnull@auto-kali:/opt/GOAD/$ ./goad.sh -t check -l GOAD -p vmware_esxi -m local -a
```

The -a option will only run the ansible playbooks and will apply the vulnerable configurations into the active directory environment. The installation will take some time to deploy. Once the deployment has completed you will get a notification in your output and you can now begin assessing the GOAD environment!

Conclusion

A huge shoutout to Viris and Fsacer for creating the vmware_esxi provider for GOAD. I think it is an exceptional environment for any infosec professional looking to sharpen their skills in Active Directory security. It provides a realistic, fully-featured Active Directory environment that is intentionally vulnerable, offering an ideal playground for practicing attack techniques, testing detection tools, and exploring defense strategies. By investing time in GOAD, you gain hands-on experience with real-world scenarios, enabling you to better understand the complexities of Active Directory exploitation and defense. It’s an invaluable resource for both learning and advancing your expertise in securing critical infrastructure.

Previous post

TJnull's guide to building a Home Lab