
KEYSIGHT
Microhard?
More like Micro*EASY*!
...to exploit...
HeadlessZeke – IoT Village – DEF CON 32

KEYSIGHT

Introduction

Who Am I?

- Principal Security Researcher at Keysight
- Focused mainly on offensive IoT research
- Hacking professionally since ’09
- Conference speaker
- Whiskey drinker, soccer watcher, comic book reader, record listener, bass guitar player

Who is Microhard?

- Canadian-based device manufacturer
- Focused on long range wireless comm equip
- Used in Oil & Gas, PoS, Transportation, etc

Industries
Oil & Gas
Water / Waste Water
SCADA
Voice Communications
Power/Electrical
Video Surveillance
Traffic and ITS
Unmanned Vehicle
Security
Agriculture
GPS Survey

KEYSIGHT

The Target

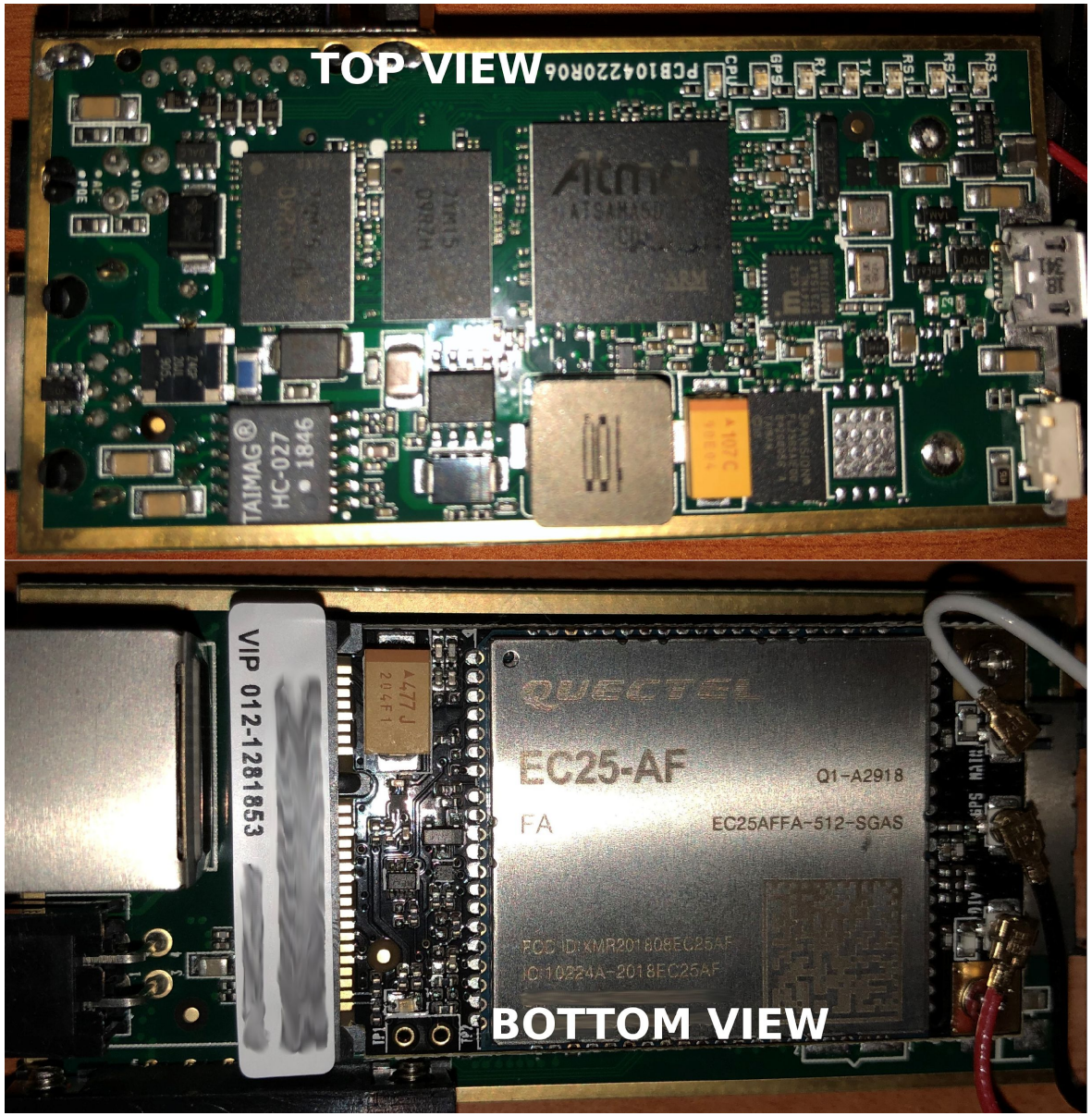
BulletLTE-NA2

- LTE bridge for adding cellular to eth/serial/io devices
- Manage remote installations from anywhere
- “Connect any device, old or new, Serial or Ethernet and provide cellular remote access using the BulletLTE-NA2.”
- Got from utility company for an audit a few years ago



BulletLTE-NA2

- Hardware
  - Atmel SAMA5D35 32bit ARM Cortex-A5
  - 128mb SPI flash for firmware
    - R/W filesystem
  - 256mb RAM
  - Quectel EC25-AF modem
  - SIM card, PoE eth, USB, serial



BulletLTE-NA2

- Firmware updates
  - Unsigned/unencrypted
  - Straightforward extraction with binwalk/unblob
  - U-boot bootloader
  - Kernel zimage
  - Full squashfs file system
  - Could easily emulate system or forge updates

BulletLTE-NA2

- Software
  - Based on OpenWRT
  - Linux 3.6.9 kernel
  - uhttpd for web (changed to lighttpd in r1112)
  - dropbear for ssh
  - telnetd for telnet
  - Login shell for admin user provided by /bin/clitest
  - Discovery service on UDP 20077, 20087, or 20097 (depends on model)

