ATTACK DEFENSE

by PentesterAcademy

Name	Inspec: Automating Compliance Checks
URL	https://attackdefense.com/challengedetails?cid=2071
Type	DevSecOps Basics: Compliance as Code

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Challenge Description

Chef Inspec is an open-source infrastructure testing framework. It can be used to define compliance, security and policy requirements using a script.

A Kali CLI machine (kali-cli) is provided to the user with inspec installed on it. The files for setting up a Tomcat server are provided in the home directory of the root user.

Objective: Audit the given web application using the Inspec utility!

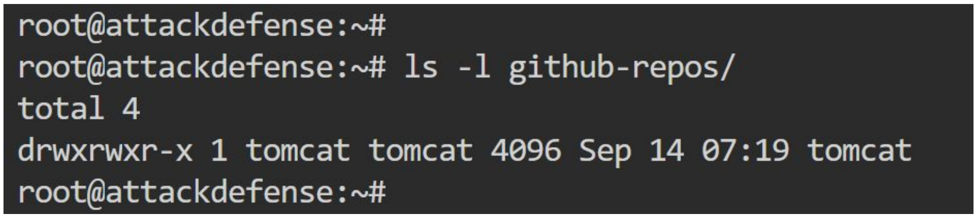
Instructions:

- The source code of web applications is provided at /root/github-repos

Solution

Step 1: Check the provided web application.

