OS Patching for a UAT/Test Server.

Note: In this document you will see both theoretical knowledge and hands on also. So stay tuned.

Theory: The criticality of OS patching in IT security management cannot be overstated, especially now when the increase in BYOD(Bring your own device) has seen more devices - operating systems connecting to company networks, increasing your attack surface.

==========

Why is OS patching important?

Done well, OS patching can be the difference between a well-supported environment and one that is susceptible to unplanned downtime and performance issues. Here are some of the critical benefits of a robust approach to OS patching:

- **Compliance**: Many organizations now have regulatory requirements or insurance directives mandating a regular patching regime. Non-compliance can lead to severe penalties.
- **Availability**: The sad truth is that as an IT professional, you are only as good as your last issue. Keeping your systems' patches will prevent extended downtime due to security threats and remedial maintenance/emergency patch activity.

- **Performance**: Devices can crash due to software defects, so keeping your services patched means they are updated with the latest bug fixes and are more secure.
- **Security**: A common cause of network security breaches is missing patches in operating systems. Having a regular patch schedule means installing updates promptly, reducing the opportunity for data loss and damage to your infrastructure.
- **New features**: Patches are not always about protection from malware or fixing bugs. Sometimes patches can include new features that can give users greater functionality.

Dos and Don’ts of OS patching

- **Don't use unsupported** or EOL (end-of-life software)..
- **Don’t install patches from** ad content.
- Do **communicate patch windows** beforehand and agree to any potential downtime with the rest of the business.
- Do **scan your environment** post patching.
- Do **understand each vendor's release schedule** for patches and updates so that you can plan and schedule maintenance work accordingly.

Lab:

1) Step 1: login to your instance.

2) Step 2: Check the POA, Pre patch and Post patch**. POA (Plan of action document which is a team/collaborative effort.)** which has information like approval from all stake holders, Ask infra team to take a backup, take pre patch reports, take required config/ property files on local machines. Do it in mentioned downtime. **Pre patch report**: it is generated before we start the patching process and it has the list of all the patch versions which will be upgraded post activity. Post patch
3) Step 3 : Generate Pre patch report:
   [ec2-user@Uat-APP-del-01 ~]$ sudo su
   [root@Uat-APP-del-01 ec2-user]# yum list updates > prepatch.txt

```
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to re
gister.

Last metadata expiration check: 0:19:03 ago on Mon 26 Feb 2024 03:27:28 PM UTC.
Available Upgrades
gnutls.x86_64                           3.7.6-23.el9_3.3            rhel-9-baseos-rhui-rpms
nspr.x86_64                             4.35.0-6.el9_3              rhel-9-appstream-rhui-rpms
nss.x86_64                              3.90.0-6.el9_3              rhel-9-appstream-rhui-rpms
nss-softokn.x86_64                      3.90.0-6.el9_3              rhel-9-appstream-rhui-rpms
nss-softokn-freebl.x86_64               3.90.0-6.el9_3              rhel-9-appstream-rhui-rpms
nss-sysinit.x86_64                      3.90.0-6.el9_3              rhel-9-appstream-rhui-rpms
nss-util.x86_64                         3.90.0-6.el9_3              rhel-9-appstream-rhui-rpms
openssl.x86_64                          1:3.0.7-25.el9_3            rhel-9-baseos-rhui-rpms
openssl-libs.x86_64                     1:3.0.7-25.el9_3            rhel-9-baseos-rhui-rpms
selinux-policy.noarch                   38.1.23-1.el9_3.2           rhel-9-baseos-rhui-rpms
selinux-policy-targeted.noarch          38.1.23-1.el9_3.2           rhel-9-baseos-rhui-rpms
sudo.x86_64                             1.9.5p2-10.el9_3            rhel-9-baseos-rhui-rpms
tzdata.noarch                           2024a-1.el9                 rhel-9-baseos-rhui-rpms
[root@Uat-APP-del-01 ec2-user]#
```

4) Take a backup of previous/ current/ pre ptach kernel version copy, luckily our kernel version is latest but we have to upgrade these packages: **uname -r > oldkernelversion.txt**
5) Before we initiate patching check the mount point by: **[root@Uat-APP-del-01 ec2-user]# mount –a > mountpointinfo.txt** and take a backup of all the mount points pre activity: **[root@Uat-APP-del-01 ec2-user]# df -hTP > mountpointinfo.txt.:**

```
[root@Uat-APP-del-01 ec2-user]# df -hTP > mountpointinfo.txt
[root@Uat-APP-del-01 ec2-user]#  cat mountpointinfo.txt
Filesystem     Type      Size  Used Avail Use% Mounted on
devtmpfs       devtmpfs  4.0M     0  4.0M   0% /dev
tmpfs          tmpfs     1.8G     0  1.8G   0% /dev/shm
tmpfs          tmpfs     729M   17M  713M   3% /run
/dev/xvda4     xfs       9.2G  1.4G  7.8G  16% /
/dev/xvda3     xfs       536M  161M  376M  31% /boot
/dev/xvda2     vfat      200M  7.0M  193M   4% /boot/efi
tmpfs          tmpfs     365M     0  365M   0% /run/user/1000
[root@Uat-APP-del-01 ec2-user]#
[root@Uat-APP-del-01 ec2-user]#
```

