ATTACK DEFENSE

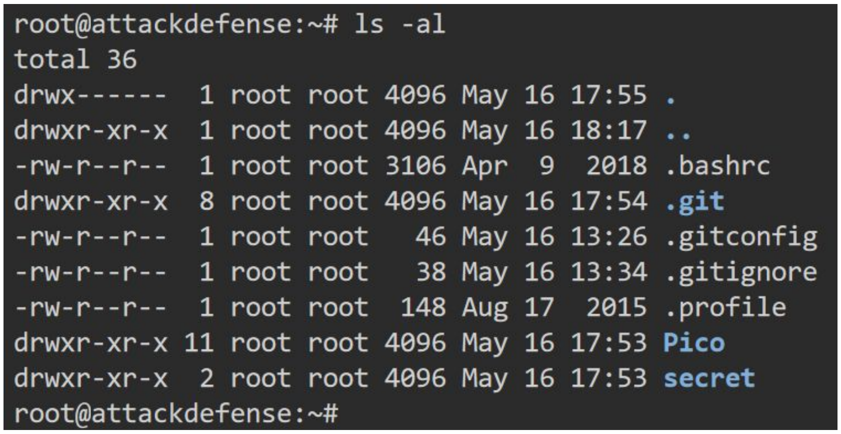
by PentesterAcademy

Name	Residue Metadata II
URL	https://www.attackdefense.com/challengedetails?cid=1041
Type	Code Repositories: Git

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.

Step 1: Viewing files in directory

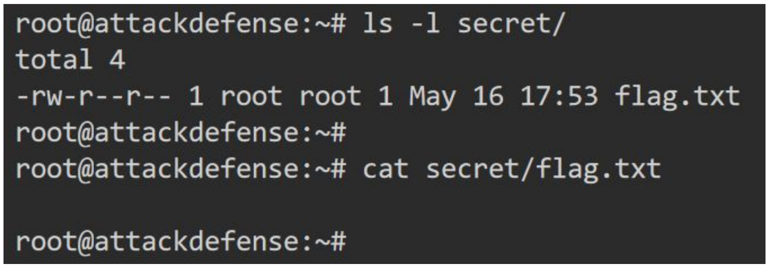
Command: ls -al



```
root@attackdefense:~# ls -al
total 36
drwx------  1 root root 4096 May 16 17:55 .
drwxr-xr-x  1 root root 4096 May 16 18:17 ..
-rw-r--r--  1 root root 3106 Apr  9  2018 .bashrc
drwxr-xr-x  8 root root 4096 May 16 17:54 .git
-rw-r--r--  1 root root   46 May 16 13:26 .gitconfig
-rw-r--r--  1 root root   38 May 16 13:34 .gitignore
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
drwxr-xr-x 11 root root 4096 May 16 17:53 Pico
drwxr-xr-x  2 root root 4096 May 16 17:53 secret
root@attackdefense:~#
```

Step 2: In the last step, we can observe a secret folder. Viewing files in secret folder.

