Unit 8 Lab – Configuration Drift and Remediation

Required Materials

Putty or other connection tool

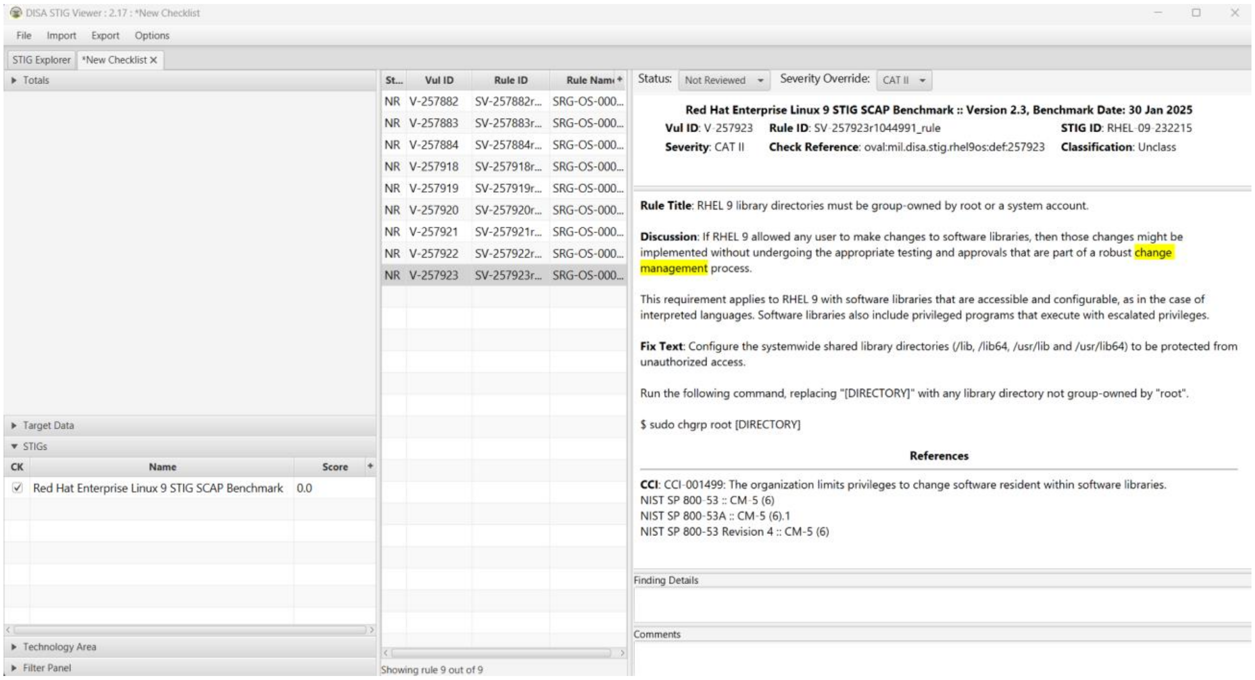
Lab Server

Root or sudo command access

LAB

These labs focus on configuration drift tracking and remediation.

Operational Activities



St...	Vul ID	Rule ID	Rule Name
NR	V-257882	SV-257882r...	SRG-OS-000...
NR	V-257883	SV-257883r...	SRG-OS-000...
NR	V-257884	SV-257884r...	SRG-OS-000...
NR	V-257918	SV-257918r...	SRG-OS-000...
NR	V-257919	SV-257919r...	SRG-OS-000...
NR	V-257920	SV-257920r...	SRG-OS-000...
NR	V-257921	SV-257921r...	SRG-OS-000...
NR	V-257922	SV-257922r...	SRG-OS-000...
NR	V-257923	SV-257923r...	SRG-OS-000...

Showing rule 9 out of 9

CK	Name	Score
☑	Red Hat Enterprise Linux 9 STIG SCAP Benchmark	0.0

Status: Not Reviewed Severity Override: CAT II

Red Hat Enterprise Linux 9 STIG SCAP Benchmark :: Version 2.3, Benchmark Date: 30 Jan 2025

Vul ID: V-257923 **Rule ID**: SV-257923r1044991_rule **STIG ID**: RHEL-09-232215
Severity: CAT II **Check Reference**: oval:mil.disa.stig.rhel9os:def:257923 **Classification**: Unclass

Rule Title: RHEL 9 library directories must be group-owned by root or a system account.

