|  |
| --- |
|  |
| Capstone Experience IST 894  Carl Laneave |
| Lab 3 Report |

Table of Contents

[1.0 Scanning ICS/SCADA Networks 3](#_Toc146488349)

[1.1 Lab Results – Scanning ICS/SCADA Networks 4](#_Toc146488350)

[2.0 Attacking the Infrastructure 9](#_Toc146488351)

[2.1 Lab Results – Attacking the Infrastructure 10](#_Toc146488352)

[3.0 Firewall Rules for SCADA 16](#_Toc146488353)

[3.1 Lab Results – Firewall rules for SCADA 17](#_Toc146488354)

[4.0 SNORT SCADA Rules 21](#_Toc146488355)

[4.1 Lab Results – SNORT SCADA Rules 22](#_Toc146488356)

[5.0 SCADA Honeypot 28](#_Toc146488357)

[5.1 Lab Results – SCADA Honeypot 29](#_Toc146488358)

[6.0 References: 34](#_Toc146488359)

[7.0 Activity Log 35](#_Toc146488360)

# Scanning ICS/SCADA Networks

During the execution of this lab, an evaluation was done to test the importance of protecting ICS and SCADA Networks. Through the usage of Modbus, we can emulate an active ICS environment and allow us to attempt different plc scans and attacks. We also use Nmap to scan the active environments to see open ports and services on Modbus. Including other actions are the usage of a honey pot to track the traffic and actions of potential attackers. Metasploit being the most used tool and application, the attempted reverse TCP shells are tracked using Metasploit payloads. Furthermore, the ability to see and scan ICS environments shows the importance of air gaping environments to prevent potential attackers from scanning or attacking ap possible vulnerability in our ICS networks.

# Lab Results – Scanning ICS/SCADA Networks

A computer screen shot of a computer program

Description automatically generated

Figure 1.0 - Start the Modbus diagnostic slave s9mulator using TCP, server will now listen to the network requests.

A computer screen shot of a program

Description automatically generated

Figure 1.1 – Change to student session and run plcscan python script targeting 502 Modbus device on target machine

A computer screen with white text and red text

Description automatically generated

Figure 1.2 – Execute Nmap targeting port 502 and Modbus. Results are captured of target.

A screen shot of a computer

Description automatically generated

Figure 1.3- Run Nmap script of Modbus discovery with same target and port.

A computer screen with white text

Description automatically generated

Figure 1.4 – Stop the SNMPs service on target machine.

A computer screen with white text

Description automatically generated

Figure 1.5 - Start conpot to create a honey pot of a realistic looking ICS environment.

A screen shot of a computer

Description automatically generated

Figure 1.6 – Rescan against the emulated ICS environment conpot/honeypot. Results look identical to actual scans on real ICS environments.

A screen shot of a computer

Description automatically generated

Figure 1.7 – Continued Nmap scanning on honeypot shows realistic ICS results.

A screen shot of a computer

Description automatically generated

Figure 1.8 - Continued Nmap scanning on honeypot shows realistic ICS results.

\

A computer screen with text and images

Description automatically generated

Figure 1.9 – Execute Metasploit against target machine running honeypot, results show it was successful while recording said attacks in the conpot session.

# 2.0 Attacking the Infrastructure

In this attack, the focus is on using phishing email attachments to show how potential attackers may use phishing emails and malicious files to access a system. Through malware, attackers can get access to systems and escalate privileges as well as create their own users with increased permissions and sudoers. When attacking infrastructure, it is critical to not only block potential threats through usage of firewalls and VPNs but also have a proper system in place to train individuals on phishing mail exercises. Most of attacks come from social engineering and phishing emails rather than exploits.

# 2.1 Lab Results – Attacking the Infrastructure

A screenshot of a computer

Description automatically generated

Figure 2.0 – Start Web Apache 2 web service.

A computer screen shot of a program

Description automatically generated

Figure 2.1 - Run Metasploit framework.

