1. Boot Options

Prefixes = ['/boot/grub/', '/boot/efi/EFI/{distro}/','/boot/grub2/', '/etc/']

Configs = ['grub.cfg', 'menu.lst', 'grub2.cfg']

1. Boot options

Verify /boot/grub2/grub.cfg Permissions or /boot/grub/menut.lst

This test verifies that 'root' user owns /boot/grub2/menu.lst (SLES 15 /boot/grub2/grub.cfg) and permissions = 600 or more restrictive.

To help protect the GRUB configuration from unauthorized changes, only 'root' user should have read and write access to /boot/grub/menu.lst file.

Shell Script

if [ -f /etc/os-release ];

then ./etc/os-release;

OS=$NAME;

VER=$VERSION\_ID;

elif [ -f /etc/SuSE-release ];

then major\_ver=$(cat /etc/SuSE-release |grep ^VERSION |awk -F'=' '{print $2}');

minor\_ver=$(cat /etc/SuSE-release |grep ^PATCHLEVEL | awk -F'=' '{print $2}');

VER=$(echo "$major\_ver.$minor\_ver" |sed 's/ //g');

OS="sles";

elif [ -f /etc/redhat-release ];

then OS=$(cat /etc/redhat-release);

else OS="unknown";

VER="unknow";

fi;

echo "OS=$OS:VER=$VER";

if [[ [$(echo "12 < $VER"|bc) -eq 1](onenote:#$(echo%20%2212%20%3c%20$VER%22|bc&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={ED1FD993-0A86-4ED3-9876-327263AF2ED7}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]];

then ls -la /boot/grub2/grub.cfg 2>/dev/null;

elif [[ $(echo "11.4 >= $VER" |bc) -eq 1 ]];

then ls -la /boot/grub/menu.lst;

fi;

Output

-rw------- 1 root root 7331

Conditions

If an element version has no content, the condition should: Fail

Regexp

(-rw-------)\s\d\s(root)\s(root)\s|VER=11\.3|VER=11\.4|(-rw-------).\*\/boot\/grub2\/grub\.cfg

grouping 0 does not Exist

1. Verify NIS Server services
2. Verify NIS Server services ypserv is disabled
4. Shell script
5. if [ -f /etc/os-release ];
6. then ./etc/os-release;
7. OS=$NAME;
8. VER=$VERSION\_ID;
9. elif [ -f /etc/SUSE-release ];
10. then major\_ver=$(cat /etc/SuSE-release | grep ^VERSION | awk -F'=' '{print $2}');
11. VER=$(echo "$major\_ver.$minor\_ver" | sed 's/ //g');
12. OS="sles";
13. elif [ -f /etc/redhat-release ];
14. then OS=$(cat /etc/redhat-release);
15. else OS="unknown";
16. VER="unknown";
17. fi;
18. echo "OS=$OS:VER=$VER";
19. sclocate=$(whereis systemctl | awk '{print $2}' 2>/dev/null);
20. cflocate=$(whereis chkconfig | awk '{print $2}' 2>/dev/null);
21. srv\_systemctl=ypbind;
22. srv\_chkconfig=ypbind;
24. if [ $(echo $sclocate | grep systemctl$|wc -l ) -gt 0 ];
25. then echo "$sclocate $srv\_systemctl $(systemctl is-enabled $srv\_systemctl 2>&1)";
27. elif [ $(echo $cflocate|grep chkconfig$|wc -l) -gt 0 ];
28. then echo "$cflocate $(chkconfig --list $srv\_chkconfig 2>&1)";
30. else echo "OS Unknown: systemctl and chkconfig both not exist";
31. fi;
33. Conditions
34. If an element has no content, the condition should: Pass
35. Style: Plain Text (Advanced)
37. Regexp:
38. .\*[\r\n\s\S-]\*
40. Case sensitive
41. (Grouping 0: NIS Contains (String) 3:off AND
42. Grouping 0: NIS Contains (String) 5:off) OR
43. Grouping 0: NIS Contains (String) unknown service OR
44. Grouping 0: NIS Contains (String) disabled OR
45. Grouping 0: NIS Contains (String) /S15Cbin/systemctl ypserv Failed to get unit file state for ypserv.service OR
46. Grouping 0: NIS Contains (String) No such file

3.9 Security Features in Kernel

3.9.1 Enable TCP SYN Cookie Protection (default in SUSE11)

Verify TCP SYN Cookie Protection is Enabled

Description

A SYN Attack is a denial of service (DoS) attack that consumes resources on your system forcing you to reboot.

This particular attack is performed by beginning TCP connection handshake (sending SYN packet), and then never completing the process to open the connection.

This leaves your system with several hundreds or thousands of half-open connections.

This is a fairly simple attack and should be blocked.

Shell script

cat /etc/sysctl.conf | egrep '^net.ipv4.tcp\_syncookies'

Conditions

If an element version has no content, the condition should Fail

regexp:

^[\ \t]\*net.ipv4.tcp\_syncookies[\ \t]=[\ \t]\*(\d+)[\ \t]\*$

case sensitive: no

multiline: no

grouping 1: net.ipv4.tcp\_syncookies equals (numeric) 1 OR

grouping 0: net.ipv4.tcp\_syncookies does not exist (String) OR

==================================================

3.9.2 Verify that net.ipv4.conf.all.accept\_source\_route is equal to 0 (Configuration State)

Description

This test checks that the net.ipv4.conf.all.accept\_source\_route parameter is set to false.

Disabling this setting helps to prevent exploitation of IP trust relationships with spoofed source packets.

Shell script Kernel parameter

cat /etc/sysctl.conf | egrep '^net.ipv4.conf.all.accept\_source\_route' | awk -F'=' '{print $2}'

Conditions

If an element version has no content, the condition should: Fail

regexp:

\d+

case-sensitive: yes

multiline: yes

grouping 0: net.ipv4.conf.all.accept\_source\_route equals (numeric) 0

=================================================

3.9.4 Enable IP Spoofing Protection

3.9.4.2 Verify net.ipv4.conf.all.rp\_filter is equal to 2 (configuration state)

Description

This test checks net.ipv4.conf.all.rp\_filter parameter is set to true.

IP spoofing is a technique where an intruder sends out packet which claim to be from another host by manipulating the source address.

Shell script

cat /etc/sysctl.conf | grep net.ipv4.conf.all.rp\_filter

Conditions

If an element version has no content, the condition should: Fail

regexp:

^[\ \t]\*net\.ipv4\.conf\.all\.rp\_filter[\ \t]\*=[\ \t]\*(\d+)[\ \t]\*$

[\ \t] = space

\s = space

\*= 0 or up to infinite

+ = 1 or up to infinite

^[\ \t]\*net =

net

\d+

^[\ \t]\*net\.ipv4 =[\ \t]\*(\d+)[\ \t]\*$

case-sensitive: no

multiline: no

grouping 1: net.ipv4.conf.all.rp\_filter equals (numeric) 2 OR

grouping 0: net.ipv4.conf.all.rp\_filter does not exist (string)

================================================

3.9.6 Enable ignoring broadcasts request

3.9.6.2 Verify net.ipv4.icmp\_echo\_ignore\_broadcasts is equal to 1

Description

This test checks the net.ipv4.icmp\_ignore\_broadcasts parameter in /etc/sysctl.conf is set to true.

This allows the system to continue to respond to normal ping packets while preventing it from participating in creating floods of echo replies

Shell script

cat /etc/sysctl.conf

Conditions

If an element version has no content, the condition should: Fail

regexp:

^[\ \t]\*net\.ipv4\.icmp\_echo\_ignore\_broadcasts[\ \t]\*=[\ \t]\*(\d+)[\ \t]\*$

case-sensitive: no

multiline: no

grouping 1: net.ipv4.icmp\_echo\_ignore\_broadcasts equals (numeric) 1

==================================================

3.9.7 Enable bad error message protection (default in SuSE Linux 11)

3.9.7.1 Verify net-ipv4.icmp\_ignore\_bogus\_error\_responses is equal to 1 (running state)

Description

This test checks net.ipv4.icmp\_ignore\_bogus\_error\_responses parameter is set 1

This let the system alert about bad error messages in the network.

Shell script

cat /etc/sysctl.conf | egrep '^net.ipv4.icmp\_ignore\_bogus\_error\_response' | awk -F'=' '{print $2}'

Conditions

If an element version has no content, the condition should: Fail

regexp:

\d+

case-sensitive: no

multiline: no

grouping 0: net.ipv4.icmp\_ignore\_bogus\_error\_response equals (numeric) 1

3.12 FTP, Telnet and rlogin (rsh)

3.12.1 Verify tftp service is Disabled

Description

This test checks tftp service is disabled.

All system daemons do not have a clear and necessary purpose should be disabled.

This greatly reduces the odds that a vulnerable system daemon will be targeted by an attack when an OS vulnerability is discovered.

Shell script

serv1="tftp";

serv2="tftp";

if [ -f /etc/os-release ];

then ./etc/os-release;

OS=$NAME;

VER=$VERSION\_ID;

elif [ -f /etc/SuSE-release ];

then major\_ver=$(cat /etc/SuSE-release | grep ^VERSION | awk -F'=' '{print $2}';

VER=$(echo "$major\_ver.$minor\_ver" | sed 's/ //g');

OS="sles";

elif [ -f /etc/redhat-release ];

then OS=$(cat /etc/redhat-release);

else OS="unknown";

VER="unknown";

fi;

echo "OS=$OS:VER=$VER";

cflocate=$(which chkconfig 2>/dev/null);

sclocate=$(which systemctl 2>/dev/null);

if [ $(echo $sclocate | awk '{print length}' ) -gt 0 ];

then echo "$sclocate $serv1 $(systemctl is-enabled $serv1 2>&1)";

elif [ $(echo $cflocate | awk '{print length}' ) -gt 0 ];

then echo "$cflocate $(chkconfig --list $serv1 2>&1)";

else echo "OS Unknown: systemctl and chkconfig both not exist";

fi;

Conditions

If an element version has no content, the condition should: Pass

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: yes

grouping 0: tftp service disabled status Contains (String) disabled OR

grouping 0: tftp service disabled status Contains (String) masked OR

grouping 0: tftp service disabled status Contains (String) unknown service OR

grouping 0: tftp service disabled status Contains (String) /usr/bin/systemctl tftp failed to get unit file state for tftp.service

==================================================

3.12.2 Verify telnet service is disabled

Description

This test checks telnet service is disabled.

Because SSH provides a mechanism for both a secure login and a secure file transfer, it is not necessary for the insecure services are supervised by xinted to be running.

NOTE: It may be necessary for an organization to have 1 or more xinetd service running.

If this is the case, only enable the services are absolutely necessary.

Shell script

if [ -f /etc/os-release ];

then source /etc/os-release;

OS=$NAME;

VER=$VERSION\_ID;

elif [ -f /etc/SuSE-release ];

then major\_ver=$(cat /etc/SuSE-release | grep ^VERSION | awk -F'=' '{print $2}');

minor\_ver=$(cat /etc/SuSE-release | grep ^PATCHLEVEL | awk -F'=' '{print $2}');

VER=$(echo "$major\_ver.$minor\_ver" | sed 's/ //g');

OS="sles";

elif [ -f /etc/redhat-release ];

then OS=$(cat /etc/redhat-release);

else OS="unknown";

VER="unknown";

fi;

echo "OS=$OS:VER=$VER";

srv1="telent";

srv1="telnet";

chkconfig --list $srv1

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: yes

grouping 0: telnet Service Status Contains (String) unknown service OR

grouping 0: telnet Service Status Contains (String) /usr/bin/systemctl telnet failed to get unit file state for telnet.service OR

grouping 0: telnet Service Status Contains (String) telnet: off

==================================================

3.12.3 Verify rsh Service is disabled

Description

This test checks rsh service is disabled.

All system daemons that do not have a clear and necessary purpose should be disabled.

This greatly reduces the odds that a vulnerable system daemon will be targeted by an attack when an operating system vulnerability is discovered.