Discussion: If RHEL 9 allowed any user to make changes to software libraries, then those changes might be implemented without undergoing the appropriate testing and approvals that are part of a robust change management process.

This requirement applies to RHEL 9 with software libraries that are accessible and configurable, as in the case of interpreted languages. Software libraries also include privileged programs that execute with escalated privileges.

Fix Text: Configure the systemwide shared library directories (/lib, /lib64, /usr/lib and /usr/lib64) to be protected from unauthorized access.

Run the following command, replacing "[DIRECTORY]" with any library directory not group-owned by "root".

```
$ sudo chgrp root [DIRECTORY]
```

References

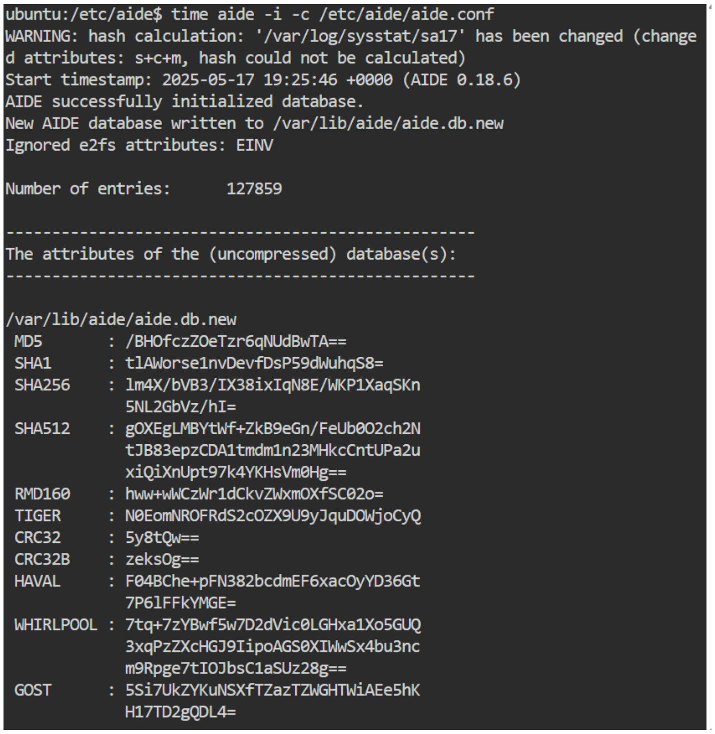
CCI: CCI-001499: The organization limits privileges to change software resident within software libraries.
NIST SP 800-53 :: CM-5 (6)
NIST SP 800-53A :: CM-5 (6).1
NIST SP 800-53 Revision 4 :: CM-5 (6)

1. Check your stig viewer and go to RHEL 9 stigs.
2. Set a filter for “change management”.
    a. How many STIGs do you see?
3. Review the wording, what is meant by a robust change management process?
    a. Do you think this can be applied in just one STIG? Why or why not?
    b. What type of control is being implemented with change management in these STIGS?

        i. Is it different across the STIGs or all the same?

