

Config files

This sends alerts to discord to know when someone connects.

First i created a script called check_wg_peer.sh in /cfg so its persistent between reboots.

```
#!/bin/sh

# ----- Configuration -----
PEER=""
NAME="Dale"
WEBHOOK_URL=""
STATE_FILE="/cfg/wg_${NAME}_state"
TIMEOUT=180 # seconds, longer than handshake interval

# ----- Get latest handshake timestamp -----
HANDSHAKE=$(wg show wg latest-handshakes | grep "$PEER" | awk '{print $2}')
NOW=$(date +%s)
DIFF=$((NOW - HANDSHAKE))

# ----- Read previous state -----
PREV_STATE="disconnected"
[ -f "$STATE_FILE" ] && PREV_STATE=$(cat "$STATE_FILE")

# ----- Determine current state -----
if [ "$DIFF" -le "$TIMEOUT" ]; then
    CURR_STATE="connected"
else
    CURR_STATE="disconnected"
fi

# ----- Send alert on transition -----
if [ "$CURR_STATE" = "connected" ] && [ "$PREV_STATE" != "connected" ]; then
    curl -s -H "Content-Type: application/json" \
        -X POST \
        -d "{\"content\": \"Peer $NAME just connected to WireGuard!\"}" \
```

```
        "$WEBHOOK_URL"
    echo "Alert sent for $NAME"
fi

# ----- Save current state -----
echo "$CURR_STATE" > "$STATE_FILE"
```

Then I added this to my rc.local within /cfg

```
# === WireGuard Discord Alert Cron Job ===
WG_ALERT_SCRIPT="/cfg/check_wg_peer.sh"

# Make sure the script is executable
chmod +x "$WG_ALERT_SCRIPT"

# Add cron job if it doesn't already exist
grep -q "$WG_ALERT_SCRIPT" /etc/crontabs/root || echo "* * * * * $WG_ALERT_SCRIPT" >>
/etc/crontabs/root
```

Revision #1

Created 2025-10-22 21:15:03 UTC by Dale

Updated 2025-10-22 21:18:01 UTC by Dale