Shell script

chkconfig --list rsh 2>/dev/null

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: yes

grouping 0: rsh Service Status Contains (String) xinetd based service is off

==================================================

3.12.5 Verify talk server is disabled

Shell script

systemctl is-enabled talk 2> /dev/null

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: yes

grouping 0: talk server Contains (String) disabled

==================================================

3.12.6 Verify talk client is not installed

Shell script

ls -la /usr/bin/talk 2>/dev/null

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: no

grouping 0: talk client Contains (String) No such file or directory

====================================================

3.12.7 Verify dhcp service is disabled

Shell script

if [ -f /etc/os-release ];

then ./etc/os-release;

OS=$NAME;

VER=$VERSION\_ID;

elif [ -f /etc/SuSE-release ];

then major\_ver=$(cat /etc/SuSE-release | grep ^VERSION | awk -F'=' '{print $2}');

VER=$(echo "$major\_ver.$minor\_ver" | sed 's/ //g');

OS="sles";

elif [ -f /etc/redhat-release ];then OS=$(cat /etc/redhat-release);

else OS="unknown";

VER="unknown";

fi;

echo "OS=$OS:VER=$VER";

sclocate=$(whereis systemctl| awk '{print $2}' 2>/dev/null);

cflocate=$(whereis chkconfig | awk '{print $2}' 2>/dev/null);

srv\_systemctl=dhcpd;

srv\_chkconfig=dhcpd;

if [ $(echo $sclocate | grep systemctl$ | wc -l ) -gt 0];

then echo "$sclocate $srv\_systemctl $(systemctl is-enabled $srv\_systemctl 2>&1)";

elif [ $(echo $cflocate | grep chkconfig$ | wc -l) -gt 0 ];

then echo "$cflocate $(chkconfig --list $srv\_chkconfig 2>&1)";

else echo "OS Unknown: systemctl and chkconfig both not exist";

fi;

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: yes

(grouping 0: dhcp Contains (String) 3:off AND

grouping 0: dhcp Contains (String) 5:off ) OR

grouping 0: dhcp Contains (String) disabled OR

grouping 0: dhcp Contains (String) unknown service OR

grouping 0: dhcp Contains (String) /usr/bin/systemctl dhcp failed to get unit file state for dhcp.service OR

grouping 0: dhcp Contains (String) No such file or directory

========================================================

3.12.8 Verify bluez-coldplug service is disabled

Shell script

sclocate=$(whereis systemctl | awk '{print $2}' 2>/dev/null);

cflocate=$(whereis chkconfig | awk '{print $2}' 2>/dev/null);

srv\_systemctl=bluez-coldplug;

srv\_chkconfig=bluez-coldplug;

if [ $(echo $sclocate | grep systemctl$ | wc -l ) -gt 0 ];

then echo "$sclocate $srv\_systemctl $(systemctl is-enabled $srv\_systemctl 2>&1)";

elif [ $(echo $cflocate | grep chkconfig$ | wc -l) -gt 0 ];

then echo "$cflocate $(chkconfig --list $srv\_chkconfig 2>&1)";

else echo "OS unknown: systemctl and chkconfig both not exist";

fi;

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: yes

(grouping 0: dhcp Contains (String) 3:off AND

grouping 0: dhcp Contains (String) 5:off ) OR

grouping 0: dhcp Contains (String) disabled OR

grouping 0: dhcp Contains (String) unknown service OR

grouping 0: dhcp Contains (String) /usr/bin/systemctl bluez-coldplug failed to get unit file state for bluez\_coldplug.service OR

grouping 0: dhcp Contains (String) No such file or directory

=======================================================

3.12.9 Verify NIS Server services (ypserv) is disabled

Shell script

if [ -f /etc/os-release ];

then ./etc/os-release;

OS=$NAME;

VER=$VERSION\_ID;

elif [ -f /etc/SuSE-release ];

then major\_ver=$(cat /etc/SuSE-release | grep ^VERSION | awk -F'=' '{print $2}');

minor\_ver=$(cat /etc/SuSE-release | grep ^PATCHLEVEL | awk -F'=' '{print $2}');

VER=$(echo "$major\_ver.$minor\_ver" | sed 's/ //g');

OS="sles";

elif [ -f /etc/redhat-release ];

then OS=$(cat /etc/redhat-release);

else OS="unknown"; VER="unknown"; fi;

echo "OS=$OS:VER=$VER";

sclocate=$(whereis systemctl | awk '{print $2}' 2>/dev/null);

srv\_systemctl=ypbind;

srv\_chkconfig=ypbind;

if [ $(echo $sclocate | grep systemctl$ | wc -l ) -gt 0 ];

then echo "$sclocate $srv\_systemctl $(systemctl is-enabled $srv\_systemctl 2>&1)";

elif [ $(echo $sclocate | grep chkconfig$ | wc -l) -gt 0 ];

then echo "$cflocate $(chkconfig --list $srv\_chkconfig 2>&1)";

else echo "OS unknown: systemctl and chkconfig both not exist;

fi;

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: yes

(grouping 0: dhcp Contains (String) 3:off AND

grouping 0: dhcp Contains (String) 5:off ) OR

grouping 0: dhcp Contains (String) disabled OR

grouping 0: dhcp Contains (String) unknown service OR

grouping 0: dhcp Contains (String) /usr/bin/systemctl ypserv failed to get unit file state for ypserv.service OR

grouping 0: dhcp Contains (String) No such file or directory

========================================================

3.12.10 Verify NIS Server services (ypbind) is disabled

Shell script

if [ -f /etc/os-release ];

then ./etc/os-release;

OS=$NAME;

VER=$VERSION\_ID;

elif [ -f /etc/SuSE-release ];

then major\_ver=$(cat /etc/SuSE-release | grep ^VERSION | awk -F'=' '{print $2}');

minor\_ver=$(cat /etc/SuSE-release | grep ^PATCHLEVEL | awk -F'=' '{print $2}');

VER=$(echo "$major\_ver.$minor\_ver" | sed 's/ //g');

OS="sles";

elif [ -f /etc/redhat-release ];

then OS=$(cat /etc/redhat-release);

else OS="unknown"; VER="unknown"; fi;

echo "OS=$OS:VER=$VER";

sclocate=$(whereis systemctl | awk '{print $2}' 2>/dev/null);

srv\_systemctl=ypbind;

srv\_chkconfig=ypbind;

if [ $(echo $sclocate | grep systemctl$ | wc -l ) -gt 0 ];

then echo "$sclocate $srv\_systemctl $(systemctl is-enabled $srv\_systemctl 2>&1)";

elif [ $(echo $sclocate | grep chkconfig$ | wc -l) -gt 0 ];

then echo "$cflocate $(chkconfig --list $srv\_chkconfig 2>&1)";

else echo "OS unknown: systemctl and chkconfig both not exist;

fi;

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: no

(grouping 0: dhcp Contains (String) 3:off AND

grouping 0: dhcp Contains (String) 5:off ) OR

grouping 0: dhcp Contains (String) disabled OR

grouping 0: dhcp Contains (String) unknown service OR

grouping 0: dhcp Contains (String) No such file

=======================================================

3.12.11 Verify rsh client service is disabled

Shell script

ls -la /usr/bin/rsh 2>/dev/null

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: yes

grouping 0: rsh Does not exist (String)

========================================================

3.12.12 Verify UsePAM yes in /etc/ssh/sshd\_config

Shell script

cat /etc/ssh/sshd\_config

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*UsePAM\s(\w+)

case-sensitive: yes

multiline: no

grouping 1: UsePAM Equals (String) yes

========================================================

3.12.13 Verify "LogLevel INFO" in /etc/ssh/sshd\_config

Shell script

cat /etc/ssh/sshd\_config

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*LogLevel\s(\w+)

case-sensitive: yes

multiline: no

grouping 1: LogLevel Equals (String) INFO OR

grouping 1: LogLevel Does not exist (String)

========================================================

3.12.14 Verify "UsePrivilegeSeparation yes" in /etc/ssh/sshd\_config

Shell script

cat /etc/ssh/sshd\_config

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*UsePrivilegeSeparation\s(.\*)

case-sensitive: yes

multiline: no

grouping 1: UsePrivilege Equals (String) yes OR

grouping 1: UsePrivilege Does not exist (String)

========================================================

3.12.15 Verify "StrictModes yes" in /etc/ssh/sshd\_config

Shell script

cat /etc/ssh/sshd\_config

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*StrictModes\s(\w+)

case-sensitive: yes

multiline: no

grouping 1: StrictMode Equals (String) yes OR

grouping 1: StrictMode Does not exist (String)

=======================================================

3.12.16 Verify alsasound Service is Disabled

Shell script

if [ -f /etc/os-release ];

then ./etc/os-release;

OS=$NAME;

VER=$VERSION\_ID;

elif [ -f /etc/SuSE-release ];

then major\_ver=$(cat /etc/SuSE-release | grep ^VERSION | awk -F'=' '{print $4}');

minor\_ver=$(cat /etc/SuSE-release | grep ^PATCHLEVEL | awk -F'=' '{print $2}');

VER=$(echo ""$major\_ver.$minor\_ver"" | sed 's/ //g');

OS="sles";

elif [ -f /etc/redhat-release ];

then OS=$(cat /etc/redhat-release);

else OS="unknown";

VER="unknown";

fi;

echo "OS=$OS:VER=$VER";

sclocate=$(whereis systemctl | awk '{print $2}' 2>/dev/null);

cflocate=$(whereis chkconfig | awk '{print $2}' 2>/dev/null);

srv\_systemctl=alsasound;

srv\_chkconfig=alsasound;

if [ $(echo $sclocate | grep systemctl$ | wc -l ) -gt 0 ];

then echo $sclocate $srv\_systemctl $(systemctl is-enabled $srv\_systemctl 2>&1);

elif [ $(echo $cflocate | grep chkconfig$ | wc -l) -gt 0 ];

then echo "$cflocate $(chkconfig --list $srv\_chkconfig 2>&1)";

else echo "OS unknown: systemctl and chkconfig both not exist";

fi;

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: yes

(grouping 0: alsasound Service Status Contains (String) 3:off AND

grouping 0: alsasound Service Status Contains (String) 5:off ) OR

grouping 0: alsasound Service Status Contains (String) disabled OR

grouping 0: alsasound Service Status Contains (String) masked

========================================================

3.16 Runlevel Services

3.16.02 Verify random Service is 3:ON

Shell script

serv1="random";

serv2="haveged";

if [ -f /etc/os-release ];

then ./etc/os-release;

OS=$NAME;

VER=$VERSION\_ID;

elif [ -f /etc/SuSE-release ];

then major\_ver=$(cat /etc/SuSE-release | grep ^VERSION | awk -F'=' '{print $2}');

minor\_ver=$(cat /etc/SuSE-release | grep ^PATCHLEVEL | awk -F'=' '{print $2}');

VER=$(echo ""$major\_ver.$minor\_ver"" | sed 's/ //g');

OS="sles";

elif [ -f /etc/redhat-release ];

then OS=$(cat /etc/redhat-release);

else OS="unknown";

VER="unknown";

fi;

echo "OS=$OS:VER=$VER";

cflocate=$(which chkconfig 2>/dev/null);

sclocate=$(which systemctl 2>/dev/null);

if [ $(echo $sclocate | awk '{print length}' ) -gt 0 ];

then echo $sclocate $serv2 $(systemctl is-enabled $serv2 2>&1)";

elif [ $(echo $cflocate | awk '{print length}' ) -gt 0 ];

then echo "$cflocate $(chkconfig --list $serv1 2>&1)";

else echo "OS unknown: systemctl and chkconfig both not exist";

fi;

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: yes

(grouping 0: alsasound Service Status Contains (String) 3:off AND

grouping 0: alsasound Service Status Contains (String) 5:off ) OR

grouping 0: alsasound Service Status Contains (String) enabled

=======================================================

3.16.05 Verify autofs Service is Disabled

Shell script

serv1="autofs";

serv2="autofs";

if [ -f /etc/os-release ];

then ./etc/os-release;

OS=$NAME;

VER=$VERSION\_ID;

elif [ -f /etc/SuSE-release ];

then major\_ver=$(cat /etc/SuSE-release | grep ^VERSION | awk -F'=' '{print $2}');

minor\_ver=$(cat /etc/SuSE-release | grep ^PATCHLEVEL | awk -F'=' '{print $2}');

VER=$(echo ""$major\_ver.$minor\_ver"" | sed 's/ //g');

OS="sles";

elif [ -f /etc/redhat-release ];

then OS=$(cat /etc/redhat-release);

else OS="unknown";

VER="unknown";

fi;

echo "OS=$OS:VER=$VER";

cflocate=$(which chkconfig 2>/dev/null);

sclocate=$(which systemctl 2>/dev/null);

if [ $(echo $sclocate | awk '{print length}' ) -gt 0 ];

then echo $sclocate $serv2 $(systemctl is-enabled $serv2 2>&1)";

elif [ $(echo $cflocate | awk '{print length}' ) -gt 0 ];

then echo "$cflocate $(chkconfig --list $serv1 2>&1)";

else echo "OS unknown: systemctl and chkconfig both not exist";

fi;

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: yes

(grouping 0: alsasound Service Status Contains (String) 3:off AND

grouping 0: alsasound Service Status Contains (String) 5:off ) OR

grouping 0: alsasound Service Status Contains (String) masked OR

grouping 0: alsasound Service Status Contains (String) disabled

=======================================================

3.16.10 Verify auditd Service is 3:ON

Shell script

serv1="auditd";

serv2="auditd";

if [ -f /etc/os-release ];

then ./etc/os-release;