Monitoring configuration drift with Aide

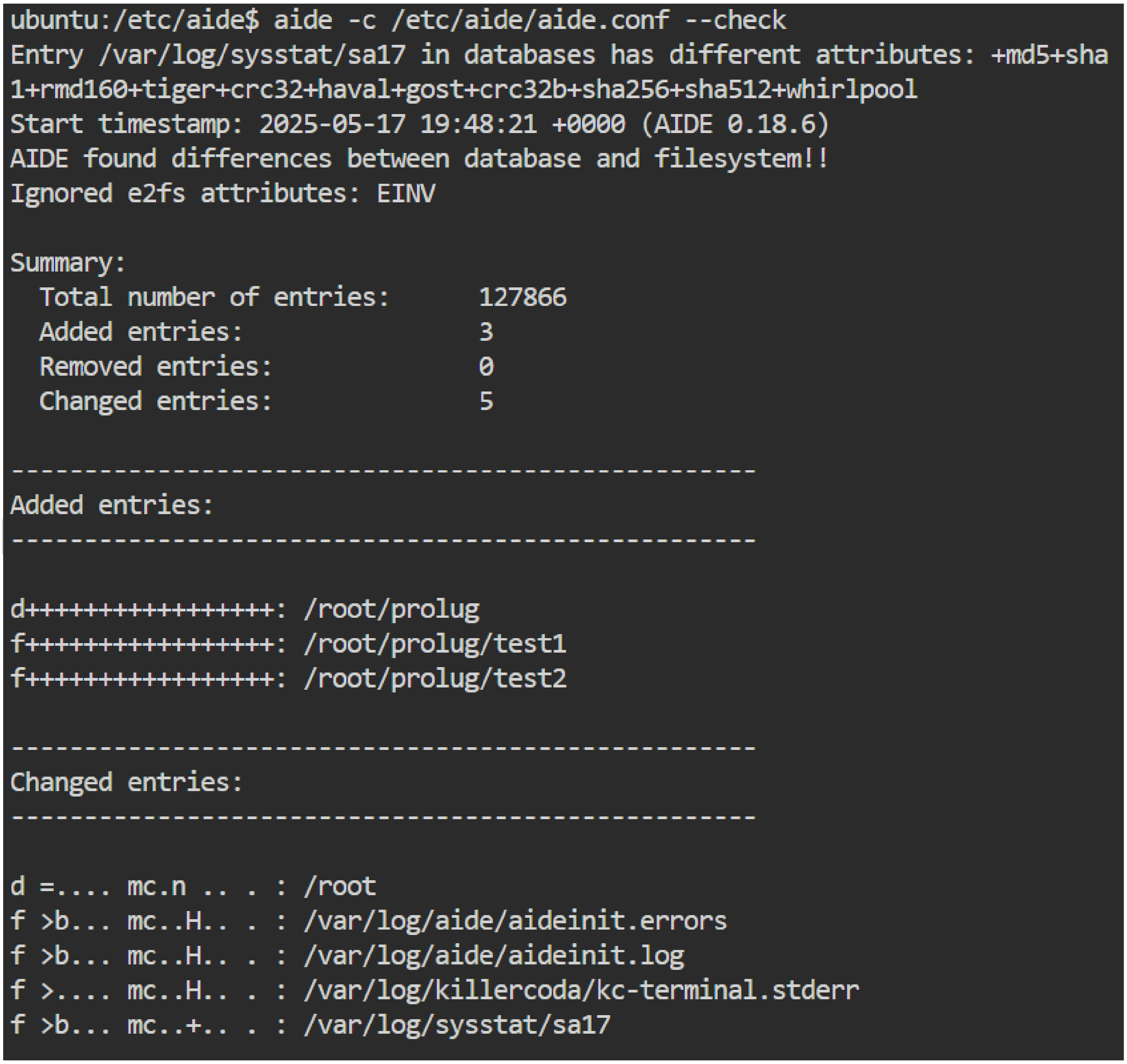
1. Go into the sandbox lab: https://killercoda.com/playgrounds/scenario/ubuntu
2. Install aide and watch the installation happen.
    a. `apt -y install aide`
    b. What is being put in the path /etc/aide/aide.conf.d/ ?
        i. How many files are in there?
3. Check your version of aide
    a. `aide -v`
4. Read the man page (first page).
    a. What does aide try to do, and how does it do it?
5. What is the configuration of cron found in /etc/cron.daily/dailyaidecheck?
    a. What does this attempt to do?
    b. What checks are there before execution?
    c. Read the man for capsh, what is it used for?
6. Set up aide according to the default configuration
    a. `time aide -i -c /etc/aide/aide.conf`
    b. How long did that take?
        i. How much time was wall clock v. system/user time?
        ii. Why might you want to know this on your systems?
        iii. What do you notice about the output?
            1. What do you need to go read about?



```
ubuntu:/etc/aide$ time aide -i -c /etc/aide/aide.conf
WARNING: hash calculation: '/var/log/sysstat/sa17' has been changed (change
d attributes: s+c+m, hash could not be calculated)
Start timestamp: 2025-05-17 19:25:46 +0000 (AIDE 0.18.6)
AIDE successfully initialized database.
New AIDE database written to /var/lib/aide/aide.db.new
Ignored e2fs attributes: EINV

Number of entries:      127859

---------------------------------------------------
The attributes of the (uncompressed) database(s):
---------------------------------------------------

/var/lib/aide/aide.db.new
 MD5       : /BHOfczZOeTzr6qNUdBwTA==
 SHA1      : tlAWorse1nvDevfDsP59dWuhqS8=
 SHA256    : lm4X/bVB3/IX38ixIqN8E/WKP1XaqSKn
             5NL2GbVz/hI=
 SHA512    : gOXEgLMBYtWf+ZkB9eGn/FeUb0O2ch2N
             tJB83epzCDA1tmdm1n23MHkcCntUPa2u
             xiQiXnUpt97k4YKHsVm0Hg==
 RMD160    : hww+wWCzWr1dCkvZWxmOXfSC02o=
 TIGER     : N0EomNROFRdS2cOZX9U9yJquDOWjoCyQ
 CRC32     : 5y8tQw==
 CRC32B    : zeksOg==
 HAVAL     : F04BChe+pFN382bcdmEF6xacOyYD36Gt
             7P6lFFkYMGE=
 WHIRLPOOL : 7tq+7zYBwf5w7D2dVic0LGHxa1Xo5GUQ
             3xqPzZXcHGJ9IipoAGS0XIWwSx4bu3nc
             m9Rpge7tIOJbsC1aSUz28g==
 GOST      : 5Si7UkZYKuNSXfTZazTZWGHTWiAEe5hK
             H17TD2gQDL4=
```

