

Greg Scharf — Development & Security

 blog.gregscharf.com/2023/04/14/goad-setup

14 апреля 2023 г.

3 minutes 535 Words

2023-04-14 15:14

📁 [Lab](#)

GOAD Lab Setup

GOAD is an Active Directory lab consisting of multiple Windows virtual machines containing many common misconfigurations and vulnerabilities that you might find in an Active Directory environment. A [visual representation](#) of the entire AD network on Orange Cyberdefense's GitHub provides an overview of the configuration along with the users, groups and running services.

I installed [GOAD](#) using Hyper-V's Quick Create with an Ubuntu 22 VM. I gave the main VM 250 GB of hard drive space (about 115 GB is all that is needed according to their documentation), 32 GB of ram and 6 processors. After everything was setup and running, including [Exegol](#) as my attack framework, [top](#) showed that I was using a total of 24 GB of memory, so configuring with a little less than 32 GB of ram will probably be fine.

After the VM is created open its settings, click on the hard drive and increase its size since by default Quick Create gives you only about a 20 GB hard drive. Once that's done start the VM. After the VM is fully booted and the install is complete you will need to use [gparted](#) to expand the drive to use all of the available hard drive space.

Before booting the VM for the first time you'll need to enable nested virtualization to allow VirtualBox to run inside of Hyper-V. Open powershell as Administrator and run the following command to identify the name of the VM you created for GOAD

```
C:\Windows\system32> get-vm
```

PowerShell

Use that name, which in my instance was "Ubuntu 22.04 LTS GOAD", and execute

```
C:\Windows\system32> get-vm | where Name -eq "Ubuntu 22.04 LTS GOAD" | set-vmprocessor -ExposeVirtualizationExtensions $true
```

PowerShell

Once the machine is booted make sure to use [gparted](#) to assign all available disc space to the main partition. This is a fairly simple task. Instructions can easily be found online.

Other than that, the installation instructions on the GitHub page worked just fine and the entire install process with Vagrant and Ansible ran without issue. Those instructions are included below along with the steps to install VirtualBox, Vagrant, Docker and other packages that weren't installed by default in the Ubuntu VM.

```
sudo apt install virtualbox git curl gnupg ca-certificates
```

```
wget -O- https://apt.releases.hashicorp.com/gpg | gpg --dearmor | sudo tee  
/usr/share/keyrings/hashicorp-archive-keyring.gpg
```

```
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg]  
https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee  
/etc/apt/sources.list.d/hashicorp.list
```

```
sudo apt update && sudo apt install vagrant
```

Bash

Install **Docker** using the [apt repository instructions](#) on the Docker site if you don't already have Docker installed. Reboot the Ubuntu VM what that's finished.

Clone the GOAD repository and run **vagrant**

```
$> git clone https://github.com/Orange-Cyberdefense/GOAD.git
```

```
$> cd GOAD
```

```
$> ./goad.sh -t check -l GOAD -p virtualbox -m docker
```

```
$> ./goad.sh -t install -l GOAD -p virtualbox -m docker
```

Bash

The installation takes quite a bit of time but it can run unattended, so once you execute the above you can leave it to do its thing and then check in on it from time to time. In case of a timeout run **./goad.sh -t install -l GOAD -p virtualbox -m docker** again and it will pick up where it left off.

For my attack machine I installed Exegol instead of a Kali VM. My instructions on installing Exegol can be viewed [here](#).
