

name = "Aperture Science Ubuntu"
title = "Aperture Science"
os = "Ubuntu 22.04"
user = "ratman"
version = "2.1.1"
local = true

[[check]]
message = "Screen timeout policy set to 5 minutes or less"
points = 1

[[check.pass]]
type = 'CommandContainsRegex'
cmd = 'sudo -u ratman gsettings get org.gnome.desktop.session idle-delay'
value = '^uint32 ([1-9][1-9][0-9]1[0-9]{2}|200|2[0-9]{2}|300)\$'

[[check]]
message = "Automatic screen lock enabled"
points = 1

[[check.pass]]
type = 'CommandContains'
cmd = 'sudo -u ratman gsettings get org.gnome.desktop.screensaver lock-enabled'
value = 'true'

[[check]]
message = "The system is configured to automatically check for updates"
points = 1

[[check.pass]]
type = 'AutoCheckUpdatesEnabled'

[[check]]
message = "Added user caroline"
points = 1

[[check.pass]]
type = 'UserExists'
user = 'caroline'

[[check]]
message = "User caroline is a member of aperturestaff and has a valid ssh key configuration"

```
[[check.pass]]  
type = 'UserInGroup'  
user = 'caroline'  
group = 'aperturestaff'
```

```
[[check.pass]]  
type = 'PathExists'  
path = '/home/caroline/.ssh'
```

```
[[check]]  
message = "User cjohnson is an administrator"  
points = 1
```

```
[[check.pass]]  
type = 'UserInGroup'  
user = 'cjohanson'  
group = 'sudo'
```

```
[[check.pass]]  
type = 'UserInGroup'  
user = 'cjohanson'  
group = 'adm'
```

```
[[check]]  
message = 'Added chell to group testsubjects'  
points = 1
```

```
[[check.pass]]  
type = 'UserInGroup'  
user = 'chell'  
group = 'testsubjects'
```

```
[[check]]  
message = 'Disabled automatic login for ratman'  
points = 1
```

```
[[check.pass]]  
type = 'FileContains'  
path = '/etc/gdm3/custom.conf'  
value = 'AutomaticLoginEnable=False'
```

```
[[check]]  
message = 'Removed unauthorized user wheatley'
```

points = 1

```
[[check.pass]]  
type = 'UserExistsNot'  
user = 'wheatley'
```

```
[[check]]  
message = 'User chell is not an administrator'  
points = 1
```

```
[[check.pass]]  
type = 'UserInGroupNot'  
user = 'chell'  
group = 'sudo'
```

```
[[check.pass]]  
type = 'UserInGroupNot'  
user = 'chell'  
group = 'adm'
```

```
[[check]]  
message = 'Changed insecure password for companioncube'  
points = 1
```

```
[[check.pass]]  
type = 'PasswordChanged'  
user = 'companioncube'  
value = '$y$9T$KSdv3iaCthqz.nxfJU5/f1$3BISl4dvR3r1le/bDhYpPfjMb14z/K5.ZdXLLa0JO23'
```

```
[[check]]  
message = 'The root account is locked'  
points = 1
```

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/shadow'  
value = '^root:[*!]'
```

```
[[check]]  
message = 'Prohibited MP3 files removed'  
points = 1
```

```
[[check.pass]]
```

```
type = 'PathExistsNot'  
path = '/home/cjohnson/Music/radio.mp3'
```

```
[[check.pass]]  
type = 'PathExistsNot'  
path = '/home/spacecore/Music/spaaaaaace.mp3'
```

```
[[check]]  
message = 'Uncomplicated Firewall (UFW) protection is enabled'  
points = 1
```