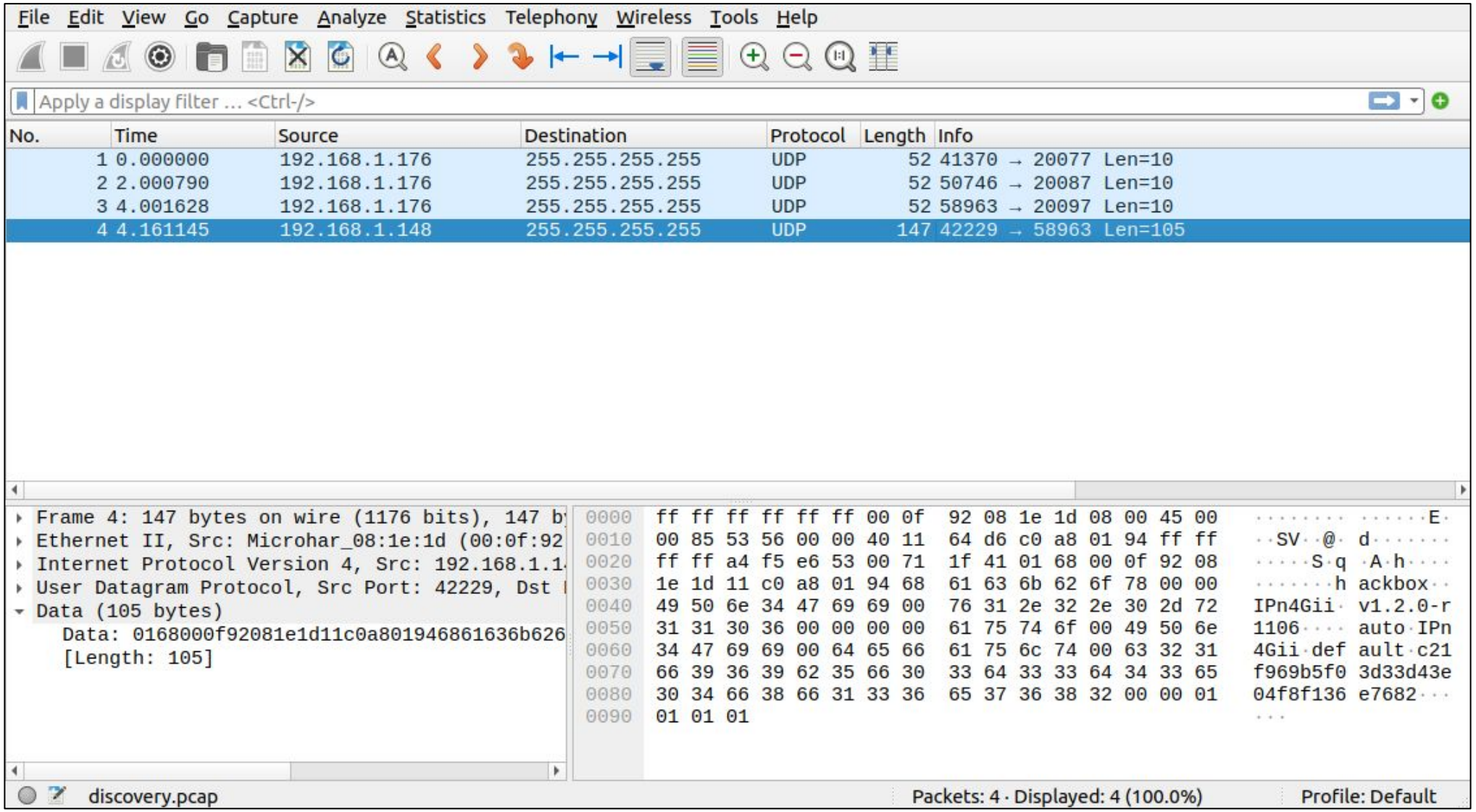
KEYSIGHT

Discovery Service

Discovery Service

- Listens for broadcast UDP packets
  - 20077 for 3G devices
  - 20087 for IP devices
  - 20097 for LTE devices
- Magic probe
  - \x70\x63\x00\x06\xff\xff\xff\xff\xff\xff

Discovery Service



Discovery Service

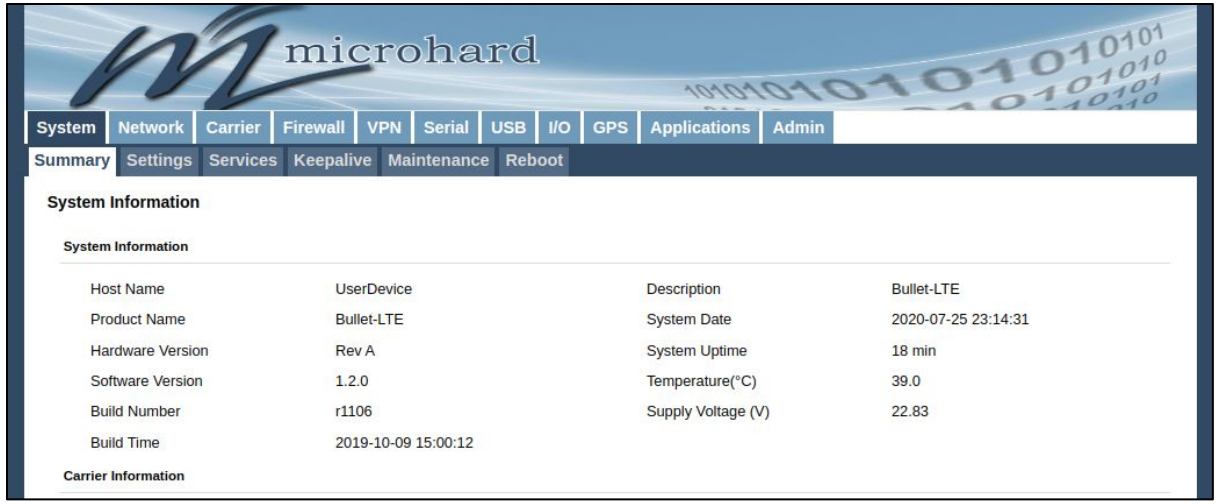
- Response has lots of info
  - Mac addr and IP addr
  - Hostname and SSID
  - Model and firmware version
  - Network Mgmt Software (NMS) domain and password hash (!!!)
    - Cloud-based portal with GPS and other data (out-of-scope for this talk)
- Scanner script and response parser will be on my github

KEYSIGHT

Web Management Portal

Web Management Portal

- uhttpd from OpenWRT circa 2014
- Used for configuring nearly every aspect of device and network
  - Firewall
  - External device routing/comms
  - Running services
- Reachable via LAN *and* WAN
  - Enabled by default
  - Disable via “remote management” toggle
- IT’S ALL SHELL SCRIPTS!!!
  - Oh, and it’s running as root