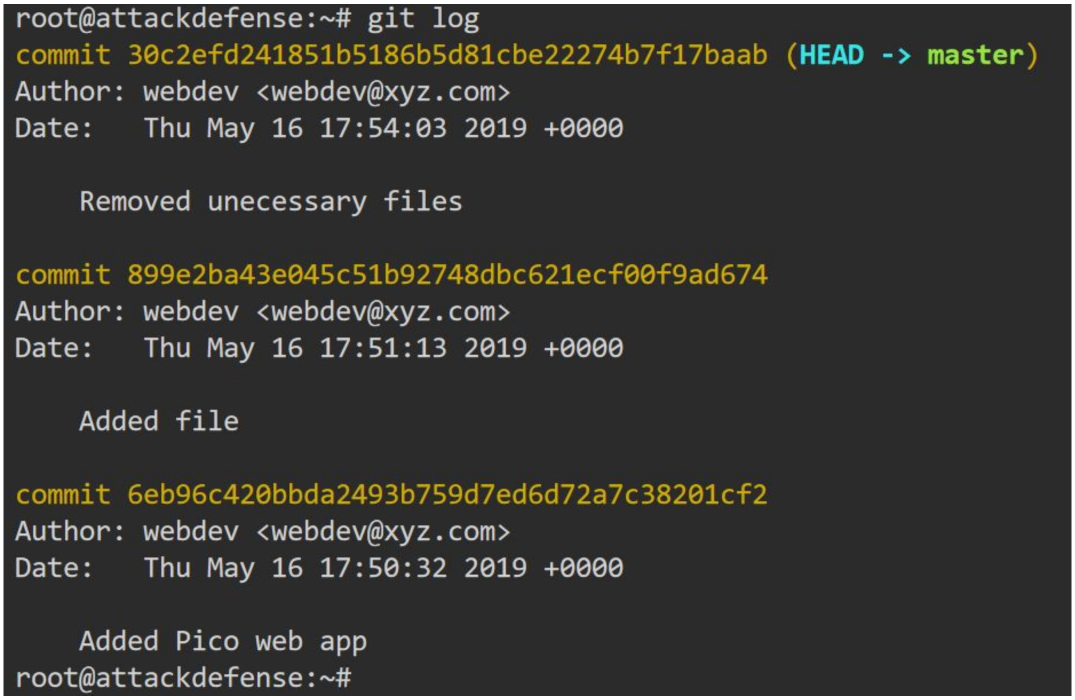
Command: ls -l secret/
cat secret/flag.txt



```
root@attackdefense:~# ls -l secret/
total 4
-rw-r--r-- 1 root root 1 May 16 17:53 flag.txt
root@attackdefense:~#
root@attackdefense:~# cat secret/flag.txt

root@attackdefense:~#
```

Step 3: The flag.txt file is empty. Viewing git log to find the modification in flag.txt

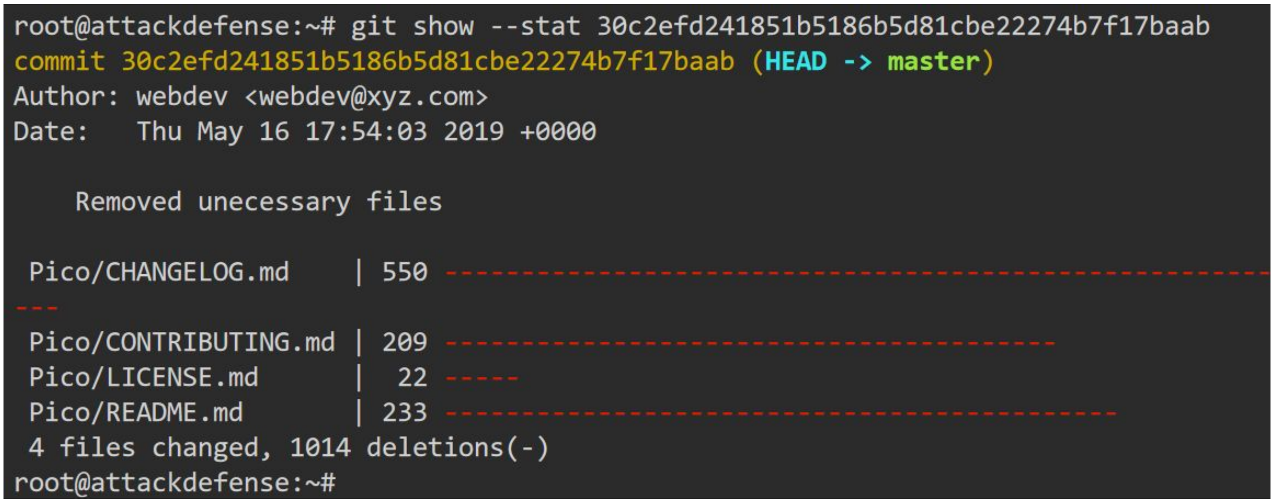
Command: git log



```
root@attackdefense:~# git log
commit 30c2efd241851b5186b5d81cbe22274b7f17baab (HEAD -> master)
Author: webdev <webdev@xyz.com>
Date:   Thu May 16 17:54:03 2019 +0000

    Removed unecessary files

commit 899e2ba43e045c51b92748dbc621ecf00f9ad674
Author: webdev <webdev@xyz.com>
Date:   Thu May 16 17:51:13 2019 +0000

    Added file

commit 6eb96c420bbda2493b759d7ed6d72a7c38201cf2
Author: webdev <webdev@xyz.com>
Date:   Thu May 16 17:50:32 2019 +0000

    Added Pico web app
root@attackdefense:~#
```

