Seguridad y Auditoría Informática

Auditoría de Sistemas Unix

Objetivos

- Definiendo Estándares para Copos de Nieve
- Scripting Básico
- Auditando Unix

Enfoque

- Buenos Controles de Configuración significan
  - Sistemas Seguros
  - Facilidad para Auditar
    - Toda la investigación está hecha por nosotros.
    - Ir de una Certificación de Seguridad a Auditoría es simple.

Especialmente con Unix

- Copos de Nieve
  - Cada uno es único
  - Sistemas únicos son como copos de nieve
- Los sistemas Unix tienden a ser copos de nieve
  - Habitualmente servidores
  - Poca gestión y administración descentralizada

Referencias de Listas de Control

- https://www.cisecurity.org/
  - Benchmarks de Windows, Unix, Cisco, Oracle, etc.
- https://public.cyber.mil/
  - Guías de Seguridad/Configuraciones
- https://www.nsa.gov/
  - Guías de Seguridad/Configuraciones
- https://www.giac.org/certification/systems-network-auditor-gsna
  - Prácticas de Auditoría de Sistemas y Redes
- https://www.giac.org/certification/certified-unix-security-administrator-gcux
  - Prácticas de Seguridad de Sistemas y Aplicaciones
- https://www.sans.org/score
  - Consenso de seguridad para la evaluación de la preparación operativa

Creación de Lista de Control

- Identificar una Fuente de Buenas Prácticas
  - Certificación Interna
  - Formularios
  - Listas de Control de Seguridad
  - Políticas
- Identificar Objetivos y buscar Controles
  - Considerar el uso de una especie de “Permiso de Edificación”

Scripting Básico

Scripting

- No es tan difícil como suena!
  - Esencialmente, scripts “Batch” para Unix
  - Muchas formas de conseguirlo
    - Perl
    - Python
    - Bash
    - Cshell

Conceptos Básicos de Scripting

- Veremos “Shell Scripting” usando Bash (o sh)
  - Bash = Bourne Again Shell
    - Versión gratuita de Bourne Shell
    - Tiene las mismas características
    - La programación es idéntica

ABC del Scripting

- Simplemente se pueden concatenar comandos como una secuencia de comandos por lote

```
#!/bin/sh
ls /etc > /tmp/resultados_auditoria
ps -xa >> /tmp/resultados_auditoria
find / -perm 04000 >> /tmp/resultados_auditoria
last >> /tmp/resultados_auditoria
```

Variables

- También pueden usarse “Variables”
  - Ideales para simplificar y generalizar

```
#!/bin/sh
RESULTADOS_AUDITORIA=/tmp/resultados_auditoria
ls /etc > $RESULTADOS_AUDITORIA
ps -xa >> $RESULTADOS_AUDITORIA
find / -perm 04000 >> $RESULTADOS_AUDITORIA
last >> $RESULTADOS_AUDITORIA
```

Echo

- Se pueden agregar comentarios a la salida con “echo”

```
#!/bin/sh
RESULTADOS_AUDITORIA=/tmp/resultados_auditoria
echo Resultados de Auditoría >> $RESULTADOS_AUDITORIA
ls /etc > $RESULTADOS_AUDITORIA
echo ---------------------- >> $RESULTADOS_AUDITORIA
ps -xa >> $RESULTADOS_AUDITORIA
```

If/Then y Corchetes

- Es posible probar condiciones
  - Quizás queremos comparar resultados y reportar variaciones

```
#!/bin/sh
netstat -an > /tmp/netstat.obs
diff netstat.base /tmp/netstat.obs > /tmp/ns.diff
if [ -s /tmp/ns.diff ];
      then mail administrador@sitio.com < /tmp/ns.diff
fi
```

Test

- ‘test’ y los corchetes son equivalentes
  - if test -z filename; then ls; fi
  - If [-z filename]; then ls; fi

Tests

-b	Dispositivo de Bloques	-O	Propiedad de EUID
-c	Dispositivo de Caracteres	-p	Es una tubería FIFO
-d	Directorio	-r	Archivo legible
-e	Existe	-s	Archivo no vacío
-f	Archivo Normal	-S	Es un socket
-g	Set GID está configurado	-t	Es una terminal
-G	Propiedad de EGID	-u	Set UID está configurado
-k	Sticky está seteado	-w	Archivo escribible
-L	Enlace simbólico	-x	Bit de ejecución está configurado
-n	Cadena no nula	-z	Cadena vacía

Otras Verificaciones Útiles

- A -nt B Si el archivo A es más nuevo que B
- A -ot B Si el archivo A es más antiguo que B
- A -ef B Si el archivo A está enlazado con B
- A = B Si la cadena A es igual a la cadena B
- A -eq B Si la expr. A es igual a la expr. B
  - -gt, -le, -ge, -lt, -ne