```
[[check.pass]]  
type = 'FirewallUp'
```

```
[[check]]  
message = 'NFS has been disabled or removed'  
points = 1
```

```
[[check.pass]]  
type = 'ServiceUpNot'  
name = 'nfs-blkmap'
```

```
[[check.pass]]  
type = 'ServiceUpNot'  
name = 'nfs-idmapd'
```

```
[[check.pass]]  
type = 'ServiceUpNot'  
name = 'nfs-mountd'
```

```
[[check.pass]]  
type = 'ServiceUpNot'  
name = 'nfsdclld'
```

```
[[check.pass]]  
type = 'ServiceUpNot'  
name = 'nfs-server'
```

```
[[check.pass]]  
type = 'ServiceUpNot'  
name = 'nfs-kernel-server'
```

```
[[check]]
```

message = 'Nginx has been disabled or removed'
points = 1

[[check.pass]]
type = 'ServiceUpNot'
name = 'nginx'

[[check]]
message = 'Apache service is started'
points = 1

[[check.pass]]
type = 'ServiceUp'
name = 'apache2'

[[check]]
message = 'The UFW application profile for Apache has been configured as "Apache Secure"'
points = 1

[[check.pass]]
type = 'CommandContainsRegex'
cmd = 'ufw status'
value = 'Apache Secure *ALLOW *Anywhere'

[[check]]
message = 'SSH server is started'
points = 1

[[check.pass]]
type = 'ServiceUp'
name = 'ssh'

[[check]]
message = 'Removed insecure sudoers rule'
points = 1

[[check.pass]]
type = 'FileContainsRegexNot'
path = '/etc/sudoers'
value = '^\\s*gman\\s+ALL=\\(ALL:ALL\\) NOPASSWD: ALL'

[[check]]
message = 'IPv4 TIME-WAIT ASSASSINATION protection enabled'

points = 1

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/sysctl.conf'
value = '^\\s*net\\.ipv4\\.tcp_rfc1337\\s*=\\s*1\\s*$'
```

```
[[check]]
message = 'IPv4 TCP SYN cookies are enabled'
points = 1
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/sysctl.conf'
value = '^\\s*net\\.ipv4\\.tcp_syncookies\\s*=\\s*1\\s*$'
```

```
[[check]]
message = 'IPv4 forwarding has been disabled'
points = 1
```

```
[[check.pass]]
type = 'FileContainsRegexNot'
path = '/etc/sysctl.conf'
value = '^\\s*net\\.ipv4\\.ip_forward\\s*=\\s*1\\s*$'
```

```
[[check]]
message = 'Source routing is disabled'
points = 1
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/sysctl.conf'
value = '^\\s*net\\.ipv4\\.conf\\.all\\.accept_source_route\\s*=\\s*0\\s*$'
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/sysctl.conf'
value = '^\\s*net\\.ipv4\\.conf\\.default\\.accept_source_route\\s*=\\s*0\\s*$'
```

```
[[check]]
message = 'Send redirects are ignored'
points = 1
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/sysctl.conf'
value = '^\\s*net\\.ipv4\\.conf\\.all\\.send_redirects\\s*=\\s*0\\s*$'
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/sysctl.conf'
value = '^\\s*net\\.ipv4\\.conf\\.default\\.send_redirects\\s*=\\s*0\\s*$'
```

```
[[check]]
message = 'Martian packet logging is enabled'
points = 1
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/sysctl.conf'
value = '^\\s*net\\.ipv4\\.conf\\.all\\.log_martians\\s*=\\s*1\\s*$'
```

```
[[check]]
message = 'Linux kernel has been updated'
points = 1
```

```
[[check.pass]]
type = 'KernelVersionNot'
value = '5.19.0-46-generic'
```

```
[[check]]
message = 'Source address verification enabled'
points = 1
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/sysctl.conf'
value = '^\\s*net\\.ipv4\\.conf\\.default\\.rp_filter\\s*=\\s*1\\s*$'
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/sysctl.conf'
value = '^\\s*net\\.ipv4\\.conf\\.all\\.rp_filter\\s*=\\s*1\\s*$'
```

```
[[check]]
message = 'ICMP redirects are ignored'
```