Web Management Portal

```
ipsec.sh:                  a /usr/bin/webif-page script, ASCII text executable
ipsec-vca-edit.sh:         a /usr/bin/webif-page script, ASCII text executable
ipsec-vca.sh:              a /usr/bin/webif-page script, ASCII text executable
ipsec-x2c-edit.sh:         a /usr/bin/webif-page script, ASCII text executable
ipsec-x2c.sh:              a /usr/bin/webif-page script, ASCII text executable
ipsec-x2s.sh:              a /usr/bin/webif-page script, ASCII text executable
ipsec-x2s-wan2.sh:         a /usr/bin/webif-page script, ASCII text executable
logoutmsie.sh:             a /usr/bin/webif-page script, ASCII text executable
logout_refresh.sh:         a /usr/bin/webif-page script, ASCII text executable
logout.sh:                 a /usr/bin/webif-page script, ASCII text executable
module-forceLN930upgrade.sh: a /usr/bin/webif-page "-U /tmp -u 32768" script, ASCII text executable
moduleL100-forceupgrade.sh:  a /usr/bin/webif-page "-U /tmp -u 32768" script, ASCII text executable
moduleL100-upgrade.sh:     a /usr/bin/webif-page "-U /tmp -u 32768" script, ASCII text executable
module-upgrade.sh:         a /usr/bin/webif-page "-U /tmp -u 32768" script, ASCII text executable
multiwan.sh:               a /usr/bin/webif-page script, ASCII text executable
multiwan-status.sh:        a /usr/bin/webif-page script, ASCII text executable
network-device-list.sh:    a /usr/bin/webif-page script, ASCII text executable
network-dhcp.sh:           a /usr/bin/webif-page script, ASCII text executable
network-gre-edit.sh:       a /usr/bin/webif-page script, ASCII text executable
network-gre.sh:            a /usr/bin/webif-page script, ASCII text executable
network-lan.sh:            a /usr/bin/webif-page script, ASCII text executable
network-localmonitor.sh:   a /usr/bin/webif-page script, ASCII text executable
network-snmp.sh:           a /usr/bin/webif-page script, ASCII text executable, with very long lines (411)
network-sroutes.sh:        a /usr/bin/webif-page script, ASCII text executable
network-wan.sh:            a /usr/bin/webif-page script, ASCII text executable, with very long lines (382)
openvpn.sh:                a /usr/bin/webif-page script, ASCII text executable
ptcheck.sh:                a /usr/bin/webif-page script, ASCII text executable
query.sh:                  a /usr/bin/webif-page script, ASCII text executable
radius.sh:                 a /usr/bin/webif-page script, ASCII text executable
request.sh:                POSIX shell script, ASCII text executable
router-smcroute.sh:        a /usr/bin/webif-page script, ASCII text executable
speedtestresult.sh:        a /usr/bin/webif-page script, ASCII text executable
speedtestui.sh:            a /usr/bin/webif-page script, ASCII text executable
```

Vuln #1 – Post-Auth Root Command Injection

- Multiple command injections throughout
- Diagnostic page as an example
  - Implements ping (!!!) and traceroute
  - “If it has ping, it has command injection.” – Me
  - Tried to sanitize input, but only hostname
  - Leave hostname blank
  - Enter count as “1 127.0.0.1; [whatever command] #” and it will run the command as root




```
! empty "$FORM_start_button" && {
        stop_Diagnostics
        if [ "$FORM_choice" = "ping" ]; then
                sanitized=$(echo "$FORM_ping_hostname" | awk -f "/usr/lib/webif/sanitize.awk")
                echo "ping -c $FORM_ping_count -s $FORM_ping_size $sanitized > "$tmpfile" 2>&1 &" >/var/run/tping
                chmod 777 /var/run/tping
                /var/run/tping &
                #ping -c $FORM_ping_count -s $FORM_ping_size $sanitized >> "$tmpfile" &
                echo "Please wait for output of \"ping -c $FORM_ping_count -s $FORM_ping_size $sanitized\"..." > "$tmpfile"
                #ping -c $FORM_ping_count -s $FORM_ping_size $sanitized 2>&1 >> "$tmpfile" &
                uci_set "Diagnostics" "ping" "ping_hostname" "$FORM_ping_hostname"
                uci_set "Diagnostics" "ping" "ping_count" "$FORM_ping_count"
                uci_set "Diagnostics" "ping" "ping_size" "$FORM_ping_size"
```