Argumentos de Línea de Comandos

- Permiten la generalización
  - Quizás especificar el archivo de salida para los resultados.

```
#!/bin/sh
if [ -z $1 ]; then
        echo Debe especificar un archivo de salida!
        exit 1
fi
echo Enviando resultados a: $1
```

```
$./audit_script /tmp/results
```

Aceptando Entrada

- Permite una auditoría repetible pero personalizada

```
#!/bin/sh
echo -n ¿Chequear puertos abiertos [s/n]?
read RESPUESTA
if [ $RESPUESTA = “s”  ] || [ $RESPUESTA = “si”  ]; then
        netstat -an > /tmp/resultados_auditoria
fi
```

¿Por qué usar Scripts?

- Simplifica tareas repetitivas
  - La auditoría es conducida exactamente de la misma manera cada vez
  - Resultados e informes pueden ser automatizados
  - Simplifica el Análisis

Ejemplo

- ¿Qué hace ésto y por qué es útil?

```
#!/bin/sh
...
echo Test 3.6: Localizar todos los archivos SUID usando find.
echo Esta es una prueba segura
echo -n ¿Realizar prueba [s/n]?
read RESPUESTA
if [ $RESPUESTA = “s”  ] || [ $RESPUESTA = “si”  ]; then
        find / -perm 0400 -type f > archivos_SUID.txt
fi
```

Otras Herramientas Útiles de Scripting

- Utilidades independientes para rebanar y cortar
  - Grep / Egrep
  - Cut
  - Sed
  - Awk

Grep/Egrep

- Estos días es mejor usar egrep
  - Grep (**G**et **R**egular **E**x**P**ression)
    - Original
    - Soporte limitado de Expresiones Regulares
  - Egrep
    - “Extendido”
    - Soporte completo de Expresiones Regulares
    - Se recomienda usar *egrep* exclusivamente
    - Considerar poner alias de egrep a grep

Expresiones Regulares

- Usar “metacaracteres” para describir lo que quiere encontrar
  - Por ejemplo un “Comodín” (*)
  - Puede ser mucho más complicado
- Regex van a coincidir tan pronto como sea posible y con todo lo que sea posible.
- Muy bueno para Análisis de Logs
  - Puede ser también usado para buscar

Metacaracteres

- Algunos de los más importantes:
  - * Coincide cero o más de lo previo
  - [ ] Describe un set
  - ^ Coincide con el principio de una línea
  - $ Coincide con el fin de la línea
  - ? Coincide con exactamente uno de lo previo
  - + Coincide con uno o más de lo previo
  - . Coincide con cualquier caractér

Cut

- No confundir Cut con Col
  - Cut permite extraer columnas
  - Col reformatea líneas recibidas y espacios en blanco
- Cut es especial para extraer información específica rápidamente
  - Extraer solamente las columnas que necesitamos
  - -f = fields

```
[~]$ free -m | grep Mem
Mem:           31827       24286         836        2930        6703        4170
[~]$ free -m | grep Mem | col
Mem:           31827       24294         848        2910        6684        4181
[~]$ free -m | grep Mem | col | cut -f 2
       31827
```

Sed

- Sed = Stream Editor
  - Rebanar y cortar el texto mientras pasa
    - Remover texto no deseado
    - Convertir el texto a algo más
    - Reformatear el texto a algo que otra herramienta pueda manejar

```
[~]$ sed -n 12,16p nodesource_setup.sh
#   or
# wget -qO- https://deb.nodesource.com/setup_14.x | bash -
#
# CONTRIBUTIONS TO THIS SCRIPT
#
```

Awk

- Awk = Aho, Weinberger y Kernighan
  - Coincidencia de patrones y lenguaje de procesamiento de texto
  - Rápido y fácil coincide y reemplaza
- Magia común de awk:
  - free | awk '/Mem/ { print $2; }'
- También se puede especificar el separador de campos!
  - awk -F: '{print $1;}' /etc/passwd
    - Mejor que:
      - sed -e 's/:/ /g' /etc/passwd | awk '{print $1;}'

Recetas

- Este tipo de Código puede ser pensado como recetas
  - Receta:
    - comando | awk ‘/termino_busqueda/ {print $<#_columna>; }’
  - Platos:
    - Memoria física: free | awk ‘/Mem/ { print $2; }’
    - Espacio libre en disco: df | awk ‘/\/$/ { print $4; }’
    - Direcciones MAC: ifconfig -a | awk ‘/ether/ { print $2; }’

Auditando Unix

Auditoría de Sistemas: El “Cómo”

