Unit 2 Lab – Network Standards and

Compliance

**Required Materials**

Putty or other connection tool Lab Server

Root or sudo command access

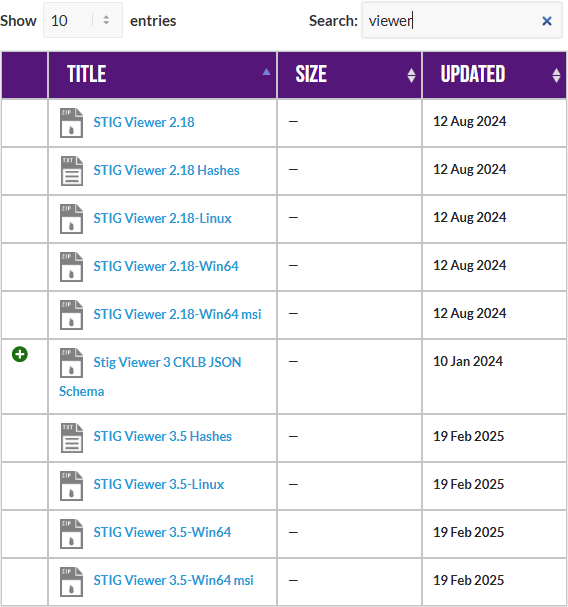
STIG Viewer 2.18 (download from <https://public.cyber.mil/stigs/downloads/> )

# EXERCISES (Warmup to quickly run through your system and familiarize yourself)

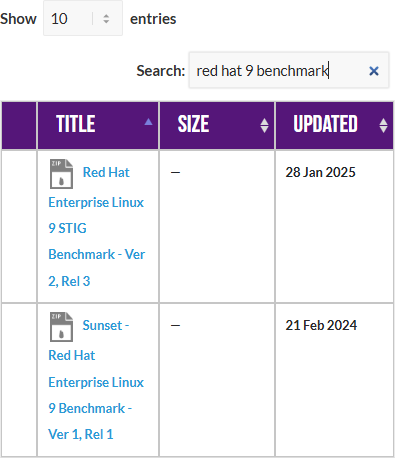
1. sysctl -a | grep -i ipv4 | grep -i forward
   1. Does this system appear to be set to forward? Why or why not?
2. sysctl -a | grep -i ipv4 | grep -i martian
   1. What are martians and is this system allowing them?
3. sysctl -a | grep -i panic
   1. How does this system handle panics?
4. sysctl -a | grep -i crypto
   1. What are the settings you see? Is FIPS enabled?
5. cat /proc/cmdline
6. fips-mode-setup --check
7. sestatus
8. cat /etc/selinux/config
   1. What information about the security posture of the system can you see here?
      1. Can you verify SELINUX status?
      2. Can you verify FIPS status?

# PreLAB

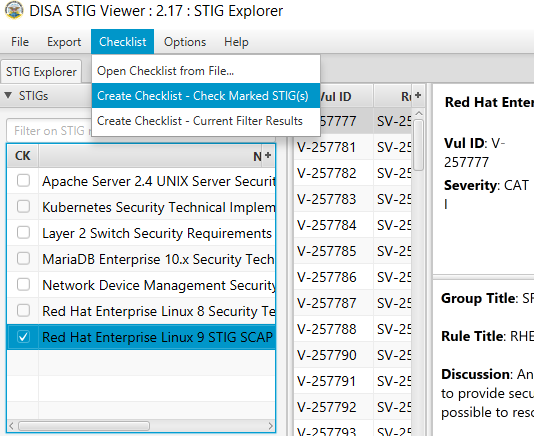
Download the STIG Viewer 2.18 from - <https://public.cyber.mil/stigs/downloads/>



Download the STIG for RHEL 9 and the import it into your STIG viewer



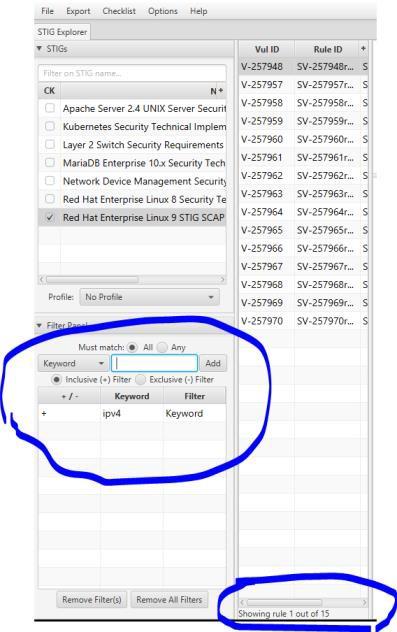
Create a checklist from the opened STIG for RHEL 9



# LAB

This lab is designed to have the engineer practice securing a Linux server or service against a set of configuration standards. These standards are sometimes called benchmarks, checklists, or guidelines. The engineer will be using STIG Viewer 2.18 to complete this lab.