points = 1

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/sysctl.conf'  
value = '^\\s*net\\.ipv4\\.conf\\.all\\.accept_redirects\\s*=\\s*0\\s*$'
```

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/sysctl.conf'  
value = '^\\s*net\\.ipv4\\.conf\\.default\\.accept_redirects\\s*=\\s*0\\s*$'
```

```
[[check]]  
message = "Prohibited software ophcrack has been removed"  
points = 1
```

```
[[check.pass]]  
type = "ProgramInstalledNot"  
name = "ophcrack"
```

```
[[check]]  
message = "Prohibited software hydra has been removed"  
points = 1
```

```
[[check.pass]]  
type = "ProgramInstalledNot"  
name = "hydra"
```

```
[[check]]  
message = "Prohibited software john has been removed"  
points = 1
```

```
[[check.pass]]  
type = "ProgramInstalledNot"  
name = "john"
```

```
[[check]]  
message = "Prohibited software nmap has been removed"  
points = 1
```

```
[[check.pass]]  
type = "ProgramInstalledNot"  
name = "nmap"
```

```
[[check]]  
message = "Prohibited software snort has been removed"  
points = 1
```

```
[[check.pass]]  
type = "ProgramInstalledNot"  
name = "snort"
```

```
[[check]]  
message = "Prohibited software wireshark has been removed"  
points = 1
```

```
[[check.pass]]  
type = "ProgramInstalledNot"  
name = "wireshark"
```

```
[[check]]  
message = 'Block dangerous and deceptive content enabled in Firefox'  
points = 1
```

```
[[check.pass]]  
type = 'DirContainsRegexNot'  
path = '/home/ratman/.mozilla/firefox'  
value = 'user_pref(\"browser\\.safebrowsing\\.malware\\.enabled\",\\s*false\\);'
```

```
[[check.pass]]  
type = 'DirContainsRegexNot'  
path = '/home/ratman/.mozilla/firefox'  
value = 'user_pref(\"browser\\.safebrowsing\\.phishing\\.enabled\",\\s*false\\);'
```

```
[[check]]  
message = 'SSH root login has been disabled'  
points = 1
```

```
[[check.pass]]  
type = 'FileContainsRegexNot'  
path = '/etc/ssh/sshd_config'  
value = '^\\s*PermitRootLogin\\s+yes\\s*$'
```

```
[[check]]  
message = "Malicious script 'sabotage' removed"  
points = 1
```

```
[[check.pass]]
type = 'PathExistsNot'
path = '/lib/.core/sabotage'
```

```
[[check]]
message = 'Forensic Question 1 correct'
points = 1
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/home/ratman/Desktop/Forensic Question 1.txt'
value = 'ANSWER:\s*Wh34tl3YRuL3s'
```

```
[[check]]
message = 'Forensic Question 2 correct'
points = 1
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/home/ratman/Desktop/Forensic Question 2.txt'
value = 'ANSWER:\s*THE CAKE IS A LIE!'
```

```
[[check]]
message = 'Forensic Question 4 correct'
points = 1
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/home/ratman/Desktop/Forensic Question 4.txt'
value = 'ANSWER:\s*hOXrgDDLpYw'
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/home/ratman/Desktop/Forensic Question 4.txt'
value = 'ANSWER:\s*I like to eat cake while excruciating\.'
```

```
[[check]]
message = 'Forensic Question 5 correct'
points = 1
```

```
[[check.passoverride]]
type = 'FileContainsRegex'
```

```
path = '/home/ratman/Desktop/Forensic Question 5.txt'
```

```
value = 'ANSWER:\s*The Aperture Science Handheld Portal Device, also known as the  
\"portal gun,\" is an experimental tool used to create two portals through which objects can pass  
from any distance, instantaneously\'. It is powered by a miniature black hole, which is kept stable  
through the use of a cooling fan at the rear of the device\'. Should the black hole diminish, two  
miniature German stick grenades, stored within the casing, can be used to restart it, by screwing  
them into two small holes in the device\'. Should the fan fail, it is recommended to open the top  
compartment of the gun to retrieve the Event Horizon Estimation Wheel, which will determine  
the minimum safe distance to survive the resulting overload\'. The gun also has a zero-point  
energy field manipulator that can pick up objects, but only those directly in front of it\'. If the  
space between the held object and the portal device is interrupted, which may occur when  
traveling through portals at high velocity and the object accidentally hits the edge of a  
portal/another object in the surrounding area, the object will detach\'. Our current data shows  
that portal surfaces cannot be drawn on certain tiled floors, metal, doors or windows, and  
miscellaneous props\'. Whenever a portal is shot at an incompatible surface, the ray will bounce  
off and form blue or orange particles that shortly disappear\'. However, it will work on security  
cameras, detaching them and making them fall on the ground\'. Portals can be created on the  
white tiles found in many test chambers or surfaces covered in conversion gel\'. And per the  
Wheatley incident, portals can also be formed on the moon itself\.'
```

```
[[check.passoverride]]
```

```
type = 'FileContainsRegex'
```

```
path = '/home/ratman/Desktop/Forensic Question 5.txt'
```

```
value = 'ANSWER:\s*The Aperture Science Handheld Portal Device, also known as the  
\"portal gun,\" is an experimental tool used to create two portals through which objects can pass  
from any distance, instantaneously\'. It is powered by a miniature black hole, which is kept stable  
through the use of a cooling fan at the rear of the device\'. Should the black hole diminish, two  
miniature German stick grenades, stored within the casing, can be used to restart it, by screwing  
them into two small holes in the device\'. Should the fan fail, it is recommended to open the top  
compartment of the gun to retrieve the Event Horizon Estimation Wheel, which will determine  
the minimum safe distance to survive the resulting overload\'. The gun also has a zero-point  
energy field manipulator that can pick up objects, but only those directly in front of it\'. If the  
space between the held object and the portal device is interrupted, which may occur when  
traveling through portals at high velocity and the object accidentally hits the edge of a  
portal/another object in the surrounding area, the object will detach\'. Our current data shows  
that portal surfaces cannot be drawn on certain tiled floors, metal, doors or windows, and  
miscellaneous props\'. Whenever a portal is shot at an incompatible surface, the ray will bounce  
off and form blue or orange particles that shortly disappear\'. However, it will work on security  
cameras, detaching them and making them fall on the ground\'. Portals can be created on the  
white tiles found in many test chambers or surfaces covered in conversion gel\'. And per the  
Wheatley incident, portals can also be formed on the moon itself\.'
```

```
[[check]]
```