- **Tarea**:
  - Evaluar la seguridad de un sistema desconocido
- **Problema**:
  - ¿Puedo confiar realmente en lo que encuentro cuando uso herramientas locales? (Rootkits)

Caso de Estudio: Lrk5 (Linux Rootkit v5)

- Oculta:
  - Archivos
  - Procesos
  - Conexiones de Red
- Borra/Edita Logs
- Limpia Registros
- Acceso Backdoor
- Escucha en la Red

Binarios reemplazados por lrk5

- chfn
- chsh
- crontab
- du
- find
- ifconfig
- inetd
- tcpd
- pidof
- killall
- login
- ls
- netstat **(2)**
- passwd
- ps **(1)**
- rshd
- syslogd **(3)**
- top **(1)**

El “Cómo” en un Sistema no confiable

- **Problema**:
  - Examinar un sistema no confiable
- **Solución**:
  - Crear un USB/CD con herramientas.
- Esto aplica a Windows también

Fuentes de CDs de Herramientas

- Knoppix/Ubuntu booteables
  - www.knoppix.org
  - www.ubuntu.com
    - Estas no son ideales. No fueron diseñadas para ser usadas en un sistema corriendo
- Helix
  - www.e-fense.com
    - Kits de respuesta en vivo
    - Contenido Digital de Forensia
    - $$
- ¿Respuesta en vivo para Solaris, HP-UX, AIX?

¿Hacer uno Propio?

- Personalizado a nuestros sistemas
- Puede crear USB/CD para cualquier sistema Unix
  - Posiblemente un USB/CD de Auditoría Universal
- Actualización instantánea
- Agregar nuestros propios scripts de auditoría

Lista de Shopping del USB/CD de Auditoría

- Librerías compartidas
- Librerías estáticas de sistema
- netstat
- lsof
- diff
- ps
- ls
- md5
- fdisk/cfdisk
- egrep/grep
- who, w, finger
- find
- df, du
- cp
- script
- dd
- sh/bash/csh
- [/test
- awk
- more/less

¿Cómo usar el Set de Herramientas?

- Montar el USB/CD como sistema de archivos
- Obtener una shell “limpia”
- Configurar los paths generales y paths para carga de librerías para saber qué librerías y binarios se están usando

```
# mount /mnt/cdrom
# /mnt/cdrom/bin/bash
# PATH=”/mnt/cdrom/bin”
# LD_LIBRARY_PATH=”/mnt/cdrom/lib”
# export PATH
# export LD_LIBRARY_PATH
```

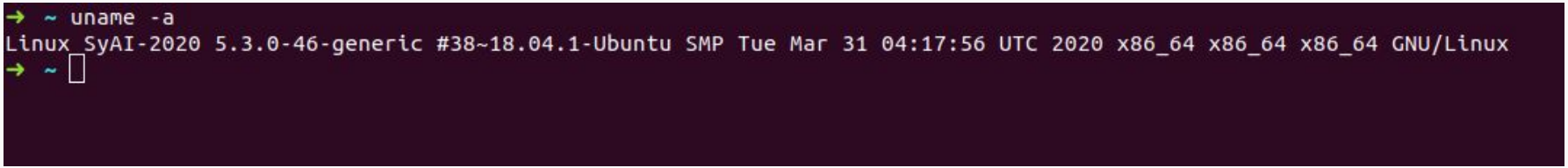
Objetivos y Actividades de la Auditoría Unix

Objetivo de Auditoría

- **Objetivo: Información del Sistema**
  - Identificar tipo de Sistema
  - Identificar nivel de actualizaciones (parches)
  - Información general del sistema
- **Actividades de Auditoría**
  - Uname
  - Patchdiag (Sun)
  - Etc

Versión del Sistema Operativo

- `uname -a`
  - Información de la arquitectura y sistema operativo
  - Disponible universalmente



Sistemas de Archivos

- `mount`
  - Sistemas de Archivos actualmente montados
  - Tipos de Sistemas de Archivos

```
→  ~ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=9800616k,nr_inodes=2450154,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=1964992k,mode=755)
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup on /sys/fs/cgroup/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
```

Sistemas de Archivos (2)

- `fdisk -l`
  - Valida montado versus real

```
→  ~ fdisk -l /dev/sda
Disk /dev/sda: 238,5 GiB, 256060514304 bytes, 500118192 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0xc55b1469

Device     Boot Start       End   Sectors   Size Id Type
/dev/sda1  *     2048 500117503 500115456 238,5G 83 Linux
```

Información General del Sistema

- `free`
  - Información de Utilización de Memoria

```
➜  ~ free
              total        used        free      shared  buff/cache   available
Mem:       19649912    13056220      430944     1820352     6162748     4436276
Swap:       2097148      950700     1146448
➜  ~
```

