口令破解实验报告

1、安装 john 使用命令：sudo apt-get install john

2、了解 Linux 口令加密保存原理：

在 linux 中，口令文件在/etc/passwd 中，早期的这个文件直接存放加密后的密码，前两位是"盐"值，是一个随机数，后面跟的是加密的密码。为了安全，现在的 linux 都提供了/etc/shadow 这个影子文件，密码放在这个文件里面，并且是只有 root 可读的。

/etc/passwd 文件，他的每个条目有 7 个域，分别是（名字:密码:用户 id:组 id:用户信息:主目录:shell） 例如：eric_hailong:x:1000:1000:Eric_hailong,,,:/home/eric_hailong:/bin/bash

/etc/shadow 文件中的记录行与/etc/passwd 中的一一对应，存放着用户的密码哈希值。

shadow 文件的情况下，密码用一个 x 表示，普通用户看不到任何密码信息。如果你仔细的看看这个文件，会发现一些奇怪的用户名，她们是系 统的缺省账号，缺省账号是攻击者入侵的常用入口，因此一定要熟悉缺省账号，特别要注意密码域是否为空。下面简单介绍一下这些缺省账号

adm 拥有账号文件，起始目录/var/adm 通常包括日志文件

bin 拥有用户命令的可执行文件

daemon 用来执行系统守护进程

games 用来玩游戏

halt 用来执行 halt 命令

lp 拥有打印机后台打印文件

mail 拥有与邮件相关的进程和文件

news 拥有与 usenet 相关的进程和文件

nobody 被 NFS（网络文件系统）使用

shutdown 执行 shutdown 命令

sync 执行 sync 命令

uucp 拥有 uucp 工具和文件

/etc/shadow 影子口令系统把口令文件分成两部分：/etc/passwd 和/etc/shadow。影子口令文件保存加密的口令；/etc/passwd 文件中的密码全部变成 x。Shadow 只能是 root 可读，从而保证了安全。/etc/shadow 文件每一行的格式如下：(用户名:加密口令:上一次修改的时间:口令在两次修改间的最小天数:口令修改之前向用户发出警告的天数:口令终止后账号被禁用的天数:从 1970 年 1 月 1 日起账号被禁用的天数:保留域)。

如：

eric_hailong:6Tiuw4Shb$Ka6bg5Ee0mrpYTh1Xc.hsvmFkFKcX9CnwQUqHWxfDGIDTTT5NzZqMwLm71qOR7W1zDzYZHaC6zx6jfs94rV3N/:17848:0:99999:7:::

Shadow 采用的 DES 的加密方式：其中的加密口令格式：idsalt$encrypted

其中 id 是指使用的哈希算法：

可取如下值：

ID	Method
1	MD5
2a	Blowfish (not in mainline glibc; added in some Linux distributions)
5	SHA-256 (since glibc 2.7)

6 | SHA-512 (since glibc 2.7)

salt：是使用上面 hash 算法对密码进行 hash 的一个干扰值。

encrypted: 这个值即 密码的 hash, 但不是直接的 hash("passwd")，而是 hash("passwd + salt") 后，再经过编码。

测试结果与上述相同：

```
>>> salt='$6$Tiuw4Shb$'
>>> m = crypt.crypt(passwd,salt)
>>> print m
$6$Tiuw4Shb$Ka6bg5Ee0mrpYTh1Xc.hsvmFkFKcX9CnwQUqHWxfDGIDTTT5NzZqMwLm71qOR7W1zDzYZHaC6zx6jfs94rV3N/
```

3、了解了相关实验原理后，进行实验，为了便于攻击，首先修改密码，将其改为简单的 123456。

```
eric_hailong@eric-ubuntu:~$ sudo passwd eric_hailong
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
eric_hailong@eric-ubuntu:~$ sudo unshadow /etc/passwd /etc/shadow >test_passwd
```