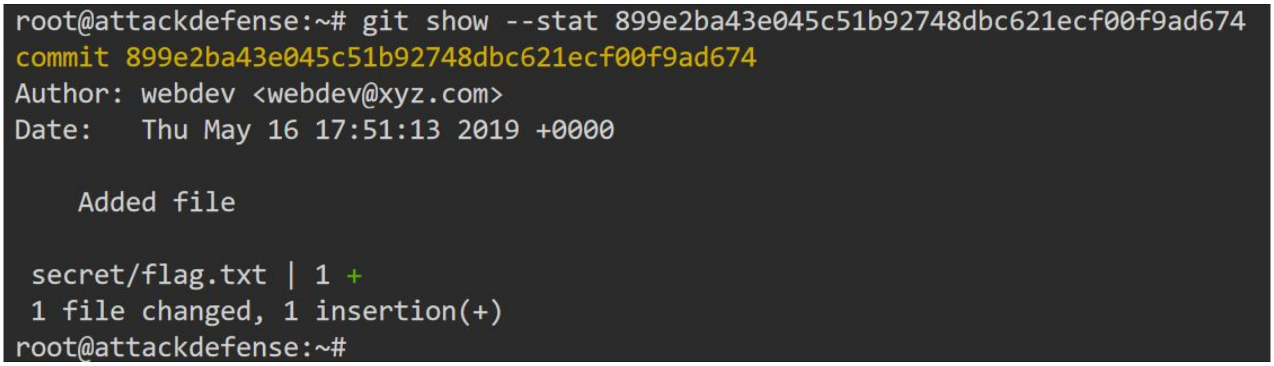
Step 4: Viewing modified files in each commit.

Command: git show --stat 30c2efd241851b5186b5d81cbe22274b7f17baab



```
root@attackdefense:~# git show --stat 30c2efd241851b5186b5d81cbe22274b7f17baab
commit 30c2efd241851b5186b5d81cbe22274b7f17baab (HEAD -> master)
Author: webdev <webdev@xyz.com>
Date:   Thu May 16 17:54:03 2019 +0000

    Removed unecessary files

 Pico/CHANGELOG.md    | 550 ------------------------------------------------------
---
 Pico/CONTRIBUTING.md | 209 ----------------------------------------
 Pico/LICENSE.md      |  22 -----
 Pico/README.md       | 233 --------------------------------------------
 4 files changed, 1014 deletions(-)
root@attackdefense:~#
```

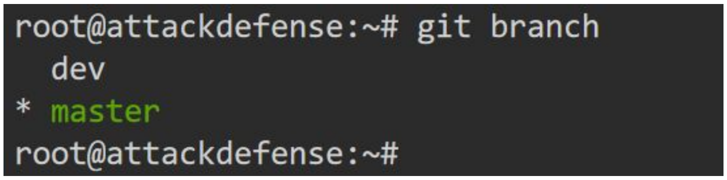
Command: git show --stat 899e2ba43e045c51b92748dbc621ecf00f9ad674



```
root@attackdefense:~# git show --stat 899e2ba43e045c51b92748dbc621ecf00f9ad674
commit 899e2ba43e045c51b92748dbc621ecf00f9ad674
Author: webdev <webdev@xyz.com>
Date:   Thu May 16 17:51:13 2019 +0000

    Added file

 secret/flag.txt | 1 +
 1 file changed, 1 insertion(+)
root@attackdefense:~#
```