A computer screen shot of a computer program

Description automatically generated

Figure 2.2 – Build out a reverse TCP payload attack against own host (home).

A screenshot of a computer

Description automatically generated

Figure 2.3 – Start Postfix mail system and send a message containing a phishing link

A computer screen shot of a computer

Description automatically generated

Figure 2.4 – Open mutt to see message.

A screenshot of a computer

Description automatically generated

Figure 2.5 – Download said attached deb file from phishing email

A screenshot of a computer

Description automatically generated

Figure 2.6 – Exploit using a reverse TCP while running the executed digitemp deb file.

A screenshot of a computer

Description automatically generated

Figure 2.7 – Digitemp file runs and message is captured.

A screen shot of a computer

Description automatically generated

Figure 2.8 – Exploit works, and attacker is on as root.

A screen shot of a computer

Description automatically generated

Figure 2.9 – Create user hacker with password hacker.

A computer screen with white text and red text

Description automatically generated

Figure 2.10 – Change over to user hacker and attempt to see /etc/shadow file. Due to lack of sudo permissions, attempt fails.

A screenshot of a computer

Description automatically generated

Figure 2.11 – Use reverse TCP exploit to add hacker to sudo group, user hacker can now see contents of /etc/shadow.

# 3.0 Firewall Rules for SCADA

One of the primary ways to protect a network is to have properly configured firewalls, for instance on our network. This is especially true for systems that are critical to a business or business function. Through IPTable configurations, Linux based systems can have their own firewall rules in place to prevent nonwhite listed ips from accessing the system. Furthermore, outbound traffic can be regulated both by IP and port as well to prevent the sending of data if a system is compromised. When a system is not properly configured, it opens itself up to malicious scans and opens for attackers to use to attempt to gain access to the system.

# 3.1 Lab Results – Firewall rules for SCADA



Figure 3.0 – Answer to start lab, which was firewalls.

A screen shot of a computer

Description automatically generated

Figure 3.1 – Show current rules on the iptables for firewall rules.

A screenshot of a computer

Description automatically generated

Figure 3.2 – Set up logging for iptables.

A black screen with white text

Description automatically generated

Figure 3.3 – Set up input destination rule to be TCP and port 502.



Figure 3.4 – Set up output rule to allow TCP and port 502 to do outbound calls.

A screen shot of a computer

Description automatically generated

Figure 3.5 – Reset the rules on the iptables and load the new changes for the firewall.

A computer screen shot of a computer program

Description automatically generated

Figure 3.6- run Modbus slave similar on address 123.

A screenshot of a computer

Description automatically generated

Figure 3.7 – Execute a call to system with firewall rules in place, they can not connect and are restricted.

A computer screen with white text

Description automatically generated

Figure 3.8 – Reset the firewall rules and open the input to allow calls on port 502 from the student machine.

A screenshot of a computer screen

Description automatically generated

Figure 3.9 – System can now talk as the firewall rules allow it to send requests.



Figure 3.10 – Running the plcscan on student is ran and is successful.

# 4.0 SNORT SCADA Rules

There are several ways to monitor actions and traffic on a system. Snort for example, allows the creation of rules to trigger alerts when attempts on certain ports, ips, etc. trigger the rule and the attempted information is then captured. There are also other tools such as conpot (honeypot) to capture the actions of a potential attacker while emulating an actual environment as well as Tshark. Tshark captures the actions of the TCP requests and packets of the requestor. This allows security network engineers to track the requests and see the patterns or attempts by a potential threat and/or threat actor.

# 4.1 Lab Results – SNORT SCADA Rules

A computer screen shot of a program

Description automatically generated

Figure 4.0 – Start snort on the system.

A screenshot of a computer

Description automatically generated

Figure 4.1 – Send a TCP replay message using Modbus, packets are successfully captured.

A screen shot of a computer

Description automatically generated

Figure 4.2 – Snort captures the actions being done using TCP replay messages from Modbus.