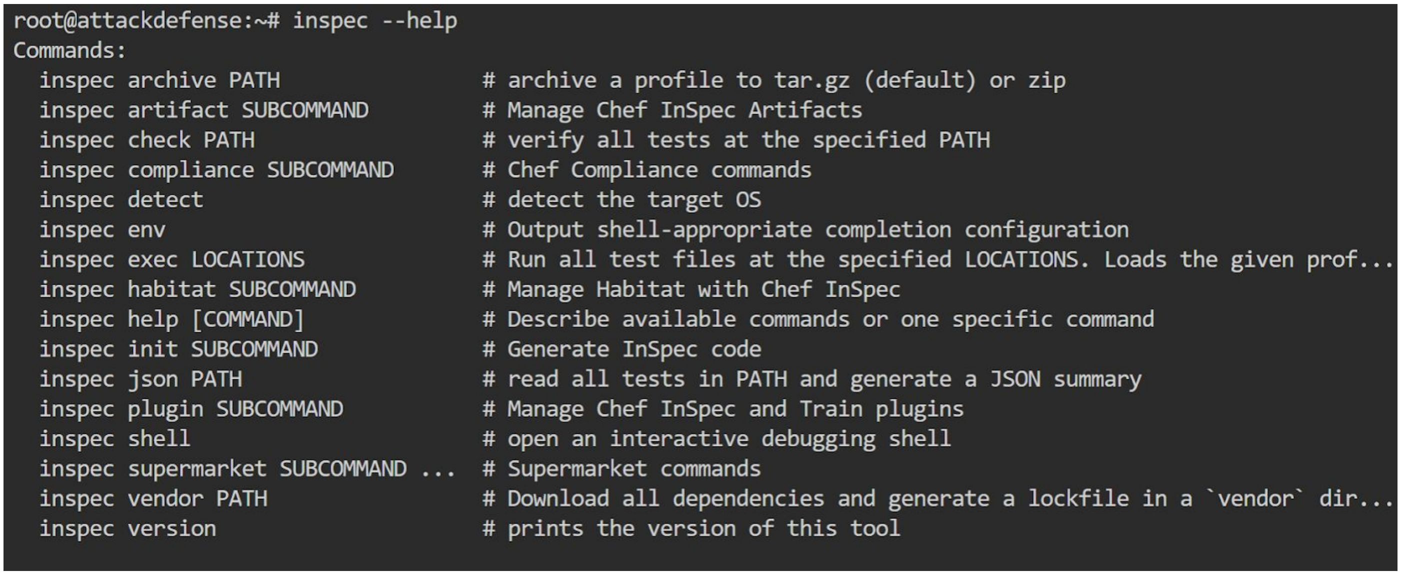
Command: ls -l github-repos



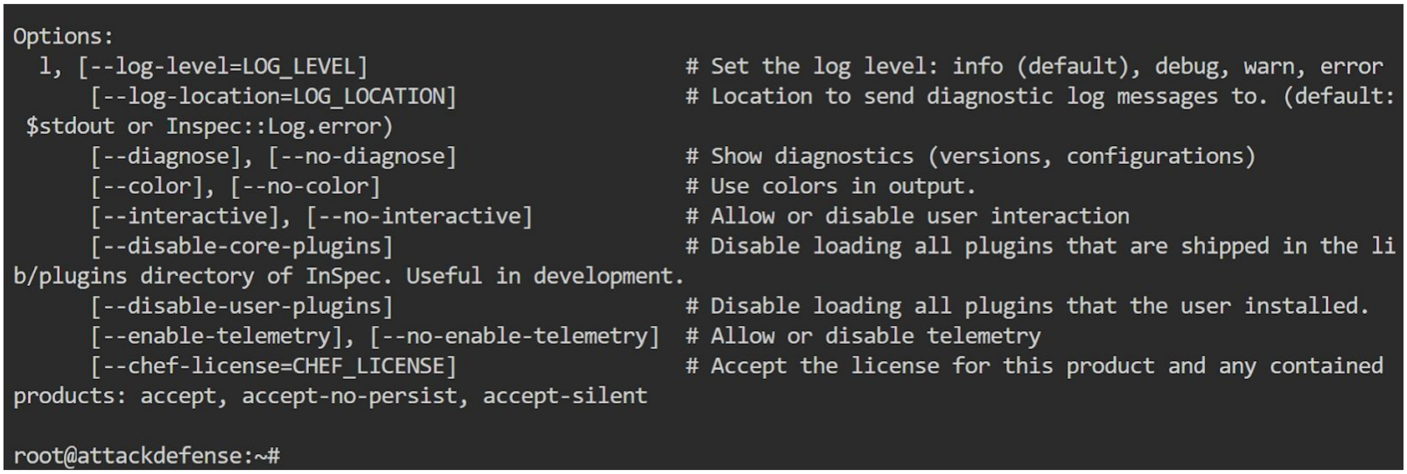
```
root@attackdefense:~#
root@attackdefense:~# ls -l github-repos/
total 4
drwxrwxr-x 1 tomcat tomcat 4096 Sep 14 07:19 tomcat
root@attackdefense:~#
```

Step 2: Check the available options of inspec tool

Command: inspec --help



```
root@attackdefense:~# inspec --help
Commands:
  inspec archive PATH                 # archive a profile to tar.gz (default) or zip
  inspec artifact SUBCOMMAND          # Manage Chef InSpec Artifacts
  inspec check PATH                   # verify all tests at the specified PATH
  inspec compliance SUBCOMMAND        # Chef Compliance commands
  inspec detect                       # detect the target OS
  inspec env                          # Output shell-appropriate completion configuration
  inspec exec LOCATIONS               # Run all test files at the specified LOCATIONS. Loads the given prof...
  inspec habitat SUBCOMMAND           # Manage Habitat with Chef InSpec
  inspec help [COMMAND]               # Describe available commands or one specific command
  inspec init SUBCOMMAND              # Generate InSpec code
  inspec json PATH                    # read all tests in PATH and generate a JSON summary
  inspec plugin SUBCOMMAND            # Manage Chef InSpec and Train plugins
  inspec shell                        # open an interactive debugging shell
  inspec supermarket SUBCOMMAND ...   # Supermarket commands
  inspec vendor PATH                  # Download all dependencies and generate a lockfile in a `vendor` dir...
  inspec version                      # prints the version of this tool
```