KEYSIGHT

DEMO TIME

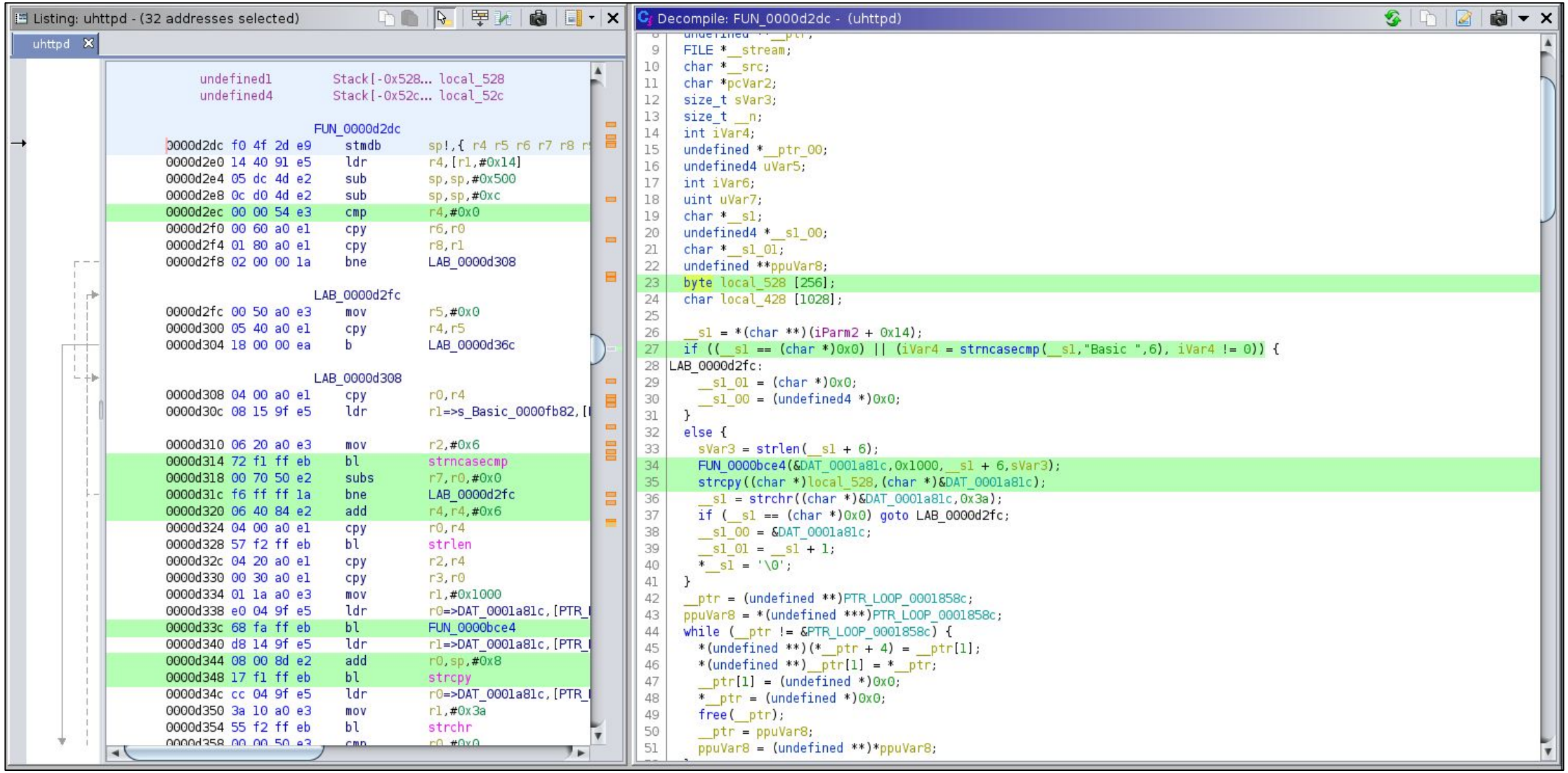
Web Management Portal

- Those cmd inj vulns were all post-auth
- Wanted auth bypass
- Web portal used basic auth code from uhttpd
  - Base64-encoded user:pass
  - Uhttpd is long-standing, well-vetted open source project
    - Surely won’t be any problems to be found there right?
- Well, about that whole “open source” thing...

