rsh-grind User Documentation

pentestmonkey@pentestmonkey.net

6 May 2007

Contents

1 Overview

rsh-grind is a tool for remotely enumerating user accounts that are able to use the Remtoe Shell (RSH) protocol. The RSH protocol requires a local and remote username to be supplied. rsh-grind simply works through all possible combinations of local and remote usernames supplied via a wordlist.

2 Installation

rsh-grind is just a stand alone PERL script, so installation is as simple as copying it to your path (e.g. /usr/local/bin). It has only been tested under Linux so far.

It depends on the following PERL modules which you may need to install them first:

- Socket
- IO::Handle
- IO::Select
- IO::Socket
- Getopt::Std
- POSIX;

If you have PERL installed, you should be able to install the modules from CPAN:

```
# perl -MCPAN -e shell
cpan> install Getopt::Std
```

3 Usage

rsh-grind simply needs to be passed a list of local users, a list of remote users and a list of target RSH servers. Here's the usage message:

```
Usage: rsh-grind.pl [options] ( -U username_file ) ( -f ips.txt | hostname )
       rsh-grind.pl [options] (-l local_user|-L file) | (-r remote_user|-R file) (-f ips.txt|host)

options are:
        -m n     Maximum number of processes (default: 10)
        -l       Local username (on the client)
        -L file  File of local usernames
        -r       Remote username (on the server)
        -R file  File or remote usernames
        -U file  File of usernames.  Each will be used as remote and local
                 user (e.g. root/root, smtp/smtp).  Permutations will not
                 be tried.
        -c cmd   Command to execute (default: id)
                 (Must be non-interactive, so don't try /bin/sh for example)
        -f file  File of target hostnames / IPs
        -t n     Wait a maximum of n seconds for reply (default: 5)
        -d       Debugging output
        -v       Verbose
        -h       This help message

The Remote Shell Protocol (RSH) allows remote users to execute commands
on the server.  Authentication is based on:
- Source IP Address
- Remote Username
- Local Username

This script can't help you with obtaining a valid source IP address, but
it can help you try different combinations of remote and local usernames.

If your source IP address is in /etc/hosts.equiv, then you just need a
valid username on the system, the remote and local usernames should be the
same.

If the remote user account you're attacking has a .rhosts file like the
following in their home directory, it doesn't matter what the local
username is:
IP +

In both of the above cases, the most efficient usage is:

$ rsh-grind.pl -U users.txt target

If the .rhosts looks like the following, though the local username must
be "foo":
IP foo

In this case you might need to try lots of permutations before you
hit on the one that works.  The correct usage is:
```

```
$ rsh-grind.pl -L localusers.txt -R remoteusers.txt target

Output lines will be something like:
10.0.0.1/localuser/remoteuser    _uid=101(remoteuser) gid=1(other)_

Non-printable characters are sqashed to _ to fit whole response on
one line.

NB: The RSH protocol requires the binding of a local privileged port, so
you need to run this script as root.
```

4 Some Examples

For the examples below we need a list of potential usernames. The following output demostrates the format for this list:

```
$ head users.txt
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
```

4.1 Assuming Remote Usernames are Same as Local Usernames

If we assume that users of the RSH protocol will use the same username locally (i.e. where the rsh client is run) and remotely (i.e. on the server which we're testing), we only need a single list of usernames.

```
 $ sudo rsh-grind.pl -U users.txt 172.16.16.7
Starting rsh-grind v0.9.2 ( http://pentestmonkey.net/tools/rsh-grind )

 ----------------------------------------------------------
|                   Scan Information                       |
 ----------------------------------------------------------

Processes ............... 10
Command ................. id
Target .................. 172.16.16.7
Usernames file ......... users.txt
Username count ......... 10
Query timeout ........... 5 secs

######## Scan started at Sun May  6 18:39:13 2007 #########
172.16.16.7/bin/bin      _uid=2(bin) gid=2(bin)_
172.16.16.7/daemon/daemon        _uid=1(daemon) gid=1(other)_
172.16.16.7/lp/lp        _uid=71(lp) gid=8(lp)_
172.16.16.7/adm/adm      _uid=4(adm) gid=4(adm)_
######## Scan completed at Sun May  6 18:39:14 2007 #########
4 results.

10 queries in 1 seconds (10.0 queries / sec)
```

Note that we have to run as root because the RSH protocol requires us to bind to ports below 1024.

The results show 4 pairs of local and remote users that can execute commands via RSH. This usually indicates that user have an entry like the following in their /.rhosts file:

```
+ myuser
```

The + here means that access is allowed from any source IP address.

However, in this case the results are caused by the host running rsh-grind being present in /etc/hosts.equiv (not very likely in the real world, but it demonstrates what results you'd get with the small list of usernames above).

4.2 What if the Local and Remote Usernames are Different?

The remote username is sometimes different to the local username, e.g. a user fred might have the following in fred/.rhosts:

```
+ fbloggs
```

This means that the remote user fbloggs would be allowed to access fred's account from any source IP address.

We simply supply two lists of usernames (often, but not necessarily the same list):

```
 $ sudo ./rsh-grind.pl -L localusers.txt -R remoteusers.txt 172.16.16.7
Starting rsh-grind v0.9.2 ( http://pentestmonkey.net/tools/rsh-grind )

 ------------------------------------------------------------
|                   Scan Information                         |
 ------------------------------------------------------------

Processes ............... 10
Command ................. id
Target .................. 172.16.16.7
Remote usernames file .. localusers.txt
Local usernames file ... remoteusers.txt
Remote username count .. 12
Local username count ... 12
Query timeout .......... 5 secs

######## Scan started at Sun May  6 18:58:03 2007 #########
172.16.16.7/fred/fbloggs        _uid=102(fred) gid=1(group)_
######## Scan completed at Sun May  6 18:58:09 2007 #########
0 results.

144 queries in 6 seconds (24.0 queries / sec)
```

5 License

This tool may be used for legal purposes only. Users take full responsibility for any actions performed using this tool. The author accepts no liability for damage caused by this tool. If these terms are not acceptable to you, then do not use this tool.

In all other respects the GPL version 2 applies:

```
This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License version 2 as
published by the Free Software Foundation.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
```