Nick's Linux Server Checklist (WIP)

Type of System: Modern SystemD based Linux Server

A: Get Acquainted With System

Collect the following info in your notes to use for injects and later work. It will also help you to be prepared for following the rest of the guide.

1. How does the server connect to the internet?
    a. Run "`ip a s`" to see all ip address
    b. Run "`ping 8.8.8.8`" or "`ping6 2001:4860:4860::8888`" to ensure connection to the internet
    c. Run "`dig google.com`" to see the dns server you are currently using. This information will be in the "SERVER" field of the output.
    d. Check out the config file for networking. Is the networking static or through DHCP? Are there any virtual interfaces? See the table below
        i. `cat /etc/os-release` for OS info

Ubuntu 18.04	/etc/netplan/*.yml
Ubuntu <16.04 and Debian	/etc/network/interfaces
Centos <= 7	/etc/sysconfig/network-scripts

    e. Run "`ip neigh list dev <Main Interface Name>`" to check arp entries.
    f. Look at "/etc/hosts" for any interesting hard coded domain names.
2. What services is the server running?
    a. As root, run "`ss -tulpn`" to see what ports are listening and the processes that started them. Take note of whether or not the services are listening only on certain ip addresses.
    b. Run "`systemctl`" and briefly scroll through the list of services
    c. If the server is running any web services or listening on any common web ports (80,443,8080, etc) try visiting those services in a web browser (or asking a friend to do so - these need to be secured with new passwords, etc if possible)
3. What user accounts are on the server?
    a. Run "`less /etc/passwd`" to see the different user accounts.
    b. For any important accounts run as root "`groups <user>`" to see the user's groups.
    c. Run "`less /etc/group`" to see who is in the sudo/wheel group
4. What are the basic capabilities of the server?
    a. Run "`free -ht`" to see the memory available.
        i. Note this, because an ELK stack requires **at least** 3 GB of RAM.
    b. Run "`df -h`" to see storage devices.

c. Run "`cat /proc/cpuinfo`" to get an understanding of your processor.
d. Run "`cat /etc/lsb-release`" to get a look at your distro.

B: Change Passwords

1. Change the password of all user accounts that have a login shell.
    a. While logged in, run "`passwd`".
2. Create a separate local admin account for future work.
    a. Make sure it can sudo and consider giving it a separate login shell.
        i. `sudo adduser <new account name>`
        ii. `sudo usermod -aG <sudo group> <account name>`
    b. Once it can sudo, make sure no other accounts have sudo by checking /etc/group and using `usermod -G username username` to remove them from sudo.
    c. If needed, add allowed only required SSH users via `AllowUsers user` in /etc/ssh/sshd_config
3. All of the services that are used locally (such as mysql on a LAMP server) should have their passwords changed immediately. Don't forget to update dependent services and restart the services after any config changes.

C: Backup Important Files

You can archive the contents of a folder with "`tar cxvz <archive name> <folder name>`". You can unpack the same archive with "`tar xvf <archive name>`". If the archives aren't too large, you can scp them back to your workstation. Make sure to stop any services running when backing up data, especially when dealing with MySQL or similar databases.

1. Basic system config files, including /etc/sudoers.
2. Config files and data for the main services you identified earlier.
3. Logs and Bash histories. (You should do this again later).

D: Check for Compromise

1. Run "w" to ensure no one is on the server with you and then "last" to look for recent strange logins.
2. Check for any strange user accounts, especially if they were created recently.
    a. `less /etc/passwd` to list out user accounts
    b. `stat /home/<user>/.bash_logout` might give an idea as to when the user was created/last logged in.
    c. `last | grep <user>` will tell you the last time a user was logged in.
3. Refer back to listening ports. Are any of them potentially malicious?
4. Check crontab for all user accounts with "`crontab -e`"
5. Check for any strange connections (possible reverse shells) with ss. Try running "`ss -tulpn`" and similar.
6. Check for modified system files with your package manager (see Linux Forensics Bits)
7. Check your binaries with rkhunter

   a. Install rkhunter with system package manager or from source (try the rkhunter package on Debian and Arch)
   b. Run "`rkhunter --update`" to fetch latest data
   c. Run "`rkhunter --check`" to verify binaries (should take about 5 mins)
8. Check for module blacklists in /etc/modprobe.d/blacklist.conf
9. Audit docker group! Check for extra users (especially if docker was preinstalled)

E: Harden Security

1. If possible disable the root account and rely on sudo.
   a. `passwd -l root`
2. Remove sudo ability from any account which don't need to be admins.
   a. Check /etc/sudoers, usually sudo ability is granted by being a member of a group, either "wheel" or "sudo"
   b. Remove a group from a user with "`deluser <user> <group>`"
3. Lock any accounts which are unnecessary with "`password -l <username>`".
4. Consider and investigate setting up a firewall. See the firewall setup guide.
5. Consider locking important system files with "`sudo chattr -i <filename>`".

F: Install useful software

1. Nmap
2. Netcat
3. HTOP
4. A light web server
5. NFTables
6. Ftp client
7. Filebeat and Auditbeat
8. Lynis security auditor
9. OpenVAS binary
10. Rootkit hunter
11. Docker
12. Update everything with package manager

Update SSH Server
Check Wicked Quick SSH
Or All about SSH
Install Host Firewall
See Wicked Quick NFTables guide.