message = 'Malicious PAM backdoor removed'
points = 1

```
[[check.pass]]  
type = 'FileContainsRegexNot'  
path = '/etc/pam.d/common-auth'  
value =  
'^\s*auth\s+sufficient\s+pam_exec\.so\s+expose_authtok\s+quiet\s+Vlib/x86_64-linux-gnu/secu  
rity/pam-bd\s*$'
```

[[check]]
message = 'Account lockout policy configured'
points = 1

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/pam.d/common-auth'  
value = '^\s*auth\s+\[default=die\]\s+pam_faillock\.so\s+authfail\s*$'
```

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/pam.d/common-auth'  
value = '^\s*auth\s+sufficient\s+pam_faillock\.so\s+authsucc\s*$'
```

[[check]]
message = 'A secure maximum password age has been set'
points = 1

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/login.defs'  
value = '^PASS_MAX_DAYS\s+(?:[1-9]|[12][0-9]|30)\s*$'
```

[[check]]
message = 'A secure minimum password age has been set'
points = 1

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/login.defs'  
value = '^PASS_MIN_DAYS\s*(?:[5-9]|[1-9][0-9]|100)\s*$'
```

[[check]]

message = 'A minimum password length has been set'
points = 1

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/security/pwquality.conf'  
value = '^minlen\s*=\s*(?:[89][1-9][0-9]{1,2}|1000)'
```

[[check]]
message = 'Password credit complexity checks added'
points = 1

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/security/pwquality.conf'  
value = 'ucredit\s*=\s*-\d+'
```

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/security/pwquality.conf'  
value = 'lcredit\s*=\s*-\d+'
```

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/security/pwquality.conf'  
value = 'ocredit\s*=\s*-\d+'
```

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/security/pwquality.conf'  
value = 'dcredit\s*=\s*-\d+'
```

[[check]]
message = 'Password dictionary check enabled'
points = 1

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/security/pwquality.conf'  
value = 'dictcheck\s*=\s*-[1-9]\d*'
```

[[check]]
message = 'Password username check enabled'

points = 1

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/security/pwquality.conf'
value = 'usercheck\s*=\s*-[1-9]\d*
```

```
[[check]]
message = 'Password encryption method set to SHA512'
points = 1
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/login.defs'
value = 'ENCRYPT_METHOD\s*SHA512'
```

```
[[check]]
message = 'A secure number of login retries is configured'
points = 1
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/login.defs'
value = 'LOGIN_RETRIES\s+[1-5]'
```

```
[[check]]
message = 'A secure password history policy is configured'
points = 1
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/pam.d/common-password'
value = 'pam_unix\.so.*?remember=(?:[5-9][1-9][0-9]{1,2}|1000)'
```

```
[[check]]
message = 'Removed unauthorized program Google Chrome'
points = 1
```

```
[[check.pass]]
type = 'ProgramInstalledNot'
name = 'google-chrome-stable'
```