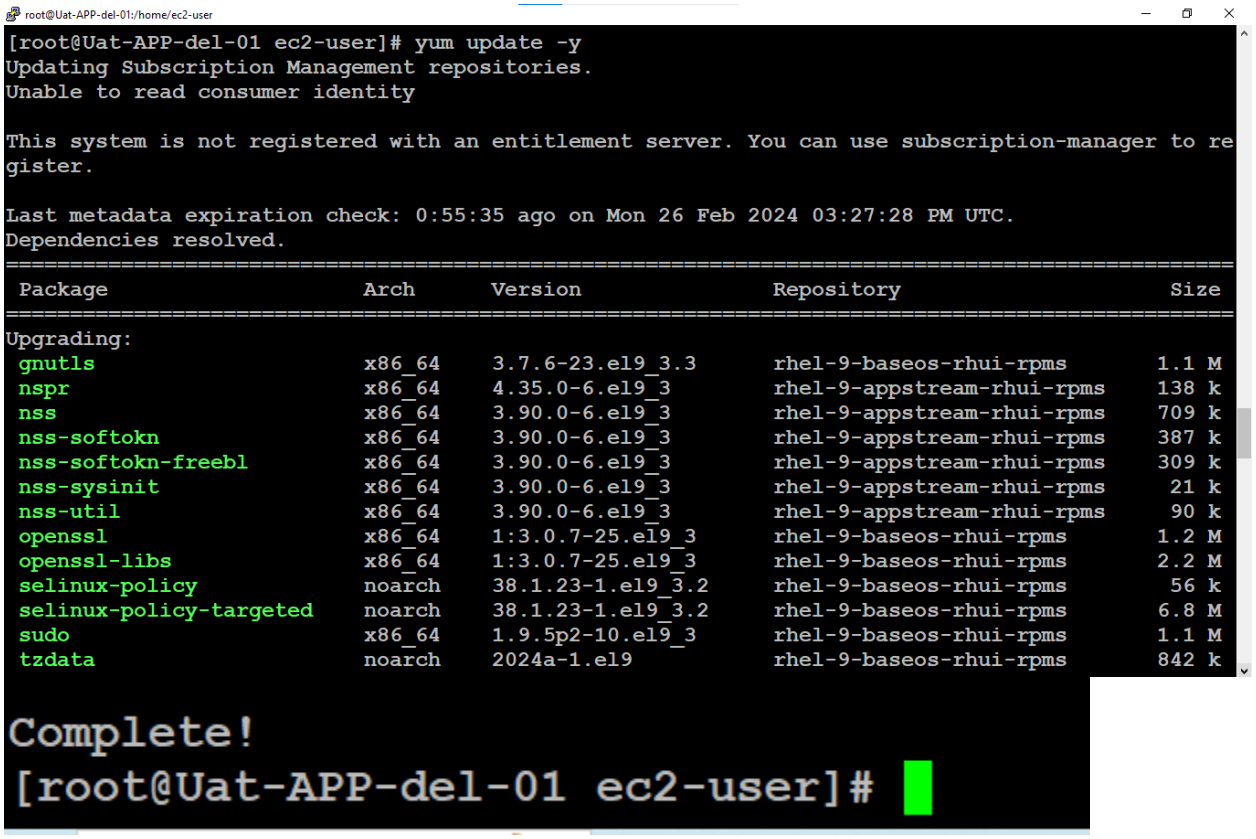
6) Take a backup of host entries: **[root@Uat-APP-del-01 ec2-user]# cat /etc/hosts > hostinfo_bkp.txt**

```
[root@Uat-APP-del-01 ec2-user]# cat  hostinfo_bkp.txt
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
[root@Uat-APP-del-01 ec2-user]#
```

7) Block storage information: **[root@Uat-APP-del-01 ec2-user]# lsblk > blkstrage-bkp.txt**

```
[root@Uat-APP-del-01 ec2-user]# lsblk > blkstrage-bkp.txt
[root@Uat-APP-del-01 ec2-user]# cat blkstrage-bkp.txt
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
xvda    202:0    0   10G  0 disk
├─xvda1 202:1    0    1M  0 part
├─xvda2 202:2    0  200M  0 part /boot/efi
├─xvda3 202:3    0  600M  0 part /boot
└─xvda4 202:4    0  9.2G  0 part /
[root@Uat-APP-del-01 ec2-user]#
```

8) Initiate manual update if you don't have automation ease which can be achieved by Ansible playbook or some crons configured earlier. BY: **[root@Uat-APP-del-01 ec2-user]# yum update –y**



9) Check module.dep file whether it is generated pre reboot or not to check whether there is kernel panic or not.

Note: The *modules.dep* as generated by **module-init-tools** **depmod**, lists the dependencies for every module in the directories under */lib/modules/version*, where *modules.dep* is. It is quite a big file.

[root@Uat-APP-del-01 ec2-user]# cat /lib/modules/5.14.0-362.18.1.el9_3.x86_64/modules.dep > module.dep_bkp.txt

10) Run: uname –r // check kernel version before the reboot in your case it might upgrade.

11) Check which kernel version will be upgraded post reboot: [**root@Uat-APP-del-01 ec2-user]# rpm -qa --last | grep kernel**

```
[root@Uat-APP-del-01 ec2-user]# rpm -qa --last | grep kernel
kernel-5.14.0-362.18.1.el9_3.x86_64           Wed 17 Jan 2024 06:50:50 PM UTC
kernel-modules-5.14.0-362.18.1.el9_3.x86_64   Wed 17 Jan 2024 06:50:29 PM UTC
kernel-modules-core-5.14.0-362.18.1.el9_3.x86_64 Wed 17 Jan 2024 06:50:28 PM UTC
kernel-core-5.14.0-362.18.1.el9_3.x86_64      Wed 17 Jan 2024 06:50:28 PM UTC
kernel-tools-5.14.0-362.18.1.el9_3.x86_64     Wed 17 Jan 2024 06:50:26 PM UTC
kernel-tools-libs-5.14.0-362.18.1.el9_3.x86_64 Wed 17 Jan 2024 06:50:23 PM UTC
```

12 Take a reboot and match all the files. With same but _new.txt

The End