Parches

- ¿Cómo determinar el nivel de parches?
  - Sistema Operativo
    - Revisar “Avisos de Seguridad” en el sitio web de soporte
- Una de las cosas más difíciles de alcanzar en sistemas Unix
  - Todo depende de cómo el software fue instalado

```
→  ~ sudo cat /var/lib/update-notifier/updates-available

0 packages can be updated.
0 updates are security updates.
```

Objetivo de Auditoría

- **Objetivo: Perfil Operativo**
  - Identificar Servicios de Red
  - Identificar Servicios Locales
  - Identificar Comportamiento de Red
- **Actividades de Auditoría**
  - Netstat
  - Lsof (Open files)
  - Ps
  - Top (Table of processes)

Identificando Servicios de Red

- `netstat`lista:
  - Conexiones activas
  - Puertos escuchando
- Algunas versiones son capaces de relacionar ésto a información de procesos

Identificando Servicios de Red (2)

```
→  ~ netstat -ntap
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:41271         0.0.0.0:*               LISTEN      12903/java
tcp        0      0 127.0.0.1:9788          0.0.0.0:*               LISTEN      16589/node
tcp        0      0 127.0.0.1:41734         0.0.0.0:*               LISTEN      13917/kbfsfuse
tcp        0      0 172.22.0.1:9993         0.0.0.0:*               LISTEN      -
tcp        0      0 192.168.0.108:9993      0.0.0.0:*               LISTEN      -
tcp        0      0 172.21.0.1:9993         0.0.0.0:*               LISTEN      -
tcp        0      0 172.20.0.1:9993         0.0.0.0:*               LISTEN      -
tcp        0      0 172.19.0.1:9993         0.0.0.0:*               LISTEN      -
tcp        0      0 172.18.0.1:9993         0.0.0.0:*               LISTEN      -
tcp        0      0 172.17.0.1:9993         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:9993          0.0.0.0:*               LISTEN      -
tcp        0      0 172.22.0.1:64683        0.0.0.0:*               LISTEN      -
tcp        0      0 192.168.0.108:64683     0.0.0.0:*               LISTEN      -
tcp        0      0 172.21.0.1:64683        0.0.0.0:*               LISTEN      -
tcp        0      0 172.20.0.1:64683        0.0.0.0:*               LISTEN      -
tcp        0      0 172.19.0.1:64683        0.0.0.0:*               LISTEN      -
tcp        0      0 172.18.0.1:64683        0.0.0.0:*               LISTEN      -
```

¿Qué es lsof?

- Generalmente instalado por defecto
- Lista archivos abiertos
- Perfecto para procesos, archivos e investigaciones de estado de red
- Puede producir salida capaz de ser consumida por otros programas

Usando lsof para Identificar Conexiones de Red

```
➜  ~ sudo lsof -i
COMMAND     PID            USER   FD   TYPE  DEVICE SIZE/OFF NODE NAME
systemd-r   684 systemd-resolve   12u  IPv4   21105      0t0  UDP localhost:domain
systemd-r   684 systemd-resolve   13u  IPv4   21106      0t0  TCP localhost:domain (LISTEN)
avahi-dae   829           avahi   12u  IPv4   30819      0t0  UDP *:mdns
avahi-dae   829           avahi   13u  IPv6   30820      0t0  UDP *:mdns
avahi-dae   829           avahi   14u  IPv4   30821      0t0  UDP *:49731
avahi-dae   829           avahi   15u  IPv6   30822      0t0  UDP *:35102
zerotier-  1938    zerotier-one    6u  IPv4   37394      0t0  TCP localhost:9993 (LISTEN)
zerotier-  1938    zerotier-one    7u  IPv6   37395      0t0  TCP ip6-localhost:9993 (LISTEN)
zerotier-  1938    zerotier-one    8u  IPv4   72932      0t0  UDP SyAI-2020:9993
zerotier-  1938    zerotier-one   13u  IPv4  774750      0t0  UDP SyAI-2020:9993
zerotier-  1938    zerotier-one   14u  IPv4  774751      0t0  TCP SyAI-2020:9993 (LISTEN)
zerotier-  1938    zerotier-one   15u  IPv4  774752      0t0  UDP SyAI-2020:64683
zerotier-  1938    zerotier-one   16u  IPv4  774753      0t0  TCP SyAI-2020:64683 (LISTEN)
zerotier-  1938    zerotier-one   17u  IPv4  774754      0t0  UDP SyAI-2020:64684
zerotier-  1938    zerotier-one   18u  IPv4  774755      0t0  TCP SyAI-2020:64684 (LISTEN)
zerotier-  1938    zerotier-one   19u  IPv4   72933      0t0  TCP SyAI-2020:9993 (LISTEN)
zerotier-  1938    zerotier-one   20u  IPv4   72934      0t0  UDP SyAI-2020:9993
zerotier-  1938    zerotier-one   21u  IPv4   72935      0t0  TCP SyAI-2020:9993 (LISTEN)
zerotier-  1938    zerotier-one   22u  IPv4   72936      0t0  UDP SyAI-2020:9993
zerotier-  1938    zerotier-one   23u  IPv4   72937      0t0  TCP SyAI-2020:9993 (LISTEN)
```

