Fundamentos de la Seguridad Informática
Seguridad de Sistemas

Grupo de Seguridad Informática
Facultad de Ingeniería
Universidad de la República

Propósito

El objetivo principal de estas prácticas es introducir a los participantes en problemas de seguridad comunes asociados a los sistemas operativos y su gestión. El estudiante deberá realizar acciones ofensivas de reconocimiento, utilizar técnicas y herramientas que le permitan obtener contraseñas y explotar vulnerabilidades para escalar privilegios en el sistema. Además, se presentan herramientas defensivas que permiten robustecer el mecanismo de autenticación en sistemas Linux.

Introducción

Las contraseñas son una de las formas más comunes en la actualidad de obtener acceso a diferentes sistemas y son ubicuas en nuestra vida cotidiana: cajeros automáticos, claves de teléfonos celulares, cuentas de redes sociales, de correo, del banco, etc. Pero no cualquier contraseña es segura ante ataques de fuerza bruta o utilizando diccionarios. A modo de referencia, en la figura 1 se presenta un gráfico con el tiempo que lleva romper una contraseña dependiendo del largo y complejidad. A continuación se brindan algunas recomendaciones o buenas prácticas para generar contraseñas más seguras.

Las contraseñas:

- deben contener letras mayúsculas y minúsculas, números y otros caracteres especiales (p.ej. #,$, %)
- deben tener un cierto largo mínimo, puesto que cuando más cortas son, más fáciles serán de descubrir
- en lo posible, ser generadas aleatoriamente o usar frases de contraseña (passphrase) con suficiente entropía

No deben:

- ser iguales al nombre de usuario
- utilizarse idénticas contraseñas para sitios distintos (p.ej: la misma para GMail y para Instagram).

TIME IT TAKES A HACKER TO BRUTE FORCE YOUR PASSWORD IN 2023

Number of Characters	Numbers Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	Instantly	Instantly
7	Instantly	Instantly	1 sec	2 secs	4 secs
8	Instantly	Instantly	28 secs	2 mins	5 mins
9	Instantly	3 secs	24 mins	2 hours	6 hours
10	Instantly	1 min	21 hours	5 days	2 weeks
11	Instantly	32 mins	1 month	10 months	3 years
12	1 sec	14 hours	6 years	53 years	226 years
13	5 secs	2 weeks	332 years	3k years	15k years
14	52 secs	1 year	17k years	202k years	1m years
15	9 mins	27 years	898k years	12m years	77m years
16	1 hour	713 years	46m years	779m years	5bn years
17	14 hours	18k years	2bn years	48bn years	380bn years
18	6 days	481k years	126bn years	2tn years	26tn years

HIVE SYSTEMS

› Learn how we made this table at hivesystems.com/password

Figura 1: Tiempo que toma romper una contraseña en el 2023 (`https://hivesystems.com/password`)

- no deben contener información personal de ningún tipo como ser número de teléfono, nombre del perro, etc.
- no deben ser palabras que puedan encontrase en cualquier diccionario de cualquier idioma.
- no deben ser mutaciones simples de palabras, como por ejemplo: `unouno` o `casa23`.
- no deben ser escritas en ningún lugar accesible a cualquier persona: por ejemplo en un papel pegado en el monitor.

Estos principios serán usados durante esta práctica.

1. Crackers de contraseñas

El objetivo principal de esta práctica es demostrar la importancia de la selección de buenas contraseñas en un entorno informático. Utiliza herramientas que permiten ilustrar diferentes técnicas utilizadas para obtener contraseñas. Los administradores de sistemas utilizan normalmente dichas herramientas para verificar que las contraseñas seleccionadas por los usuarios no sean triviales y por ello que no se puedan adivinar fácilmente. También es utilizada a veces por personas con otras intenciones.

Escenario

El escenario simula la red interna de una empresa y está compuesto por:

- Una máquina Linux `A` previamente comprometida y equipada con herramientas de ataque.
- Un puesto de trabajo víctima `V` a atacar.