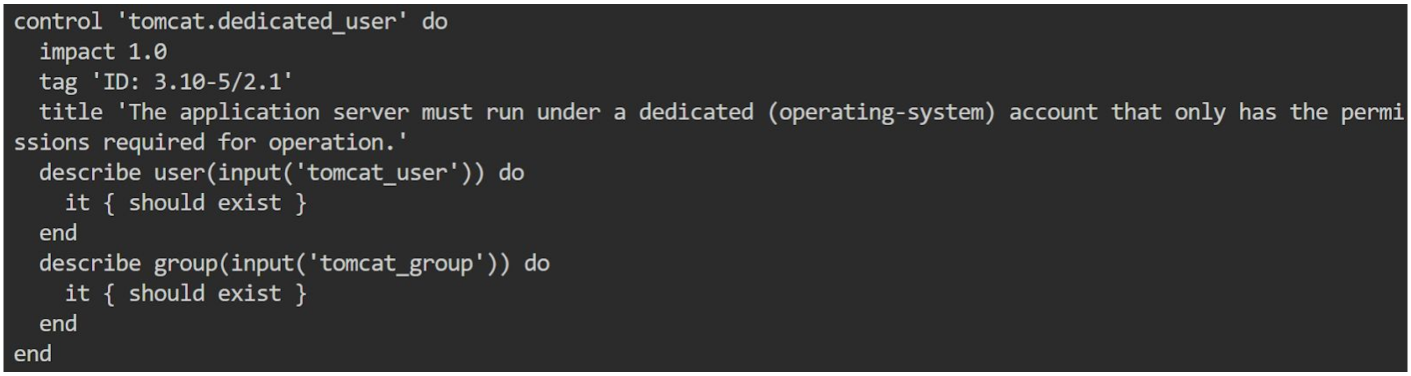


```
Options:
  l, [--log-level=LOG_LEVEL]                         # Set the log level: info (default), debug, warn, error
      [--log-location=LOG_LOCATION]                  # Location to send diagnostic log messages to. (default:
 $stdout or Inspec::Log.error)
      [--diagnose], [--no-diagnose]                  # Show diagnostics (versions, configurations)
      [--color], [--no-color]                        # Use colors in output.
      [--interactive], [--no-interactive]            # Allow or disable user interaction
      [--disable-core-plugins]                       # Disable loading all plugins that are shipped in the li
b/plugins directory of InSpec. Useful in development.
      [--disable-user-plugins]                       # Disable loading all plugins that the user installed.
      [--enable-telemetry], [--no-enable-telemetry]  # Allow or disable telemetry
      [--chef-license=CHEF_LICENSE]                  # Accept the license for this product and any contained
products: accept, accept-no-persist, accept-silent

root@attackdefense:~#
```

Step 3: The inspec tests for tomcat are defined in the “tomcat.rb” file. Check these tests.

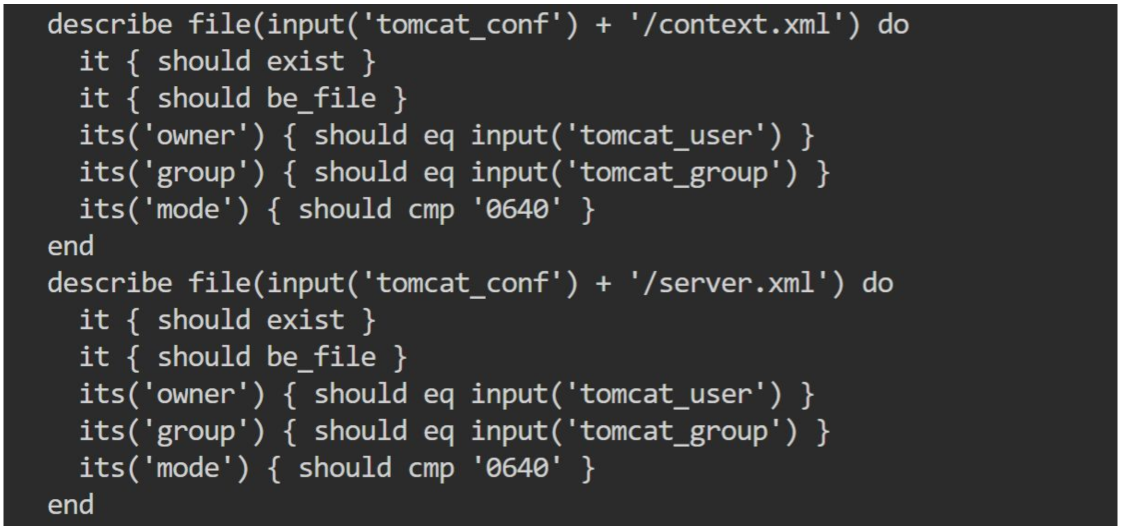
Commands: cat tomcat.rb

Test 1: Check if the ‘tomcat’ user and group exists or not?



```
control 'tomcat.dedicated_user' do
  impact 1.0
  tag 'ID: 3.10-5/2.1'
  title 'The application server must run under a dedicated (operating-system) account that only has the permi
ssions required for operation.'
  describe user(input('tomcat_user')) do
    it { should exist }
  end
  describe group(input('tomcat_group')) do
    it { should exist }
  end
end
```

