Invasión de OpenWrt en Xiaomi Mi Router 4C

Este es mi tercer intento de instalar OpenWrt. La primera vez fue en 2019, cuando utilicé un puerto UART para conectar. La segunda vez, en 2023, utilicé un método remoto similar al descrito aquí.

El código de explotación se puede encontrar en https://github.com/acecilia/OpenWRTInvasion.

Primero, instala los requisitos:

```
pip install -r requirements.txt --break-system-packages
```

Después de ejecutar la explotación, puedes acceder a la interfaz web del router en una URL similar a esta (el valor de `stok` variará):

```
http://192.168.1.28/cgi-bin/luci/;stok=fe9b14c5c4dee48709fbdf00e048d5ec/web/home
```

“bash lzwjava@anonymous OpenWRTInvasion % python remote_command_execution_vulnerability.py Dirección IP del router [presiona enter para usar el valor predeterminado 'miwifi.com']: 192.168.1.28 Ingresa la contraseña del administrador del router: ... Hay dos opciones para proporcionar los archivos necesarios para la invasión: 1. Usar un servidor de archivos TCP local ejecutándose en un puerto aleatorio para proporcionar archivos en el directorio localscript_tools‘. 2. Descargar los archivos necesarios desde el repositorio de GitHub remoto. (elige esta opción solo si GitHub es accesible dentro del dispositivo del router.) ¿Cuál opción prefieres? (predeterminado: 1)1 ********** router_ip_address: 192.168.1.28 stok: 08f4f22fed20b94580cb8e70703c941c proveedor de archivos: servidor de archivos local ********** comenzando a subir el archivo de configuración…comenzando a ejecutar el comando… el servidor de archivos local está ejecutándose en 0.0.0.0:63067. root=‘script_tools’el servidor de archivos local está obteniendo ‘busybox-mipsel’para 192.168.1.28. el servidor de archivos local está obteniendo ‘dropbearStaticMipsel.tar.bz2’para 192.168.1.28. ¡Hecho! Ahora puedes conectarte al router usando varias opciones: (usuario: root, contraseña: root) * telnet 192.168.1.28 * ssh -oKexAlgorithms=+diffie-hellman-group1-sha1 -oHostKeyAlgorithms=+ssh-rsa -c 3des-cbc -o UserKnownHostsFile=/dev/null root@192.168.1.28 * ftp: usando un programa como cyberduck

root@XiaoQiang:/tmp# wget “https://downloads.openwrt.org/releases/24.10.0/targets/ramips/mt76x8/openwrt-24.10.0-ramips-mt76x8-xiaomi_mi-router-4c-squashfs-sysupgrade.bin”wget: no es una URL http o ftp: https://downloads.openwrt.org/releases/24.10.0/targets/ramips/mt76x8/openwrt-24.10.0-ramips-mt76x8-xiaomi_mi-router-4c-squashfs-sysupgrade.bin

scp -oKexAlgorithms=+diffie-hellman-group1-sha1 -oHostKeyAlgorithms=+ssh-rsa -c 3des-cbc openwrt-24.10.0-ramips-mt76x8-xiaomi_mi-router-4c-squashfs-sysupgrade.bin root@192.168.1.28:/tmp/ ash: /usr/libexec/sftp-server: no encontrado scp: Conexión cerrada

cat openwrt-24.10.0-ramips-mt76x8-xiaomi_mi-router-4c-squashfs-sysupgrade.bin | ssh -oKexAlgorithms=+diffie-hellman-group1-sha1 -oHostKeyAlgorithms=+ssh-rsa root@192.168.1.28 “cat > /tmp/openwrt-24.10.0-ramips-mt76x8-xiaomi_mi-router-4c-squashfs-sysupgrade.bin”

root@XiaoQiang:/tmp# ls 2541.bootcheck.log oui TZ rc.done appStoreRule.json rc.timing arrays resolv.conf authenfailed-cache resolv.conf.auto busybox root daemon rr datalist run dropbear script.sh dropbear.tar.bz2 speedtest_urls.xml etc spool ftpd startscene_crontab.lua.PID lock stat_points_privacy.log log stat_points_rom.log logexec state luci-indexcache sysapihttpd luci-nonce sysapihttpdconf luci-sessions sysinfo messages syslog-ng.ctl miqos.lock syslog-ng.pid mnt taskmonitor mt76xx2.sh.log uci2dat_mt7628.log network.env uploadfiles nginx_check.log upnp.leases ntp.status web_config_list openwrt-24.10.0-ramips-mt76x8-xiaomi_mi-router-4c-squashfs-sysupgrade.bin wifi_analysis.log

root@XiaoQiang:/tmp# mtd -r write openwrt-24.10.0-ramips-mt76x8-xiaomi_mi-router-4c-squashfs-sysupgrade.bin OS1 Desbloqueando OS1 ...

Escribiendo desde openwrt-24.10.0-ramips-mt76x8-xiaomi_mi-router-4c-squashfs-sysupgrade.bin a OS1 ... [w]

Conéctate al router a través de una conexión por cable. Luego puedes acceder a la interfaz web en 192.168.1.1 o usar SSH ejecutando `ssh root@192.168.1.1`.