GoibhniUWE

GoibhniUWE is a container based cyber range that allows end users to spin up complex attack scenarios spread across multiple networks quickly and easily.

GoibhniUWE offers built in logging and monitoring to allow for data from attacks to be analysed at a later date. This includes: * IDS - Suricata * Traffic Capture - Tshark * EFK - Elasticsearch, Filebeats and Kibana (containerised) * Container logs

This allows end users to setup live monitoring of attacks or capture data and load it back into an Elasticsearch stack afterwards. The EFK stack can also be turned off for a more lightweight expereince based around log and traffic capture analysis.

Initial Setup and Requirements

The range itself is built using python3 and comes with a requirements file. As it needs access to the underlying network and requires elevated privileges the range needs to be run with sudo permissions. As a result the required python modules need to be installed so that the root user can access / utilise them. For ease of use we suggest using `sudo`.

```
sudo pip3 install -r requirments.txt
```

The range also makes use of tshark - Enusre this is installed beforehand. For a Ubuntu / Debian envrionment (the target OS) use:

```
sudo apt install tshark
```

Container Usage and Setup

To run the container range move to the `GUI` folder and run:

```
sudo python3 ./manage runserver
```

This should launch the container range (which has been built using Django) and will accessible via the link provided in the terminal.

GoibhniUWE uses a combination of custom built containers and pre-existing vulnerable container enviornments found on vulnhub. Custom containers such as the attackbox, target OS and IDS are deployed by default while other custom environments can be selected or even created based on end user needs and requirements.

The pre-existing custom containers are hosted on DockerHub and are pulled down as and when required. While the container environments from vulnhub are loaded via a git clone and update process.

Attacks and Traffic

GoibhniUWE comes complete with the following offensive containers: * attackbox - Built on parrot OS this is the main offensive “actor” * auto_attack - An optional container that can be set to randomly scan / probe a target container * traffic_generator - An optional container that can be set to generate web traffic for a target container

The attackbox comes with a `auto_attack.py` script which takes commands in the form of a .json file. This can be used to automate attacks allowing for quick and easy replication of attacks as and when required.

GUI

GoibhniUWE uses a simple Django based API that is used to view: * Vulnhub Containers * Select and view containers based on their CWE and subsequent application or CVE

- Custom containers
  - Select and view pre-existing custom containers
- Environment and Setup Deployment
  - Allows end users to customise their container range topography, adding in auxillary containers, move containers between networks and save scenarios once setup and created. It also provides a graphical representation of the current container network deployment so end users can make note of assigned IP addresses and container connextability.
- Edit dockerfiles
  - Edit dockerfiles for containers selected from either Vulnhub or pre-existing custom containers. These can then be saved as new custome containers, selectable for future use and deployment

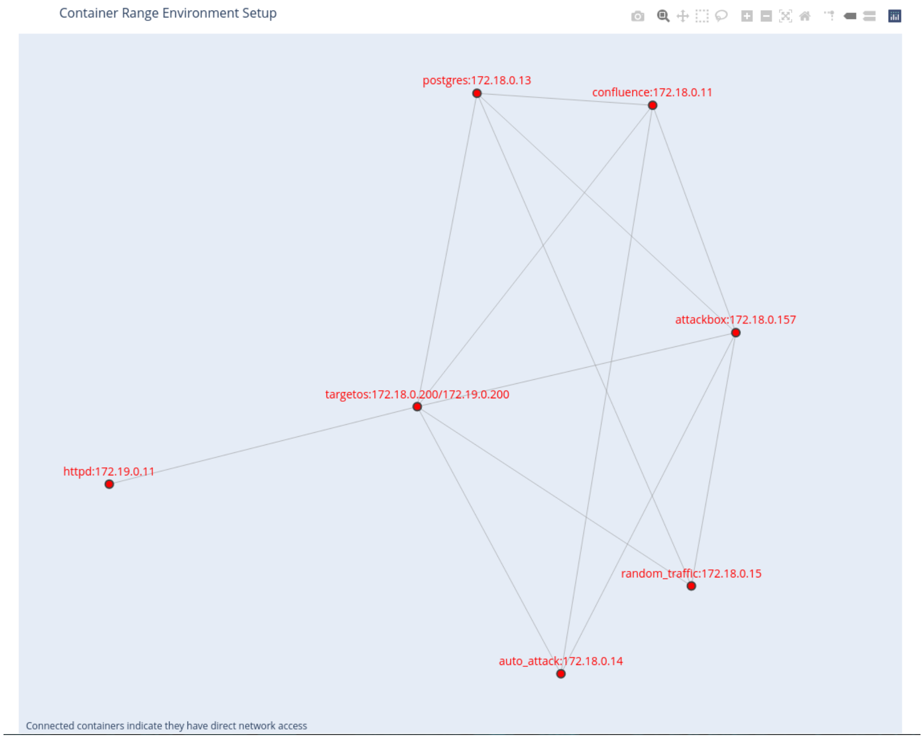
Container Range Environment Setup
postgres:172.18.0.13
confluence:172.18.0.11
attackbox:172.18.0.157
targetos:172.18.0.200/172.19.0.200
httpd:172.19.0.11
random_traffic:172.18.0.15
auto_attack:172.18.0.14
Connected containers indicate they have direct network access

Figure 1: Network Diagram Example