¿Cómo son iniciados los Servicios?

- Inetd
  - “Super Daemon” original
  - Sin Control de Acceso incorporado
- Xinetd
  - Versión moderna de inetd
  - Control de Acceso incorporado
- Scripts de Inicio

Inetd

```
# /etc/inetd.conf:  see inetd(8) for further informations.
#
# Internet superserver configuration database
#
#
# Lines starting with "#:LABEL:" or "#<off>#" should not
# be changed unless you know what you are doing!
#
# If you want to disable an entry so it isn't touched during
# package updates just comment it out with a single '#' character.
#
# Packages should modify this file by using update-inetd(8)
#
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
#
#:INTERNAL: Internal services
#discard                 stream  tcp     nowait  root    internal
#discard                 dgram   udp     wait    root    internal
#daytime                 stream  tcp     nowait  root    internal
#time           stream  tcp     nowait  root    internal
```

XInetd.conf

```
defaults
{
        instances               = 60
        log_type                = SYSLOG authpriv
        log_on_success          = HOST PID
        log_on_failure          = HOST
        cps                     = 25 30
}

includedir /etc/xinetd.d
```

Archivo de Servicio de XInetd

```
service rsync
{
        disable         = yes
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/bin/rsync
        server_args     = --daemon
        log_on_failure  += USERID
}
```

Scripts de Inicio

- /etc/rc.d
- /etc/init.d
  - /etc/rc*.d
- /etc/rc.local
- /etc/inittab

```
→ ~ ls -las /etc/init.d/
total 208
 4 drwxr-xr-x   2 root root  4096 abr 10 06:29 .
12 drwxr-xr-x 139 root root 12288 abr 15 12:37 ..
 4 -rwxr-xr-x   1 root root  2269 abr 22  2017 acpid
 8 -rwxr-xr-x   1 root root  5336 ene 23  2017 alsa-utils
 4 -rwxr-xr-x   1 root root  2014 may 29  2017 anacron
 8 -rwxr-xr-x   1 root root  4335 mar 22  2018 apparmor
 4 -rwxr-xr-x   1 root root  2805 feb 27 00:18 apport
 4 -rwxr-xr-x   1 root root  2401 ago 22  2018 avahi-daemon
 4 -rwxr-xr-x   1 root root  2968 sep 10  2019 bluetooth
 4 -rwxr-xr-x   1 root root  1904 nov 17  2015 cgroupfs-mount
 4 -rwxr-xr-x   1 root root  1232 abr 19  2018 console-setup.sh
 4 -rwxr-xr-x   1 root root  3049 nov 16  2017 cron
 4 -rwxr-xr-x   1 root root  2804 mar 27  2018 cups
 4 -rwxr-xr-x   1 root root  1961 feb 26  2018 cups-browsed
 4 -rwxr-xr-x   1 root root  2813 nov 15  2017 dbus
```

Servicios de Inicio de OS X

- Almacenados en archivos “plist”
  - **/System/Library/LaunchDaemons**: daemons de todo el sistema
  - **/Library/LaunchDaemons**: daemons de inicio de todo el sistema controlados por el Administrador
  - **/Library/LaunchAgents**: agentes de lanzamiento de todo el sistema
  - **/System/Library/LaunchAgents**: agentes de lanzamiento de todo el sistema
  - **~/Library/LaunchAgents**: agentes de lanzamiento específicos del usuario
- La manera fácil de inventariarlos es usando “**launchctl**” para obtener información

El Arranque de Unix es “Determinístico”