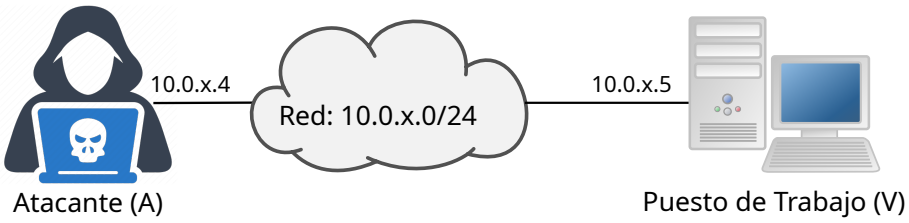

Figura 2: Descripción del Escenario

Herramientas

Para realizar esta práctica se dispondrá de las herramientas *John the Ripper* [1] y *THC-Hydra* [2].

Guía de la práctica

El objetivo de esta primera práctica es obtener las credenciales de un usuario en la máquina víctima que permita acceso por SSH.

1. Descargar de la página del laboratorio el archivo `practica1.tar.gz`, que contiene los archivos necesarios para realizar esta práctica:
   - El listado de personal de la empresa: `listadopersonal.html`
   - Diccionario: `dic.txt`
2. Observando la información que la empresa publica sobre los empleados, ¿qué puede deducir sobre la política para la asignación de nombres de usuarios de la empresa?
3. Se pudo saber que la empresa NO aplica ninguna política de asignación de contraseñas, pudiendo existir contraseñas débiles.

Utilizando las herramientas provistas y la información que la empresa publica sobre sus empleados, obtener acceso al puesto de trabajo con un usuario de mínimo privilegio.

Sugerencias:

- Deducir los nombres de usuario para aquellos que no se publica la dirección de correo electrónico.
- Tenga en cuenta como *NO DEBEN* ser las contraseñas.
- Se sugiere utilizar herramientas de reconocimiento adecuadas para descubrir los distintos servicios brindados por el puesto de trabajo.
- Se sugiere intentar conseguir la mayor cantidad de credenciales posibles, utilizando los diferentes tipos de ataque de las herramientas de *cracking*.

Entregables:

- En un archivo de texto (`usuarios.txt`) ingresar los nombres de usuario y contraseñas encontradas, la forma y el programa que utilizó para obtener cada una.

Objetivos de aprendizaje

En esta práctica se deberán comprender lo siguientes puntos:

- Buenas prácticas para la gestión de las contraseñas.
- Los distintos modos de ejecución de las herramientas de cracking de contraseñas.
- Pros, contras y escenarios de uso de cracking local y remoto.
- Para qué sirven las *salt* en contraseñas.

2. Escalada de privilegios

La escalada de privilegios consiste en que un usuario incremente sus privilegios, a través de un error de programación o configuración, permitiendole acceder a recursos del sistema que no tiene permitidos.

Vulnerabilidades comunes asociadas a la escalada de privilegio son buffer overflow o shell injection. En Linux es común, además, explotar vulnerabilidades sobre comandos que tienen activado el bit SUID (Set UserID) o que pueden ser ejecutados mediante el comando `sudo`. Ambos mecanismos permiten la ejecución de un programa con los privilegios efectivos de otro usuario (típicamente `root`).

Esta práctica permite visualizar la escalada de privilegios a través de ataques locales sobre un sistema y a su vez, muestra malas prácticas en la administración de sistemas operativos que hacen a un sistema vulnerable.

Escenario

Esta práctica usará el mismo puesto de trabajo atacado en la práctica anterior.

Herramientas

Para realizar esta práctica se dispondrá de la herramienta de reconocimiento *linPEAS* de la suite PEASS-ng [3]. La misma está disponible en la ruta `/usr/local/bin/linpeas.sh` de la máquina atacante.

Guía de la práctica

1. Hacer un reconocimiento sobre el puesto de trabajo atacado.
2. Recabar la máxima información posible del sistema (versión de sistema operativo, aplicaciones, procesos, usuarios, etc.).
3. Investigar posibles vulnerabilidades de escalada local de privilegios que se puedan ejecutar sobre la instalación. Se sugiere seguir alguna checklist (por ejemplo [4, 5]), y utilizar la herramienta de reconocimiento provista.

Explotando alguna de las vulnerabilidades encontradas, obtener acceso como usuario `root` del sistema. Descargar el archivo `/root/flag.txt` y guardarlo, ya que será la evidencia de que se obtuvo permisos de super-usuario.