OS=$NAME;

VER=$VERSION\_ID;

elif [ -f /etc/SuSE-release ];

then major\_ver=$(cat /etc/SuSE-release | grep ^VERSION | awk -F'=' '{print $2}');

minor\_ver=$(cat /etc/SuSE-release | grep ^PATCHLEVEL | awk -F'=' '{print $2}');

VER=$(echo ""$major\_ver.$minor\_ver"" | sed 's/ //g');

OS="sles";

elif [ -f /etc/redhat-release ];

then OS=$(cat /etc/redhat-release);

else OS="unknown";

VER="unknown";

fi;

echo "OS=$OS:VER=$VER";

cflocate=$(which chkconfig 2>/dev/null);

sclocate=$(which systemctl 2>/dev/null);

if [ $(echo $sclocate | awk '{print length}' ) -gt 0 ];

then echo $sclocate $serv2 $(systemctl is-enabled $serv2 2>&1)";

elif [ $(echo $cflocate | awk '{print length}' ) -gt 0 ];

then echo "$cflocate $(chkconfig --list $serv1 2>&1)";

else echo "OS unknown: systemctl and chkconfig both not exist";

fi;

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: yes

grouping 0: auditd Contains (String) enabled

=======================================================

3.16.11 Verify cron Service is 3:ON

Shell script

serv1="cron";

serv2="cron";

if [ -f /etc/os-release ];

then ./etc/os-release;

OS=$NAME;

VER=$VERSION\_ID;

elif [ -f /etc/SuSE-release ];

then major\_ver=$(cat /etc/SuSE-release | grep ^VERSION | awk -F'=' '{print $2}');

minor\_ver=$(cat /etc/SuSE-release | grep ^PATCHLEVEL | awk -F'=' '{print $2}');

VER=$(echo ""$major\_ver.$minor\_ver"" | sed 's/ //g');

OS="sles";

elif [ -f /etc/redhat-release ];

then OS=$(cat /etc/redhat-release);

else OS="unknown";

VER="unknown";

fi;

echo "OS=$OS:VER=$VER";

cflocate=$(which chkconfig 2>/dev/null);

sclocate=$(which systemctl 2>/dev/null);

if [ $(echo $sclocate | awk '{print length}' ) -gt 0 ];

then echo $sclocate $serv2 $(systemctl is-enabled $serv2 2>&1)";

elif [ $(echo $cflocate | awk '{print length}' ) -gt 0 ];

then echo "$cflocate $(chkconfig --list $serv1 2>&1)";

else echo "OS unknown: systemctl and chkconfig both not exist";

fi;

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: yes

grouping 0: cron Contains (String) enabled

=======================================================

3.16.12 Verify crond Service is 3:ON

Shell script

serv1="network";

serv2="network";

if [ -f /etc/os-release ];

then ./etc/os-release;

OS=$NAME;

VER=$VERSION\_ID;

elif [ -f /etc/SuSE-release ];

then major\_ver=$(cat /etc/SuSE-release | grep ^VERSION | awk -F'=' '{print $2}');

minor\_ver=$(cat /etc/SuSE-release | grep ^PATCHLEVEL | awk -F'=' '{print $2}');

VER=$(echo ""$major\_ver.$minor\_ver"" | sed 's/ //g');

OS="sles";

elif [ -f /etc/redhat-release ];

then OS=$(cat /etc/redhat-release);

else OS="unknown";

VER="unknown";

fi;

echo "OS=$OS:VER=$VER";

cflocate=$(which chkconfig 2>/dev/null);

sclocate=$(which systemctl 2>/dev/null);

if [ $(echo $sclocate | awk '{print length}' ) -gt 0 ];

then echo $sclocate $serv2 $(systemctl is-enabled $serv2 2>&1)";

elif [ $(echo $cflocate | awk '{print length}' ) -gt 0 ];

then echo "$cflocate $(chkconfig --list $serv1 2>&1)";

else echo "OS unknown: systemctl and chkconfig both not exist";

fi;

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: yes

grouping 0: network Contains (String) enabled

=======================================================

3.16.13 Verify syslog Service is 3:ON

Shell script

serv1="syslog";

serv2="syslog";

if [ -f /etc/os-release ];

then ./etc/os-release;

OS=$NAME;

VER=$VERSION\_ID;

elif [ -f /etc/SuSE-release ];

then major\_ver=$(cat /etc/SuSE-release | grep ^VERSION | awk -F'=' '{print $2}');

minor\_ver=$(cat /etc/SuSE-release | grep ^PATCHLEVEL | awk -F'=' '{print $2}');

VER=$(echo ""$major\_ver.$minor\_ver"" | sed 's/ //g');

OS="sles";

elif [ -f /etc/redhat-release ];

then OS=$(cat /etc/redhat-release);

else OS="unknown";

VER="unknown";

fi;

echo "OS=$OS:VER=$VER";

cflocate=$(which chkconfig 2>/dev/null);

sclocate=$(which systemctl 2>/dev/null);

if [ $(echo $sclocate | awk '{print length}' ) -gt 0 ];

then echo $sclocate $serv2 $(systemctl is-enabled $serv2 2>&1)";

elif [ $(echo $cflocate | awk '{print length}' ) -gt 0 ];

then echo "$cflocate $(chkconfig --list $serv1 2>&1)";

else echo "OS unknown: systemctl and chkconfig both not exist";

fi;

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: yes

grouping 0: syslog Contains (String) enabled

=======================================================

3.16.14 Verify sshd Service is 3:ON

Shell script

serv1="sshd";

serv2="sshd";

if [ -f /etc/os-release ];

then ./etc/os-release;

OS=$NAME;

VER=$VERSION\_ID;

elif [ -f /etc/SuSE-release ];

then major\_ver=$(cat /etc/SuSE-release | grep ^VERSION | awk -F'=' '{print $2}');

minor\_ver=$(cat /etc/SuSE-release | grep ^PATCHLEVEL | awk -F'=' '{print $2}');

VER=$(echo ""$major\_ver.$minor\_ver"" | sed 's/ //g');

OS="sles";

elif [ -f /etc/redhat-release ];

then OS=$(cat /etc/redhat-release);

else OS="unknown";

VER="unknown";

fi;

echo "OS=$OS:VER=$VER";

cflocate=$(which chkconfig 2>/dev/null);

sclocate=$(which systemctl 2>/dev/null);

if [ $(echo $sclocate | awk '{print length}' ) -gt 0 ];

then echo $sclocate $serv2 $(systemctl is-enabled $serv2 2>&1)";

elif [ $(echo $cflocate | awk '{print length}' ) -gt 0 ];

then echo "$cflocate $(chkconfig --list $serv1 2>&1)";

else echo "OS unknown: systemctl and chkconfig both not exist";

fi;

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: yes

grouping 0: sshd Contains (String) enabled

=======================================================

3.16.15 Verify dbus Service is 3:ON

Shell script

serv1="dbus";

serv2="dbus";

if [ -f /etc/os-release ];

then ./etc/os-release;

OS=$NAME;

VER=$VERSION\_ID;

elif [ -f /etc/SuSE-release ];

then major\_ver=$(cat /etc/SuSE-release | grep ^VERSION | awk -F'=' '{print $2}');

minor\_ver=$(cat /etc/SuSE-release | grep ^PATCHLEVEL | awk -F'=' '{print $2}');

VER=$(echo ""$major\_ver.$minor\_ver"" | sed 's/ //g');

OS="sles";

elif [ -f /etc/redhat-release ];

then OS=$(cat /etc/redhat-release);

else OS="unknown";

VER="unknown";

fi;

echo "OS=$OS:VER=$VER";

cflocate=$(which chkconfig 2>/dev/null);

sclocate=$(which systemctl 2>/dev/null);

if [ $(echo $sclocate | awk '{print length}' ) -gt 0 ];

then echo $sclocate $serv2 $(systemctl is-enabled $serv2 2>&1)";

elif [ $(echo $cflocate | awk '{print length}' ) -gt 0 ];

then echo "$cflocate $(chkconfig --list $serv1 2>&1)";

else echo "OS unknown: systemctl and chkconfig both not exist";

fi;

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: yes

grouping 0: dbus Contains (String) enabled OR

grouping 0: dbus Contains (String) static

=======================================================

3.16.19 Verify postfix Service is Enabled

Description

SMTP server for send system message

Shell script

cat /etc/SuSE-release;

echo "postfix $(systemctl is-enabled postfix 2>&1)";

chkconfig --list postfix 2>&1;

ps -ef | grep master 2>&1;

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: yes

(grouping 0: postfix Contains (String) 3:on AND

grouping 0: postfix Contains (String) 5:on) OR

grouping 0: postfix Contains (String) enabled OR

grouping 0: postfix Contains (String) /usr/lib/postfix/bin/master OR

grouping 0: postfix Contains (String) /usr/lib/postfix/bin//master OR

grouping 0: postfix Contains (String) /usr/lib/postfix/master OR

========================================================

 3.18 Reviewing InitTab and Boot Scripts

3.18.1 Verify Ctrl+Alt+Del key Sequence is Disabled

Shell script

if [ -f /etc/os-release ];

then ./etc/os-release;

OS=$NAME;

VER=$VERSION\_ID;

elif [ -f /etc/SuSE-release ];

then major\_ver=$(cat /etc/SuSE-release | grep ^VERSION | awk -F'=' '{print $2}');

minor\_ver=$(cat /etc/SuSE-release | grep ^PATCHLEVEL | awk -F'=' '{print $2}');

VER=$(echo "$major\_ver.$minor\_ver" | sed 's/ //g');

OS="sles";

elif [ -f /etc/redhat-release ];

then OS=$(cat /etc/redhat-release);

else OS="unknown";

fi;

echo "OS=$OS:VER=$VER";

if [ -f /etc/inittab ];

then echo "/etc/inittab : $(egrep '^ca::ctrlaltdel' /etc/inittab)";

else ls -la /etc/systemd/system/ctrl-alt-del.target 2>&1;

fi;

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: yes

(grouping 0: Ctrl-Alt-Del Contains (String) 12 AND

grouping 0: Ctrl-Alt-Del Contains (String) /etc/systemd/system/ctrl-alt-del.target -> /dev/null) OR

(grouping 0: Ctrl-Alt-Del Contains (String) 11 AND

grouping 0: Ctrl-Alt-Del Contains (String) /etc/inittab) OR

grouping 0: Ctrl-Alt-Del Contains (String) /etc/systemd/system/ctrl-alt-del.target -> /dev/null

========================================================

3.18.2 Verify GUI is Disabled

Shell script

if [ -f /etc/os-release ];

then ./etc/os-release;

OS=$NAME;

VER=$VERSION\_ID;

elif [ -f /etc/SuSE-release ];

then major\_ver=$(cat /etc/SuSE-release | grep ^VERSION | awk -F'=' '{print $2}');

minor\_ver=$(cat /etc/SuSE-release | grep ^PATCHLEVEL | awk -F'=' '{print $2}');

VER=$(echo "$major\_ver.$minor\_ver" | sed 's/ //g');

OS="sles";

elif [ -f /etc/redhat-release ];

then OS=$(cat /etc/redhat-release);

else OS="unknown";

VER="unknown";

fi;

echo "OS=$OS:VER=$VER";

sclocate=$(whereis systemctl | awk '{print $2}' 2>/dev/null);

if [ $(echo $sclocate | grep systemctl$ | wc -l ) -gt 0 ];

then echo "$sclocate $srv\_systemctl $(systemctl get-default 2>&1)";

elif [ -f /etc/inittab ];

then echo "$(egrep '^id:[12345]:initdefault' /etc/inittab;

else echo "OS unknown: systemctl and /etc/inittab both not exist";

fi;

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: yes

grouping 0: GUI Login Setting Contains (String) multi-user.target

========================================================

3.20 Securing SSH

3.20.1 Verify sshd\_config Disables PermitRootLogin

Shell script

cat /etc/ssh/sshd\_config

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*PermitRootLogin[\ \t]+(\w+)[\ \t]\*$

case-sensitive: no

multiline: yes

grouping 1: PermitRootLogin Equals (String) no

=======================================================

3.20.2 Verify Privilege Separation is Enabled

Shell script

cat /etc/os-release /etc/SuSE-release /etc/ssh/sshd\_config 2>/dev/null | egrep -v '^#' | egrep -i 'UsePrivilegeSeparation|VERSION\_ID'