Test 2: The file permission of context.xml and server.xml is set to 0640 or not?



```
describe file(input('tomcat_conf') + '/context.xml') do
  it { should exist }
  it { should be_file }
  its('owner') { should eq input('tomcat_user') }
  its('group') { should eq input('tomcat_group') }
  its('mode') { should cmp '0640' }
end
describe file(input('tomcat_conf') + '/server.xml') do
  it { should exist }
  it { should be_file }
  its('owner') { should eq input('tomcat_user') }
  its('group') { should eq input('tomcat_group') }
  its('mode') { should cmp '0640' }
end
```

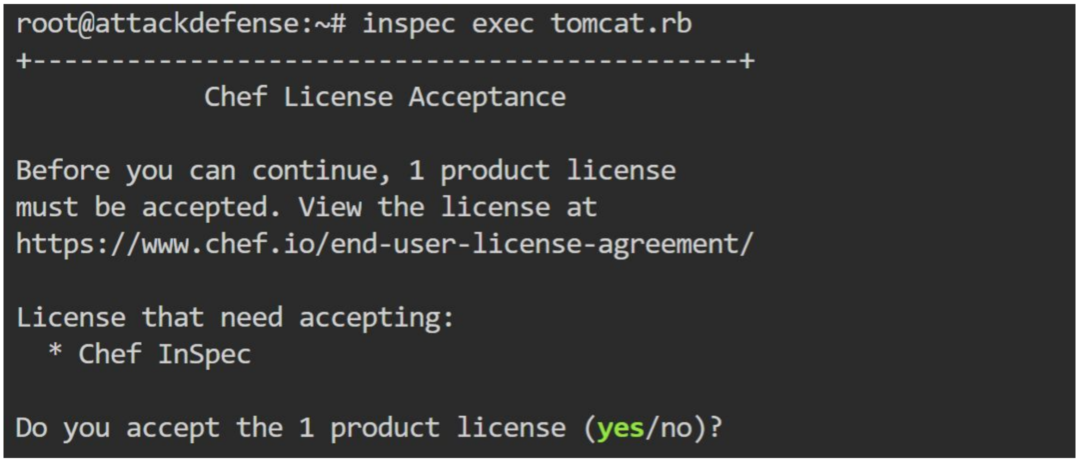
Note: There are more tests defined. But here only these two tests are covered.

Step 4: Run the inspec tool while passing tomcat.rb as an argument.

Command: inspec exec tomcat.rb

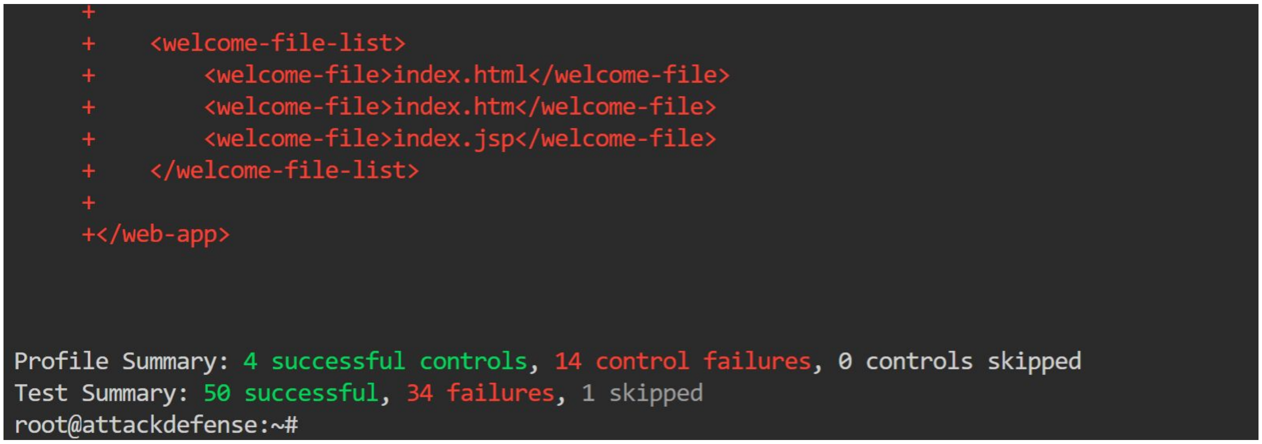
Command Explanation:

- exec - This command will run all the tests defined in the configuration file passed to it.



```
root@attackdefense:~# inspec exec tomcat.rb
+---------------------------------------------+
            Chef License Acceptance

Before you can continue, 1 product license
must be accepted. View the license at
https://www.chef.io/end-user-license-agreement/

License that need accepting:
  * Chef InSpec

Do you accept the 1 product license (yes/no)?
```

Type **yes** and press enter to continue the tool.

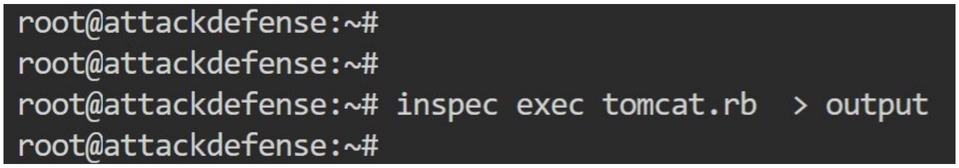


```
     +
     +    <welcome-file-list>
     +        <welcome-file>index.html</welcome-file>
     +        <welcome-file>index.htm</welcome-file>
     +        <welcome-file>index.jsp</welcome-file>
     +    </welcome-file-list>
     +
     +</web-app>



Profile Summary: 4 successful controls, 14 control failures, 0 controls skipped
Test Summary: 50 successful, 34 failures, 1 skipped
root@attackdefense:~#
```

The tests executed successfully and printed a lengthy report on the console. Save this report in a file.

Step 5: Run the tests again and save the report in a file.

Command: inspec exec tomcat.rb > output

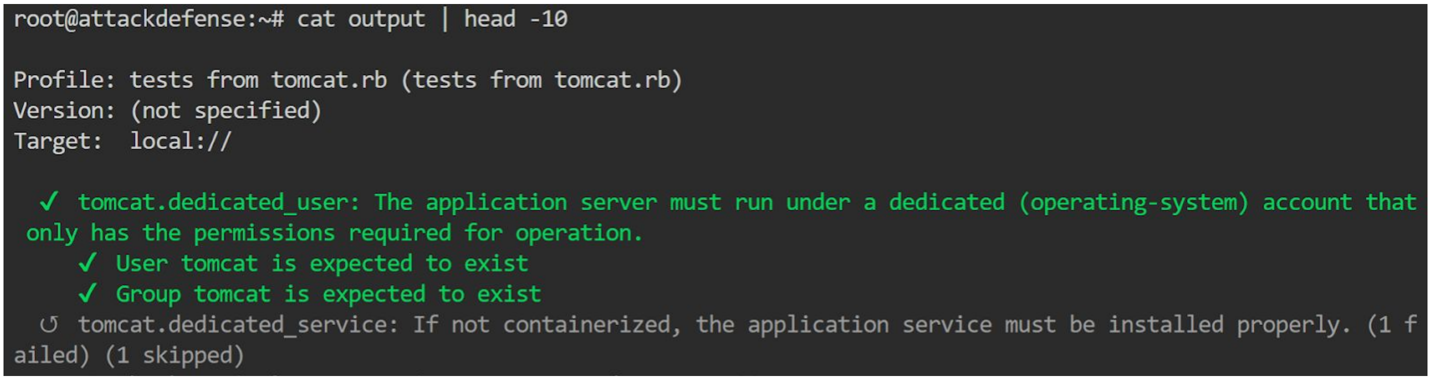


```
root@attackdefense:~#
root@attackdefense:~#
root@attackdefense:~# inspec exec tomcat.rb  > output
root@attackdefense:~#
```

Step 6: Check the content of the “output” file.