```
[[check]]
```

message = 'GRUB signature checks enabled'
points = 1

```
[[check.pass]]  
type = 'FileContains'  
path = '/etc/grub.d/40_custom'  
value = 'set check_signatures=enforce'
```

```
[[check.pass]]  
type = 'FileContains'  
path = '/etc/grub.d/40_custom'  
value = 'export check_signatures'
```

[[check]]
message = 'Unauthorized superuser wheatley removed from GRUB'
points = 1

```
[[check.pass]]  
type = 'FileContains'  
path = '/etc/grub.d/40_custom'  
value = 'set superusers="root"'
```

```
[[check.pass]]  
type = 'FileContainsNot'  
path = '/etc/grub.d/40_custom'  
value = 'password wheatley EVIL'
```

[[check]]
message = '/etc/shadow is not world-readable'
points = 1

```
[[check.pass]]  
type = 'PermissionIsNot'  
path = '/etc/shadow'  
value = '???????r??'
```

[[check]]
message = '/etc/passwd is not world-writable'
points = 1

```
[[check.pass]]  
type = 'PermissionIsNot'  
path = '/etc/passwd'
```

value = '???????w?'

[[check]]

message = 'cp is not a SUID binary'

points = 1

[[check.pass]]

type = 'PermissionIsNot'

path = '/usr/bin/cp'

value = '???s??????'

[[check]]

message = 'Removed insecure /etc/hosts entries'

points = 1

[[check.pass]]

type = 'FileContainsRegexNot'

path = '/etc/hosts'

value = '^s*127\.\0\.\0\.\1s+google\.\com\s*\$'

[[check.pass]]

type = 'FileContainsRegexNot'

path = '/etc/hosts'

value = '^s*127\.\0\.\0\.\1s+duckduckgo\.\com\s*\$'

[[check]]

message = 'Password authentication disabled in SSH'

points = 1

[[check.pass]]

type = 'FileContains'

path = '/etc/ssh/sshd_config'

value = '^PasswordAuthentication\s*no\$'

[[check]]

message = 'SSH port set to 1382'

points = 1

[[check.pass]]

type = 'FileContainsRegex'

path = '/etc/ssh/sshd_config'

value = '^Port\s*1382\$'

```
[[check]]
message = 'SSH disabled for all members of group testsubjects'
points = 1
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/ssh/sshd_config'
value = '^DenyGroups\s*testsubjects$'
```

```
[[check]]
message = 'Apache response header set to prod'
points = 1
```

```
[[check.pass]]
type = 'FileContainsRegexNot'
path = '/etc/apache2/conf-enabled/security.conf'
value = '^ServerTokens\s*Full$'
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/apache2/conf-enabled/security.conf'
value = '^ServerTokens\s*Prod$'
```

```
[[check]]
message = 'Apache server signature disabled'
points = 1
```

```
[[check.pass]]
type = 'FileContainsRegexNot'
path = '/etc/apache2/conf-enabled/security.conf'
value = '^ServerSignature\s*On$'
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/apache2/conf-enabled/security.conf'
value = '^ServerSignature\s*Off$'
```

```
[[check]]
message = 'SSH server UFW application profile is enabled'
points = 1
```

```
[[check.pass]]
```

```
type = 'CommandContainsRegex'  
cmd = 'ufw status'  
value = 'OpenSSH *ALLOW *Anywhere'
```

```
[[check]]  
message = 'IPv6 has been disabled'  
points = 1
```

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/sysctl.conf'  
value = '^net.ipv6.conf.all.disable_ipv6\s*=\s*1$'
```

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/sysctl.conf'  
value = '^net.ipv6.conf.default.disable_ipv6\s*=\s*1$'
```

```
[[check]]  
message = 'Malicious configuration file 99autotrust removed from apt'  
points = 1
```

```
[[check.pass]]  
type = 'PathExistsNot'  
path = '/etc/apt/apt.conf.d/99autotrust'
```

```
[[check]]  
message = 'FTP server is started'  
points = 1
```

```
[[check.pass]]  
type = 'ServiceUp'  
name = 'vsftpd'
```

```
[[check]]  
message = 'FTP server UFW application profile is enabled'  
points = 1
```