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: yes

grouping 0: UsePrivilegeSeparation Finds (String) ^\s\*UsePrivilegeSeparation\s+(\w+)\s\*#?.\*$ OR

grouping 0: UsePrivilegeSeparation Finds (String) VERSION\_ID="15.1" OR

grouping 0: UsePrivilegeSeparation Finds (String) VERSION\_ID="15.2" OR

grouping 0: UsePrivilegeSeparation Finds (String) VERSION\_ID="15.3" OR

grouping 0: UsePrivilegeSeparation Finds (String) VERSION\_ID="15.4" OR

========================================================

3.20.3 Verify sshd\_config uses Protocol 2 only

Shell script

cat /etc/ssh/sshd\_config

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*Protocol[\ \t]+(\d+).\*$

case-sensitive: no

multiline: yes

grouping 1: SSH Protocol Version Equals (Numeric) 2

=======================================================

3.20.4 Verify TCP Forwarding is Disabled

Shell script

cat /etc/ssh/sshd\_config

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*AllowTcpForwarding[\ \t]+(\w+)[\ \t]\*$

case-sensitive: no

multiline: yes

grouping 1: AllowTcpForwarding Equals (String) no OR

grouping 1: AllowTcpForwarding Does not exist (String)

=======================================================

3.20.5 Verify X11 Forwarding is Disabled

Shell script

cat /etc/ssh/sshd\_config

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*X11Forwarding[\ \t]+(\w+)[\ \t]\*$

case-sensitive: no

multiline: yes

grouping 1: X11Forwarding Equals (String) no OR

grouping 1: X11Forwarding Does not exist (String)

=======================================================

3.20.6 Verify StrictModes is Enabled

Shell script

cat /etc/ssh/sshd\_config

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*StrictModes[\ \t]+(\w+).\*$

case-sensitive: yes

multiline: yes

grouping 1: StrictModes Equals (String) yes OR

grouping 1: StrictModes Does not exist (String)

=======================================================

3.20.7 Disable All Host-based Authentications

3.20.7.1 Verify sshd\_config Enables IgnoreRhosts

Shell script

cat /etc/ssh/sshd\_config

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*IgnoreRhosts[\ \t]+(\w+)[\ \t]\*$

case-sensitive: no

multiline: yes

grouping 1: IgnoreRhosts Equals (String) yes OR

-----------------------------------------------------------------------------------------

3.20.7.2 Verify sshd\_config Disables HostbasedAuthentication

Shell script

cat /etc/ssh/sshd\_config

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*HostbasedAuthentication[\ \t]+(\w+)[\ \t]\*$

case-sensitive: no

multiline: yes

grouping 1: HostbasedAuthentication Status Equals (String) no OR

----------------------------------------------------------------------------------------

3.20.7.3 Verify sshd\_config Disables RhostsRSAAuthentication

This test verifies that host-based authentication is disabled.

Host-based authentication allows authentication to occur without any user challenge.

This form of authentication is inherently insecure.

Shell script

cat /etc/ssh/sshd\_config

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*HostbasedAuthentication[\ \t]+(\w+)[\ \t]\*$

case-sensitive: no

multiline: yes

grouping 1: RhostsRSAAuthentication Equals (String) no OR

3.24 Checking File Permissions and Ownership

3.24.33 Verify /etc/pam.d/common-auth and /etc/sssd/sssd.conf (3-tier members)

Shell script

sssd=$(systemctl is-enabled sssd 2>/dev/null);

winbind=$(chkconfig --list winbind 2>/dev/null|grep 3:on | wc -l);

if [[ [$sssd == "enabled"](onenote:#$sssd%20==%20%22enabled%22&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={017E381C-D49B-4A57-AE0F-D8BDFCF1144D}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]];

then echo "systemctl is-enabled sssd = enabled";

echo "/etc/sssd/sssd.conf";

cat /etc/sssd/sssd.conf | grep -P "^simple\_allow\_groups\s{0,}=\s{0,}(?=.\*linux\_group1)(?=.\*linux\_group2)(?=.\*linux\_group3)";

elif [[ [$winbind == "1"](onenote:#$winbind%20==%20%221%22&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={45015D7E-2431-443A-8C2A-F25686DF8F56}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]];

then chkconfig --list winbind;

echo "/etc/pam.d/common-auth-pc";

cat /etc/pam.d/common-auth-pc | grep -P "^auth\s+required\s+pam\_winbind.so\s+require\_membership\_of=(?=.\*linux\_group1)(?=.\*linux\_group2)(?=.\*linux\_group3)";

else echo "sssd or winbind not enabled, not join any domain";

fi;

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: yes

grouping 0: 3-tier members Contains (String) require\_membership\_of OR

grouping 0: 3-tier members Contains (String) simple\_allow\_groups OR

grouping 0: 3-tier members Contains (String) not join any domain

========================================================

3.24.34 Verify /etc/sudoers

3.24.34.1 Verify /etc/sudoers or /etc/sudoers.d/\* [\\linux\_group1](file:///\\linux_group1) ALL =

Description

Verify what accounts can sudo

Shell script

if [ $(grep "^#includedir /etc/sudoers.d" /etc/sudoers | wc -l) -gt 0];

then echo "#includedir /etc/sudoers.d exist in /etc/sudoers";

subdir=$(ls /etc/sudoers.d/\* | sed ':a;N;$!ba;s/\n/ /g');

else echo "#includedir not exist in /etc/sudoers";

echo "#/etc/sudoers";

subdir="";

fi;

cat /etc/sudoers $subdir | egrep -v '^#|^$' | egrep 'linux\_group1\s\*ALL=' | grep -P '(?=.\*DAEMON)(?=.\*KILL)(?=.\*TAR)(?=.\*HWINFO)(?=.\*SUPPORTCONFIG)(?=.\*READFILE)(?=.\*DU)(?=.\*BACKUP)'

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: no

grouping 0: linux\_group1 exists (string)

------------------------------------------------------------------------------------------

3.24.34.2 Verify /etc/sudoers or /etc/sudoers.d/\* [\\linux\_superadmins](file:///\\linux_superadmins) ALL =

Description

Verify what linux superadmins group can do

[\\linux\_superadmins](file:///\\linux_superadmins) ALL=(ALL) ALL

Shell script

cat /etc/sudoers /etc/sudoers.d/\* 2>/dev/null

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

linux\_superadmins\s+ALL\=(.\*)

case-sensitive: no

multiline: yes

grouping 1: linux\_superadmins ALL Contains (string) (ALL) ALL

------------------------------------------------------------------------------------------

3.24.34.3 Verify /etc/sudoers or /etc/sudoers.d/\*

[\\linux\_group1\_supervisors](file:///\\linux_group1_supervisors) ALL =

Description

Verify what linux\_group1\_supervisors can sudo

Shell script

if [ $(grep "^#includedir /etc/sudoers.d" /etc/sudoers | wc -l) -gt 0 ];

then subdir=$(ls /etc/sudoers.d/\* | sed ':a;N;$!ba;s/\n /g';

else echo "#includedir not exist in /etc/sudoers";

echo "#/etc/sudoers";

subdir="";

fi;

cat /etc/sudoers $subdir | egrep -v '^#|^$' | egrep 'linux\_group1\_supervisors\s\*ALL' | grep -P '(?=.\*PASSWD)(?=.\*CHMOD)(?=.\*YAST)(?=.\*FDISK)(?=.\*MOUNT)(?=.\*UMOUNT)(?=.\*CRONTAB)(?=.\*INIT)(?=.\*DATE)(?=.\*SHUTDOWN)(?=.\*QUEST\_STATUS)(?=.\*REMOVEFILE)(?=.\*!PASSWD\_ROOT)'

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: yes

grouping 0: linux\_group1\_supervisors ALL Contains (string) linux\_group1\_supervisors

--------------------------------------------------------------------------------------------

3.24.34.4 Verify /etc/sudoers or /etc/sudoers.d/\* root All=

Description

root ALL=(ALL) ALL

Shell script

cat /etc/sudoers /etc/sudoers.d/\* 2>/dev/null

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*root[\ \t]+ALL=(.\*)

case-sensitive: no

multiline: yes

grouping 1: root ALL Contains (string) (ALL) ALL

========================================================

3.25 Default umask

3.25.1 Verify Default umask for Users in /etc/profile

Description

This test determines if the default umask for users in /etc/profile is set to 022(default) or 027 or 077 or does not exist.

With this umask, files and directories created by users are not readable by any other user on the system.

This restrictive umask setting has been known to cause issues with installation package scripts that attempt to use the default umask.

The workaround is to manually issue a less restrictive umask. e.g. umask 022 to the shell session used by the installer.

Shell script

egrep '^umask\s{1,}' /etc/profile | awk '{print $2}' 2>/dev/null

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*umask[\ \t](\d+)

case-sensitive: yes

multiline: yes

grouping 0: Default umask setting deviation Does not exist (String) OR

grouping 1: Default umask setting deviation (string) 077 OR

grouping 1: Default umask setting deviation (string) 027 OR

grouping 1: Default umask setting deviation (string) 022 OR

--------------------------------------------------------------------------------------------

3.25.2 Verify Default umask for Users in /etc/csh.cshrc

Description

This test determines if the default umask for users in /etc/csh.cshrc is set to 022(default) or 027 or 077 or does not exist.

With this umask, files and directories created by users are not readable by any other user on the system.

Please note that this restrictive umask setting has been unknown to cause issues with installation package scripts that attempt to use the default umask.

The workaround is to manualy issue a less restrictive umask e.g. umask 022 to the shell session used by the installer.

Shell script

egrep '^umask\s{1,}' /etc/csh.cshrc | awk '{print $2}' 2>/dev/null

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*umask[\ \t](\d+)

case-sensitive: no

multiline: yes

grouping 0: Default umask setting deviation Does not exist (String) OR

grouping 1: Default umask setting deviation (string) 077 OR

grouping 1: Default umask setting deviation (string) 027 OR

grouping 1: Default umask setting deviation (string) 022 OR

-------------------------------------------------------------------------------------------

3.25.3 Verify Default umask for Users /etc/bash.bashrc

Description

This test determines if the default umask for users in /etc/bash.bashrc is set to 022(default) or 027 or 077 or does not exist.

With this umask, files and directories created by users are not readable by any other user on the system.

Please note that this restrictive umask setting has been unknown to cause issues with installation package scripts that attempt to use the default umask.

The workaround is to manualy issue a less restrictive umask e.g. umask 022 to the shell session used by the installer.

Shell script

egrep '^umask\s{1,}' /etc/bash.bashrc | awk '{print $2}' 2>/dev/null

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*umask[\ \t](\d+)

case-sensitive: no

multiline: yes

grouping 0: Default umask setting deviation Does not exist (String) OR

grouping 1: Default umask setting deviation (string) 077 OR

grouping 1: Default umask setting deviation (string) 027 OR

grouping 1: Default umask setting deviation (string) 022 OR

--------------------------------------------------------------------------------------------

3.25.4 Verify Default umask for Users /etc/csh.login

Description

This test determines if the default umask for users in /etc/csh.login is set to 022(default) or 027 or 077 or does not exist.

With this umask, files and directories created by users are not readable by any other user on the system.

Please note that this restrictive umask setting has been unknown to cause issues with installation package scripts that attempt to use the default umask.

The workaround is to manualy issue a less restrictive umask e.g. umask 022 to the shell session used by the installer.

Shell script

egrep '^umask\s{1,}' /etc/csh.login | awk '{print $2}' 2>/dev/null

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*umask[\ \t](\d+)

case-sensitive: no

multiline: yes

grouping 0: Default umask setting deviation Does not exist (String) OR

grouping 1: Default umask setting deviation (string) 077 OR

grouping 1: Default umask setting deviation (string) 027 OR

grouping 1: Default umask setting deviation (string) 022 OR

------------------------------------------------------------------------------------------

3.25.6 Verify Default umask for Users in /etc/login.defs

Description

This test determines if the default umask for users in /etc/login.defs is set to 022(default) or 027 or 077 or does not exist.

With this umask, files and directories created by users are not readable by any other user on the system.

Please note that this restrictive umask setting has been unknown to cause issues with installation package scripts that attempt to use the default umask.

The workaround is to manualy issue a less restrictive umask e.g. umask 022 to the shell session used by the installer.

Shell script

egrep '^umask\s{1,}' /etc/login.defs | awk '{print $2}' 2>/dev/null

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*umask[\ \t](\d+)

case-sensitive: no

multiline: yes

grouping 0: Default umask setting deviation Does not exist (String) OR

grouping 1: Default umask setting deviation (string) 077 OR

grouping 1: Default umask setting deviation (string) 027 OR

grouping 1: Default umask setting deviation (string) 022 OR

-------------------------------------------------------------------------------------------

3.25.7 Verify Default umask for Users in /etc/bash.bashrc, if /etc/bash.bashrc exists

Description

This test determines if the default umask for users in /etc/login.defs is set to 022(default) or 027 or 077 or does not exist.

With this umask, files and directories created by users are not readable by any other user on the system.