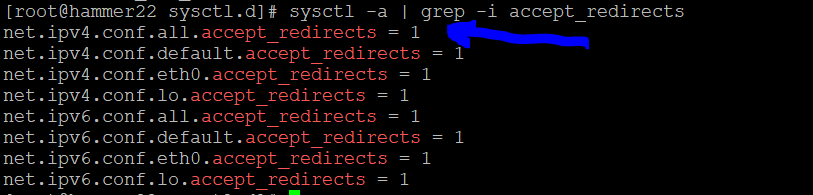
# Network Service configuration:

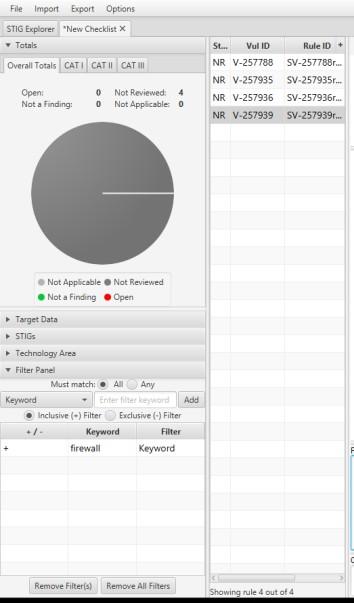
1. Connect to a hammer server
2. Filter by ipv4 and see how many STIGS you have.
3. Examine STIG V-257957
   1. What is the problem?
   2. What is the fix?
   3. What type of control is being implemented?
   4. Is it set properly on your system?
      1. sysctl -a | grep -i ipv4 | grep -i syncookiesA black background with white text  AI-generated content may be incorrect.
      2. Can you remediate this finding?

In this case it’s already correctly set.

But if we needed to, we would set that value in /etc/sysctl.d/00- remediate.conf

And then reload sysctl with `sysctl --system`

1. Check and remediate V-257958 STIG
   1. What is the problem?
   2. What is the fix?
   3. What type of control is being implemented?
   4. Is it set properly on your system?
   5. How would you go about remediating this on your system?
2. Check and remediate V-257960 and V-257961 STIGs
   1. What is the problem? How are they related?
   2. What is the fix?
   3. What type of control is being implemented?
   4. Is it set properly on your system?
3. Filter by firewall
   1. How many STIGS do you see?



* 1. What do these STIGS appear to be trying to do? What types of controls are they?

# Firewall port exposure

Your team needs to use node\_exporter with Prometheus to allow scraping of system information back to your network monitoring solution. You are running a firewall, so you need to expose the port that node\_exporter runs on to the network outside of your system.

1. Expose a network port through your firewall
   1. Verify that your firewall is running systemctl status firewalld
   2. Verify that your firewall has the service defined firewall-cmd –get-services | grep -i node

ls /usr/lib/firewalld/services | grep -i node

* 1. Verify that the service is not currently enabled for node\_exporter fireall-cmd –list-services
  2. Examine the structure of the firewall .xml file

cat /usr/lib/firewalld/services/prometheus-node-exporter.xml

* 1. Enable the service through your firewall

firewall-cmd --permanent --add-service=prometheus-node-exporter firewall-cmd –reload

* 1. Verify that the service is currently enabled for node\_exporter firewall-cmd --list-services

# Automate STIG remediation on a system

There are many options and the STIG remediation steps are well known. Here the learner will examine a few ways to generate Ansible and Shell fixes to your system. Then one can apply all of them, or just

some of them. This is the real value of a security engineer focused Linux engineer, the tradeoff between security and productivity.

1. Download and extract a STIG remediation tool

cd /root mkdir stigs cd stigs

wget -O U\_RHEL\_9\_V2R3\_STIG\_Ansible.zip https://dl.dod.cyber.mil/wp- content/uploads/stigs/zip/U\_RHEL\_9\_V2R3\_STIG\_Ansible.zip

unzip U\_RHEL\_9\_V2R3\_STIG\_Ansible.zip mkdir ansible

cp rhel9STIG-ansible.zip ansible/ cd ansible

unzip rhel9STIG-ansible.zip

1. Examine the default values for stigs

cd /root/stigs/ansible/roles/rhel9STIG/defaults/ vim main.yml

Search for a few of the STIG numbers you used earlier and see their default values. #use /257784 to search

1. Examine the playbook to see how those are applied in a running system.

vim /root/stigs/ansible/roles/rhel9STIG/tasks/main.yml

#use /257784 to search for the STIG from above and see how it is fixed in the playbook.

1. Create an Ansible playbook from openscap.

dnf -y install openscap-scanner openscap-utils openscap-scanner scap-security-guide

#Generate the Ansible

oscap xccdf generate fix --profile ospp --fix-type ansible

/usr/share/xml/scap/ssg/content/ssg-rhel9-ds.xml > draft-disa- remediate.yml

#Examine the file

vim draft-disa-remediate.yml

#Generate a BASH version

oscap xccdf generate fix --profile ospp --fix-type bash

/usr/share/xml/scap/ssg/content/ssg-rhel9-ds.xml > draft-disa- remediate.sh

#Examine the file

vim draf-disa-remediate.sh