4、为了符合 john 解密的格式，合并 passwd 和 shadow 文件，输出文件 test_passwd 文件，只是将 passwd 文件里面 x 还原成原来的密码 hash 值。

```
hplip:*:117:7:HPLIP system user,,,:/var/run/hplip:/bin/false
geoclue:*:118:124::/var/lib/geoclue:/usr/sbin/nologin
gnome-initial-setup:*:119:65534::/run/gnome-initial-setup/:/bin/false
gdm:*:120:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
eric_hailong:$6$Tiuw4Shb$Ka6bg5Ee0mrpYTh1Xc.hsvmFkFKcX9CnwQUqHWxfDGIDTTT5NzZqMwLm71qOR7W1zDzYZHaC6zx6jfs94rV3N/:1000:1000:Eric_hailong,,,:/hom
e/eric_hailong:/bin/bash
cups-pk-helper:*:121:116:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
mysql:!:122:127:MySQL Server,,,:/nonexistent:/bin/false
```

5、使用 john 所提供的密码列表进行攻击，结果马上攻击成功。

```
eric_hailong@eric-ubuntu:~$ sudo john --wordlist=/usr/share/john/password.lst te
st_passwd
Loaded 1 password hash (crypt, generic crypt(3) [?/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
123456           (eric_hailong)
1g 0:00:00:00 100% 3.030g/s 290.9p/s 290.9c/s 290.9C/s 123456..pepper
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

观察 password.lst 列表，显然 123456 在列表里面。

```
eric_hailong@eric-ubuntu:~$ cat /usr/share/john/password.lst
#!comment: This list has been compiled by Solar Designer of Openwall Project
#!comment: in 1996 through 2011.  It is assumed to be in the public domain.
#!comment:
#!comment: This list is based on passwords most commonly seen on a set of Unix
#!comment: systems in mid-1990's, sorted for decreasing number of occurrences
#!comment: (that is, more common passwords are listed first).  It has been
#!comment: revised to also include common website passwords from public lists
#!comment: of "top N passwords" from major community website compromises that
#!comment: occurred in 2006 through 2010.
#!comment:
#!comment: Last update: 2011/11/20 (3546 entries)
#!comment:
#!comment: For more wordlists, see http://www.openwall.com/wordlists/
123456
12345
password
password1
123456789
12345678
```

6、使用 john test_passwd 进行暴力破解，发现几分钟过去也无法破解，于是我将密码设置更简单一点，设成 123，使用暴力破解，结果破解成功。

```
eric_hailong@eric-ubuntu:~$ sudo john test_passwd
Loaded 1 password hash (crypt, generic crypt(3) [?/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
123              (eric_hailong)
1g 0:00:00:44 100% 2/3 0.02245g/s 316.2p/s 316.2c/s 316.2C/s 123456..pepper
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

7、试用其他的攻击 MODE，-single 主要以用户名产生的变化进行攻击，以及遍历模式都不成功。

8、讨论：

我认为虽然存在好多已经破解的密码算法，但是一般使用的工具里面的密码用到的加密算法都是目前没法破解的。是相对安全的，所以密码的安全问题，主要集中于口令的猜测，进行选择破解，常常由于社会工程学，这些选择大大提高了破解密码的成功率。因此口令的选择很重要，即使很强的加密算法，如果是弱口令，那么安全也将无法保证。如何设置一个好的口令呢？我根据 john 的破解模式以及规则，提出以下几种建议：

1 、不可让账号与密码相同（相似），john 的-sigle 模式可以很快破解。

2、不适用简单的数字以及英文字符，长度至少 4 个以上。因为很多简单的密码在 password.lst 里面，攻击将特别容易。

3、不可使用自己的姓名或特定特定意义的日期，因为社会工程学上这些很容易获取。

所以建议选择一种自己可以记忆的自己的密码系统，包括大小写字母、数字和字符。根据自己资料的类型或重要性，使用不同的密码系统。比如，对于自己的社交账户的密码，可以选一个自己与这个平台相联系的诗的拼音进行一定规则的相关的变化得到，这样容易记，有安全。对于自己的银行密码可以使用不同的规则。前提满足上述条件。

这样就极大地降低了猜测的成功率，所以有选择的暴力破解将很难成功。