Command: cat output | head -10

Test 1: Check if the ‘tomcat’ user and group exists or not?



```
root@attackdefense:~# cat output | head -10

Profile: tests from tomcat.rb (tests from tomcat.rb)
Version: (not specified)
Target:  local://

  ✓  tomcat.dedicated_user: The application server must run under a dedicated (operating-system) account that
 only has the permissions required for operation.
     ✓  User tomcat is expected to exist
     ✓  Group tomcat is expected to exist
  ↺  tomcat.dedicated_service: If not containerized, the application service must be installed properly. (1 f
ailed) (1 skipped)
```

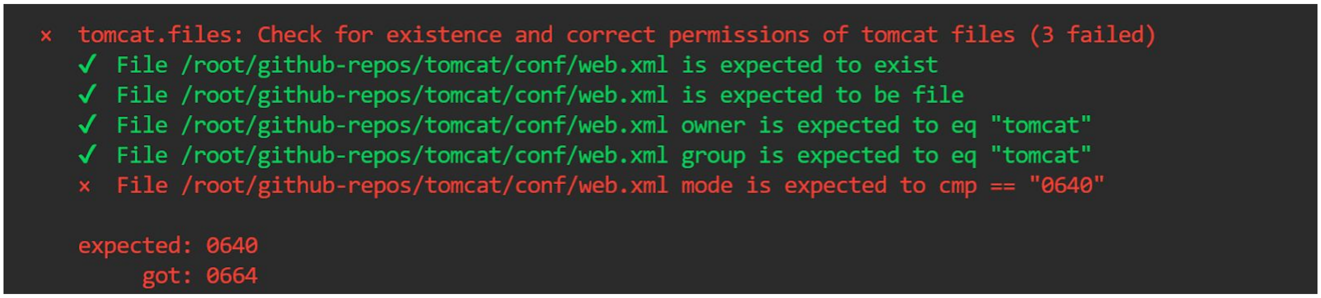
The tomcat user exists.

Result: Test passed

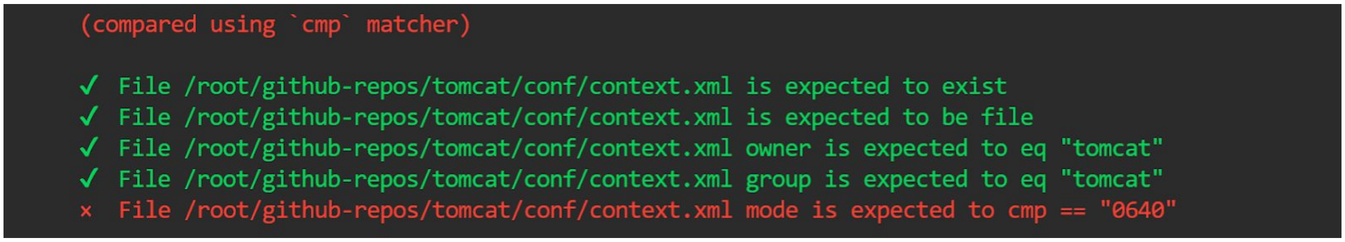
Test 2: The file permission of context.xml and server.xml is set to 0640 or not?

Command: cat output | less

Note: Scroll through the output to find the results.



```
  ×  tomcat.files: Check for existence and correct permissions of tomcat files (3 failed)
     ✓  File /root/github-repos/tomcat/conf/web.xml is expected to exist
     ✓  File /root/github-repos/tomcat/conf/web.xml is expected to be file
     ✓  File /root/github-repos/tomcat/conf/web.xml owner is expected to eq "tomcat"
     ✓  File /root/github-repos/tomcat/conf/web.xml group is expected to eq "tomcat"
     ×  File /root/github-repos/tomcat/conf/web.xml mode is expected to cmp == "0640"

     expected: 0640
          got: 0664
```



```
    (compared using `cmp` matcher)

  ✓  File /root/github-repos/tomcat/conf/context.xml is expected to exist
  ✓  File /root/github-repos/tomcat/conf/context.xml is expected to be file
  ✓  File /root/github-repos/tomcat/conf/context.xml owner is expected to eq "tomcat"
  ✓  File /root/github-repos/tomcat/conf/context.xml group is expected to eq "tomcat"
  ×  File /root/github-repos/tomcat/conf/context.xml mode is expected to cmp == "0640"
```

The file permissions are wrong for both context.xml web.xml files.

Issues Detected:

- web.xml file does not have 0640 permission but 0664.
- context.xml file does not have 0640 permission

Learnings

Perform audits on the infrastructure using inspec tool.