- ¿Qué significa ésto y como es útil para nosotros?

```
→  ~ sudo ps -aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.1  0.0 225956  9048 ?        Ss   abr14   1:59 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    abr14   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   abr14   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   abr14   0:00 [rcu_par_gp]
root           9  0.0  0.0      0     0 ?        I<   abr14   0:00 [mm_percpu_wq]
root          10  0.0  0.0      0     0 ?        S    abr14   0:03 [ksoftirqd/0]
root          11  0.1  0.0      0     0 ?        I    abr14   1:53 [rcu_sched]
root          12  0.0  0.0      0     0 ?        S    abr14   0:00 [migration/0]
root          13  0.0  0.0      0     0 ?        S    abr14   0:00 [idle_inject/0]
root          14  0.0  0.0      0     0 ?        S    abr14   0:00 [cpuhp/0]
root          15  0.0  0.0      0     0 ?        S    abr14   0:00 [cpuhp/1]
root          16  0.0  0.0      0     0 ?        S    abr14   0:00 [idle_inject/1]
root          17  0.0  0.0      0     0 ?        S    abr14   0:00 [migration/1]
root          18  0.0  0.0      0     0 ?        S    abr14   0:01 [ksoftirqd/1]
root          20  0.0  0.0      0     0 ?        I<   abr14   0:00 [kworker/1:0H-kb]
root          21  0.0  0.0      0     0 ?        S    abr14   0:00 [cpuhp/2]
root          22  0.0  0.0      0     0 ?        S    abr14   0:00 [idle_inject/2]
```

Comportamiento de Red

- ¿Qué clase de opciones de red están disponibles que nos importen?
  - ¿Enrutamiento de Origen?
  - ¿Enrutamiento en General?
  - ¿Pueden las tablas de enrutamiento ser reconfiguradas a través de redirecciones?
  - ¿Alguna protección de denegación de servicio para los servidores?

Linux como Ejemplo

- La reconfiguración es posible en todos, pero tenemos Linux a mano
  - /etc/sysctl.conf

net.ipv4.ip_forward

net.ipv4.conf.all.accept_source_route

net.ipv4.tcp_max_syn_backlog

net.ipv4.conf.all.rp_filter

net.ipv4.conf.all.default.accept_redirects

net.ipv4.tcp_syncookies

net.ipv4.conf.all.send_redirects

net.ipv4.conf.all.accept_redirects

¿Podemos escribir un script para estas configuraciones?

Objetivo de Auditoría

- **Objetivo: Acceso No Autorizado**
  - Examinar Control de Accesos a Nivel de Red para Equipos
- **Actividades de Auditoría**
  - Hosts.allow
  - Hosts.deny

Listas de Control de Acceso (ACLs)

- ¿Qué es lo que las ACLs dicen?
  - /etc/hosts.deny debería incluir ALL:ALL
  - /etc/hosts.allow debería listar equipos individuales (confiables) servicio por servicio
- Por defecto, tcpd está instalado, pero el archivo hosts.deny está vacío
  - Inutilizado

```
→  ~ ls -lp /etc | grep -v / | grep hosts
-rw-r--r--  1 root root      329 mar 27 18:00 hosts
-rw-r--r--  1 root root      411 feb  3 15:25 hosts.allow
-rw-r--r--  1 root root      711 feb  3 15:25 hosts.deny
→  ~
```

Objetivo de Auditoría

- **Objetivo: Administración/Acceso de Usuarios**
  - Asegurar cuentas de usuario únicas
  - Identificar usuarios autorizados
  - Examinar configuraciones de contraseñas
  - Asegurar que se están usando contraseñas fuertes
- **Actividades de Auditoría**
  - Examinar /etc/passwd, /etc/shadow
  - John the Ripper

Seguridad en el Momento de Arranque

- Los sistemas Unix son particularmente vulnerables con acceso físico
  - Es posible cambiar/borrar contraseñas si podemos reiniciar el sistema
- Buscar controles para mitigar
  - Contraseñas para el Momento de Arranque
  - Restringir teclas de acceso rápido para reinicio

El “Cómo Hacerlo” Varía

- Dependiendo del Sistema Operativo - Veamos en Linux
  - Dos cargadores de inicio principales: LILO/Grub
  - Grub permite protección por contraseña del proceso de inicio
    - Por defecto la contraseña es almacenada en texto plano en **/etc/grub.conf**
    - **/sbin/grub-md5-crypt** permite crear un hash cifrado
    - Verificar que la línea de contraseña en **/etc/grub.conf** es correcta