A screenshot of a computer

Description automatically generated

Figure 4.3 - Execute [cat /etc/snort/rules/all‑quickdraw.rules | more +/MODBUS] to see rules with Modbus

A screenshot of a computer

Description automatically generated

Figure 4.4 – Run [sudo snort ‑A full ‑i lo ‑k none ‑c /etc/snort/snort.conf1] to start snort and block requests using new rules

A computer screen shot of a computer program

Description automatically generated

Figure 4.5 – Running Modbus server to listen on the network.

A screenshot of a computer

Description automatically generated

Figure 4.6 – running Tshark to listen on port 502

A computer screen shot of a computer screen

Description automatically generated

Figure 4.7 – Run Metasploit with tshark, Modbus and Snort all listening to port 502.

A computer screen with text and images

Description automatically generated

Figure 4.8 – Set host target as student and run Modbus and reverse TCP exploit.

A computer screen with white text

Description automatically generated

Figure 4.9 – Show the results of snort and the requests from Metasploit actions.

A screenshot of a computer

Description automatically generated

Figure 4.10 – Run [sudo tshark ‑r /tmp/cap ‑V ‑x | more +/Modbus] to see the captured results of tshark from the Metasploit request.

# 5.0 SCADA Honeypot

One of the best ways to better understand attackers is sees what attackers are doing during their reconnaissance and exploits. Security professionals are now using Honeypots to emulate active environments while providing no real access to their system. Honeypots emulate real environments while reading and tracking all the actions of a potential threat actor. By watching the commands and requests an attacker is attempting, valuable information can be gained in their attack attempts. Furthermore, they track the meta-information about the said attacker to use in identifying who is attempting said attack. Conpot is an excellent example of a tool used to emulate a honeypot on a server.

# 5.1 Lab Results – SCADA Honeypot

A screenshot of a computer

Description automatically generated

Figure 5.0 – Answer the question to start the lab, which is honeypot



Figure 5.1 – Stop the snmpd service.

A computer screen with white text

Description automatically generated

Figure 5.2 – Start conpot to be a honey pot and listener emulating an ICS network.

A screenshot of a computer

Description automatically generated

Figure 5.3 – Run Metasploit framework on attacker system.

A screen shot of a computer

Description automatically generated

Figure 5.4 – Start a reverse TCP shell payload attack against the honey pot. The session was not successful.

A computer screen shot of a black screen

Description automatically generated

Figure 5.5 – We can see the attackers attempt in our conpot session.

A computer screen shot of a computer program

Description automatically generated

Figure 5.6 = Attempt to run smod Python script which loads the modbus penetration testing framework.

A computer screen shot of a program

Description automatically generated

Figure 5.7 - Set the hosts to the Ips of the honey pot, we can see that the ICS honeypot network is picked up.

A computer screen with black and white text

Description automatically generated

Figure 5.8 – The actions attempted using the penetration test framework and exploit are all captured on the honeypot.

A screen shot of a computer

Description automatically generated

Figure 5.9 – Add a firewall rule to block the IP of the attacker on port 502.

A screen shot of a computer

Description automatically generated

Figure 5.10 – Attacker can no longer access and hit the honey pot and is blocked by the firewall.

# 6.0 References:

[1] Team, S. (2020, February 18). What’s the difference between OT, ICS, Scada and DCS? - insight from securicon. <https://www.securicon.com/whats-the-difference-between-ot-ics-scada-and-dcs/>

[2] Conpot. (n.d.). Conpot. <http://conpot.org/>

[3] Metasploit. (n.d.). Metasploit Documentation Penetration Testing Software, Pen Testing Security. https://docs.metasploit.com/

[4] The Latest Phishing Statistics (updated January 2023) | AAG IT Support. (n.d.). Aag-It.com. https://aag-it.com/the-latest-phishing-statistics/

‌

# 7.0 Activity Log

| **Member Name** | **Task Date** | **Task Details** |
| --- | --- | --- |
| Carl Laneave | 9/23/2023 | Created Template, executed all labs, took screenshots, and completed report |
|  |  |  |
|  |  |  |