(Mine took 5 minutes 8 seconds to run on the lab system)

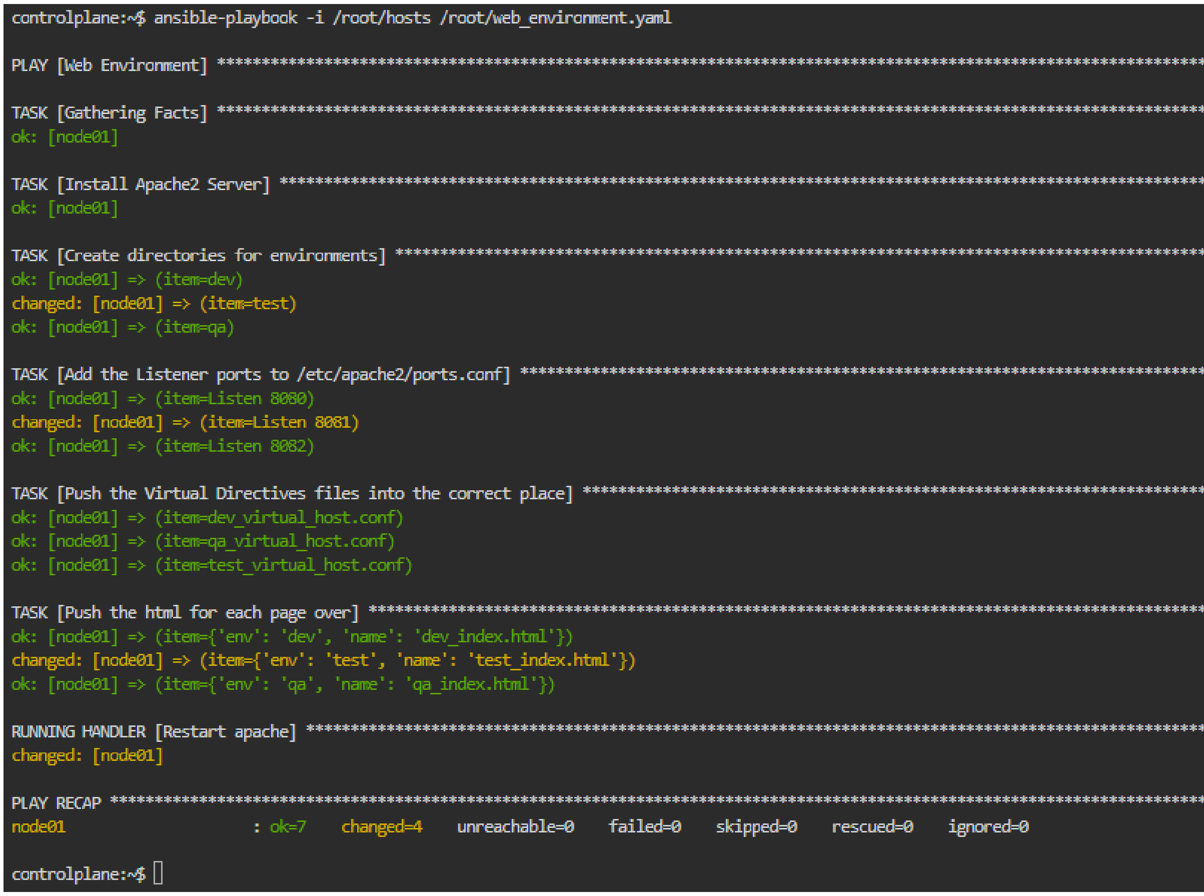
7. Set the database up properly
    a. `cp /var/lib/aide/aide.db.new /var/lib/aide/aide.db`
    b. `update-aide.conf`
8. Test aide by making files in a tracked directory
    a. `mkdir /root/prolug`
    b. `touch /root/prolug/test1`
    c. `touch /root/prolug/test1`
    d. `time aide -c /etc/aide/aide.conf –check`
        i. Did you see your new files created?
        ii. How long did this take to run?
            1. What type of usage do you see against user/system space?



```
ubuntu:/etc/aide$ aide -c /etc/aide/aide.conf --check
Entry /var/log/sysstat/sa17 in databases has different attributes: +md5+sha
1+rmd160+tiger+crc32+haval+gost+crc32b+sha256+sha512+whirlpool
Start timestamp: 2025-05-17 19:48:21 +0000 (AIDE 0.18.6)
AIDE found differences between database and filesystem!!
Ignored e2fs attributes: EINV

Summary:
  Total number of entries:      127866
  Added entries:                3
  Removed entries:              0
  Changed entries:              5

---------------------------------------------------
Added entries:
---------------------------------------------------

d+++++++++++++++++: /root/prolug
f+++++++++++++++++: /root/prolug/test1
f+++++++++++++++++: /root/prolug/test2

---------------------------------------------------
Changed entries:
---------------------------------------------------

d =.... mc.n .. . : /root
f >b... mc..H.. . : /var/log/aide/aideinit.errors
f >b... mc..H.. . : /var/log/aide/aideinit.log
f >.... mc..H.. . : /var/log/killercoda/kc-terminal.stderr
f >b... mc..+.. . : /var/log/sysstat/sa17
```