Deshabilitando Teclas de Acceso Rápido

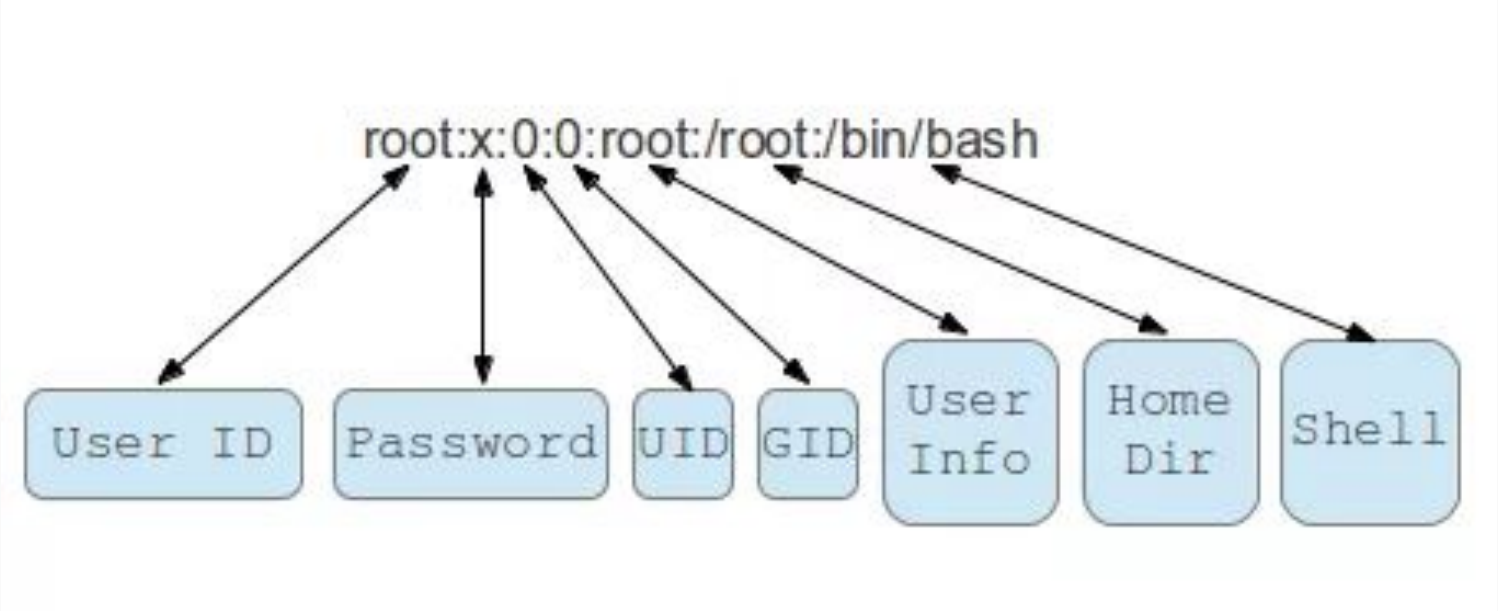
- ¿Cómo podríamos prevenir los reinicios?
  - Muchos sistemas Unix x86 mapean Control-Alt-Delete para un reinicio automático
  - Controlado a través de **/etc/inittab**
  - Comentar la línea para prevenir el comportamiento
    - (Agregar un “#” al inicio de la línea para comentarla)

Limitando el Acceso Remoto

- ¿Necesita el usuario “root” realmente iniciar sesión directa desde ubicaciones remotas?
  - Muchos sistemas Unix permiten restringir ésto usando el archivo de configuración **/etc/securetty**
  - Verificar que solamente terminales conectadas localmente/físicamente permiten iniciar sesión directamente con root.

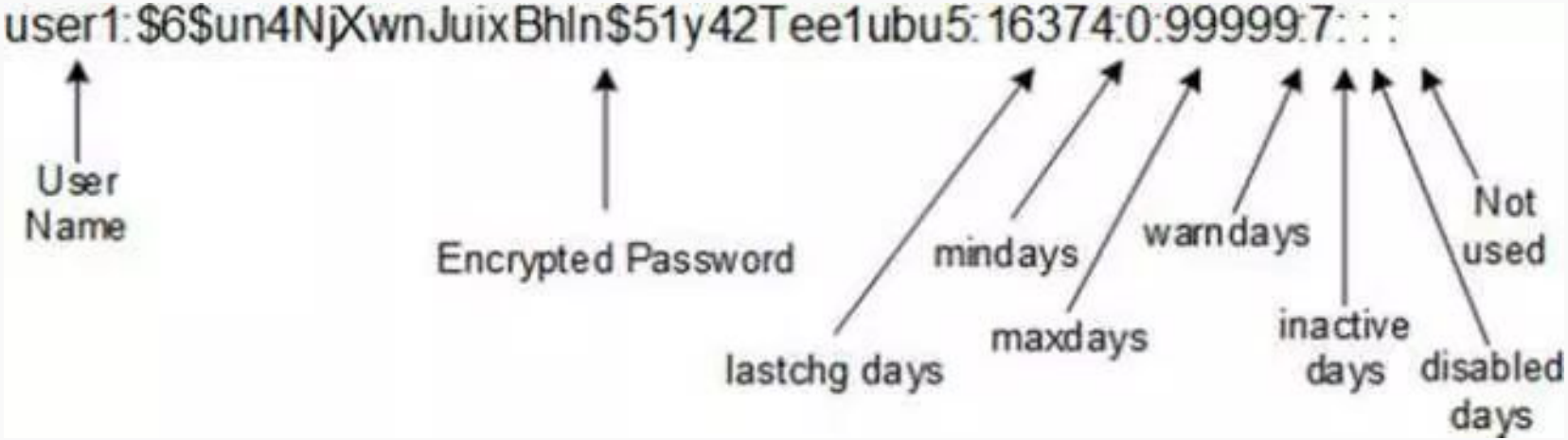
Archivo ‘passwd’