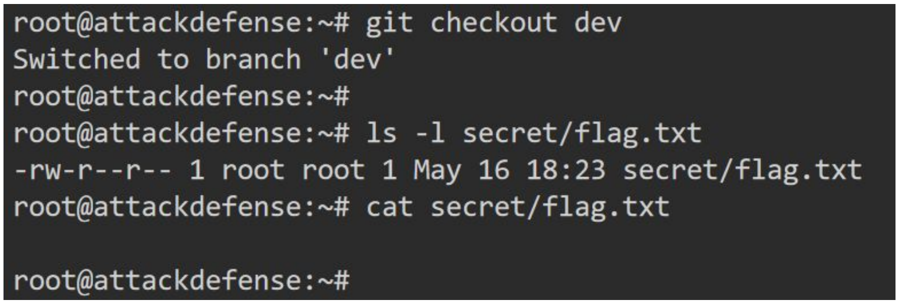
Step 5: Since flag.txt file was not modified in the latest commit. We can interpret that an empty flag.txt was created in the 2nd commit. Checking presence of other branches.

Command: git branch



```
root@attackdefense:~# git branch
  dev
* master
root@attackdefense:~#
```

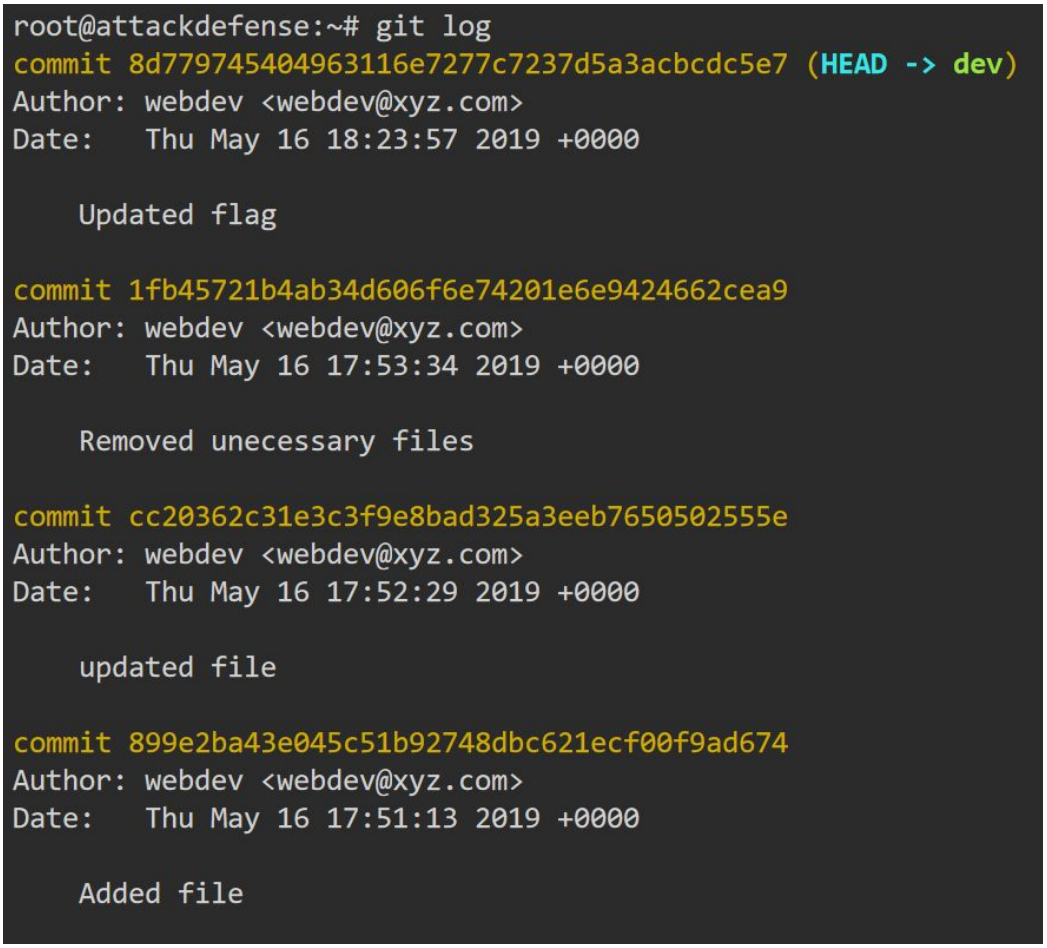
Step 6: Switching to “dev” branch and viewing flag.txt file



```
root@attackdefense:~# git checkout dev
Switched to branch 'dev'
root@attackdefense:~#
root@attackdefense:~# ls -l secret/flag.txt
-rw-r--r-- 1 root root 1 May 16 18:23 secret/flag.txt
root@attackdefense:~# cat secret/flag.txt

root@attackdefense:~#
```

