Reconnaissance Attacks

# Objective

A reconnaissance attack is when an attacker targets a victim’s system to gather information about its vulnerabilities. In this lab, we will look for SCADA devices in ICS. Recall that SCADA systems are in various types of applications, including electrical transmission, power plants, substations, HVAC systems, chemical plants, water treatment facilities, and gas pipelines. These systems have PLCs, which are susceptible to attacks. Keep in mind that SCADA systems are a concern for national defense agencies since compromised SCADA systems would jeopardize our capacity for successfully defending in times of war. Endangering our populace while simultaneously conducting physical attacks would be catastrophic for the nation.

In this assignment, you will use three tools to perform tasks: NMAP, SHODAN and S-MOD. This exercise has three parts to design and implement a Reconnaissance attack on a SCADA system.

1. Discover information about devices connected to the internet using SHODAN.
2. Perform a network scan (IP address & MAC) to find all the IP addresses and MAC addresses of your Virtual Network. Use NMAP to also find open ports in your virtual network.
3. Use S-MOD to identify registers, coils, and function codes available in a system.

# Finding SCADA systems using SHODAN (20 points)

SHODAN is a search engine that scans the world’s IP addresses, pulls the [banner](https://blog.shodan.io/what-is-a-banner/) - or metadata about the services on that IP - and indexes the banners. Searching here provides parameters specific to the IP addresses of a particular device type.

There are multiple communication protocols used in SCADA or ICS. Unlike IT Protocols, the industrial control industry uses multiple protocols that are often unique to the manufacturer of the device. Some popular protocols are shown below:

|  |  |
| --- | --- |
| **Protocol** | **Port** |
| Modbus | 502 |
| DNP3 | 20000 |
| Fieldbus | 1089 - 1091 |
| Ethernet/IP | 2222 |
| EtherCat | 34980 |
| Profinet | 34962 - 34964 |

## Getting Started with SHODAN

1. **!!DO NOT INTERACT WITH ANY DEVICE YOU FIND ON SHODAN!! USE INFORMATION PROVIDED BY SHODAN ONLY**
2. Navigate to Shodan and search for IP addresses that have port 502 open. To use shodan you will have to either create your own account, or login with a Google/Twitter/Windows Live account. When searching in Shodan, note that you must type a parameter that you intend to search for, followed by a colon, and then the value of that parameter.
3. You can proceed by typing into the Shodan search:  
   Port:502
4. Take a screenshot of your results and paste it into your lab report. (10 points)
5. Note how many results first appear from your search. Randomly pick a device and note its IP. Beneath it, we can clearly see the country in which the IP address is located; clicking on it will also reveal the admin interface of the device with port 502 open. Go ahead and search for the type of device. Note: if the device you pick does not respond to Modbus function code 43 (Device Identification) on SHODAN, pick a different device.
6. Record the IP address, the product, its manufacturer, which country the system is based in, what organization it is part of, what type of device it is, and what they are typically used for. (10 points)
7. **!!DO NOT INTERACT WITH ANY DEVICE YOU FIND ON SHODAN!! USE INFORMATION PROVIDED BY SHODAN ONLY**

# Reconnaissance of your traffic lights exercise

## Install a Kali Linux VM

1. Make sure you have Kali Linux installed on a VM. If you don’t have it, please click on the below and go to part 5: <https://sites.google.com/uah.edu/openplctipsandtricks/initial-vm-setup>
2. Run your traffic lights exercise from HW3 and start your Kali Linux VM. Make sure all VMs are on the same host only network in Virtualbox, otherwise Kali won't be able to communicate with your PLC and HMI.

## Find all IP addresses and MAC addresses on your Virtual Network

1. Kali Linux comes with some pre-installed tools such as NMAP. So, you **do not** need to install it.
2. Open the terminal and type the command: “**sudo** **nmap –sn [subnet/CIDR]**”.
3. *Using the information that you have found, answer the post-exercise questions.*

## Use NMAP to find open ports in your virtual network.

1. Open the terminal and type: **“sudo nmap -sT -p [range of ports] [subnet/CIDR]”.**

## Using S-MOD to read coils and registers (Initial Setup)

1. Open Kali terminal and install S-MOD, type: **“git clone https://github.com/theralfbrown/smod-1”**
2. Next, type: “**cd smod-1**”
3. Run smod-1: “**python2 smod.py**”
4. After you run, type **“show modules”** to get the list containing all the functions available using this tool.