Entregables:

- El archivo `flag.txt` encontrado en el homedir de root.
- Los comandos utilizados y el nombre del usuario que los ejecutó para realizar la escalada de privilegios.

Objetivos de Aprendizaje

- Entender la vulnerabilidad explotada, el funcionamiento del ataque, y por qué éste es exitoso.
- Cómo mitigar este problema:
  - desde el punto de vista del programador (mínimo privilegio, separación de funciones)
  - desde el punto de vista del administrador

3. Fortalecimiento de los mecanismos de autenticación

El objetivo de esta práctica es mostrar herramientas que contribuyen a asegurar los mecanismos de autenticación en un sistema Linux. Para esto se trabajará en dos líneas: por un lado herramientas que fuercen que las contraseñas seleccionadas para el acceso a los sistemas no sean triviales y que cumplen con determinados requisitos que las hacen computacionalmente más seguras; por otro lado, se trabajará con herramientas que permitan agregar un segundo factor de autenticación (2FA) en base de algo que se sabe (la contraseña) y algo que se tiene.

Herramientas

Para la realización de esta práctica se utilizará el framewok de módulos PAM (Pluggable Authentication Modules) de linux, en particular los siguientes módulos:

- `pam_passwdqc` [6]
- `pam_google_authenticator` [7].

3.1. Presentación

El módulo PAM `pam_passwdqc` brinda al administrador la posibilidad de exigir mayor fortaleza sobre las contraseñas de los usuarios, al momento de asignarla por primera vez o cambiarla. Por su parte, el módulo `pam_google_authenticator` permite configurar un segundo factor de autenticación basado en TOTP/HOTP. En el RFC 6238 [8] se puede ver la especificación del Time-based One-time Password (TOTP), y en el RFC 4226 [9] la del HMAC-Based One-time Password (HOTP).

Linux PAM
Los módulos PAM[10] permiten al administrador de sistemas elegir cómo autentican las aplicaciones a los usuarios y realizar diferentes controles. Permiten también la integración de varios mecanismos de autenticación, como pueden ser kerberos, radius, etc, sin necesidad de recompilar las aplicaciones.

Dependiendo del sistema operativo, puede que tengamos un único archivo de configuración para todas las aplicaciones `/etc/pam.conf`; o un directorio donde podemos encontrar un archivo de configuración para cada aplicación `/etc/pam.d`.

3.2. Escenario

Esta práctica se ejecutará sobre una maquina con el sistema operativo Linux, y los módulos `pam_passwdqc` y `pam_google_authenticator` instalados. Aclarar el sistema operativo en el cual se realizó la tarea.

3.3. Guía de la práctica

Configurar las contraseñas de ingreso al sistema según la siguiente política:

- Las contraseñas deberán tener al menos 2 de las 4 clases de caracteres (dígitos, letras minúsculas, letras mayúsculas y caracteres especiales)
- Las contraseñas deberán tener un largo mínimo de 12 caracteres.
  - si el largo es de 12 a 14 caracteres, debe tener caracteres de las 4 clases
  - si el largo es de 15 a 18 caracteres, debe tener caracteres de al menos 3 clases
- Cuando se cambia de contraseña, la nueva contraseña no puede ser similar a la anterior.
- No se podrá usar información del login del usuario como parte de la contraseña.
- El usuario podrá hacer hasta 4 intentos de ingreso de una nueva contraseña, que cumpla con la política establecida.