Step 7: Viewing git log

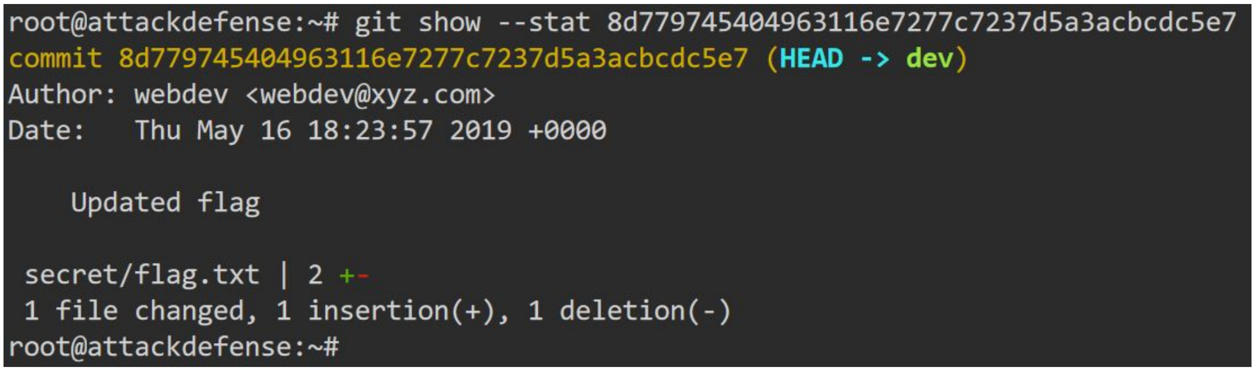
Command: git log



```
root@attackdefense:~# git log
commit 8d779745404963116e7277c7237d5a3acbcdc5e7 (HEAD -> dev)
Author: webdev <webdev@xyz.com>
Date:   Thu May 16 18:23:57 2019 +0000

    Updated flag

commit 1fb45721b4ab34d606f6e74201e6e9424662cea9
Author: webdev <webdev@xyz.com>
Date:   Thu May 16 17:53:34 2019 +0000

    Removed unecessary files

commit cc20362c31e3c3f9e8bad325a3eeb7650502555e
Author: webdev <webdev@xyz.com>
Date:   Thu May 16 17:52:29 2019 +0000

    updated file

commit 899e2ba43e045c51b92748dbc621ecf00f9ad674
Author: webdev <webdev@xyz.com>
Date:   Thu May 16 17:51:13 2019 +0000

    Added file
```

Step 8: Viewing modified files in each commit.