Vuln #2 – Buffer Overflow in Basic Auth Header

- ...it means anyone can make changes to the code base
- Logins were unmodified, but *logout* algorithm was added by Microhard
  - Logout button sets Basic Auth header to “bG9nb3V0OmxvZ291dA==”
  - Basic Auth header gets Base64 decoded like normal
  - Decoded string gets copied into a 256-byte buffer
  - Copied buffer is later compared to the string “logout:logout”
- String is copied with strcpy() into a statically-sized buffer with no bounds checking
  - The only change they made, and it’s Intro to Buffer Overflows

Vuln #2 – Buffer Overflow in Basic Auth Header

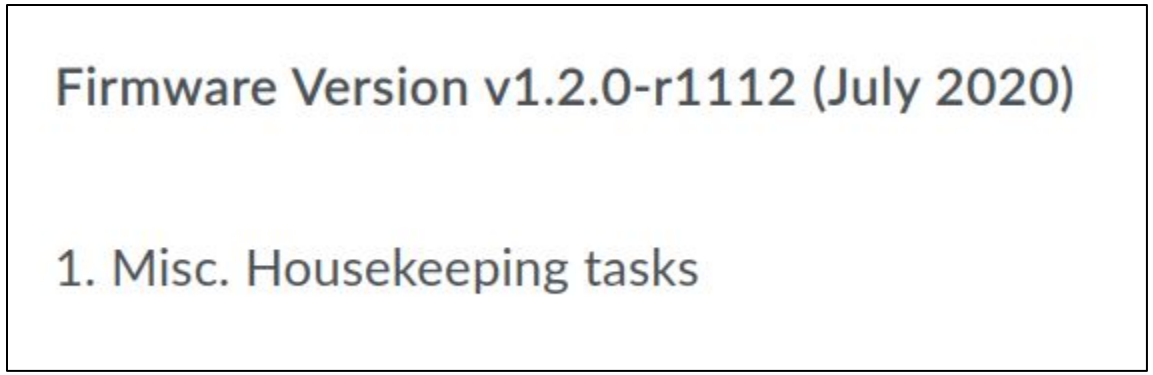


KEYSIGHT

DEMO TIME AGAIN

Web Management Portal

- Disclosed findings to vendor
  - CVE-2020-17406, CVE-2020-17407
  - Fixed in v1.2.0-r1112
    - Added more sanitization
    - Uses lighttpd now instead of uhttpd
    - Still all shell scripts
- Audit ended, put it in a box with other random devices
- Didn’t think about it again until a couple months ago



KEYSIGHT

Telnet Portal

Telnet Portal

- Restricted debug shell for admin user
- Couple of networking config commands
- A TON of AT+ modem commands for interacting with Quectel EC25-AF
- Entering any other command gives “Invalid command” error

Telnet Portal

```
Command Line Interface
UserDevice> ?
  help                                  Show available commands
  history                               Show a list of previously run commands
  info                                  System info
  status                                Display the system status
  system                                Setting system configurations
  network                               Set or Get network config
  AT                                    AT Echo OK
  ATE0                                  Disable Echo
  ATE1                                  Enable Echo
  AT+TEST                               AT Echo TEST
  ATH                                   Show a list of previously run AT commands
  ATL                                   List all available AT commands
  AT&R                                  Reserved
  AT&V                                  Display modem active profile
  AT&W                                  Reserved
  AT+MREB                               Reboot the modem
  ATA                                   Quit
  ATO                                   Quit
  AT+CMGS                               Send SMS
  AT+CMGR                               Read SMS
  AT+CMGL                               List SMSs
  AT+CMGD                               Delete SMSs
  AT+GMR                                Get modem Record Information
  AT+GMI                                Get manufacturer Identification
  AT+CNUM                               Get modem's Phone Number
  AT+CIMI                               Get modem's IMEI and IMSI
  AT+CCID                               Get modem's SIM Card Number
  AT+MSYSI                              Get system summary information
  AT+MMNAME                             Get/Set modem Name Setting
  AT+MLEIP                              Get/Set the IP address of the modem LAN Ethernet interface
  AT+MDHCP                              Get/Set DHCP server running on the Ethernet interface
```

Telnet Portal

- Handled by /bin/clitest
  - 32bit ARM binary
  - Spawned by /etc/m_cli/m_cli.sh when admin user connects
  - Defines all of the AT+ commands
    - And I mean ALL...including a few hidden cmds that didn’t show up in help...