This is a setting agreed upon by CIS, DISA, and NSA.

This restrictive umask setting has been known to cause issues with installation package scripts that attempt to use the default umask.

The workaround is to manualy issue a less restrictive umask e.g. umask 022 to the shell session used by the installer.

Shell script

cat /etc/bash.bashrc

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*umask[\ \t](\d+)

case-sensitive: no

multiline: yes

grouping 0: Default umask setting deviation Does not exist (String) OR

grouping 1: Default umask setting deviation (string) 077 OR

grouping 1: Default umask setting deviation (string) 027 OR

grouping 1: Default umask setting deviation (string) 022 OR

grouping 0: Default umask setting deviation Contains (String) No such file

========================================================

3.27 World-writable Files

3.27.1 World-writable Directories should have Sticky Bit Set

Description

This test verifies that 'sticky bit' is set on all world-writable directories.

When 'sticky bit' is set o na directory, only the owner of a file may remove that file from the directory.

(only for /boot, /root, /sbin, /bin, /lib64 and /lib except /lib/udev/devices)

All World-writable Files except home:

Detects the installed anti-virus software has recently updated the virus definitions.

Shell script

ver=$(cat /etc/SuSE-release | egrep ^VERSION | awk -F'=' '{print $2}'|tr -d " ");

chkdir="/boot /root /sbin /bin /lib64 /lib";

for i in $chkdir;

do if [ $ver=="11" ];

then ppath="/lib/udev/devices";

else ppath="/proc";

fi;

find $i -path $ppath -prune -o -perm -2 ! -type l -ls ;

done;

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: yes

grouping 0: World-writable Directories without Sticky Bit Does not exist (String) OR

grouping 0: World-writable Directories without Sticky Bit Contains (String) No such file

-------------------------------------------------------------------------------------------

3.27.2 World-writable Files (/home)

Description

This test verifies that 'sticky bit' is set on all world-writable directories.

When 'sticky bit' is set on a directory, only the owner of a file may remove that file from the directory.

(only for /home)

Shell script

find /home -path /proc -prune -o -perm -2 ! -type l -ls

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: yes

grouping 0: World-writable Directories without Sticky Bit Does not exist (String) OR

========================================================

3.28 Orphaned or Unowned Files

Orphaned or Unowned Files (for folders /boot /root /sbin /bin /lib64 /lib /home)

Description

This test checks for the presence of unowned files on the file system.

Any unowned files found on the file system should be carefully reviewed by the sysadmin.

Unowned files may be an indication of unauthorized system access or improper package maintenance/installation.

Shell script

tmpfile=/tmp/orphantmp$$.txt;

touch $tmpfile;

chkpath="/boot /root /sbin /bin /lib64 /lib /home";

for p in $chkpath;

do find $p -path /proc prune -o -nouser -o -nogroup 2>/dev/null | egrep -v '^\/proc'>>$tmpfile;

done;

for i in $(cat $tmpfile);

do ls -la $i;

done;

rm -rf $tmpfile

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: yes

grouping 0: Unowned Files Does not exist (String) OR

grouping 0: Unowned Files Contains (String) /proc OR

grouping 0: Unowned Files Contains (String) /var/lib/ntp/proc OR

3.29 Various Account Checks

3.29.1 Unlocked Accounts

3.29.1.1 Block System Accounts

-------------------------------------------------------------------------------------------

3.29.1.2 Verify that there are NO accounts with Empty Password Fields

Description

This test determines if any individual accounts listed in /etc/shadow have empty passwords.

All accounts should have strong passwords or the account should be locked.

Shell script

awk -F':' '{print $2,$1}' /etc/shadow | egrep '^\s'

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: yes

grouping 0: Empty Password Accounts Does not exist (String) OR

--------------------------------------------------------------------------------------------

3.29.1.3 Verify that all accounts use Shadow Password

Description

This test verifies that all accounts use shadow password.

Shadow password is used to increase the security level of passwords by restricting all but highly privileged users' access to hashed password data.

Shell script

cat /etc/passwd | awk -F':' '{print $2,$1}' | egrep -v '^x'

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: yes

grouping 0: Password Setting Deviation Does not exist (String) OR

3.30 Single User Mode Password for ROOT

This test verifies that single user mode requires root-level access.

Authentication should always be required for root-level access.

On some Linux systems user mode is entered using the 'Linux Single' command in the GRUB boot-editing menu, which represents a security risk since no authentication is required.

Shell script

if [ -f /etc/os-release ];

then ./etc/os-release;

OS=$NAME;

VER=$VERSION\_ID;

elif [ -f /etc/SuSE-release ];

then major\_ver=$(cat /etc/SuSE-release | grep ^VERSION | awk -F'=' '{print $2}');

minor\_ver=$(cat /etc/SuSE-release | grep ^PATCHLEVEL | awk -F'=' '{print $2}');

VER=$(echo "$major\_ver.$minor\_ver" | sed 's/ //g');

OS="sles";

elif [ -f /etc/redhat-release ];

then OS=$(cat /etc/redhat-release);

else OS="unknown";

VER="unknown";

fi;

echo "OS=$OS:VER=$VER";

if [[ $(echo "11.4 < $VER" | bc) -eq 1 ]];

then egrep '^ExecStart=' /usr/lib/systemd/system/rescue.service | grep sulogin 2>/dev/null;

elif [[ $(echo "11.4 >=$VER" | bc) -eq 1 ]];

then cat /etc/inittab | egrep "~~:S:respawn:/sbin/sulogin";

fi;

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

(~~:S:respawn:\/sbin\/sulogin)|(\/usr\/sbin\/sulogin)|(\/usr\/lib\/systemd\/systemd-sulogin-shell)

case-sensitive: yes

multiline: yes

grouping 0: Single User Mode Setting Exists (String) OR

3.31 Enabling Password Aging

3.31.1 Verify PASS\_MIN\_DAYS Parameter in /etc/login.defs

Description

This test verifies that /etc/login.defs is configured to prevent password changes for 1 day.

This setting is used for the creation of new accounts.

Preventing frequent password resets helps protect against brute-force password cracking programs.

Shell script

cat /etc/login.defs

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*PASS\_MIN\_DAYS[\ \t]+(\d+)(?:$|\#)

case-sensitive: no

multiline: yes

grouping 1: Passwd\_min\_days >= (Numeric) 1 OR

-------------------------------------------------------------------------------------------

3.31.2 Verify PASS\_MIN\_DAYS Setting for Non-system Accounts

Description

This test verifies that all non-system accounts are configured to prevent password changes for at least 1 day.

Preventing frequent password resets helps protect against brute-force password cracking programs.

Shell script

sysacct=$(cut -d';' -f1,3 /etc/passwd | egrep ':[1-9][0-9]{3}$|:[1-5][0-9]{3,4}$|:60000$' | cut -d':' -f1);

nonsysusers=$(echo $sysacct);

minday\_e=${pmind[$servtype]};

maxday\_e=${pmaxd[$servtype]};

inactive\_e=${inactiveday[$servtype]};

for user1 in root $nonsysusers;

do

is\_locked=$(passwd -S $user1 | awk '{print $2}');

usrshell=$(cat /etc/passwd | egrep ""^$user1:"" | awk -F':' '{print $7}');

usr\_uid=$(cat /etc/passwd | egrep ""^$user1:"" | awk -F':' '{print $3}');

err\_per\_user=0;

minday=$(passwd -S $user1 | awk '{print $4}');

if [[ $minday -lt $minday\_e ]];

then ((err\_per\_user++));

fi;

maxday=$(passwd -S $user1 | awk '{print $5}');

if [[ $maxday -gt $maxday\_e ]];

then (err\_per\_user++));

fi;

inactive=$(passwd -S $user1 | awk '{print $7}');

is\_locked=$(passwd -S $user1 | awk '{print $2}');

usrshell=$(cat /etc/passwd | egrep ""^$user1:"" | awk -F':' '{print $7}');

lastchange=$(passwd -S $user1 | awk '{print $3}');

lastchange\_epoch=$(date +%s -d""$lastchange"");

if [[ $inactive -lt $inactive\_e ]];

then ((err\_per\_user++));

fi;

if [[ $is\_locked != ""L"" && $is\_locked !=""LK"" && $usrshell != ""/bin/false"" && $usrshell != ""/sbin/nologin"" ]];

then next\_pw\_change=$((lastchange\_epoch+maxday\*86400));

npwc=$(date +%m/%d/%Y --date=""@next\_pw\_change"");

echo ""$user1:PASS\_MIN\_DAYS=$minday PASS\_MAX\_DAYS=$maxday INACTIVE=$inactive IS\_LOCKED=$is\_locked LASTPWCHANGE=$lastchange USERSHELL=$usrshell"";

else echo ""$user1:PASS\_MIN\_DAYS=$minday PASS\_MAX\_DAYS=$maxday INACTIVE=$inactive IS\_LOCKED=$is\_locked LASTPWCHANGE=$lastchange USERSHELL=$usrshell"";

fi;

done;

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

(.\*PASS\_MIN\_DAYS=1\s+.\*\s+IS\_LOCKED=(P|PS))|(.\*IS\_LOCKED=(L|LK))|(.\*USERSHELL=(\/bin\/false|\/bin\/nologin))

case-sensitive: yes

multiline: yes

grouping 0: Passwd\_min\_days >= Exists (String)

-------------------------------------------------------------------------------------------

3.31.3 Verify PASS\_MAX\_DAYS Parameter in /etc/login.defs

Description

This test verifies that /etc/login.defs is configured to force password change after 60 days or less.

This setting is used for the creation of new accounts.

Requiring regular password changes ensures that if a password is cracked, it will only be valid temporarily.

Shell script

cat /etc/login.defs

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*PASS\_MAX\_DAYS[\ \t]+(\d+)(?:$|\#)

case-sensitive: no

multiline: yes

grouping 1: Passwd\_max\_days >= (Numeric) 1 AND

grouping 1: Passwd\_max\_days <= (Numeric) 60 AND

------------------------------------------------------------------------------------------

3.31.4 Verify PASS\_MAX\_DAYS Setting for Non-system Accounts

Description

This test verifies that all non-system accounts are configured to expire every 90 days or less.

Requiring regular password changes ensures that if a password is cracked, it will only be valid temporarily.

Shell script

sysacct=$(cut -d':' -f1,3 /etc/passwd | egrep ':[1-9][0-9]{3}$|:[1-5][0-9]{3,4}$|:60000$' | cut -d':' -f1);

nonsysusers=$(echo $sysacct);

for user1 in root $nonsysusers;

do is\_locked=$(passwd -S $user1 | awk '{print $2}');

usrshell=$(cat /etc/passwd | egrep ""^$user1:"" | awk -F':' '{print $7}');

usr\_uid=$(cat /etc/passwd | egrep ""^$user1:"" | awk -F':' '{print $3}');

minday=$(passwd -S $user1 | awk '{print $4}');

maxday=$(passwd -S $user1 | awk '{print $5}');

inactive=$(passwd -S $user1 | awk '{print $7}');

is\_locked=$(passwd -S $user1 | awk '{print $2}');

usrshell=$(cat /etc/passwd | egrep ""^$user1:"" | awk -F':' '{print $7}');

lastchange=$(passwd -S $user1 | awk '{print $3}');

echo ""$user1:PASS\_MIN\_DAYS=$minday PASS\_MAX\_DAYS=$maxday INACTIVE=$inactive IS\_LOCKED=$is\_locked LASTPWCHANGE=$lastchange USERSHELL=$usrshell UID=$usr\_uid"";

done

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

(.\*PASS\_MAX\_DAYS=(\b([1-5][0-9])|60|[1-9])\s+.\*\s+IS\_LOCKED=(P|PS))|(.\*IS\_LOCKED=(L|LK))|(.\*USERSHELL=(\/bin\/false|\/bi\/nologin))

case-sensitive: no

multiline: yes

grouping 0: Passwd\_max\_days Exists (String)

 3.32 Stronger Password Enforcement

3.32.1 Password Length

Description

This test verifies that the system is configured to use a minimum password length of 8 characters.

Using longer passwords hinders the ability of an attacker to use brute-force methods in trying to gain access to the system.

