**Honeypot**

A Honeypot is a information security terminology that pertains to a mechanism that is intended to counteract, deflect and detect directed efforts pf unauthorized usage of systems. In a general sense, a honeypot contains data or a set of data that may seem to appear as a legitimate component of a network or system but in reality is actually a standalone system that is monitored. It lures potential attackers as these honeypots may seems to contain resources or information that could be of value to attackers.

**HoneyD**

HoneyD is an open source honeypot tool that behave as a daemon. Its capable of creating virtual networks and hosts. These hosts are able to be configured to perform arbitrary computer services and is capable of adapting personalities so they are presented as running specific operating systems. HoneyD is able to configure a single host to claim a vast amount of private IP addresses on a network simulated LAN. HoneyD promotes cyber security by implementing a mechanism for threat assessment, detection. It is also used to deter threats by masking actual systems inn the midst of the virtual systems.

Figure 7 below shows a basic network topology design using Honeyd. It shows two virtual host with HoneyD using the IP space 10.40.231.146 for Host 1 and 10.40.231.147 for Host 2. These two host are monitoring any activity that has these 2 IP address as the destination.

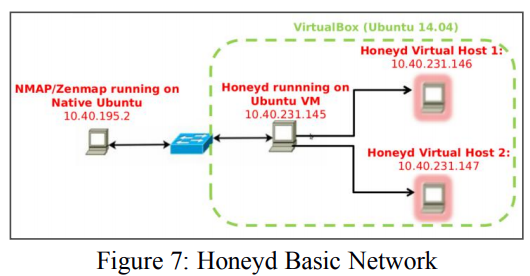
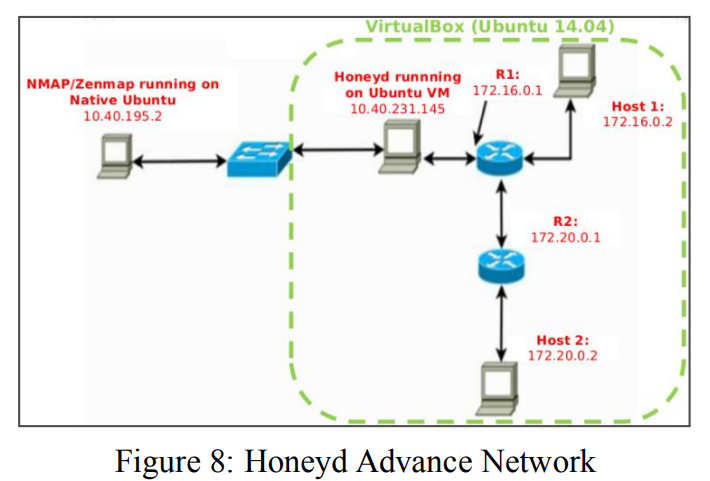
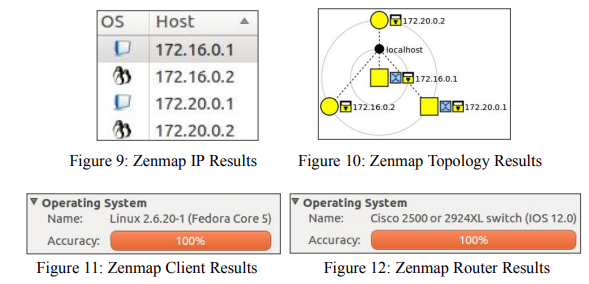


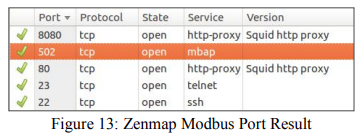
Figure 8 below displays another virtual network topology designed using HoneyD. As seen below, the network also includes two routers interconnected with two hosts. To increase the deception factor, configuration of the routers and hosts generated what is called a “personality” to emulate real machines and OS. The routers to act like Cisco 2500 (IOS 12.0) routers and the hosts to act like Linux 2.6.20-1 Fedora Core 5.



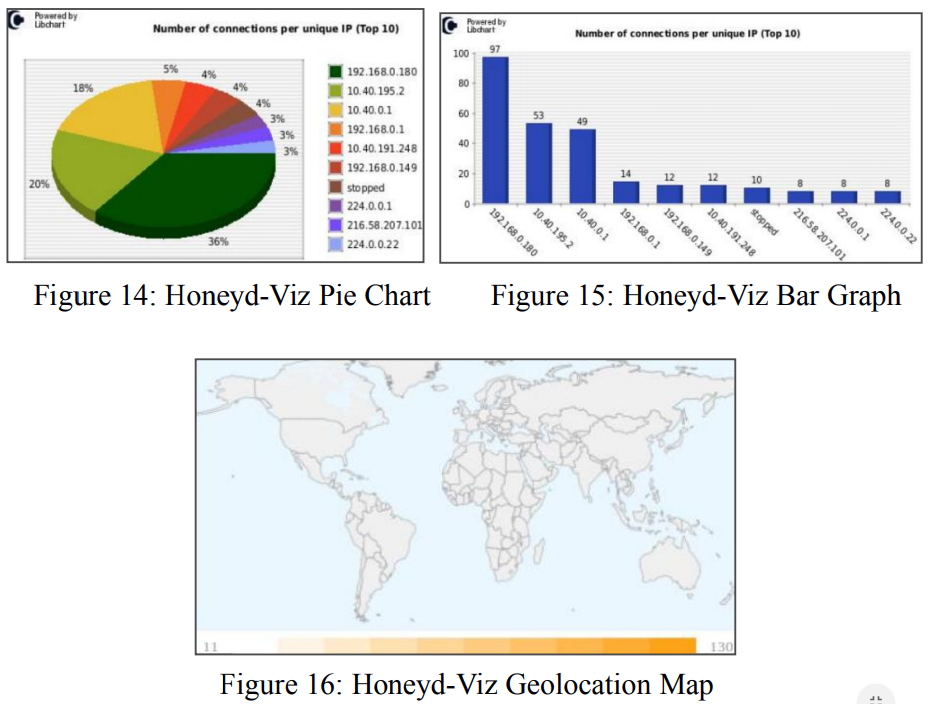
To ensure that the HoneyD setup was working, a simulated attack was carried out using an external machine. A GUI based network exploration tool called Zenmap was used to attempt to collect valuable information from the HoneyD setup. Figure 9, 10, 11, and 12 below displays the Zenmap results. It shows that a potential attacker will detect the virtual network setup upon a simple Zenmap search. Furthermore, Figure 10 below shows that Zenmap was able to also detect the exact virtual network. In addition, Figure 11 and Figure 12 below shows that Zenmap was able to detect the configured personality of the virtual hosts and the virtual routers routers respectively.