Telnet Portal

- AT+MBASHCMD
  - Secret command for running bash utils!
  - Well, specifically two utils – ps and cat – but still, progress

```
undefined4 cmd_bash_cmd(undefined4 param_1,undefined4 param_2,char **param_3,int param_4)

{
  int iVar1;
  size_t sVar2;
  size_t sVar3;
  char *pcVar4;
  char acStack_a128 [256];
  undefined auStack_a028 [40964];

  if ((param_4 < 2) || (iVar1 = strcmp(*param_3,"="), iVar1 != 0)) {
    pcVar4 = "+MBASHCMD: Invalid parameters";
  }
  else {
    pcVar4 = param_3[1];
    iVar1 = strcmp(pcVar4,"?");
    if (iVar1 == 0) {
      cli_print(param_1,"+MBASHCMD: Command Syntax:AT+MBASHCMD=<Bash Command>\n");
      cli_print(param_1,"Bash Commands Supported:\n");
      cli_print(param_1,"  ps\n");
      cli_print(param_1,"  cat <filename with full path>\n");
      pcVar4 = "OK";
    }
    else {
      sVar2 = strlen(&DAT_000284e0);
```

```
Entering character mode
Escape character is '^]'.

Command Line Interface
UserDevice> at+mbashcmd=?
+MBASHCMD: Command Syntax:AT+MBASHCMD=<Bash Command>
Bash Commands Supported:
  ps
  cat <filename with full path>
OK
```

Telnet Portal

- AT+MBASHCMD
  - How does it work?
    - Uses strncasecmp() to verify input
    - Compares n-bytes of user input against the strings “ps” and “cat”
    - If return != 0, outputs “+MBASHCMD: Unsupported bash command” and bails
    - Otherwise, puts “ > /var/run/cmd_output.txt” on the end and passes to system()
  - So where does the value of n-bytes come from?

Vuln #3 – Post-Auth Root Command Injection...Again...

- N-bytes from strlen() of “ps” or “cat”
- Only checking the beginning of user input
- As long as it *starts* with one of those, it runs
- Just put “ps;” in front of whatever command you actually want to run
- And once again, service runs as root

```
    else {
      sVar2 = strlen(&DAT_000284e0);
->    iVar1 = strncasecmp(pcVar4,&DAT_000284e0,sVar2);
      if (iVar1 != 0) {
        sVar2 = strlen(&DAT_00028512);
->      iVar1 = strncasecmp(pcVar4,&DAT_00028512,sVar2);
        if (iVar1 != 0) {
          pcVar4 = "+MBASHCMD: Unsupported bash command\n";
          goto LAB_0001e9c0;
        }
      }
      memset(acStack_a128,0,0x100);
      iVar1 = 1;
      while( true ) {
        param_3 = param_3 + 1;
        pcVar4 = *param_3;
        sVar2 = strlen(pcVar4);
        sVar3 = strlen(s__var_run_cmd_output_txt_00028544);
        if (0xff < sVar2 + sVar3 + 1) break;
        iVar1 = iVar1 + 1;
        strcat(acStack_a128,pcVar4);
        if (iVar1 == param_4) {
          strcat(acStack_a128,">");
          strcat(acStack_a128,s__var_run_cmd_output_txt_00028544);
->        system(acStack_a128);
          memset(auStack_a028,0,(size_t)"cho ok");
```

KEYSIGHT

DEMO TIME ONE MORE TIME

Telnet Portal

- MBASHCMD removed in v1.2.0-r1132
- Released in May 2024, maybe 2 weeks before I found it...whomp whomp

2024-05-03

IPn4Gii/BulletLTE - v1.2.0-r1132

==========

1. Improved NMS2 and system upgrade states
2. Updated SNMP and AT commands
3. VPN updates
4. Updated modbus support
5. Updated IoT support
6. Carrier connection manager updates



KEYSIGHT

Wrapping Up

Wrapping Up

- Could see an effort to secure things
  - Fell miserably short every time
  - Lack of complete understanding of protection methods
  - Need secure development process or external code audits
- Hiding vulns behind vague release notes is negligent
  - Minimizes impact
  - Customers, especially in critical env will ignore, leaving them vulnerable
- IF IT HAS PING, IT HAS COMMAND INJECTION


KEYSIGHT