Shell script

directory="/etc/pam.d";

files="passwd";

files=$files$(/bin/cat /etc/pam.d/passwd 2>/dev/null | /bin/awk -F"#" ' $0 ~ /^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]+include[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]].\*/ {print $1}' | /bin/awk 'BEGIN {ORS=";"} {print $3}');

SavedIFS=$IFS;

IFS=";";

for file in $files;

do

if [ "`/usr/bin/dirname $file 2>/dev/null`" !="." ];

then if [ -f "$file" ];

then /bin/echo "$file contains:";

/bin/cat $file 2>/dev/null | /bin/egrep "^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password";

fi;

else full\_path=$directory"/"$file;

fi;

if [ -f "$full\_path" ];

then /bin/echo "$full\_path contains:";

/bin/cat $full\_path 2>/dev/null | /bin/egrep "^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password";

fi;

done;

IFS=$SavedIFS;

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*password[\ \t]+(?:requisite|required)[\ \t]+[^\#&&\S]\*\bpam\_cracklib\.so[\ \t]+[^\#]\*\bminlen=(\d+)\b.\*

case-sensitive: yes

multiline: yes

grouping 1: Minimum Password Length >= (Numeric) 12

------------------------------------------------------------------------------------------------

3.32.2 Password Character Mix: At least 1 Lowercase Character

Description

This test verifies that passwords include at least 1 lowercase alphabetic character.

Forcing users to use complex passwords makes it more difficult for attackers to gain access to the system.

Shell script

directory="/etc/pam.d";

files="passwd";

files=$files$(/bin/cat /etc/pam.d/passwd 2>/dev/null | /bin/awk -F"#" ' $0 ~ /^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]+include[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]].\*/ {print $1}' | /bin/awk 'BEGIN {ORS=";"} {print $3}');

SavedIFS=$IFS;

IFS=";";

for file in $files;

do

if [ "`/usr/bin/dirname $file 2>/dev/null`" !="." ];

then if [ -f "$file" ];

then /bin/echo "$file contains:";

/bin/cat $file 2>/dev/null | /bin/egrep "^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password";

fi;

else full\_path=$directory"/"$file;

fi;

if [ -f "$full\_path" ];

then /bin/echo "$full\_path contains:";

/bin/cat $full\_path 2>/dev/null | /bin/egrep "^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password";

fi;

done;

IFS=$SavedIFS;

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*password[\ \t]+(?:requisite|required)[\ \t]+[^\#&&\S]\*\bpam\_cracklib\.so[\ \t]+[^\#]\*\blcredit=(\d+)\b.\*

case-sensitive: yes

multiline: yes

grouping 1: Minimum Lowercase Characters >= (Numeric) 1

--------------------------------------------------------------------------------------------------

3.32.3 Password Character Mix: At least 1 Uppercase Character

Description

This test verifies that passwords include at least 1 uppercase alphabetic character.

Forcing users to use complex passwords makes it more difficult for attackers to gain access to the system.

Shell script

directory="/etc/pam.d";

files="passwd";

files=$files$(/bin/cat /etc/pam.d/passwd 2>/dev/null | /bin/awk -F"#" ' $0 ~ /^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]+include[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]].\*/ {print $1}' | /bin/awk 'BEGIN {ORS=";"} {print $3}');

SavedIFS=$IFS;

IFS=";";

for file in $files;

do

if [ "`/usr/bin/dirname $file 2>/dev/null`" !="." ];

then if [ -f "$file" ];

then /bin/echo "$file contains:";

/bin/cat $file 2>/dev/null | /bin/egrep "^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password";

fi;

else full\_path=$directory"/"$file;

fi;

if [ -f "$full\_path" ];

then /bin/echo "$full\_path contains:";

/bin/cat $full\_path 2>/dev/null | /bin/egrep "^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password";

fi;

done;

IFS=$SavedIFS;

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*password[\ \t]+(?:requisite|required)[\ \t]+[^\#&&\S]\*\bpam\_cracklib\.so[\ \t]+[^\#]\*\bucredit=-(\d+)\b.\*

case-sensitive: yes

multiline: yes

grouping 1: Minimum Uppercase Characters >= (Numeric) 1

--------------------------------------------------------------------------------------------------

3.32.4 Password Character Mix: At least 1 Numerical Character

Description

This test verifies that passwords include at least 1 numerical character.

Forcing users to use complex passwords makes it more difficult for attackers to gain access to the system.

Shell script

directory="/etc/pam.d";

files="passwd";

files=$files$(/bin/cat /etc/pam.d/passwd 2>/dev/null | /bin/awk -F"#" ' $0 ~ /^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]+include[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]].\*/ {print $1}' | /bin/awk 'BEGIN {ORS=";"} {print $3}');

SavedIFS=$IFS;

IFS=";";

for file in $files;

do

if [ "`/usr/bin/dirname $file 2>/dev/null`" !="." ];

then if [ -f "$file" ];

then /bin/echo "$file contains:";

/bin/cat $file 2>/dev/null | /bin/egrep "^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password";

fi;

else full\_path=$directory"/"$file;

fi;

if [ -f "$full\_path" ];

then /bin/echo "$full\_path contains:";

/bin/cat $full\_path 2>/dev/null | /bin/egrep "^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password";

fi;

done;

IFS=$SavedIFS;

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*password[\ \t]+(?:requisite|required)[\ \t]+[^\#&&\S]\*\bpam\_cracklib\.so[\ \t]+[^\#]\*\bdcredit=-(\d+)\b.\*

case-sensitive: yes

multiline: yes

grouping 1: Minimum Uppercase Characters >= (Numeric) 1

--------------------------------------------------------------------------------------------------

3.32.5 Password Character Mix: At least 1 Special Character

Description

This test verifies that passwords include at least 1 special character.

Forcing users to use complex passwords makes it more difficult for attackers to gain access to the system.

Shell script

directory="/etc/pam.d";

files="passwd";

files=$files$(/bin/cat /etc/pam.d/passwd 2>/dev/null | /bin/awk -F"#" ' $0 ~ /^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]+include[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]].\*/ {print $1}' | /bin/awk 'BEGIN {ORS=";"} {print $3}');

SavedIFS=$IFS;

IFS=";";

for file in $files;

do

if [ "`/usr/bin/dirname $file 2>/dev/null`" !="." ];

then if [ -f "$file" ];

then /bin/echo "$file contains:";

/bin/cat $file 2>/dev/null | /bin/egrep "^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password";

fi;

else full\_path=$directory"/"$file;

fi;

if [ -f "$full\_path" ];

then /bin/echo "$full\_path contains:";

/bin/cat $full\_path 2>/dev/null | /bin/egrep "^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password";

fi;

done;

IFS=$SavedIFS;

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*password[\ \t]+(?:requisite|required)[\ \t]+[^\#&&\S]\*\bpam\_cracklib\.so[\ \t]+[^\#]\*\bocredit=-(\d+)\b.\*

case-sensitive: yes

multiline: yes

grouping 1: Minimum Uppercase Characters >= (Numeric) 1

3.33 Leveraging an Effective pam-stack

3.33.2 Restricting Use of Previous Passwords (Password History)

3.33.2.2 Password Reuse (History)

Description

This test verifies that passwords cannot be reused until after 24 changes have been made.

Preventing users from reusing passwords makes it more difficult for attackers to gain access to the system.

Shell script

directory="/etc/pam.d";

files="passwd";

files=$files$(/bin/cat /etc/pam.d/passwd 2>/dev/null | /bin/awk -F"#" ' $0 ~ /^[[[:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one)]]\*password[[[:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one)]]+include[[[:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one)]].\*/ {print $1}' | /bin/awk 'BEGIN {ORS=";"} {print $3}');

SavedIFS=$IFS;

IFS=";";

for file in $files;

do if [ "`/usr/bin/dirname $file 2>/dev/null`" != "." ];

then if [ -f "$file" ];

then /bin/echo "$file contains:";

/bin/cat $file 2>/dev/null | /bin/egrep "^[[[:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one)]]\*password";

fi;

else full\_path=$directory"/"$file;

fi;

if [ -f "$full\_path" ];

then /bin/echo "$full\_path contains:";

/bin/cat $full\_path 2>/dev/null | /bin/egrep "^[[[:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one)]]\*password";

fi;

done;

IFS=$SavedIFS;

Shell script2

cat /etc/pam.d/common-password

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*password[\ \t]+(?:requisite|required)[\ \t]+(pam\_unix\_passwd\.so|pam\_pwhistory\.so)[\ \t]+use\_authtok[\ \t]+remember=(\d+)

case-sensitive: no

multiline: yes

grouping 2: Password History >= (Numeric) 24

----------------------------------------------------------------------------------------------------

3.33.2.5 Verify /etc/security/opasswd Permissions

Description

This test verifies that permissions of the /etc/security/opasswd file are equal to 600.

It is recommended to allow only 'root' to have read and write access on this file if you choose to prohibit unprivileged users from mounting removable media on the system.

Shell script

cat /etc/security/opasswd

Conditions

If a condition's attribute is missing from an element version, the condition should: Pass

Permissions Matches ^-..-{7}.\*$ AND

User Matches ^root\s\(\d+\)\s\*$ AND

Group Matches ^root\s\(\d+\)\s\*$ AND

----------------------------------------------------------------------------------------------------

3.33.2.6 Verify /var/log/tallylog Permissions

Description

This test verifies that permissions of the /var/log/tallylog file are equal to 600.

It is recommended to allow only 'root' to have read and write access on this file if you choose to prohibit unprivileged users from mounting removable media on the system.

Shell script

ls -la /var/log/tallylog;

ls -la /var/log/faillog;

Conditions

If an element version has no content,

Permissions Matches ^-..-{7}.\*$ AND

User Matches ^root\s\(\d+\)\s\*$ AND

Group Matches ^root\s\(\d+\)\s\*$ AND

============================================================

3.33.3 Locking User Accounts after too many login failures

3.33.3.1 Failed Login Attempts

Description

This test verifies that accounts will be disabled after no more than 3 failed login attempts.

Locking accounts hinders the ability of an attack to use brute-force methods to try to gain access to the system.

(deny > 0 and deny <= 5)

Shell script - PAM common-auth Configuration

directory="/etc/pam.d";

files="passwd";

files=$files$(/bin/cat /etc/pam.d/passwd 2>/dev/null | /bin/awk -F"#" ' $0 ~ /^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]+include[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]].\*/ {print $1}' | /bin/awk 'BEGIN {ORS=";"} {print $3}');

SavedIFS=$IFS;

IFS=";";

for file in $files;

do

if [ "`/usr/bin/dirname $file 2>/dev/null`" !="." ];

then if [ -f "$file" ];

then /bin/echo "$file contains:";

/bin/cat $file 2>/dev/null | /bin/egrep "^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password";

fi;

else full\_path=$directory"/"$file;

fi;

if [ -f "$full\_path" ];

then /bin/echo "$full\_path contains:";

/bin/cat $full\_path 2>/dev/null | /bin/egrep "^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password";

fi;

done;

IFS=$SavedIFS;

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*auth[\ \t]+(?:requisite|required|sufficient)[\ \t]+[^\#&&\S]\*\bpam\_tally2?\.so[\ \t]+[^\#\n]\*\bdeny=(\d+)\b.\*

case-sensitive: yes

multiline: yes

(grouping 1 deny Settings <= (Numeric) 5 AND

grouping 1 deny Settings > (Numeric) 0 ) OR

(grouping 1 deny Settings <= (Numeric) 3 AND

grouping 1 deny Settings > (Numeric) 0 ) OR

---------------------------------------------------------------------------------------------------

3.33.3.2 Verify lock\_time is Set to >= 1800 seconds

Description

This test verifies that lock\_time is set to >= 1800 seconds

Setting the lock\_time=1800 (30 mins) would deny access for 1800 seconds after a failed attempt.

Shell script - PAM common-auth Configuration

directory="/etc/pam.d";

files="passwd";

files=$files$(/bin/cat /etc/pam.d/passwd 2>/dev/null | /bin/awk -F"#" ' $0 ~ /^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]+include[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]].\*/ {print $1}' | /bin/awk 'BEGIN {ORS=";"} {print $3}');

SavedIFS=$IFS;

IFS=";";

for file in $files;

do

if [ "`/usr/bin/dirname $file 2>/dev/null`" !="." ];

then if [ -f "$file" ];

then /bin/echo "$file contains:";

/bin/cat $file 2>/dev/null | /bin/egrep "^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password";

fi;

else full\_path=$directory"/"$file;

fi;

if [ -f "$full\_path" ];

then /bin/echo "$full\_path contains:";

/bin/cat $full\_path 2>/dev/null | /bin/egrep "^[[ [:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one) ]]\*password";

fi;

done;

IFS=$SavedIFS;

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^[\ \t]\*auth[\ \t]+(?:requisite|required|sufficient)[\ \t]+[^\#&&\S]\*\bpam\_tally2?\.so[\ \t]+[^\#\n]\*\bunlock\_time=(\d+)\b.\*

case-sensitive: yes

multiline: yes

grouping 1 lock\_time Setting >= (Numeric) 1800

3.35 Displaying Login Banners

3.35.1 Verify that sshd\_config Contains a Banner for Network Access

Description

This test verifies that the SSH server is configured to display a login banner message when it is accessed.

The presence of a login banner is useful when prosecuting trespassers of the computer system.