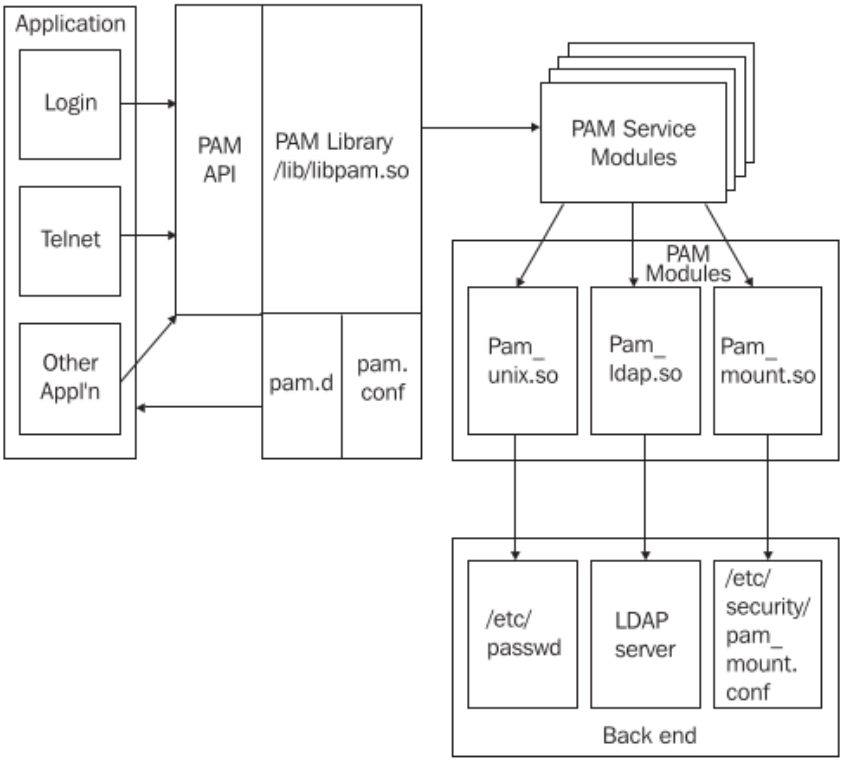

Figura 3: Módulos PAM

Además, configurar el sistema para que permita a los usuarios configurar un segundo factor de autenticación basado en TOTP.

3.4. Entregables:

1. El o los archivos que se modificaron para que se cumpla con la política de contraseña indicada.
2. El o los archivos que se modificaron para configurar el 2FA en el sistema.
3. Un archivo de texto plano, donde se presente de forma concisa a un usuario final cómo debe de configurar el 2FA y discutir las buenas prácticas de los diferentes parámetros que pueda configurar el usuario.

3.5. Objetivos de Aprendizaje

- Entender los distintos tipos de módulos PAM y las distintas situaciones en las cuáles pueden ser utilizados.
- Entender el funcionamiento del stack de módulos PAM.
- Entender la necesidad de definir políticas de contraseñas en las organizaciones.

- Métodos de autenticación alternativos al uso de contraseñas.
- El uso de OTP como segundo factor de autenticación.

4. Desafío

El desafío de esta práctica es descubrir la contraseña del usuario root en la máquina víctima.

Forma de Entrega

Se debe entregar el archivo `entrega2.tar.gz` conteniendo lo solicitado en cada práctica.

La entrega se realizará a través del EVA. Se deberá entregar el archivo solicitado en la actividad `Laboratorio 2`.

5. Referencias

Referencias

[1] Openwall Project. John the Ripper password cracker. `http://www.openwall.com/john`.

[2] The Hackers Choice. THC Hydra. `https://github.com/vanhauser-thc/thc-hydra`.

[3] Carlos Polop Martin. PEASS-ng. `https://github.com/carlospolop/PEASS-ng`.

[4] Carlos Polop Martin. Hacktricks linux privilege escalation checklist. `https://book.hacktricks.xyz/linux-hardening/linux-privilege-escalation-checklist`.

[5] Emilio Pinna and Andrea Cardaci. GTFOBins. `https://gtfobins.github.io`.

[6] Openwall Project. Pluggable password strength checking for your servers. `http://www.openwall.com/passwdqc`.

[7] Google. Google authenticator pam module. `https://github.com/google/google-authenticator-libpam/`.

[8] D. M'Raihi S. Machani M. Pei J. Rydell. Totp: Time-based one-time password algorithm. `https://datatracker.ietf.org/doc/html/rfc6238`.

[9] D. M'Raihi M. Bellare F. Hoornaert D. Naccache O. Ranen. Hotp: An hmac-based one-time password algorithm. `https://datatracker.ietf.org/doc/html/rfc4226`.

[10] K. Geisshirt. *Pluggable Authentication Modules: The Definitive Guide to PAM*. From technologies to solutions. Packt Pub., 2006.