Using Ansible to fix drift

1. Complete the lab here: https://killercoda.com/het-tanis/course/Ansible-Labs/16-Ansible-Web-Server-Env-Deploy
2. When you finish ensure that you see broken output for 8081, as required.
   a. `curl node01:8080`
3. One of the dev teams figured out they could modify the test and qa environments because a previous engineer left them in the sudoers file. You can address that separately with the security team, but for now you need to get those environments back to working. Run your original deployment command to see if it sets the environment back properly.
   a. `ansible-playbook -i /root/hosts /root/web_environment.yaml`



```
controlplane:~$ ansible-playbook -i /root/hosts /root/web_environment.yaml

PLAY [Web Environment] ************************************************************************************************

TASK [Gathering Facts] ************************************************************************************************
ok: [node01]

TASK [Install Apache2 Server] *****************************************************************************************
ok: [node01]

TASK [Create directories for environments] ****************************************************************************
ok: [node01] => (item=dev)
changed: [node01] => (item=test)
ok: [node01] => (item=qa)

TASK [Add the Listener ports to /etc/apache2/ports.conf] **************************************************************
ok: [node01] => (item=Listen 8080)
changed: [node01] => (item=Listen 8081)
ok: [node01] => (item=Listen 8082)

TASK [Push the Virtual Directives files into the correct place] *******************************************************
ok: [node01] => (item=dev_virtual_host.conf)
ok: [node01] => (item=qa_virtual_host.conf)
ok: [node01] => (item=test_virtual_host.conf)

TASK [Push the html for each page over] *******************************************************************************
ok: [node01] => (item={'env': 'dev', 'name': 'dev_index.html'})
changed: [node01] => (item={'env': 'test', 'name': 'test_index.html'})
ok: [node01] => (item={'env': 'qa', 'name': 'qa_index.html'})

RUNNING HANDLER [Restart apache] **************************************************************************************
changed: [node01]

PLAY RECAP ************************************************************************************************************
node01                     : ok=7    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

controlplane:~$ 
```

   b. Did this force the system back into a working configuration?
      i. If it worked, would it always work, or could they the system need to be manually intervened?
      ii. What is your test? (hint: curl 8080 8081 and 8082 from previous commands)
   c. Could this cause potential problems in the environment?
      i. If so, is that problem based on technology or operational practices? Why?

Digging Deeper challenge (not required for finishing lab)

1. Complete this lab: https://killercoda.com/het-tanis/course/Ansible-Labs/19-Ansible-csv-report
    a. Can you think about how you'd use this to verify that a system was stamped according to your build process?
        i. You may have to tie it in with something like this lab and add some variables to your custom facts files, maybe the date of deployment: https://killercoda.com/het-tanis/course/Ansible-Labs/12-Ansible-System-Facts-Grouping