## Use S-MOD to find supported function codes

1. Type: “**use modbus/scanner/getfunc**” and configure the module if it needs some additional settings before running (Check Reconnaissance slide deck for instructions).
2. *You will need this task completed in order to answer the post-exercise questions.*

## Use S-MOD to read coils and register values

1. To read register values, **use modbus/function/readHoldingRegister**
2. After you run, you need to know if the values you obtained are correct or not – register values should be in front of the registerVal list. To find out that**:**
3. Go to your *HMI* and open *SCADABR*.
4. Click on Data Sources.
5. Click on the Edit icon for the Traffic Light.
6. Under *Modbus read data,* choose *Holding Register* for Register range, change the number of registers to 8 and click on *Read data*.
7. Note what you see and answer the post-exercise questions.
8. To read coil values, **use modbus/function/readCoils**
9. After you run, you need to know if the values you obtained are correct or not – coil values should be in front of coilStatus list. To do this,
10. Go to your *HMI* and open *SCADABR*.
11. Click on Data Sources.
12. Under *Modbus read data,* choose *Coil Status* for Register range, change the number of registers to 8 and click on *Read data*.
13. *Note what you see and answer the post-exercise questions.*

# Post Exercise Report from Section 3 (30 points)

Submit your answers to the following questions.

## Using what you found in section 3.2, please fill out the table below: (5 points)

|  |  |
| --- | --- |
| IP ADDRESS | MAC ADDRESS |
|  |  |
|  |  |
|  |  |
|  |  |

## Using what you found in section 3.3, please fill out the table below: (5 points)

|  |  |  |  |
| --- | --- | --- | --- |
| IP | MAC | Port | Service |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## What are the function codes used in your system? (5 points)

## Is it possible to create an attack to change the values of registers or coils using function codes 5 through 16? Why or why not? Explain. (10 points) HINT: Remember that the values are in decimal, but they are stored in binary/hexadecimal.

## For Section 3.6, you read the registers and coils. For each of them, compare what you see on the HMI with what you see using S-Mod. Are they the same? Why or why not? Explain. (5 points) Hint: Consider how coils are read either in Little Endian or Big Endian. binary/hexadecimal.

# Reconnaissance of an unknown system (50 points)

A reconnaissance attack is when an attacker targets a victim’s system to gather information about its vulnerabilities. In this part of the lab, we will scan a network, find relevant IP addresses, and conduct reconnaissance for a given topology. Unlike the previous section, it is expected that you work independently and draw from lecture slides to appropriately attempt the prompts.

Download and import the SystemSecurityLabs.ova VM.

Link: <https://drive.google.com/file/d/111kvR5cuFI-in25_r4JcCvOfJeSuddkT/view?usp=sharing>

**Please note that this VM takes a minimum of 4GB RAM. For this exercise, you do not need to run your previous VMs, so feel free to close your PLC, HMI and Kali Linux VMs.**

Once the VM finishes booting, login with username ccre and password ccre

Then open terminal and run the provided bash script:

sudo ~/Desktop/ReconLab/systemstart.sh

The bash script included in this lab creates a network topology and several containers that run OpenPLC, ScadaBR, and other software. It will be your job to:

1. Identify the network(s) that the systems are running on. (Hint: use the command "ip a" to see all networks your VM is connected to. Ignore the "veth" adapters) (5 points)
2. Find all active hosts and their IP addresses on each network(s). (5 points)
3. Find all open ports for each device you can find at each network. (5 points)
4. Determine what nodes are PLCs, what nodes are HMIs, and what nodes - if any - are neither. (5 points)
5. Draw (or digitally create) a picture of the network topology that you determined. Clearly denote the IPs, different network interfaces, and presumed roles of active nodes. (15 points)
6. Once you’ve correctly identified the network(s) and their connected nodes, determine all nonzero holding registers on each PLC node using S-MOD, Pymodbus or any other tool you would like.   
   *Note*: S-MOD and Pymodbus are not installed by default. Follow instructions on section 3.4 to install S-MOD in your virtual machine (15 points)  
   *Hint*: you may find scripting to be very useful for this section.

*Provide evidence of how you arrived at each of your answers. You may take screenshots of what you see using the tools mentioned in the lecture slides, discuss how you logically came to your conclusion, and reason through the consequences of elaborate reconnaissance.*

If, for any reason, you need to restart the containers, you can run the cleanup.sh script in the same directory as the systemstart.sh script, before running systemstart.sh again.

Additionally, you may want to consult a cheat sheet for Nmap flags to aid you in your tasks:  
<https://www.stationx.net/nmap-cheat-sheet/>