Additionally, it can have the effect of obfuscating important operating system information.

Shell script

SSHFileNames=`/usr/bin/find -L /usr/local/etc/sshd\_config /usr/lbin/ssh/sshd\_config /opt/OPENSSH/sshd\_config /opt/ssh/etc/sshd\_config /etc -type f -name sshd\_config 2>/dev/null | /usr/bin/tr "\n" ";"`;

IFS=";";

RealSSHFileNames="";

for SSHPath in $SSHFileNames;

do DirName=$SSHPath;

unset BaseName;

unset LinkedSSHPath;

while [ "/" != "$DirName" ];

do LinkedName=`/bin/ls -dl $DirName | awk -F" -> " '$0 ~ /^l/ && $2 !~ /^[[[:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one)]]\*$/ {print $2}' 2>/dev/null`;

if [ -n "$LinkedDirName" ];

then LinkedSSHPath=$LinkedDirName"/"$BaseName;

break;

else BaseName=`/usr/bin/basename $DirName 2>/dev/null`"/"$BaseName;

DirName=`/usr/bin/dirname $DirName 2>/dev/null`;

fi;

done;

LinkedSSHPath=`/bin/echo $LinkedSSHPath | /bin/sed 's/\/\//\//g' | /bin/sed 's/\/$//g'`;

if [ -n "$LinkedSSHPath" ];

then RealSSHFileNames=$LinkedSSHPath";"$RealSSHFileNames;

else RealSSHFileNames=$SSHPath";"$RealSSHFileNames;

fi;

done;

SSHFileNames=$RealSSHFileNames;

SSHFilesNumbers=`for File in $SSHFileNames;

do /bin/ls -li $File | /bin/awk '{print $1}' 2>/dev/null;

done | /usr/bin/uniq | /usr/bin/wc -l | /bin/sed 's/ //g'`;

IFS=" ";

case $SSHFileNumber in 0 ) /bin/echo "SSHD Config File = Does Not Exist";; 1) SSHFileName =`/bin/echo $SSHFileNames | /bin/awk -F";" '{print $1}' 2>/dev/null`;

BannerFileName=`egrep -i "^[[[:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one)]]\*Banner[[[:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one)]]+" $SSHFileName 2>/dev/null | awk '{print $2}' | head -n 1`;

if [ -f "$BannerFileName" ];

then BannerMessage=`egrep -v "^[[[:space:](onenote:#space&section-id={927497EB-07A6-4914-B732-579B2CCD9AC5}&page-id={B261C6BA-32C2-4648-B10F-E7C259F31F03}&end&base-path=https://unisydneyedu-my.sharepoint.com/personal/blee3344_alumni_sydney_edu_au/Documents/Berlin%20@%20The%20University%20of%20Sydney%20(Students)/Linux.one)]]\*$" $BannerFileName 2>/dev/null`;

if [ -n "$BannerMessage" ];

then /bin/echo "Banner Entry = Exist";

else /bin/echo "Banner Entry = Does Not Exist";

fi;

else /bin/echo "Banner File = Does Not Exist"; fi;; \* ) /bin/echo "SSHD Config File = Have more than 1 file";; esac

Conditions

If an element version has no content, the condition should: Fail

Style: Java-style properties

Key: Banner Entry Equals (String) Exist

--------------------------------------------------------------------------------------------------

3.35.2 Verify Warning Banners in /etc/motd

Description

This test verifies that the a warning banner is present in /etc/motd.

Presenting a statutory warning message prior to the normal user login may assist the prosecution of trespassers on the computer system.

Shell script

cat /etc/motd

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

\S+.\*

case-sensitive: no

multiline: no

grouping 0: Logon Warning Banner Setting Equals (String) This system is restricted to authorized users only

--------------------------------------------------------------------------------------------------

3.35.3 Verify Warning Banners in /etc/issue

Description

This test verifies that the a warning banner is present in /etc/issue.

Presenting a statutory warning message prior to the normal user login may assist the prosecution of trespassers on the computer system.

Shell script

cat /etc/issue

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

\S+.\*

case-sensitive: no

multiline: no

grouping 0: Logon Warning Banner Setting Equals (String) This system is restricted to authorized users only

--------------------------------------------------------------------------------------------------

3.35.4 Verify Warning Banners in /etc/issue.net

Description

This test verifies that the a warning banner is present in /etc/issue.net

Presenting a statutory warning message prior to the normal user login may assist the prosecution of trespassers on the computer system.

Shell script

cat /etc/issue.net

Output

This system is restricted to authorized users only

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

\S+.\*

case-sensitive: no

multiline: no

grouping 0: Logon Warning Banner Setting Equals (String) This system is restricted to authorized users only

3.36 Miscellaneous

3.36.3.2 Verify /etc/ntp.conf >= 2 NTP Servers defined

Description

This test verifies NTP is configured to use 2 NTP pool servers.

Using multiple NTP pool servers is a good practice for failover purposes.

Proper NTP configuration is a Key Factor in maintaining accurate log records used for incident handling.

Shell script

cat /etc/chrony.conf $(egrep '^\s\*include' /etc/chrony.conf 2>/dev/null | awk '{print $2}') /etc/ntp.conf 2>/dev/null | egrep -v '^!|^;|^#|^%' | egrep '^\s\*server|^\s\*pool'

Output

pool <IP> iburst

pool <IP> iburst

pool <2.suse.pool.ntp.org> iburst

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^\s\*pool\s+\d+\.\d+\.\d+\.\d+[\ \t]\*.\*(.\*[\r\n\s\S-]\*)\s\*pool\s+\d+\.\d+\.\d+\.\d+

case-sensitive: no

multiline: yes

grouping 0: NTP Servers Exists (String)

--------------------------------------------------------------------------------------------------

3.36.3.4 Verify Time-servers are local or authoritative sources

Description

This test verifies that the time servers are either local or authoritative time servers.

The presence of a network time protocol server allows for the synchronization of system clocks.

This can be helpful in the correlation of system events when problems occur.

Shell script

cat /etc/chrony.conf $(egrep '^\s\*include' /etc/chrony.conf 2>/dev/null | awk '{print $2}') /etc/ntp.conf 2>/dev/null | egrep -v '^!|^;|^#|^%' | egrep '^\s\*server|^\s\*pool'

Output

pool <IP> iburst

pool <IP> iburst

pool <2.suse.pool.ntp.org> iburst

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

(?s).+

case-sensitive: yes

multiline: yes

grouping 0: Non-local/Non-authoritative Time-Server settings Finds (String) IP1 OR

grouping 0: Non-local/Non-authoritative Time-Server settings Finds (String) IP2 OR

...

grouping 0: Non-local/Non-authoritative Time-Server settings Finds (String) IP\_n OR

--------------------------------------------------------------------------------------------------

3.36.4.2 Check DES password

3.36.4.2.1 Check password in /etc/login.defs SHA512/SHA256

Description

Check password should be either sha256 or sha512 in /etc/login.defs (SLES 15) configuration.

Shell script

cat /etc/login.defs | egrep '^ENCRYPT\_METHOD\s{1,}' | awk -F' ' '{print $2}';

cat /etc/pam.d/common-password-pc | grep sha | egrep -v '^#';

cat /etc/default/passwd | egrep -v '^#' | grep CRYPT\_FILES

Output

SHA512

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| password | sufficient | pam\_unix.so | use\_authtok nullok | sha512 try\_first\_pass |

cat: /etc/default/passwd: No such file or directory

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: yes

grouping 0: SHApassword Contains (String) sha512 OR

grouping 0: SHApassword Contains (String) SHA512 OR

grouping 0: SHApassword Contains (String) sha256 OR

grouping 0: SHApassword Contains (String) SHA256 OR

--------------------------------------------------------------------------------------------------

3.36.4.2 Check DES password

3.36.4.2.2 Check password in /etc/login.defs SHA512/SHA256

Description

Check password should be either sha256 or sha512 in /etc/shadow (current account).

Shell script

cat /etc/shadow | awk -F':' '{print $1,$2}' | awk '{printf "%s:%s;\n", $1, $2}' | egrep -v '!|\\*' | egrep -v '\$5\$|\$6\$' 2>/dev/null

Output

N/A

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: yes

grouping 0: SHApassword Does not exist (String)

============================================================

--------------------------------------------------------------------------------------------------

3.36.4.5 Autologout

Description

Configure the system to automatically logout remote shell session after a period of inactivity.

Minimum of auto-logout should be set to activate after 30 mins of inactivity.

This would reduce the risk of unauthorized access to the system in an unlocked state if it is left unattended.

Shell script

cat /etc/profile.d/autologout.sh

Output

######################################################

# /etc/profile.d/autologout.sh #

# TMOUT = 1800 in seconds, when no application running #

# for 30 minutes, it will logout session automatically #

#####################################################

#TMOUT=1800 i.e. 30 minutes

TMOUT=1800

readonly TMOUT

export TMOUT

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*TMOUT[\ \t]\*=[\ \t]\*(\d+).\*

case-sensitive: yes

multiline: yes

grouping 1: autologout <= (Numeric) 1800

3.37 New requirements

3.37.1 Syslog Log rotate Max Age

/etc/logrotate.d/syslog scheme for /var/log/messages

Description

/etc/logrotate.d/syslog scheme for /var/log/messages

Expected Pass : maxage 365

Shell script

logfile=/var/log/messages;

rotate\_config=/etc/logrotate.d/syslog;

ls -la $rotate\_config;

echo "logfile=$logfile rotate\_config=$rotate\_config";

for i in maxage size rotate;

do var1=$i;

echo $var1=$(cat $rotate\_config | egrep -v '^#' | egrep -A50 $logfile | tr '\n' ' ' | awk -F$var1 '{print $2}' | awk '{print $1}');

done

Output

-rw-r--r-- 1 root root 1079 Jan 12 07:41 /etc/logrotate.d/syslog

logfile=/var/log/messages rotate\_config=/etc/logrotate.d/syslog

maxage=365

size=+4096k

rotate=99

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

maxage=(\d+)

case-sensitive: yes

multiline: yes

grouping 1: Syslog Max Age <= (Numeric) 365

--------------------------------------------------------------------------------

3.37.2 Syslog Log rotate Max Size

/etc/logrotate.d/syslog scheme for /var/log/messages

Description

/etc/logrotate.d/syslog scheme for /var/log/messages

Expected Pass : size >= 4096k

Shell script

logfile=/var/log/messages;

rotate\_config=/etc/logrotate.d/syslog;

echo "config=$(ls -la $rotate\_config)";

echo "logfile=$logfile";

var1=size;

size=$(cat $rotate\_config | egrep -v '^#' | egrep -A50 $logfile | tr '\n' ' ' | awk -F$var1 '{print $2}' | awk '{print $1}');

actual\_size\_byte=$(echo $size | tr 'kmg' 'KMG' | sed -e 's/+/ /g' | sed -e 's/K/\\*1024/g' -e 's/M/\\*1048576/g' -e 's/G/\\*1073741824/g' | bc);

echo "size=$size";

echo "size=$actual\_size\_byte";

Output

config=-rw-r--r-- 1 root root 1079 Jan 12 07:41 /etc/logrotate.d/syslog

logfile=/var/log/messages

size=+4096k

size=4194304

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

size=(\d+)

case-sensitive: yes

multiline: yes

grouping 1: Syslog Max Size <= (Numeric) 4194304 OR

grouping 1: Syslog Max Size Equals (String) +4096k

------------------------------------------------------------------------------

3.37.3 Syslog Log rotate Max Number

Description

"/etc/logrotate.d/syslog scheme for /var/log/messages

Passed : max. no. <= 99

"

Shell script

logfile=/var/log/messages;

rotate\_config=/etc/logrotate.d/syslog;

ls -la $rotate\_config;

echo "logfile=$logfile rotate\_config=$rotate\_config";

for i in maxage size rotate;

do var1=$i;

echo $var1=$(cat $rotate\_config | egrep -v '^#' | egrep -A50 $logfile | tr '\n' ' ' | awk -F$var1 '{print $2}' | awk '{print $1}';

done

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

rotate=(\d+)

case-sensitive: yes

multiline: yes

grouping 1: Syslog Max Age <= (Numeric) 99

---------------------------------------------------------------------------------

3.37.4 Wtmp Log rotate Max Age

Description

"/etc/logrotate.d/wtmp scheme for /var/log/wmtp

expected maxage <= 365

"

Shell script

logfile=/var/log/wtmp;

rotate\_config=/etc/logrotate.d/wtmp;

ls -la $rotate\_config;

echo "logfile=$logfile rotate\_config=$rotate\_config";

for i in maxage size rotate;

do var1=$i;

echo $var1=$(cat $rotate\_config | egrep -v '^#' | egrep -A50 $logfile | tr '\n' ' ' | awk -F$var1 '{print $2}' | awk '{print $1}');

done

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

rotate=(\d+)

case-sensitive: yes

multiline: yes

grouping 1: wtmp Max Age <= (Numeric) 365

-----------------------------------------------------------------------------------