```
[[check.pass]]  
type = 'CommandContainsRegex'  
cmd = 'ufw status'  
value = 'vsftpd *ALLOW *Anywhere'
```

[[check]]

message = 'Unauthorized PPA persepolis removed from apt'

points = 1

[[check.pass]]

type = 'PathExistsNot'

path = '/etc/apt/sources.list.d/persepolis-ubuntu-ppa-jammy.list'

[[check]]

message = 'Anonymous FTP disabled'

points = 1

[[check.pass]]

type = 'FileContainsRegex'

path = '/etc/vsftpd.conf'

value = '^anonymous_enable\s*=\s*NO\$'

[[check.pass]]

type = 'FileContainsRegexNot'

path = '/etc/vsftpd.conf'

value = '^anonymous_enable\s*=\s*YES\$'

[[check]]

message = 'UFW denies incoming connections by default'

points = 1

[[check.pass]]

type = 'CommandContains'

cmd = 'ufw status verbose'

value = 'deny (incoming)'

[[check]]

message = 'Test subject turret is chrooted in FTP'

points = 1

[[check.pass]]

type = 'FileContainsRegexNot'

path = '/etc/vsftpd.chroot_list'

value = '^turret\$'

[[check]]

message = 'Unnecessary daemon CUPS no longer running'

points = 1

```
[[check.pass]]  
type = 'ServiceUpNot'  
name = 'cups'
```

```
[[check.pass]]  
type = 'ServiceUpNot'  
name = 'cups-browsed'
```

```
[[check]]  
message = 'Unnecessary daemon Avahi no longer running'  
points = 1
```

```
[[check.pass]]  
type = 'ServiceUpNot'  
name = 'avahi-daemon'
```

```
[[check]]  
message = 'FTP server configured to only use TLS'  
points = 1
```

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/vsftpd.conf'  
value = '^ssl_enable=YES$'
```

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/vsftpd.conf'  
value = '^ssl_tlsv1=YES$'
```

```
[[check.pass]]  
type = 'FileContainsRegexNot'  
path = '/etc/vsftpd.conf'  
value = '^ssl_sslv2=YES$'
```

```
[[check.pass]]  
type = 'FileContainsRegexNot'  
path = '/etc/vsftpd.conf'  
value = '^ssl_sslv3=YES$'
```

```
[[check]]  
message = "GLaDOS's home directory is not world-readable by testsubjects"
```

points = 1

```
[[check.pass]]
type = 'CommandContains'
cmd = 'getfacl /home/glados'
value = 'group:testsubjects:---'
```

```
[[check]]
message = 'The employees directory of the Apache server is not world-readable by Black Mesa'
points = 1
```

```
[[check.pass]]
type = 'CommandContains'
cmd = 'getfacl /var/www/aperture-apache-server/employees/'
value = 'group:blackmesa:---'
```

```
[[check]]
message = "Anonymous TLS/SSL has been disabled"
points = 1
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/vsftpd.conf'
value = '^ssl_enable=YES$'
```

```
[[check.pass]]
type = 'FileContainsRegexNot'
path = '/etc/vsftpd.conf'
value = '^allow_anon_ssl=YES$'
```

```
[[check]]
message = "A passive port range of 50000-50100 has been configured"
points = 1
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/vsftpd.conf'
value = '^pasv_min_port=50000$'
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/vsftpd.conf'
value = '^pasv_max_port=50100$'
```

```
[[check.pass]]  
type = 'CommandContainsRegex'  
cmd = 'sudo ufw status'  
value = '50000:50100/tcp *ALLOW *Anywhere'
```

```
[[check]]  
message = 'gdm3 does not run under rick'  
points = 1
```

```
[[check.pass]]  
type = 'FileContainsRegexNot'  
path = '/etc/gdm3/custom.conf'  
value = '^User=rick$'
```

```
[[check.pass]]  
type = 'FileContainsRegexNot'  
path = '/etc/gdm3/custom.conf'  
value = '^Group=rick$'
```

```
[[check]]  
message = 'gdm3 disallows TCP connections'  
points = 1
```

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/gdm3/custom.conf'  
value = '^DisallowTCP=true$'
```

```
[[check.pass]]  
type = 'FileContainsRegexNot'  
path = '/etc/gdm3/custom.conf'  
value = '^DisallowTCP=false$'
```

```
[[check]]  
message = 'System core dumps are disabled'  
points = 1
```