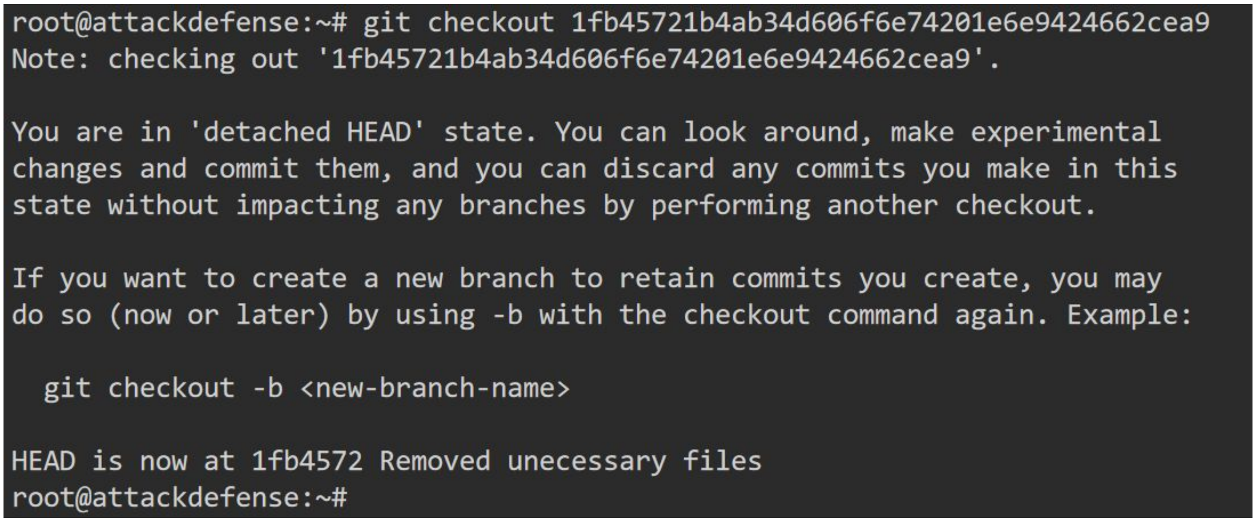
Command: git show --stat 8d779745404963116e7277c7237d5a3acbcdc5e7



```
root@attackdefense:~# git show --stat 8d779745404963116e7277c7237d5a3acbcdc5e7
commit 8d779745404963116e7277c7237d5a3acbcdc5e7 (HEAD -> dev)
Author: webdev <webdev@xyz.com>
Date:   Thu May 16 18:23:57 2019 +0000

    Updated flag

 secret/flag.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
root@attackdefense:~#
```

Step 9: The flag.txt file was modified in the latest commit. Reverting the git repository to previous commit.

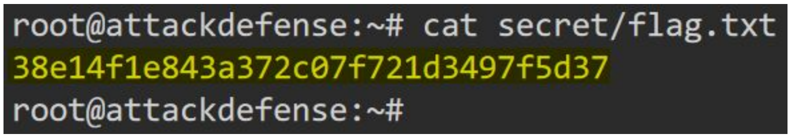
Command: git checkout 1fb45721b4ab34d606f6e74201e6e9424662cea9



```
root@attackdefense:~# git checkout 1fb45721b4ab34d606f6e74201e6e9424662cea9
Note: checking out '1fb45721b4ab34d606f6e74201e6e9424662cea9'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b <new-branch-name>

HEAD is now at 1fb4572 Removed unecessary files
root@attackdefense:~#
```

Step 10: Viewing flag.txt

Command: cat secret/flag.txt



```
root@attackdefense:~# cat secret/flag.txt
38e14f1e843a372c07f721d3497f5d37
root@attackdefense:~#
```

This reveals the flag to us.

Flag: 38e14f1e843a372c07f721d3497f5d37

References:

1. Git (https://git-scm.com/docs)