3.37.5 Wtmp Log rotate Max Size

Description

"/etc/logrotate.d/wtmp scheme for /var/log/wtmp

expected maxsize <= 400k

"

Shell script

logfile=/var/log/wtmp;

rotate\_config=/etc/logrotate.d/wtmp;

echo "config=$(ls -la $rotate\_config)";

echo "logfile=$logfile";

var1=size;

size=$(cat $rotate\_config | egrep -v '^#' | egrep -A50 $logfile | tr '\n=' ' ' | awk -F$var1 '{print $2}' | awk '{print $1}');

actual\_size\_byte=$(echo $size | tr 'kmg' 'KMG' | sed -e 's/+/ /g' | sed -e 's/K/\\*1024/g' -e 's/M/\\*1048576/g' -e 's/G/\\*1073741824/g' | bc);

echo "size=$size";

echo "size=$actual\_size\_byte";

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

size=(\d+)

case-sensitive: yes

multiline: yes

grouping 1: wtmp Max Size <= (Numeric) 409600 OR

grouping 1: wtmp Max Size Equals (String) +400k OR

-------------------------------------------------------------------------------------

3.37.6 /etc/logrotate.d/wtmp scheme for /var/log/wtmp

wtmp Log rotate Max Number

"/etc/logrotate.d/wtmp scheme for /var/log/wtmp

max. no. or expected version <= 99

"

Shell script

logfile=/var/log/wtmp;

rotate\_config=/etc/logrotate.d/wtmp;

ls -la $rotate\_config;

echo "logfile=$logfile rotate\_config=$rotate\_config";

for i in maxage size rotate;

do var1=$i;

echo $var1=$(cat $rotate\_config | egrep -v '^#' | egrep -A50 $logfile | tr '\n' ' ' | awk -F$var1 '{print $2}' | awk '{print $1}');

done

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

rotate=(\d+)

case-sensitive: yes

multiline: yes

grouping 1: wtmp Max Number <= (Numeric) 99

----------------------------------------------------------------------------------

3.37.7 Zypp History Log rotate Max Age

Description

/etc/logrotate.d/zypp-history.lr scheme for /var/log/zypp/history -

zypp-history Log rotate Max age

Shell script

logfile=/var/log/zypp/history;

rotate\_config=/etc/logrotate.d/zypp-history.lr;

ls -la $rotate\_config;

echo "logfile=$logfile rotate\_config=$rotate\_config";

for i in maxage size rotate;

do var1=$i;

echo $var1=$(cat $rotate\_config | egrep -v '^#' | egrep -A50 $logfile | tr '\n' ' ' | awk -F$var1 '{print $2}' | awk '{print $1}');

done

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

maxage=(\d+)

case-sensitive: yes

multiline: yes

grouping 1: zypp-history Max <= (Numeric) 1827

---------------------------------------------------------------------------------------

3.37.8 Zypp History Log rotate Max Size

Description

/etc/logrotate.d/zypp-history.lr scheme for /var/log/zypp/history -

Zypp-history Log rotate Max size

"/etc/logrotate.d/zypp-history.lr scheme for /var/log/zypp/history

Failed : maxsize 11M >= expected 10M

"

Shell script

logfile=/var/log/zypp/history;

rotate\_config=/etc/logrotate.d/zypp-history.lr;

echo "config=$(ls -la $rotate\_config)";

echo "logfile=$logfile";

var1=size;

size=$(cat $rotate\_config | egrep -v '^#' | egrep -A50 $logfile | tr '\n' ' ' | awk -F$var1 '{print $2}' | awk '{print $1}');

actual\_size\_byte=$(echo $size | tr 'kmg' 'KMG' | sed -e 's/+/ /g' | sed -e 's/K/\\*1024/g' -e 's/M/\\*1048576/g' -e 's/G/\\*1073741824/g' | bc);

echo "size=$size";

echo "size=$actual\_size\_byte";

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

size=(\d+)

case-sensitive: yes

multiline: yes

grouping 1: zypp Max size <= (Numeric) 10485760 OR

grouping 1: zypp Max size Equals (String) 10M

------------------------------------------------------------------------------------

3.37.9 Zypp History Log rotate Max Number

Description

/etc/logrotate.d/zypp-history.lr scheme for /var/log/zypp/history -

Zypp-history Log rotate Max Number

"/etc/logrotate.d/zypp-history.lr scheme for /var/log/zypp/history

Passed : max. no. or version 99 =< expected 99

"

Shell script

logfile=/var/log/zypp/history;

rotate\_config=/etc/logrotate.d/zypp-history.lr;

ls -la $rotate\_config;

echo "logfile=$logfile rotate\_config=$rotate\_config";

for i in maxage size rotate;

do var1=$i;

echo $var1=$(cat $rotate\_config | egrep -v '^#' | egrep -A50 $logfile | tr '\n' ' ' | awk -F$var1 '{print $2}' | awk '{print $1}');

done

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

rotate=(\d+)

case-sensitive: yes

multiline: yes

grouping 1: zypp Max Number >= (Numeric) 99

-----------------------------------------------------------------------------------

3.37.16 Do NOT allow outsiders to alter routing tables-net.ipv4.conf.all.secure\_redirects

net.ipv4.conf.all.secure\_redirects = 0

Shell script

cat /etc/sysctl.conf | egrep '^net.ipv4.conf.all.secure\_redirects' | awk -F'=' '{print $2}'

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

\d+

case-sensitive: no

multiline: yes

grouping 0: net.ipv4.ip\_forward Equals (Numeric) 0

-------------------------------------------------------------------------------

3.37.17 Do NOT allow outsiders to alter routing tables-net.ipv4.conf.default.secure\_redirects

net.ipv4.conf.default.secure\_redirects = 0

Shell script

cat /etc/sysctl.conf | egrep '^net.ipv4.conf.default.secure\_redirects' | awk -F'=' '{print $2}'

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

\d+

case-sensitive: no

multiline: yes

grouping 0: net.ipv4.conf.default.secure\_redirects Equals (Numeric) 0

------------------------------------------------------------------------------

3.37.19 Ensure only 'root' account has UID0 with full permissions to access the system

Ensure only 'root' account has UID0

Shell script

cat /etc/group | egrep '^root' | awk -F':' '{print $4}'

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*

case-sensitive: yes

multiline: yes

grouping 0: ROOT Does not exist (String)

------------------------------------------------------------------------------

3.37.20 Ensure only 'root' account has UID0 with full permissions to access the system

Ensure only 'root' account has full permissions to access the system.

(no account under root group except root)

Shell script

cat /etc/passwd | awk -F':' '{print $3,$1}' | egrep '^0' | egrep -v '^0 root$'

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*

case-sensitive: yes

multiline: yes

grouping 0: ROOT UID Does not exist (String)

--------------------------------------------------------------------------------

3.37.21 Prevent traffic bypass to other installed NICs

net.ipv4.conf.all.send\_redirects = 0

Shell script

cat /etc/sysctl.conf | egrep '^net.ipv4.conf.all.send\_redirects' | awk -F'=' '{print $2}'

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

\d+

case-sensitive: no

multiline: yes

grouping 0: NIC-net Equals (Numeric) 0

--------------------------------------------------------------------------------

3.37.22 Prevent traffic bypass to other installed NICs

net.ipv4.conf.default.send\_redirects = 0

Shell script

cat /etc/sysctl.conf | egrep '^net.ipv4.conf.default.send\_redirects' | awk -F'=' '{print $2}'

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

\d+

case-sensitive: no

multiline: yes

grouping 0: prevent traffic Equals (Numeric) 0

--------------------------------------------------------------------------------

3.37.23 Prevent traffic bypass to other installed NICs

net.ipv4.ip\_forward = 0

Shell script

cat /etc/sysctl.conf | egrep '^net.ipv4.ip\_forward' | awk -F'=' '{print $2}'

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

\d+

case-sensitive: yes

multiline: yes

grouping 0: net.ipv4.ip\_forward Equals (Numeric) 0

------------------------------------------------------------------------------

3.37.24 Verify /etc/pam.d/common-account Permissions (pam\_unix)

"Grep ``pam common-account`` file output Result: account required pam\_unix2.so"

Shell script

cat /etc/pam.d/common-account | egrep -v '^#';

cat /etc/pam.d/common-account | egrep '^accounts\s{1,}required\s{1,}pam\_unix\*'

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^.\*account\s+(required|requisite)\s+pam\_unix2?\.so.\*$

case-sensitive: yes

multiline: yes

grouping 0: common account Permissions Exists (String)

------------------------------------------------------------------------------

3.37.25 Verify /etc/pam.d/common-account Permissions (pam\_tally\*.so)

account required pam\_tally2.so onerr=fail......

Otherwise, all user password update will fail as pam\_unix cannot write the history to the file.

Shell script

cat /etc/pam.d/common-account | egrep '^account\s{1,}required\s{1,}pam\_tally2{0,1}.so'

Conditions

If an element version has no content, the condition should: Fail

Style: Plain Text (Advanced)

regexp:

^.\*account\s+required\s+pam\_tally2?\.so.\*$

case-sensitive: no

multiline: yes

grouping 0: /etc/pam.d common account Permissions Exists (String)

------------------------------------------------------------------------------

3.37.26 Verify the Chargen service is disabled

Shell script

chkconfig --list chargen 2>/dev/null | grep 3:

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: yes

(grouping 0: Chargen services Contains (String) 3:off AND

grouping 0: Chargen services Contains (String) 5:off ) OR

grouping 0: Chargen services Does not exist (String)

------------------------------------------------------------------------------

3.37.27 Verify the Chargen service is disabled (xinetd)

Shell script

cat /etc/xinetd.d/chargen | egrep '^\s{0,}disable\s{1,}=\s{0,}yes' 2>&1

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: yes

grouping 0: Chargen services Matches (String) ^\s\*disable\s\*=\s\*yes\s\* OR

grouping 0: Chargen services Contains (String) No such file

------------------------------------------------------------------------------

3.27.28 Verify the daytime service is disabled

Shell script

chkconfig --list daytime 2>/dev/null | grep 3:

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: yes

( grouping 0: daytime service Contains (String) 3:off AND

grouping 0: daytime service Contains (String) 5:off ) OR

grouping 0: daytime service Does not exist (String) OR

grouping 0: daytime service Contains (String) No such file

------------------------------------------------------------------------------

3.27.29 Verify the daytime service is disabled (xinetd)

Shell script

cat /etc/xinetd.d/daytime | egrep '^\s{0,}disable\s{1,}=\s{0,}yes' 2>&1

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: yes

grouping 0: Daytime service Matches (String) ^\s\*disable\s\*=\s\*yes\s\* OR

grouping 0: Daytime services Contains (String) No such file

------------------------------------------------------------------------------

3.27.30 Verify the discard service is disabled

Shell script

chkconfig --list discard 2>/dev/null | grep 3:

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: yes

( grouping 0: daytime service Contains (String) 3:off AND

grouping 0: daytime service Contains (String) 5:off ) OR

grouping 0: daytime service Does not exist (String) OR

grouping 0: daytime service Contains (String) No such file

------------------------------------------------------------------------------

3.27.31 Verify the discard service is disabled (xinetd)

Shell script

cat /etc/xinetd.d/discard | egrep '^\s{0,}disable\s{1,}=\s{0,}yes' 2>&1

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: yes

grouping 0: discard service Matches (String) ^\s\*disable\s\*=\s\*yes\s\* OR

grouping 0: discard services Contains (String) No such file OR

grouping 0: discard services Does not exist (String)

------------------------------------------------------------------------------

3.27.32 Verify Echo service is disabled

Shell script

chkconfig --list echo 2>/dev/null | grep 3:

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: no

multiline: yes

( grouping 0: Echo service Contains (String) 3:off AND

grouping 0: Echo service Contains (String) 5:off ) OR

grouping 0: Echo service Does not exist (String) OR

grouping 0: Echo service Contains (String) No such file

------------------------------------------------------------------------------

3.27.33 Verify Echo (ICMPv4) service is disabled

Shell script

cat /etc/xinetd.d/echo | egrep '^\s{0,}disable\s{1,}=\s{0,}yes' 2>&1

Conditions

If an element version has no content, the condition should: Pass

Style: Plain Text (Advanced)

regexp:

.\*[\r\n\s\S-]\*

case-sensitive: yes

multiline: yes

grouping 0: echo service Matches (String) ^\s\*disable\s\*=\s\*yes\s\* OR

grouping 0: echo services Contains (String) No such file OR

grouping 0: echo services Does not exist (String)