In order to emulate protocol communications, HoneyD uses python scripts that emulates several protocol communications such as Telnet and FTP. Since this project is concerned about securing and protecting SCADA and CPS protocols, vendor specific SCADA/CPS protocols such as ftp-Schneider, telnet-Schneider, telnet-Siemens and Modbus emulation scripts was used in the HoneyD virtual hosts. Once again, the simulated attack used an external machine to test the protocol emulation. The test was successful and the simulated attack was able to communicate with the HoneyD virtual host using SCADA/CPS ports. Figure 13 below shows how Zenmap was used to detect Modbus among other open ports in one of my HoneyD virtual host.



With regards to data visualization of the gathered logs, there is a Perl script that automatically converts HoneyD logs into a tabulated MySQL table called Honeyd2MySQL. This MySQL data is used by the web interface called Honeyd-Viz which creates a variety of bar graphs, pie charts, linear graphs, and geolocation map regarding the HoneyD logs collected. Figure 14, 15 and 16 were all the results of Honeyd-Viz which displays the number of connections per unique IP in pie chart and bar graph and the geolocation of the source according to the source IP respectively



**Conpot**

Conpot is an open source server side low-interaction type of SCADA/CPS/ICS honeypot easily deployed, extended and modified. Its provides a variety of different commonly used standard protocols for industrial control systems. Conpot is able to emulate sophisticated infrastructures to delude attackers into thinking that they have just found a massive industrial complex. Conpot is built upon scripts that mirror how certain SCADA/CPS ports communicate. When an adversary attempts to communicate with the sensors, Conpot will reply as if they were real connections and reply in a specific manner to certain queries or probes. Conpot enables the user to fashion their own template to create a specific decoy system of their design. In addition, Conpot also enables the use of extended real hardware to improve system emulations. Conpot is a designed and developed by the Honeynet Project.

The default template emulates the Siemens SIMATIC S7-200 hardware (commonly used in ICS/SCADA/CPS). Other templates include the IEC104 (power plants) and the Kamstrup (water and electricity meter) and Guardian\_AST (gas tank meters).

The Conpot sensors have gone through extensive and deliberate connectivity tests to ensure that potential attackers could be convinced that these ports are coming from real a SCADA/CPS. The tests included a ping, traceroute, wget (port 80/HTTP), NMAP TCP, NMAP NSE, and several Modbus command tests.

There are several screenshots below of the input and output of these tests. Figure 21 and Figure 22 displays the basic ping test by pinging the public address and waiting on the Conpot end to receive the traffic. Figure 23 and Figure 24 displays the HTTP test using the wget command to which the Conpot receive and logs the traffic. Figure 25 and Figure 26 display the basic NMAP test with a twist of specifically checking for the Modbus port 502 to which the Conpot responds with implying that the port is in an open state even though the port is actually fabricated. Figure 27 and Figure 28 displays the NSE at work to check on the bits and proprietary device identification of the Modbus to which the Conpot replies with “Siemens SIMATIC S7-200” to elude attackers in to thinking this is a real machine with a running Modbus. Finally, Figure 29 and Figure 30 displays the Modbus command to probe into Modbus registry bits.



**Dionaea**

Dionaea is a special type of honeypot that is able to capture Malware. Its intention is to seize Malware exploitation vulnerabilities disclosed by services available over networks and to ultimately possess a copy of the Malware.

The current progress of Dionaea is rudimentary. Basic installation is completed and is fully up and running to collect data.

Major issues were encountered such as inflation of SMB error in the logs which caused the sensor to be incapacitated due to insufficient storage. The cause and solution to this issue is yet to be found. Furthermore, the Dionaea FR (web interface for visualization of data) was not appearing for some reason. The cause and solution is also yet to be found for this issue.

**Honeypot Comparison**

Overall, there were three honeypot tools tried and tested in total. The primary honeypot used was Conpot as It was more geared towards SCADA and CPS. HoneyD had major issues that prevented it from being deployed with a public IP address as well as the limitation of the number of SCADA/CPS ports readily available for emulation. Dionaea also had major problems that prevented it from being deployed and ultimately this Honeypot was focused on Malware rather than SCADA/CPS operations. Table 1 below summarizes the pro and con of each Honeypot.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| HoneyD | | Conpot | | Dionaea | |
| Pro | Con | Pro | Con | Pro | Con |
| Less processing resource needed | Issues with assigning public IPs | CPS oriented Honeypot | One Conpot per IP, so more resource needed | Great for traffic with Malware | Has the SMB log issue that inflates the log file |

Table 1: Honeypot Tools Pro and Con