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/sysctl.conf'  
value = '^fs.suid_dumpable\s*=\s*0$'
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/etc/security/limits.conf'
value = '^*\s*hard\s*core\s*0$'
```

```
[[check]]
message = 'Service AppArmor is running'
points = 1
```

```
[[check.pass]]
type = 'ServiceUp'
name = 'apparmor'
```

```
[[check]]
message = 'Firefox is up to date'
points = 1
```

```
[[check.pass]]
type = 'CommandContainsNot'
cmd = 'sudo -u ratman firefox -v'
value = '115.0.2'
```

```
[[check]]
message = 'Keylogger kernel module kisni is unloaded'
points = 1
```

```
[[check.pass]]
type = 'CommandContainsNot'
cmd = 'ls /lib/modules/$(uname -r)/updates/dkms'
value = 'kisni.ko'
```

```
[[check]]
message = 'Kernel lockdown is enabled'
points = 1
```

```
[[check.passoverride]]
type = 'FileContainsRegex'
path = '/sys/kernel/security/lockdown'
value = '\[integrity\]'
```

```
[[check.passoverride]]
type = 'FileContainsRegex'
path = '/sys/kernel/security/lockdown'
```

value = '[confidentiality]'

[[check]]

message = 'Bind shell systemd service r00t has been removed or disabled'

points = 1

[[check.pass]]

type = 'ServiceUpNot'

name = 'r00t'

[[check]]

message = 'Auditd logs local events'

points = 1

[[check.pass]]

type = 'FileContainsRegex'

path = '/etc/audit/auditd.conf'

value = '^local_events\s*=\s*yes\$'

[[check]]

message = 'Removed maliicious bash alias GOODBYE'

points = 1

[[check.pass]]

type = 'FileContainsNot'

path = '/etc/bash.bashrc'

value = 'alias GOODBYE='

[[check]]

message = 'A hard limit of 2500 processes or less has been configured'

points = 1

[[check.pass]]

type = 'FileContainsRegex'

path = '/etc/security/limits.conf'

value = '^*\s*hard\s*nproc\s*(?:[1-9]|[1-9][0-9]{1,2}|1[0-9]{3}|2[0-4][0-9]{2}|2500)\$'

[[check]]

message = 'System account syslog has a nologin or false shell'

points = 1

[[check.pass]]

type = 'FileContainsRegex'

```
path = '/etc/passwd'  
value = '^syslog:[^:]*:[^:]*:[^:]*:[^:]*:[^:]*:(VusrVsbinVnologin|VbinVfalse)$'
```

```
[[check]]  
message = '/home is owned by root and not spacecore'  
points = 1
```

```
[[check.pass]]  
type = 'FileOwner'  
path = '/home'  
name = 'root'
```

```
[[check]]  
message = '/swapfile is only readable and writable by root'  
points = 1
```

```
[[check.pass]]  
type = 'Permissions'  
path = '/swapfile'  
value = '-rw?-----'
```

```
[[check]]  
message = 'Kernel address space layout randomization (ASLR) is enabled'  
points = 1
```

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/sysctl.conf'  
value = '^\\s*kernel\\.randomize_va_space\\s*=\\s*[12]\\s*$'
```

```
[[check]]  
message = 'Kernel ExecShield enabled'  
points = 1
```

```
[[check.pass]]  
type = 'FileContainsRegex'  
path = '/etc/sysctl.conf'  
value = '^\\s*kernel\\.exec-shield\\s*=\\s*1\\s*$'
```

```
[[check]]  
message = 'Forensic Question 3 correct'  
points = 1
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/home/ratman/Desktop/Forensic Question 3.txt'
value = '^\\s*ANSWER:\\s*0x13d9\\s*$'
```

```
[[check.pass]]
type = 'FileContainsRegex'
path = '/home/ratman/Desktop/Forensic Question 3.txt'
value = '^\\s*ANSWER:\\s*0x13f4\\s*$'
```

```
[[check]]
message = 'Removed LD_PRELOAD rootkit EVIL_RABBIT'
points = 1
```

```
[[check.pass]]
type = 'PathExistsNot'
path = '/usr/src/evil_rabbit.so'
```

```
[[check]]
message = 'Removed malicious script victim in /usr/sbin'
points = 1
```

```
[[check.pass]]
type = 'PathExistsNot'
path = '/usr/sbin/victim'
```