- Ubicación Tradicional de Información de Autenticación



Archivo ‘shadow’

- Solamente root puede leerlo



Shadow Recargado

- Revisamos los campos:
  - Nombre de Usuario
  - Hash de la Contraseña
  - Días desde 1/1/1970 que la contraseña fue cambiada
  - Días que deben pasar para que la contraseña pueda ser cambiada
  - Días después que la contraseña debe ser cambiada
  - Días antes de la expiración que el usuario es advertido
  - Días después de la expiración que la cuenta es deshabilitada
  - Días desde 1/1/1970 que la cuenta fue deshabilitada
- Generalmente configurado por **/etc/default/useradd**
  - Opciones para cada una de las configuraciones arriba

Herramientas de Evaluación de Contraseñas

- John the Ripper
  - Cracking distribuído
  - Corre en Windows y Unix
  - Contraseñas del estilo BSD
  - Contraseñas basadas en DES
  - Contraseñas basadas en Twofish
  - Hashes NTLM

Objetivo de Auditoría

- **Objetivo: Acceso No Autorizado**
  - Asegurar que solamente los archivos necesarios tienen los bits **set-user** o **set-group** configurados
  - Identificar archivos modificados recientemente
- **Actividades de Auditoría**
  - Find
  - Ls

¿Qué Buscamos?

- Archivos SUID y SGID
- Binarios recientemente modificados
- Archivos ocultos
- Entradas extra o incorrectas en /etc/passwd
- Cualquier cosa “fuera de lo ordinario”

Comando ‘find’

- Dada una expresión, find busca el árbol de directorios y realiza alguna acción en los archivos cuyos atributos coinciden
- Puede buscar:
  - Creación, modificación y fechas de acceso
  - Patrones de nombre
  - Tipo de archivo, tamaño, permisos, dueño, grupo

Encontrar Archivos SUID

- Obtener una lista de todos los archivos suid y sgid
- Encontrar archivos SUID debería ser parte de la línea de base
- Usar el comando find:

```
→  ~ sudo find / -perm /4000 -type f
/opt/keybase/chrome-sandbox
/opt/google/chrome/chrome-sandbox
/bin/umount
/bin/fusermount
/bin/ping
/bin/mount
/bin/su
/var/lib/docker/overlay2/e06ae0834c56257e15f4704bdf95d0456a1462bb8dea4e3cc1f3960e9c911eb1/diff/bin/umount
/var/lib/docker/overlay2/e06ae0834c56257e15f4704bdf95d0456a1462bb8dea4e3cc1f3960e9c911eb1/diff/bin/ping
/var/lib/docker/overlay2/e06ae0834c56257e15f4704bdf95d0456a1462bb8dea4e3cc1f3960e9c911eb1/diff/bin/mount
/var/lib/docker/overlay2/e06ae0834c56257e15f4704bdf95d0456a1462bb8dea4e3cc1f3960e9c911eb1/diff/bin/su
/var/lib/docker/overlay2/e06ae0834c56257e15f4704bdf95d0456a1462bb8dea4e3cc1f3960e9c911eb1/diff/usr/bin/passwd
/var/lib/docker/overlay2/e06ae0834c56257e15f4704bdf95d0456a1462bb8dea4e3cc1f3960e9c911eb1/diff/usr/bin/gpasswd
/var/lib/docker/overlay2/e06ae0834c56257e15f4704bdf95d0456a1462bb8dea4e3cc1f3960e9c911eb1/diff/usr/bin/newgrp
```

Localizar Binarios Modificados Recientemente

```
# touch -m 04072021 /tmp/tstamp

# find / -newer /tmp/tstamp -type f
```

- Encuentra, imprime y ordena por fecha:
  - Todos los archivos regulares
  - Más nuevos que 7 de Abril de 2021

En Síntesis

- Estándares para Copos de Nieve
- Recomendaciones para Scripting Básico
- Puntos Importantes de Auditoría Unix

¿Preguntas?

¡Muchas Gracias!